Audio Source Separation Using Perceptual Principles for Content-Based Coding and Information Management

by

Kathy Melih

A dissertation submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

November 2003.

School of Information Technology, Faculty of Engineering and Information Technology Griffith University, Gold Coast

Principal supervisor: Dr. Ruben Gonzalez
Associate supervisor: Dr Keith Carter
Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

_______________________________

Kathy Melih,

21st November, 2003
Abstract

The information age has brought with it a dual problem. In the first place, the ready access to mechanisms to capture and store vast amounts of data in all forms (text, audio, image and video), has resulted in a continued demand for ever more efficient means to store and transmit this data. In the second, the rapidly increasing store demands effective means to structure and access the data in an efficient and meaningful manner. In terms of audio data, the first challenge has traditionally been the realm of audio compression research that has focused on statistical, unstructured audio representations that obfuscate the inherent structure and semantic content of the underlying data. This has only served to further complicate the resolution of the second challenge resulting in access mechanisms that are either impractical to implement, too inflexible for general application or too low level for the average user. Thus, an artificial dichotomy has been created from what is in essence a dual problem.

The founding motivation of this thesis is that, although the hypermedia model has been identified as the ideal, cognitively justified method for organising data, existing audio data representations and coding models provide little, if any, support for, or resemblance to, this model. It is the contention of the author that any successful attempt to create hyperaudio must resolve this schism, addressing both storage and information management issues simultaneously. In order to achieve this aim, an audio representation must be designed that provides compact data storage while, at the same time, revealing the inherent structure of the underlying data. Thus it is the aim of this thesis to present a representation designed with these factors in mind.

Perhaps the most difficult hurdle in the way of achieving the aims of content-based audio coding and information management is that of auditory source separation. The MPEG committee has noted this requirement during the development of its MPEG-7 standard, however, the mechanics of "how" to achieve auditory source separation were left as an open research question. This same committee proposed that MPEG-7 would "support descriptors that can act as handles referring directly to the data, to allow manipulation of the multimedia material." While meta-data tags are a part solution to this problem, these cannot allow manipulation of audio material down to the level of individual sources when several simultaneous sources exist in a recording. In order to achieve this aim, the data themselves must be encoded in such a manner that allows these descriptors to be formed. Thus, content-based coding is obviously required. In the case of audio, this is impossible to achieve without effecting auditory source separation.

Auditory source separation is the concern of computational auditory scene analysis (CASA). However, the findings of CASA research have traditionally been restricted to a limited domain. To date, the only real application of CASA research to what could loosely be classified as information management has been in the area of signal enhancement for automatic speech
recognition systems. In these systems, a CASA front end serves as a means of separating the target speech from the background “noise”. As such, the design of a CASA-based approach, as presented in this thesis, to one of the most significant challenges facing audio information management research represents a significant contribution to the field of information management.

Thus, this thesis unifies research from three distinct fields in an attempt to resolve some specific and general challenges faced by all three. It describes an audio representation that is based on a sinusoidal model from which low-level auditory primitive elements are extracted. The use of a sinusoidal representation is somewhat contentious with the modern trend in CASA research tending toward more complex approaches in order to resolve issues relating to co-incident partials. However, the choice of a sinusoidal representation has been validated by the demonstration of a method to resolve many of these issues. The majority of the thesis contributes several algorithms to organise the low-level primitives into low-level auditory objects that may form the basis of nodes or link anchor points in a hyperaudio structure. Finally, preliminary investigations in the representation’s suitability for coding and information management tasks are outlined as directions for future research.
Acknowledgements

The author wishes to express her sincere appreciation to all those people and organisations who have helped to bring this work into existence. First and foremost many thanks to Dr. Ruben Gonzalez for setting a “PhD trap” and for teaching that, especially when the road of duty gets rough, giving up is never an option. His advice and 'big picture' view helped to keep the goal in focus when the way was obscured by a tangle of details.

Secondly, the author thanks Dr Keith Carter for agreeing to act as associate supervisor for a student whom he had never seen and for his continued interest in her progress. Gratitude is also expressed for the brief period of supervision provided by Dr Philip Ogunbona at Wollongong University. His comments and advice provided a completely different perspective that was much appreciated, as has been his continued interest and encouragement.

Various people have provided assistance or advice on a number of issues both specific and general. Among these have been Paul Jardine and Olena Kravchuk. Also, for her assistance in proofreading the manuscript, the author’s sister, Elizabeth Melih is gratefully acknowledged.

The indirect contribution of several tutors, who lifted a major portion of the burden of marking and student consultation while the author was juggling a full-time teaching load with PhD research, also deserves mention.

The financial assistance of the Australian Government, Griffith University and the School of Information Technology, Griffith University, Gold Coast via various scholarships and related grants is acknowledged. The School of Information Technology is also gratefully acknowledged for providing the facilities and resources for completion of the project and funding conference travel throughout its course.

Finally, the author wishes to express her deepest gratitude to her closest friends and family who have endured (for longer than originally expected) broken dates or turned down invitations, roller-coaster moods, incomprehensible mealtime chatter, and the various other symptoms of PhD-itis.
Works Published in Pursuit of this Research


**Table of Contents**

Statement of Originality ........................................................................................................ iii

Abstract ........................................................................................................................................ v

Acknowledgements .................................................................................................................. vii

Works Published in Pursuit of this Research ........................................................................ ix

Table of Contents ...................................................................................................................... xi

Table of Figures ........................................................................................................................ xvi

Table of Abbreviations ............................................................................................................. xxviii

Preface .......................................................................................................................................... xxix

Chapter 1 ...................................................................................................................................... 29

*Introduction* ............................................................................................................................... 29

1.1. Introduction ......................................................................................................................... 29

1.2. Motivation and Context ..................................................................................................... 30

1.3. Problem Definition ............................................................................................................. 41

1.4. Hypothesis .......................................................................................................................... 42

1.5. Scope .................................................................................................................................. 45

1.6. Statement of Contribution ................................................................................................. 46

1.7. Outline of this thesis .......................................................................................................... 50

1.8. Summary ............................................................................................................................. 51

Chapter 2 ..................................................................................................................................... 53

*Background Theory and Previous Work* ............................................................................... 53

2.1. Introduction ......................................................................................................................... 53

2.2. Auditory Perception ........................................................................................................... 53

2.3. Auditory Scene Analysis ..................................................................................................... 62

2.4. Audio’s Perceptual Characteristics ................................................................................... 66

2.5. Audio Signal Analysis Techniques ..................................................................................... 81

2.6. Audio Coding Methods ....................................................................................................... 94

2.7. Audio Information Management Techniques .................................................................. 105

2.8. Summary ............................................................................................................................. 105

Chapter 3 ..................................................................................................................................... 111

*The Data Representation: A Basis for Auditory Source Separation* ................................. 111

3.1. Introduction ......................................................................................................................... 111

3.2. Basic Architecture for Auditory Source Separation ......................................................... 111

3.3. Models of the Auditory Periphery .................................................................................... 113

3.4. Time-Frequency Distribution ........................................................................................... 115

3.5. Masking Thresholds ............................................................................................................ 131
# Chapter 8

**Conclusions and Future Work**

8.1. Introduction

8.2. General Conclusions

8.3. Conclusions Drawn from Grouping Experiments

8.4. Comparison with Other Contributions in the Literature

8.5. Directions for Future Work

8.6. Final Remarks

---

**Appendix A**

*Units of Measurement in Acoustics*

**Appendix B**

*Full Results for Frequency and Amplitude Contour Grouping*

**Appendix C**

*Full Results for Harmonicity Grouping*

**Appendix D**

*Full Results for Intersection-based Grouping*

**Appendix E**

*Full Results for Distance-Vector-based Grouping*

References
Table of Figures

Figure 1-1: Architecture of hypermedia system ........................................................... 33
Figure 1-2: Organizing the auditory scene into a collection of objects is the basic aim of auditory scene analysis ................................................................. 35
Figure 1-3: Possible anchor points within an audio node ............................................... 36
Figure 1-4: Model of mid-level auditory signal representation .................................... 39
Figure 1-5: Structuring audio using a perceptually based representation ................... 40
Figure 2-1: Anatomy of the peripheral auditory system ............................................. 54
Figure 2-2: Cross section of cochlea ......................................................................... 54
Figure 2-3: Absolute threshold of hearing ................................................................. 55
Figure 2-4: Equal loudness contours ....................................................................... 56
Figure 2-5: Frequency sensitivity of Basilar Membrane ............................................ 57
Figure 2-6: Plot of several auditory filter responses .................................................... 58
Figure 2-7: Masking pattern for a tone ..................................................................... 59
Figure 2-8: Masking pattern calculation for a tone at 1 kHz ..................................... 60
Figure 2-9: Spectrogram of a mixed source signal .................................................... 61
Figure 2-10: Mid-level auditory signal representation ................................................ 62
Figure 2-11: Elementary example of auditory stream segregation ......................... 63
Figure 2-12: The principle of closure ....................................................................... 64
Figure 2-13: Effect of rate of presentation and frequency separation on streaming of two alternating tones ................................................................. 65
Figure 2-14: Speech production model used by cepstral pitch detection algorithms .................................................................................................................... 72
Figure 2-15: Basic multipitch estimation algorithm developed by Klapuri et al ....... 73
Figure 2-16: A simplified short time Fourier transform .......................................... 83
Figure 2-17: Oscillatory network employed by Wang in his neural network approach to CASA ................................................................. 92
Figure 2-18: General form of a perceptual audio codec ......................................... 96
Figure 2-19: Basic 'noise' signal extraction for sinusoidal coding.......................... 103
Figure 2-20: Multimedia archival architecture ....................................................... 105
Figure 2-21: Overview of Klapuri’s blackboard architecture .................................. 106
Figure 3-1: Basic model adopted by CASA systems ............................................. 112
Figure 3-2: Architecture adopted for source separation in this thesis .................... 112
Figure 3-3: Basic mapping between transform window size and resolution of a constant-Q transform ................................................................. 116
Figure 3-4: Resolution of hybrid low-pass filter-bank followed by a fixed resolution transform in each band........................................117
Figure 3-5: Impulse response of filter.................................................................122
Figure 3-6: Filter frequency response .................................................................122
Figure 3-7: Peaks generated for 175Hz tone using MDCT-based analysis.................................................................124
Figure 3-8: Peaks generated for 175Hz tone using FFT based-analysis ..........124
Figure 3-9: Window alignment to take compensate for filter delay...............128
Figure 3-10: Apparent window alignment..........................................................129
Figure 3-11: FFT analysis of a 1 kHz tone with commonly used windows applied ................................................................130
Figure 3-12: Architecture of TFD generation system........................................131
Figure 3-13: Absolute threshold of hearing curve used to remove inaudible noise from the TFD.................................................132
Figure 3-14: Example of voiced speech spectrum before and after masking has been applied.................................................133
Figure 3-15: Results of the original Terhardt algorithm.................................134
Figure 3-16: Peak-picking result of the Terhardt algorithm with threshold reduced to 0.5dB..........................................................135
Figure 3-17: Peaks detected by the compromise algorithm.............................136
Figure 3-18: Horizontal bias of McAulay and Quatieri peak tracking algorithm ..............................................................................139
Figure 3-19: Results of the M&Q tracking algorithm for a mixed source file indicating the tracking errors that occurred.........................140
Figure 3-20: Example of the crossed partial problem in CASA .....................140
Figure 3-21: Example of potential incorrect tracking due to simple proximity rule ..........................................................................141
Figure 3-22: Results of the extended M&Q algorithm for a mixed source file indicating tracking improvements over the original algorithm.........................................................143
Figure 3-23: Multi-band tracking for a signal displaying FM typical of musical vibrato .................................................................144
Figure 3-24: Multi-band tracking for purely tonal signal..................................144
Figure 3-25: Architecture of the initial track and group formation procedure ......................................................................................145
Figure 3-26: Results for band collapsing for the FM signal depicted in Figure 3-23 .................................................................................146
Figure 3-27: Results of band collapsing for tonal signal depicted in Figure 3-24 .................................................................................146
Figure 3-28: Collapsed FM track after smoothing.............................................147
Figure 3-54: Results of extended track consolidation algorithm for mixed source file ......................................................... 169
Figure 3-55: Algorithm for track inversion ................................................................. 170
Figure 3-56: Algorithm for deriving the residual ..................................................... 170
Figure 4-1: Illustration of the problem of shared harmonics .................................. 173
Figure 4-2: Comparison of the data driven (traditional) CASA architecture and the prediction driven approach employed by Ellis ................................................................. 176
Figure 4-3: Group number ratio for example case .................................................. 181
Figure 4-4: Hausdorff distance grouping accuracy metric ....................................... 182
Figure 4-5: Group and track count grouping accuracy metrics for the example case ........................................................................ 184
Figure 4-6: Example of a file containing tracks that are temporally remote from one another ......................................................... 186
Figure 4-7: Example of the histogram used for relative comparison of algorithms ........................................................................ 187
Figure 4-8: Alphabet of possible track shapes ............................................................ 194
Figure 4-9: Classification of track shapes .................................................................. 194
Figure 4-10: Results of classification based grouping procedure for single and mixed source files .......................................................... 196
Figure 4-11: Tracks derived for a male speaker saying "g" ....................................... 197
Figure 4-12: Close-up of fundamental track in Figure 4-11 ..................................... 197
Figure 4-13: Track produced for a piano note .......................................................... 198
Figure 4-14: Illustration of threshold calculation ..................................................... 200
Figure 4-15: Illustration of how scaling affects MSE similarity determination ....................................................................... 202
Figure 4-16: Results of scaling tracks to a common average frequency of 2kHz ........................................................................ 203
Figure 4-17: Scaling tracks to a common frequency .................................................. 204
Figure 4-18: Results of shape normalisation ............................................................. 206
Figure 4-19: Grouping error due to transience in the fundamental ......................... 207
Figure 4-20: Tracks of Figure 4-19 normalised to shape ........................................ 207
Figure 4-21: Results of track trimming ................................................................. 208
Figure 4-22: Result of applying shape normalisation to trimmed tracks ................ 208
Figure 4-23: Summary of results for frequency contour MSE-based grouping ........................................................................ 209
Figure 4-24: Number of files for which grouping performance was perfect ................. 210
Figure 4-25: Results for MSE amplitude contour-based grouping

Figure 4-26: Comparison of amplitude contours normalised to mean and shape

Figure 4-27: Count of files with perfect grouping performance based on amplitude contour MSE grouping

Figure 4-28: Comparison of MSE-based frequency and amplitude contour grouping

Figure 4-29: Comparison of scaled amplitude contours and frequency contours

Figure 4-30: Illustration of grouping considerations for tonal signals

Figure 4-31: Shape parameters for some typical track shapes

Figure 4-32: Time-normalised shape parameters

Figure 4-33: Results for Frequency Contour Shape Parameter-Based Grouping

Figure 4-34: Amplitude Contour Parameter-based Grouping Results

Figure 4-35: Results of Fourier shape descriptor for frequency contour-based grouping

Figure 4-36: Results of Fourier shape descriptor-based amplitude contour grouping compared to the shape parameter-based method

Figure 4-37: Summary of results for frequency contour grouping

Figure 4-38: Summary of results for amplitude contour grouping

Figure 4-39: Comparison of results of the optimum frequency and amplitude contour-based grouping algorithms

Figure 5-1: Illustration of problems associated with using a key frequency for harmonicity metric calculation

Figure 5-2: Problems with using a key frequency as a harmonicity measure

Figure 5-3: Simplified example of a pair of harmonic sieves used in harmonicity grouping

Figure 5-4: Illustration of important histogram bin characteristics

Figure 5-5: Accuracy of harmonic number estimation for each algorithm

Figure 5-6: Grouping results for Bregman and Euclid based harmonicity grouping algorithms

Figure 5-7: Results for track-based harmonic number determination and grouping with respect to the harmonic number estimator threshold

Figure 5-8: Results for harmonicity grouping with respect to main grouping threshold using the track-based Bregman algorithm
Figure 5-9: Comparison of the peak- and track-based versions of the Bregman inspired harmonicity grouping algorithm................................. 241
Figure 5-10: Accuracy of harmonic number estimation using the methods of continuing fractions and table lookup................................. 244
Figure 5-11: Harmonicity-based grouping results for frequency ratio approach using the method of continuing fractions and table lookup......................................................... 246
Figure 5-12: Grouping performance versus harmonicity threshold for the continuing fractions technique .......................................................... 247
Figure 5-13: Summary of grouping results for harmonicity-based algorithms.......................................................................................... 248
Figure 6-1: Summary of performance of each of the distance metrics tested.......................................................................................... 249
Figure 6-2: Maximal region from the four-dimensional relative performance histogram showing performance of the intersection-based grouping technique ................................................. 254
Figure 6-3: Region of best performance from the four-dimensional frequency, amplitude, harmonicity, mis-grouped tracks function........................................................................................................ 255
Figure 6-4: Results for intersection based grouping as a function of the various thresholds............................................................................................................ 255
Figure 6-5: Example of undergrouped case of track grouping using the intersection based combined grouping strategy........................................... 256
Figure 6-6: Example of incorrect track grouping using the intersection based combined grouping strategy......................................................... 257
Figure 6-7: Results of distance-vector based grouping as a function of the minimum threshold value........................................................................................................... 258
Figure 6-8: Results for the distance-vector based combined grouping approach........................................................................................................... 259
Figure 7-1: Example of segmentation for a file containing mixed source data............................................................................................. 270
Figure 7-2: Use of shape parameters in track type identification .......................................................................................................................... 271
Figure 7-3: Results of shape-parameter based track type identification compared with the simple method .......................................................... 272
Figure 8-1: Example of coding an artificial chirp and a male speech signal ................................................................................................. 285
Figure 8-3: Allowed track trajectories for 3-D case............................................................................................................................... 288
Figure 8-4: A basic sketch of a visual display for hyper-audio .................................................................................................................. 293
Table of Abbreviations

Unless otherwise noted in the text, the following abbreviations are used as defined in the table.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPCM</td>
<td>Adaptive Differential Pulse Code Modulation</td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td>Amplitude Modulation</td>
<td></td>
</tr>
<tr>
<td>ASA</td>
<td>Auditory Scene Analysis</td>
<td></td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
<td></td>
</tr>
<tr>
<td>BSS</td>
<td>Blind Source Separation</td>
<td></td>
</tr>
<tr>
<td>CASA</td>
<td>Computational Auditory Scene Analysis</td>
<td></td>
</tr>
<tr>
<td>CELP</td>
<td>Code Excited Linear Prediction</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
<td></td>
</tr>
<tr>
<td>DHT</td>
<td>Discrete Hartley Transform</td>
<td></td>
</tr>
<tr>
<td>DPCM</td>
<td>Differential Pulse Code Modulation</td>
<td></td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
<td></td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
<td></td>
</tr>
<tr>
<td>FOHM</td>
<td>Fundamental Open Hypermedia Model</td>
<td></td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
<td></td>
</tr>
<tr>
<td>HILN</td>
<td>Harmonic and Individual Lines plus Noise</td>
<td></td>
</tr>
<tr>
<td>JND</td>
<td>Just Noticeable Differences</td>
<td></td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Prediction (or Predictive) Coding</td>
<td></td>
</tr>
<tr>
<td>M&amp;Q</td>
<td>McAulay and Quatieri – usually in reference to [73]</td>
<td></td>
</tr>
<tr>
<td>MDCT</td>
<td>Modified Discrete Cosine Transform</td>
<td></td>
</tr>
<tr>
<td>MIDI</td>
<td>Musical Instrument Digital Interface</td>
<td></td>
</tr>
<tr>
<td>MM</td>
<td>Multimedia</td>
<td></td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Pictures Expert Group</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
<td></td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
<td></td>
</tr>
<tr>
<td>PDCASA</td>
<td>Prediction Driven Computational Auditory Scene Analysis</td>
<td></td>
</tr>
<tr>
<td>pdf</td>
<td>probability density function</td>
<td></td>
</tr>
<tr>
<td>RTP</td>
<td>Real Time Protocol</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>Sensation Level</td>
<td></td>
</tr>
<tr>
<td>SPL</td>
<td>Sound Pressure Level</td>
<td></td>
</tr>
<tr>
<td>STFT</td>
<td>Short-Time Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>STN</td>
<td>Sinusoids, Transients and Noise</td>
<td></td>
</tr>
<tr>
<td>TF</td>
<td>Time-Frequency</td>
<td></td>
</tr>
<tr>
<td>TFD</td>
<td>Time-Frequency Distribution</td>
<td></td>
</tr>
</tbody>
</table>
Preface

We live in a noisy world. A never-ending, low-level drone punctuates even what we term “peace and quiet”. The gentle rustle of the breeze through long grass, the happy chatter of various birds about their business, the whir of a pigeon in flight, the incessant hum of cicadas, the gurgle of a fast flowing shallow stream over its rocky bed, etc. All these bombard the ear in what by all rights should be a totally incomprehensible confusion. Yet, miraculously, we are able to make sense of the confusion and attend (or not, if we choose) to whichever instrument in this natural orchestra we choose. Not only are we able to make sense of the confusion, but we also seem capable of doing so without expending much, if any, conscious effort. Therefore we may be lead to conclude that the task must be simple. Or is it?

After some ten years of research in computational auditory scene analysis (CASA), attempts to mimic this amazing ability of the human perceptual system are still very much in their infancy. Early attempts at CASA were simply pure research for its own sake or to gain a better understanding of how the perceptual system works. However, potential applications were quickly identified and most CASA systems have been developed with a specific application in mind. Perhaps the most common, and certainly oldest, application area of interest to CASA researchers has been the development of more robust automatic speech recognition systems. However, the full potential of CASA is not realised in these systems, being relegated to play the role of little more than a sophisticated noise cancellation tool.

At the same time we have been facing a quiet invasion. The dawn of the information age having long given way to the full heat of the day has left us literally drowning in a sea of data in every form, text, image, audio and video. Data is raw, unorganised information and that is precisely what the “information age” has given us. We have the ability to store on a plastic and metal film disk about the size of saucer more information than the average man living 100 years ago could have accessed in his entire life as well as ready access to devices that can very quickly fill this space with data. Yet the mechanisms we have available to organise and access this data are, for the large part, merely the electronic counterparts of those that Vannevar Bush criticised as being “as old as the square rigged ships” as far back as 1945.

With his keen foresight, Vannevar Bush identified a dual relationship that nevertheless seems to have escaped notice. Data compression and information management go hand in hand. Attempting to solve either problem in isolation exacerbates the other. We have already noted that the ability to store vast amounts of data without a suitable access and organisation mechanism for it is more of a loss than a gain. Compressing the data simply allows the conceptual size of the record to grow further still, with no relief to the access problem in sight. Attempting to impose an artificial organisation on the data via some form of meta-data or labelling scheme necessarily increases the data volume. This not only runs the risk of exhausting the available storage space
(which admittedly is becoming less of a problem) but perhaps more importantly, is likely to physically increase the time required to retrieve the data, whether the delay be caused by having to physically change a DVD or for a data stream to travel down a communications link.

Thus we have identified a pressing need: the ability to simultaneously support data compression and information management for all data types. Each domain will present its own unique challenges in solving this problem, hence, we shall focus on audio data hereafter. The first step in addressing this dual problem is to identify the requirements imposed by each partner in our dual relationship as well as the limitations it imposes on the other. To support compressed data representation without impeding information management, we should identify all areas of data redundancy and eliminate them as far as possible without destroying access to useful indexing or organisational cues. To support information management without violating the requirements of compression, we should seek a method to arrange the data so that it is self-indexing. We should also define mechanisms that allow for one to identify the inherent structure of the underlying audio data in its coded form. To use the analogy of textual data we can think of “self-indexing” as being equivalent to the ability to perform a key word or string search. The mechanism for identifying structure in a text document refers to the punctuation and formatting used to identify sentences, paragraphs, headings, etc.

At first glance the above analogy would seem to suggest a transcription-based solution to the problem. This might provide a part solution for speech and musical data, however it is far from ideal. In the first place, it is contrary to the main limitation imposed by the need to support compressed data storage because the index would necessarily have to take on the form of meta-data. Secondly, while moderate levels of success have been achieved with automatic speech recognition and monophonic melody transcription, neither of these mechanisms is robust in a generic unconstrained environment. Further, a transcription of general audio data, such as the one that appears at the beginning of this discussion, is, for the most part, of little general use as the description must necessarily be subjective, and therefore subject to much variation among and between users and implementers of such a system. Further, there is much semantic content in an audio signal that is difficult, if not impossible, to describe with a transcription. For example, a person’s tone of voice often is a better indicator of their state of mind than the words that they utter. Similarly, a single melody line can at one time be played in a light, “happy” tone and at another in a darker, more serious fashion. At the very least, adequately describing such information with a transcription would increase the volume of the meta-data beyond that of the data it is describing.

Thus we need to extend our analogy further. The structure in text is composed of individual semiotic elements (letters) that are grouped together according to syntactical rules into semantic units (words) which, in turn, are grouped together via grammatical rules into cognitive units (phrases, sentences and paragraphs). Hence, we have now clearly defined the problem. In order to
create an audio representation that is self-indexing and explicitly reveals the structure of the underlying data, we must determine the audio equivalent of a letter, word and phrase or sentence. For the purposes of this discussion we shall call the audio equivalent of a letter, a low-level object or partial; of a word, a mid-level object or group; and a phrase a high-level object or simply an audio object.

At this juncture we find it necessary to turn our gaze to the psychology of hearing and the precursor to CASA, auditory scene analysis (ASA). What is known of the process of audition suggests that the noisy mix of compression waves bombarding the ear drum are eventually transformed, via several mechanical and transduction mechanisms into an electrical signal resembling a three-dimensional function known as a time-frequency distribution. This function reveals the variation of the signal characteristics in both time and frequency. Further it is apparent that the local maxima of this function are of particular significance to the perceptual system. Finally, it has been clinically determined that the perceptual system “tracks” peaks that are in close proximity on the time-frequency plane and that neurones exist to code at least three special features derived from this analysis: the tone, the sweep and the noise burst.

ASA is the term coined by Bregman to describe the study of how the perceptual system organises these temporal-spectral features to ultimately make sense of the auditory stimulus that gave rise to these elements. CASA is the name given to the computational science that seeks to implement the findings of ASA in order to 1) better understand these findings and 2) exploit the benefits of auditory source separation in various applications. In our discussion of the lowest levels of auditory perception we have already achieved our first aim in that we have identified the low-level objects, or atomic units in our structure. ASA will lead us to recognise the objects at the higher levels of our structure and to identify the rules used to form the higher level elements from the lower ones while CASA will provide us with computational tools in order to implement these rules.

Many controlled experiments have led to the proposal of a series of principles believed to govern the process of mid-level object, or group, formation. Briefly, in order for partials to fuse into groups they must bear some resemblance in the time-frequency plane. Clearly temporal proximity is a strong grouping cue. Two other strong grouping cues are the existence of a harmonic relationship between the frequencies of the partials and correlated frequency or amplitude modulation in contemporaneous sections of the partials. The formation of auditory objects is more complex and can involve more than one level based on contextual factors. This is borne out in the textual analogy in that phrases, sentences and paragraphs were all mentioned at the same level of the structural hierarchy, yet clearly they represent different levels of organisation. In any case, at the lowest levels of auditory object formation, Gestalt principles are heavily involved.
Hence, we have identified an alphabet, syntax and grammar for our required audio structure. We may have confidence that it will be self-indexing because we have identified atomic units whose characteristics will presumably be unique to individual sounds (and, incidentally, this is a presumption that has been validated). We can also be certain that it will be semantically relevant since we have modelled it on the very perceptual process responsible for organising audio data. Thus, we are left with but one task: to implement it without the direct help of the master computer that revealed it to us.

As has already been intimated, this is no mean feat. However, if we are to ever truly solve the duel compression-information management problem, this is a challenge we must meet head on. Ten years or so of work on CASA arms us with the knowledge and intuition of what does work, what does not and what might work better. Thus we have a firm platform from which to commence our study.

Given the evidence of this history, it might be surprising to find that a representation that has been branded with the “doesn’t work” label of late has been selected as the basis of the current work, that is a classical McAulay and Quatieri style sinusoidal representation. However, there are several arguments that may be raised in its defence. Firstly, the bias against this representation has in part been motivated by the assertion that it is at the least very difficult, if not impossible, to deal with coincident partials from simultaneous sources. It will be shown in this thesis that at least in the case of harmonic sources, where at least some partials are independent of each other, this need not be the case. Secondly, as the primary motivation of the current work was to develop a representation that would eventually support content-based coding and information management, the requirements of this were an overarching concern. As such, the more computationally intensive systems that are gaining popularity in the CASA community which aim to describe the stimulus first and invert it as a secondary aim, if at all, were not deemed to be suitable for this purpose. Finally, while it is recognised that the formation of even the lowest level audio objects benefits greatly from higher level cognitive processes, a recognition driven approach is not suitable for developing a generic audio coding representation, since the former is highly context sensitive while the latter should ideally remain unencumbered by contextual considerations.

A further departure from the norm is evident in the time-frequency resolution of transform underlying the sinusoidal representation that was designed to provide the maximum possible resolution in both time and frequency planes simultaneously. This resulted in a multi-band track representation that required consolidation. Initial attempts at this consolidation using a simple local averaging procedure proved discouraging hence an alternative was sought. The search resulted in a completely new architecture for the CASA problem. Like many previous systems it employed feedback to improve on decisions made in previous stages based on those made in latter
stages. However, its uniqueness lies in the source and destination of the feedback, being much lower down the chain than is traditionally adopted.

At the heart of the solution lie the algorithms that implement the grouping of the primitives to form groups. Since only the tonal and sweep primitives were considered in this study, the groups can correctly be called harmonic groups. Given the inherent characteristics readily available in the primitives derived from the sinusoidal representation, three grouping cues were natural candidates for the task: frequency contour similarity, amplitude contour similarity and harmonic relationship. A fourth cue, loosely implied in the implementation of grouping algorithms, was common temporal extent. Various algorithms for grouping based on each of the cues individually were developed and tested along with two algorithms that combined all three cues. Some of these were similar to those of previous or contemporary research while others were, to the author’s knowledge, unique to the current work.

It was found that a classical Mean-Squared Error-based approach, with one novel variation in the manner in which the tracks were scaled, performed best for frequency contour grouping. An entirely new algorithm based on average gradients of portions of the track performed best for amplitude contour grouping and a unique algorithm for harmonicity grouping, based on a theory of pitch perception suggested by Bregman, was also found to be optimal. For the combined approaches, a common Euclidean distance in a distance-vector-based approach was found to be superior to a unique set intersection based method.

The final conclusions drawn from the experimental work were that it is indeed possible to achieve at least the preliminary level of auditory source separation based on a sinusoidal representation. With this validation, it is possible to plan future work in all three fields that have influenced the current work. The conspicuous omission of a noise model in the sinusoidal representation poses a difficult CASA challenge and an opportunity for quality enhancement for coding concerns. Further, the formation of low-level groups and the track based representation itself provides the basis for a compressed data representation with scope for research as to how best to quantise the elements of this representation without compromising the unique properties and accessibility of the representation. Finally, this representation also provides the initial building blocks from which to build prototype audio information management systems, including query-by-example and hyperaudio systems.
Chapter 1

Introduction

1.1. Introduction
We are constantly surrounded by auditory confusion, yet the brain has an amazing ability to make sense of the cacophony. Inspired by the analogy to recognising objects in a visual scene, Bregman [1] coined the term “auditory scene analysis” to describe this phenomenon. Exactly how the auditory system achieves this task is a subject of ongoing research, however, we already know enough about the underlying principles to attempt to model the process.

There are many reasons why we would like to be able to model the process of auditory scene analysis electronically. Some have long been areas of concern, while interest in others is only just emerging. To cite an example of the former, it has long been recognised that practical automatic speech recognition systems would greatly benefit from some unconstrained method to separate the speech of interest from the background ‘noise’. An example of the latter would be as an aid to research in the auditory processes underlying this ability. In other words, it is hoped that a greater understanding of the perceptual system will be gained by attempting to model its behaviour.

Two somewhat newer potential applications of an electronic model of the auditory scene analysis process are structured, content-based coding, and content-based retrieval and browsing of audio data. These applications are the focus of this thesis. As will be discussed in the literature review, a number of different audio signal representations fall under the broad definition of “structured representations”. However, for content-based coding of generic audio signals, the definition of a structured representation must be significantly narrowed. One of the principle features of such a representation will be the ability to encode the audio stream as a series of audio objects. Each of these objects will represent a single, cognitively significant acoustic event, such as a single word. The need for auditory signal separation is obvious where several events occur simultaneously. Similarly, content-based audio retrieval benefits greatly from source separation in that it allows the user to search for an acoustic event that is temporally co-located with others.

Classically the problem of audio source separation has been approached using one of three broad methods: via signal enhancement techniques, blind source separation (BSS), or computational auditory scene analysis (CASA). Signal enhancement is a binary
system where it is assumed that only one source is of interest and the remainder of the signal is noise. Hence, signal enhancement has limited utility for content-based retrieval and virtually none in the context of content-based coding. Blind source separation is a purely statistical approach that assumes that the incoming audio signal is a mixture of statistically independent signals. CASA approaches the problem by attempting to model and mimic the human auditory perceptual system. Both BSS and CASA have their advantages and disadvantages and one method may be more appropriate than the other in a given application. For reasons that will be discussed later, it is the contention of the author that, for the purposes of content-based audio coding and information management, the CASA approaches are the most appropriate.

This brief introduction has set the scene for the work that is reported here. The remainder of this chapter begins by further detailing the motivation and context of the work, followed by a formal statement of the problem addressed and details of the hypothesis explored. Brief outlines of the solution and the validation procedures employed are then presented before the scope of the work performed and the contributions that have been made are detailed. Finally, an outline of the remainder of this thesis is given.

1.2. Motivation and Context

1.2.1. The WWW: A Square-rigged Ship in Disguise?

In 1945 Vannevar Bush published his vision for the ideal information storage, retrieval and management system [2]. In his seminal work, Bush described a device that he called the “memex”, which was intended to augment the memory of the researcher. This device would be fed a constant stream of data via a myriad of input devices. A small camera mounted on the researcher’s forehead would take photographs of important objects in his field of view; a document camera would quickly capture a text document while a voice recorder would dutifully take down the researcher’s audible thoughts. Further, this data would be compressed before being stored to minimise storage and transmission costs. While in 1945 all these capabilities were virtually science fiction, they are now readily available to the average consumer at relatively low cost, with the repository of all this data (a PC) being considerably smaller than the desk-sized object of Bush’s vision.

With this in mind, one may well ask, has Bush’s vision finally become a reality? So far as it has been described in the previous paragraph, the answer is in the affirmative. However, Bush’s assessment of the above situation is perhaps even more poignant today than it was nearly 60 years ago [2]:
Thus far we are worse off than we were before – for we can enormously increase the record; yet even in its present bulk we cannot consult it.

The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of the square-rigged ships.

Bush blamed the situation on “the artificiality of systems of indexing” that run contrary to the cognitive process which is associative in nature. The key concept that Bush’s memex introduced was the ability to link items together such that one item could be made to bring up another item automatically. He predicted that a new form of encyclopaedia would be developed “with a mesh of associative trails running through them.”

It should be obvious that Bush was describing what has come to be known as “hypermedia.” Afraid of “loosing touch with reality,” Bush refrained from extending the predicted capabilities of the memex to the electronic exchange of data. However, today the electronic exchange of data is a very familiar reality and the hypermedia data model has been embraced as a powerful information management tool. Nevertheless, despite these advances, information management of audio data is still riding the “square-rigged ships” with even the concept of hyperaudio still very much in its infancy. Some of the reasons for this lag lie in short-comings of Bush’s original vision that have persisted while others are the result of technological hurdles, which are somewhat unique to audio data, and remain to be addressed.

Despite his incredible foresight, Bush neglected to recognise one very important factor that is still commonly ignored by many researchers in multimedia information retrieval. That is, the information content of multimedia signals is often encapsulated in their presentation, this is especially the case for audio data. For example, a person’s tone of voice often gives a more accurate description of their thoughts or mood than the words that they are actually speaking. Hence, a textual transcription of the form suggested by Bush, and still used in many modern audio retrieval systems, is insufficient for a truly general audio information retrieval system. So, to be a complete and accurate record of his thoughts, the memex would have to note down the researcher’s tone of voice along with his words. As will be evident later, any form of textual description is generally insufficient for this task.
Audio data poses yet another problem that was not addressed by Bush, and, although sometimes recognised as a challenge, is rarely addressed in the literature on audio information retrieval systems. The problem lies in the observation, made in the introduction, that auditory events rarely, if ever, occur in temporal isolation. The implications of this observation for the memex are that, not only would it have to be making a faithful record of all that was being said, but, in order for it to be able to do so, it would first have to separate the researcher’s voice from the surrounding cacophony. Further, consider the example of a memex-based encyclopaedia of music. Here there would be many instances where links to individual parts of audio recordings would be appropriate. For example, an article on a particular composer’s style might link to various examples of a commonly occurring motif played by individual instruments and separated from their ensemble setting. One way to achieve this would obviously be to record these separately, however, this is not always possible, especially if one wishes to use existing recordings. Thus, auditory source separation is a mandatory requirement if true hypermedia is ever to be realised.

It would appear to many that the World Wide Web is already a modern-day, global version of the memex and, as far as Bush’s system was described, this observation may indeed be correct. However, the foregoing observations have shown that true hypermedia, and hyperaudio in particular, will not exist until some unique challenges have been met. In order to establish what developments are required to generate true hyperaudio, a formal description of the hypermedia data model will now be presented.

1.2.2. **The Hypermedia Data Model**

The hypermedia data model is fundamentally very simple. The underlying data are organized into a set of self-contained units called nodes that are connected by links that represent the relationships between them. The nodes are defined independently of the links such that links can be formed or removed at will with no impact on the actual data or nodes themselves. It is also possible to define anchor points within nodes that act as either the source or destination of links. Neither the node or link layers contain any information as to how the data should be presented to the user (such as font selection or text layout, or in the case of audio, playback volume or rate). Rather, this is defined within the user interface layer that also provides the navigation interface. Hypermedia engines typically manage all three layers simultaneously. Figure 1-1 illustrates the three layers of the hypermedia model.
Abstracting access to the data in this way has a number of powerful advantages. Separating the link structure from the nodes themselves allows multiple references to a particular node to be made with only a single copy of the data itself being stored. This has obvious benefits both from the point of view of expediting access and optimising storage requirements. Abstracting the presentation layer from both of the lower levels has similar advantages in that it allows the same nodes, and even link structures, to be presented in any number of user defined, context-dependent manners.

Thus, the basic requirements of any hypermedia system are:

- **Node definition**: a means to chunk the data into self-contained conceptual units that can contain anchor points for the source or destination of links;
- **Links**: a means to define and manage relationships between the nodes; and
- **Interface**: a presentation engine that is independent of the node-link structure.

Despite its simplicity, the hypermedia data model is a powerful tool for information management. Representing a given document, for example, as conceptual units (nodes) rather than lexical units (letters or words) allows it to be viewed as a structured collection of related ideas and not just as a string of letters or words. Thus, the focus can be on what message is being conveyed rather than on how it is being conveyed. Further, the information structure can be manipulated without having to deal with the underlying data. The power of this level of abstraction becomes even more apparent in the case of
audio data where the basic lexical units (the instantaneous amplitude values of a waveform) convey very little useful content information and where manipulation of the raw data representation is tedious and requires a high level of user expertise.

1.2.3. The State of the Art in Hypermedia Systems
Thus, hypermedia systems would obviously be a powerful information management resource. Yet the question remains, does a true hypermedia system exist? In the case of textual and graphical data, the WWW offers a partial answer in the affirmative. However, as far as audio data is concerned, the situation has progressed little from the position it occupied in Bush’s day. A single audio file can serve only as a complete node at the destination of a link. No mechanism is available to either source or sink a link from within a node and the only controls that are available over the access and presentation of the node are those that are found on a standard tape player, that is start, stop, pause, etc. This is because audio data in its raw form is highly unstructured and gives very little indication of the inherent structure or semantic content of the underlying signal. Because of this obvious deficiency, the WWW would more accurately be described as a “multimedia enhanced” hypertext system than a hypermedia system.

Given the vast amounts of audio data now hidden away in electronic repositories, it is simply infeasible to expect that extracting nodes from, as well defining potential source and destination link anchor points in, this data be done manually. Indeed, this is a nearly impossible task for even a single audio recording because of the temporal nature of audio signals and the fact that several nodes and/or anchor points would commonly coexist in time. Hence, there is a clear need for automated mechanisms to decompose audio data into conceptual units that might form nodes in a hypermedia system.

1.2.4. What is an Audio Node?
Before we can devise mechanisms to extract nodes from a given audio recording, we must first determine exactly what constitutes an audio node. Referring to the definition of a general hypermedia node, we note that it must be a self-contained unit embracing a single concept. The basic idea of a self-contained audio unit is encapsulated in the audio objects of auditory scene analysis (ASA) [1]. The primary aim of ASA is to determine how we, as humans, separate out, without necessarily identifying, each of the constituent sounds that make up an auditory “scene”. Each of the constituent sounds identified by the perceptual system is termed an auditory object or stream.

The exact nature of an auditory object is context dependent. For example, a recording of a piece of music played back through a speaker may at one time form the primary focus
of attention and be considered as a single object that is distinct from the background noise “object”. At another instant in time this object may be combined into the background noise object to make way for a new object that represents the voice of a person that has just entered the room. In both instances, the listener will likely be aware that the background noise object is actually composed of several sub-objects. However, they will generally want to ignore this for the rather pragmatic reason of reducing information processing requirements.

This concept of context dependence leads to a recursive, hierarchical definition for an audio node. However, the nodes at each level will still embrace a single concept. These concepts will range from general to specific between the lower and higher levels of the hierarchy. Thus this hierarchical definition of an audio node is in keeping with that of a general hypermedia node. This concept is illustrated in Figure 1-2.

![Figure 1-2: Organizing the auditory scene into a collection of objects is the basic aim of auditory scene analysis. These same objects are ideal candidates for nodes in a hyper-audio system.](Figure_1-2.png)

Hence an audio node is a self-contained, randomly accessible, unit that contains all the information required to recreate the sound event from whence it originated. It exists in a hierarchy of related audio nodes. The nature of this structure is time varying and given by defining links between and within nodes.

1.2.5. **Linking Nodes**

In order to define links within as well as between nodes, we must be able to identify and access link anchor points within each audio node. An anchor point may be an entire node or it may be a section within the node. Given the hierarchical nature of audio nodes, internal anchor points can be located along two dimensions: time or stream (or “sub-object”). The concept is illustrated in Figure 1-3.
In contrast, existing, unstructured, audio representations can only allow for anchor points at specifically stated temporal locations such as “five minutes, 35.23 seconds from the beginning of the recording” or, more precisely, “25,579 bytes from the beginning of the file”. In reality, however, users are more likely to think of temporal locations with respect to specific auditory events (i.e. the origin of auditory objects). For example, “… about 2 bars in from the beginning of the flute solo.”

Even for expert users accustomed to manipulating raw audio waveforms, the task of locating a specific event is a matter of tedious trial and error since there is virtually no content information of any cognitive significance present in a raw audio waveform. In this instance, the prior decomposition of the audio into a collection of audio objects simplifies the task from one of “hunting” for the exact sample value in an approximate temporal location to auditioning a small set of candidate audio objects. Indeed, if any form of automatic recognition has been applied during the node generation phase, the set of candidates may be further reduced. Once the correct audio object has been isolated, its own description will provide details of the exact start time should explicit knowledge of this be required.

With many statistically based coding schemes, even this relatively simple requirement is difficult to accommodate, as direct random access to the data is impossible. For example, consider the case of data encoded using a predictive scheme where each successive value depends on previous values. While keeping a separate list of temporal locations that identify the sources and destinations of links is relatively trivial, if a user wished to gain access to the destination of one such “link”, the entire track up to and including the section identified as the link destination would need to be decoded before being presented to the user. This fact would tend to discourage the use of compressed data.

**Figure 1-3:** Possible anchor points within an audio node

In this figure, we see a possible setup for anchor points within an audio node. The node includes an anchor point for a recorded lecture about music, with a focus on a specific phrase/passage. The node also includes two streams: a recording of an instrumental duet, with comments on the role of a particular instrument in musical history. The diagram illustrates the timeline of the entire recording, showing how anchor points can be used to navigate through the audio content.
representations in a primitive hyperaudio system where nodes and links are defined by lists of temporal locations, rather like a cue list. Thus, it is manifest that, even at the primitive level, unstructured audio data representations result in a dichotomy between compressed data storage and non-linear access.

1.2.6. Content-based Retrieval
Although Bush considered browsing and associative access superior to index-based retrieval, there are still instances where the latter kind of access is desirable. It may be that a user wishes to retrieve all instances of a particular melodic motif in a collection of music, for example. A major issue that must be addressed to allow for this level of access is the nature of the index keys used. This selection influences both the nature and scope of queries allowed. Possibilities range from manually generated text labels to automatic semantic (content) analysis.

Manual index generation suffers many drawbacks; the most obvious being that it is tedious and not practical for large collections. Another significant problem is that manually generated indexes will be highly subjective and context dependent. For example, what one person defines as modern music another may refer to as noise. This problem is exacerbated by the fact that it is virtually impossible to predict the intentions and desired search criteria of future users. For example, there would be no way for a user to find a recording based on the predominant instrument if this was not one of the keys defined at index creation. Yet another limitation of textual indexes is that audio data contains a great deal of information that cannot easily be described verbally. Examples of this include prosodics in speech and timbre in music.

A major problem associated with automatic content analysis is that, at present, it can generally only operate in highly constrained environments. Usually, the data must first be segmented such that each section contains only one type of sound (e.g., speech, music or a single environmental sound). Also, in the case of automatically generated transcriptions for music and speech, manual intervention is often required to obtain accurate transcriptions. Further, the problems mentioned for manually generated textual indexes, such as restrictions on search criteria, also apply.

Statistical feature vectors lie somewhere between the two extremes of manually generated text labels and fully automated semantic analysis. They offer the advantage of being simple to generate automatically. However, existing statistical descriptions are too low level to be of any significance to the user. For example, the terms “bandwidth” or “spectral envelope”, often used in such systems, mean little to a general user. The other
disadvantage of this method is that many features need to be analysed, often in completely independent calculations, to make the index sufficiently general. Although these may be performed automatically, the processing required may become excessive for large collections.

It was mentioned earlier that audio data contains much information that would not be easy to include in a transcription-based index. This highlights a challenge for multimedia indexing: the information content of MM data is often encapsulated in its presentation. An example of this is the difference between two musicians playing the same piece of music. If one is a beginner, one is likely to play the piece “as written” with strict meter, constant dynamic and a steady tone. The more experienced player, however, is likely to add interpretive variations in tempo, dynamic, stress, vibrato etc. In both cases, a melody transcription would be identical. However, a human listener could easily distinguish between the two performances. Hence, the index keys should ideally be able to describe both the ‘transcribable’ contents of the data as well as the audio characteristics.

Arguably, the most useful form of index key for any data type will allow queries in the form of a ‘sketch’; that is, query-by-example. This is analogous to key-word search in a textual database. In terms of audio data, a sketch could include a sung melody, a clapped rhythm, a sample of a person’s voice, or an imitation of a general sound. The last two examples, in particular show the importance of a description that goes beyond a simple transcription.

Finally, it should be stressed that providing indexing mechanisms for audio data should not preclude compressed storage of this data. Conversely, the need for compressed data representation should not be allowed to hinder the ability to effectively index the data. These two statements have major implications for the nature of the index mechanism selected. The first is that the storage overhead due to the index should be kept to a minimum. Transcription-based schemes are obviously far from ideal in this respect. The second consideration means that most traditional audio coding schemes would not be suitable for index-based retrieval, as they do not allow random access to the data.

**1.2.7. An Audio Data Representation for Hyperaudio**

From the foregoing discussion, it is clear that a key requirement for audio information retrieval is a structured audio representation that possesses a number of characteristics. Firstly, the data must be divided into semantically relevant units. Secondly, these units should be individually decodable and randomly accessible. It is also necessary to provide mechanisms to extract these units automatically from a raw audio stream.
Useful information about the content of the unit should be co-located with the unit itself to avoid the disadvantages of transcription-based approaches. Finally, the representation should provide for data compression, without compromising any of the foregoing characteristics.

Given that auditory object formation is a task that the human auditory system performs routinely, a natural starting point on our quest for a representation capable of meeting the above requirements would be to model the auditory system’s native audio representation. The details of this are given in the literature, so only a very brief summary will be provided now. At the mid-level of the audition process, the cochlea transforms the physical, 2-dimensional audio signal into a 3-dimensional electrical signal representing the time-frequency distribution of the original signal. As far as the auditory system is concerned, it is only the amplitude peaks in this distribution that are important. The resulting representation is illustrated in Figure 1-4.

![Figure 1-4: Model of mid-level auditory signal representation](image)

Sinusoidal coding produces a representation that is somewhat similar to the representation shown in Figure 1-4. Figure 1-5 shows the structural decomposition of an audio scene that may be achieved using such a representation.
The lowest level contains the perceptually relevant atomic features of an audio signal. These represent the features in the mid-level auditory representation shown in Figure 1-4. Essentially two main classes of feature are extracted: a track and a noise burst. The auditory system distinguishes two track classes: tone and sweep. While a single track or noise burst carries little cognitive significance from a user’s perspective, there are several reasons why maintaining access to the tracks would be desirable. One reason is that this allows query-by-example resource discovery using the track parameters as search keys. Another is that the predominant track class in a given group or auditory object may be exploited to classify the type of signal that the group represents into one of three categories: speech, music or general environmental sound. Yet another reason would be to allow audio editing at a level that is impossible with raw audio waveforms. An application where this kind of editing might be useful would be in generating stimuli for psychoacoustic testing.

At the next level up in the structure shown in Figure 1-5 are found harmonic groups and noise frames. The harmonic groups consist of tracks that are co-located in time and bear some resemblance to one another. This similarity may be in any one or more of the following domains: harmonicity, amplitude contour, frequency contour, or temporal extent. Noise frames consist of temporally adjacent noise bursts with similar characteristics with respect to bandwidth, RMS power, spectral envelope, etc. Even with the track and noise burst representation at the lowest level of the hierarchy, extracting the track groups and noise bursts are non-trivial exercises. This thesis concerns itself with the formation of the harmonic groups while methods for extracting the noise bursts are suggested as areas for future work.
The next level up in the hierarchy shown in Figure 1-5 contains audio objects. This level could itself be expanded, depending on the context-dependent definition of an audio object. In speech, for example, the lowest level audio object might be a phoneme or single word. At the next level we might find a single spoken phrase and, at a higher level still, a long monologue. Which level the user interacts with depends on what he aims to do with the objects. If automatic transcription were the aim, then the simplest level audio objects (phonemes) would suffice. On the other hand, someone wishing to interactively skim (or edit) a lengthy recorded meeting would interact with higher-level objects (monologues/phrases).

In compliance with the requirements stated earlier, the elements of the structure are randomly accessible and individually decodable. From a user’s perspective, the relevance of this increases significantly at each step up the hierarchy. While the elements at lower levels form perceptually significant units, they contain incomplete cognitive information. Decoding a single track at the lowest level results only in a single sinusoid that may or may not be frequency modulated. This obviously has very little audition value. Inverting a single harmonic group may well result in an intelligible signal, however, very few naturally occurring sounds are purely tonal or noise-like. Hence, it is much more likely that a combination of at least two mid-level elements is required to create an inversion that is both intelligible and of acceptable quality. It follows that inverting an individual audio object should result in a signal that is both intelligible and of reasonable perceptual quality.

1.3. Problem Definition

Traditionally audio coding and audio information management have been treated as two separate research problems. On the one hand, this has resulted in the possibility to store conceptually vast amounts of data relatively compactly with only limited means available for resource discovery. On the other hand, what information retrieval mechanisms do exist for audio data, rely on analysis of the raw audio waveform and introduce potentially bulky indexes that do not provide a complete description of the underlying data. Further, despite its utility, very little support for audio browsing exists with virtually no support for hyperaudio presently in existence.

As has already been made clear, the chief reason that these issues remain a problem is that, traditionally, both audio coding and information management have been
approached from a purely statistical perspective. On the one hand, this leads to representations that obfuscate the inherent structure, provide little information about the semantic content, and generally preclude random access to the underlying data. On the other hand, purely statistical indexing has little utility to the general user. Finally, the general lack of support for random access in most audio coding representations hinders the development of hyperaudio, which, as we have seen, is a powerful information access model.

1.4. **Hypothesis**

It is possible to develop a structured audio representation to support content-based audio coding and information management by providing intrinsic support for audio data compression, audio content description and auditory source separation. The most appropriate basis for this representation is a model of the mid-level representation of audio in the human auditory system allowing for a computational auditory scene analysis (CASA) approach to the problem of auditory source separation. Further, while many researchers in the CASA field have adopted a precise modelling of each component of the auditory system, it is hypothesised that a carefully designed “black box” approach is sufficient to adequately solve the problem of source separation as it applies to content-based audio coding and data management.

This hypothesis raises a number of questions including: What is a structured audio representation and what characteristics must such a representation possess in order to support content-based coding and retrieval, what characteristics of the human perceptual system will be useful in the design of this representation; and how might a black box design approach be superior to a more faithful model of this system? These questions will now be briefly addressed.

1.4.1. **Structured Audio Representation**

The classic definition of a structured audio representation “description formats that are made up of semantic information about the sounds they represent and that make use of high-level or algorithmic models” [4] encompasses many representations that do not fill the requirements imposed by generic audio management. One such example is a MIDI event list. While this representation may be a useful basis for melody retrieval [5][6], it cannot be applied to audio of a more generic nature, neither is it a trivial matter to generate such a representation automatically from recorded music. Thus, in order for an
audio representation to support content-based retrieval of general audio, it must possess a number of characteristics.

Firstly, despite the ever-increasing bandwidth of our networked world, efficient representation of storage hungry multimedia data is still considered essential. Secondly, in order to allow for content-based retrieval both by example and by “keyword” or semantic descriptions, the data must be divided into individual semantically relevant units. Clearly, these units should also be individually decodable and randomly accessible. Further, it is desirable that a mechanism be provided to allow these units to be extracted automatically from a raw audio stream. Indeed, in the case of separating two simultaneously occurring units (speech over background music, for example) it is impossible to do so manually. Finally, useful information about the content of the unit should be co-located with the unit itself with useful content information being made available without the need for manual labelling.

1.4.2. Perceptual Considerations
The principal requirements of content-based audio coding and information management are efficient data representation, semantically relevant content description, and source separation. Given that the human auditory system is adept at performing all of these functions, it is a natural conclusion that a representation based on it should be able to at least approximate this performance.

The exploitation of peculiarities of the perceptual system to achieve multimedia data compression is by no means a new concept. However, the conventional focus is on removing perceptually redundant information from the data rather than mimicking the auditory system’s native representation. This is a pragmatic approach when data rate reduction is the sole aim but is counterproductive for content-based information management.

To a large extent, the provision of semantically relevant and useful content descriptions of audio data still relies heavily on manually generated transcriptions. For several years much research and development in content-based audio retrieval has focused on automatically generating statistical descriptions of the data [7]. However, these descriptions are generally very low level and of little relevance to the naïve user (for example, fundamental frequency or RMS value). By extracting features from an audio signal that are perceptually important, it is easier to generate descriptions that are more relevant to the generic user (perceived pitch or loudness, for example).
Source separation has traditionally been considered a mathematical problem with certain constraints being placed on the input to the system. These constraints assume much about the nature of the audio signal that cannot be practically applied to generic audio. On the other hand, the best performing auditory source separation system in existence is the actual perceptual system. Hence, it is logical that the optimum general solution to this problem will be based on this system.

1.4.3. The Black-Box Design Approach
There are a number of ways in which a perceptually motivated representation may be generated. These methods range from a very “true-to-life” model of each component of the auditory system through to extremely loose black-box approximations that seek merely to produce an output that is similar to what the auditory system would produce for a given input. As will be evinced in this section, neither method satisfies the requirements of a representation suitable for auditory source separation applicable to content-based audio retrieval and information management.

The first method suffers two major drawbacks. Firstly, the finer details of how the auditory system functions are the subject of ongoing research; hence any model of this process, however precisely it fits the current theories, will only ever be an approximation of the true system. Secondly, whether because of our limited understanding of the process or the level of extant detail within the actual system, “true-to-life” models of the human perceptual system are computationally intensive. For most information management applications, the processing time required by a computationally intensive algorithm is simply not acceptable.

Extremely loose black-box approximations that attempt to solve the problem from a purely analytical perspective have also failed. Examples of the analytic black-box approach include the various blind source separation methods (see section 2.5.5). In general these systems suffer from being over-constrained thus fail in a ‘real-world’ environment and would thus not be useful to generic information management applications.

In essence, one can say that, from the perspective of information management and content-based retrieval, existing systems lie at two extremes, neither of which provides a satisfactory solution to the problems faced in these applications. True-to-life models present an overly complex solution while loose black-box approximations are insufficiently general. It is therefore logical that a compromise between these two
extremes would provide the most appropriate solution to the problem of auditory source separation in the context of information management applications.

1.5. Scope
In order to substantiate or refute the hypothesis it is necessary to develop an audio representation and demonstrate its suitability for content-based coding and information management. Specifically, it must be shown that, at least in principle, the representation can implicitly support:

1) Compressed data storage;
2) Audio content description; and
3) Auditory source separation.

A thorough investigation of each one of these aspects represents a major undertaking, hence, it was necessary to limit the scope of the investigation because of time and resource limitations. The last of the aspects mentioned above has received the least research attention from the audio information management community, despite being perhaps the most pressing hindrance to the development of a truly useful and general audio information management system. As such, this aspect was selected to be the focus of the investigation reported here. Nevertheless, in order to ensure that support for auditory source separation did not impede the representation’s support for the other two required factors, it was necessary to perform initial investigations into these aspects as well.

The specific boundaries of the investigation undertaken were:

1) **Overall design goals.** The overriding concern in the design of the representation and source separation algorithms was to achieve a balance between the needs of information management and those of computational auditory scene analysis. Computational efficiency is one concern of importance in information management while perceptual accuracy is of greater import in CASA. Hence, in deciding between basic techniques, it was necessary to determine which of the two concerns played the overriding role for the particular aspect in question and then to select the technique that most adequately addressed that concern.

2) **Audio representation design.** Various existing audio data representation schemes were investigated to determine their suitability and support for each of the three required factors mentioned earlier. From this investigation it
was determined that sinusoidal representation would be appropriate. However, it was necessary to design a variation of a sinusoidal coder in order for these criteria to be met. The resultant design and its justification appear in Chapter 3.

3) **Auditory source separation.** The area of auditory source separation is very much in the early stages of development. Further, application of auditory source separation to audio information management and content-based coding is virtually unknown. Hence, the main focus of this thesis is to investigate the suitability of the developed representation to auditory source separation. To this end, a number of algorithms were developed and compared against one another.

Given the magnitude of the task, and the paucity of previous work in the area, the aims of the investigation were modest with a “proof of concept” being sought. To this end, the system was required to separate simple two source mixtures of various sounds (male and female speech, artificial tonal and chirped signals, and notes played by various orchestral instruments). A “successful” separation was determined to have occurred if the reconstructed signal was intelligible and the source identifiable (i.e. male or female speaker, instrument).

### 1.6. Statement of Contribution

A key feature of this thesis is its cross-disciplinary approach to solving the stated research problem with the primary contribution being in the field of CASA while being motivated by the needs of content-based coding and information management of audio data. In pursuing this goal, a number of contributions have been made to the fields of audio coding and information management as well as CASA. The significance of these contributions will be presented in the remainder of this section.

#### 1.6.1. New Approach to Computational Auditory Scene Analysis

The approach to auditory source separation that is reported in this thesis adopts a top-level architecture similar to most existing CASA systems. However many of the details of individual components are unique. Most importantly, the track grouping algorithms and subsequent track shape refinement represent a significantly new approach to the problem. Another key contribution is the underlying representation itself, which is designed to serve as the foundation for the scene analysis and coding and information management as a parallel concerns. The reasons for, and significance of, this dual focus
were discussed in the previous sub-section. From the perspective of CASA research, this has resulted in new algorithms for the field.

The most significant contribution to the field of CASA is a consequence of all the abovementioned considerations, and lies in the overall approach employed. The various proposed grouping methods represent analytic approximations of the most important heuristic principles proposed by Bregman [1] to be responsible for the task in the human auditory system. While others have attempted to apply these heuristic principles, they have generally used heuristic rule-based approaches and abandoned them after only initial investigation. Further, the manner in which feedback is employed in the system is significantly different from other systems that employ some feedback mechanism. Specifically, most existing CASA systems that employ feedback do so from “high-level” processes that model the cognitive process back to the low-level track and group formation stages. In contrast, the feedback employed in the system presented in this thesis is within the mid-level grouping stage back to the low-level tracking stage. While it is recognised that some high-level cognitive process will be required to refine groupings in certain applications as well as to form true “auditory streams”, it is contended that for the purposes of general content-based audio coding, such feedback may not be necessary and may, in fact, cause a loss of generality.

1.6.2. Secondary and specific contributions

A number of specific contributions are presented in this thesis. These are termed “secondary contributions,” not because of their significance, but rather because they are in support of either or both of the two main contributions. Table 1-1 summarises each of the specific contributions, refers to where in the thesis details of the contribution are located and gives a brief statement of its significance.

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Significance</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture for CASA</td>
<td>The particular architecture adopted in this thesis is distinct from previous attempts in the manner in which feedback is employed.</td>
<td>Figure 3-2, p. 112</td>
</tr>
<tr>
<td>Overlapped filterbank</td>
<td>The full-band overlap filterbank resolution is novel for a CASA system.</td>
<td>Figure 3-4, p.117</td>
</tr>
<tr>
<td>Contribution</td>
<td>Significance</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Final filterbank resolution</strong></td>
<td>The final filterbank resolution, and the practical justification for the band boundaries is novel in a CASA system.</td>
<td>Table 3-3, p. 120</td>
</tr>
<tr>
<td><strong>Effect of transform choice and phase considerations on peak extraction.</strong></td>
<td>The effect that the choice of transform, and in particular the obfuscation of phase information, has on peak picking performance is an issue that, to the author’s knowledge, has not been explored by either the sinusoidal coding or CASA communities.</td>
<td>Section 3.4.4, pp. 122-126</td>
</tr>
<tr>
<td><strong>Peak picking algorithm</strong></td>
<td>A new compromise between the McAulay and Quatieri approach and that of Terhardt.</td>
<td>Equation 3 – 20, p. 135</td>
</tr>
<tr>
<td><strong>First order prediction for peak tracking</strong></td>
<td>Although the concept of prediction has been mentioned by Cooke [118], extending the algorithm of McAulay and Quatieri to include first order prediction is, to the author’s knowledge, novel for CASA work.</td>
<td>Section 3.7.3, p.142</td>
</tr>
<tr>
<td><strong>Observation of optimum transform resolution with respect to signal characteristics</strong></td>
<td>CASA researchers do not traditionally cite the observation that the optimum transform resolution is dependent on the actual signal characteristics. This is understandable from the point of view of remaining true to the physical system; however, it is undesirable from the point of view of information management.</td>
<td>Section 3.8.1, p. 143</td>
</tr>
<tr>
<td><strong>Band consolidation procedures</strong></td>
<td>The need to consolidate the data from the various bands is unique to the overlapped band approach taken in this thesis and, as a result, so are the solutions to this.</td>
<td>Sections 3.8.2, 3.8.3 and 3.8.4, pp. 145-149</td>
</tr>
<tr>
<td><strong>Modified Hough transform-based spline fitting proposal</strong></td>
<td>The initial proposal for track shape approximation using a modified Hough transform is a completely unique procedure. While determined to be impractical in its current design, the concept warrants further investigation.</td>
<td>Section 3.8.5, p. 151</td>
</tr>
<tr>
<td><strong>Least-squares spline fitting</strong></td>
<td>The application of least-squares spline fitting for track shape refinement is novel in both the CASA and sinusoidal coding domains.</td>
<td>Section 3.8.6, p. 154</td>
</tr>
<tr>
<td><strong>Model track calculation based on grouped track estimates</strong></td>
<td>The notion of generating a model track based on grouped track estimates is different from the traditional view of attempting to generate a fundamental from all available peak or track data in order to generate and/or group the tracks.</td>
<td>Section 3.8.7, p. 161</td>
</tr>
<tr>
<td>Contribution</td>
<td>Significance</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Algorithm for final track extraction</td>
<td>The concept of using the model to derive the final set of track shapes is somewhat similar to some traditional CASA approaches. However, the extended algorithm developed to achieve this is unique.</td>
<td>Section 3.9.2 p. 168</td>
</tr>
<tr>
<td>Quantitative partial-based evaluation of grouping performance</td>
<td>Evaluating the accuracy of grouping based on the initial track estimates without requiring access to the separated source data is an entirely novel approach.</td>
<td>Section 4.2.1, p. 177</td>
</tr>
<tr>
<td>Grouping evaluation metrics</td>
<td>The proposal of the Hausdorff distance as one of the accuracy metrics is particularly significant in that it is classically viewed as an image, rather than audio, processing tool. While some of the other metrics are similar to those employed by others, none are identical to previous work and thus represent a contribution.</td>
<td>Table 4-1 and Table 4-2 pp. 179-180</td>
</tr>
<tr>
<td>Technique for pair-wise comparison of algorithms</td>
<td>This method of comparison is also unique to the present work.</td>
<td>Figure 4-7, p. 187</td>
</tr>
<tr>
<td>Model-based frequency contour grouping</td>
<td>This algorithm bears more similarity to image classification techniques than to any previous audio processing or CASA work.</td>
<td>Section 4.4, p. 194</td>
</tr>
<tr>
<td>Automatic threshold calculation for distance-vector based grouping</td>
<td>The method developed for determining the threshold in the distance-vector algorithm is unique to the present work.</td>
<td>Section 4.5.2, p. 199</td>
</tr>
<tr>
<td>Shape normalisation for contour-based grouping</td>
<td>Although scaling track contours to a common mean frequency is a popular technique in CASA research, shape normalisation has, to the author’s knowledge, never been employed in CASA work.</td>
<td>Section 4.6.2, p. 204</td>
</tr>
<tr>
<td>Method to account for onset and offset transients</td>
<td>Pre-processing the track shape estimates to remove onset and offset transients is a novel approach.</td>
<td>Section 4.6.4, p. 206</td>
</tr>
<tr>
<td>Shape parameters for contour-based grouping</td>
<td>The shape parameters which were derived as a comparison metric for the track contours are entirely novel.</td>
<td>Equations 4 – 20 to 4 – 29, pp. 218-219</td>
</tr>
<tr>
<td>Application of Fourier shape descriptors to contour-based grouping</td>
<td>Fourier shape descriptors have traditionally found application in image processing. Their application here to the track-grouping problem is unique.</td>
<td>Section 4.6.7, p. 221</td>
</tr>
<tr>
<td>Contribution</td>
<td>Significance</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Proportionally spaced histogram</td>
<td>The histogram developed for harmonicity grouping used proportionally spaced bins to mirror auditory frequency resolution. This is a novel method for generating a histogram.</td>
<td>Section 5.5.1, p. 233</td>
</tr>
<tr>
<td>Modification of Euclid’s algorithm to determine the harmonicity of two frequencies.</td>
<td>Extending the classical algorithm for determining the highest common factor of two numbers to operate on real numbers, instead of integers, is novel.</td>
<td>Section 5.5.2, p. 234</td>
</tr>
<tr>
<td>Bregman inspired fundamental frequency calculation</td>
<td>To the author’s knowledge, no previous implementation exists of Bregman’s loosely sketched hypothesis of pitch perception.</td>
<td>Section 5.5.3, p. 235</td>
</tr>
<tr>
<td>Track-based harmonic number calculation</td>
<td>The extension of the Bregman technique to determine the harmonic number of each track in a global fashion is novel since traditionally this problem is approached on a localised frame-by-frame basis.</td>
<td>Section 5.6 p. 239</td>
</tr>
<tr>
<td>Application of the method of continuing fractions to harmonicity-based grouping</td>
<td>While some researchers have used a lookup table method to determine whether two frequencies are harmonically related, application of the method of continuing fractions to solve this problem is novel.</td>
<td>Section 5.7.2, p. 242</td>
</tr>
<tr>
<td>Set-theory based combination of individual dimension grouping results</td>
<td>The set intersection based method for combining the grouping results along each individual dimension (frequency contour, amplitude contour and harmonicity) is novel.</td>
<td>Section 6.3, p. 250</td>
</tr>
<tr>
<td>Algorithms for source type identification</td>
<td>The algorithms developed for source type identification both serve as an initial validation of the audio information management potential of the representation that has been developed as well as contributing to the limited body of research in this field.</td>
<td>Section 7.3, p. 266</td>
</tr>
</tbody>
</table>

1.7. Outline of this thesis
The remainder of this thesis is organised into four broad sections as follows:

- **Background information**: Chapter 2 presents the relevant theory in auditory perception as well as a review of existing work in the areas of audio information management, audio coding, and auditory source separation.

- **Audio representation**: Chapter 3 describes the representation developed and details its extraction from raw audio.
• **Perceptual grouping:** Chapters 4-6 describe the various algorithms developed to perform perceptual grouping using the representation and the experimental results derived for each algorithm.

• **Conclusions and directions for future work:** Chapter 7 explores the potential applications of the representation to audio information management and content-based coding. Final conclusions are also drawn.

### 1.8. Summary

It has been shown that the hypermedia model proposed by Vannevar Bush in 1945 is a powerful information management tool. It has also been noted that the current situation for audio information management and coding falls far short of this model. It was stated that the main reason for this is the traditional reliance on statistical representations that compress the data at the expense of obfuscating the structure of the underlying audio information. This led to a formal definition of the problem to which the work reported here seeks to offer a solution. Finally, it was hypothesised that a perceptually based audio representation, in conjunction with algorithms for auditory source separation, would provide the necessary basis to solve the problem.
Audio Source Separation... for Content-based Coding and Management
Chapter 2

Background Theory and Previous Work

2.1. Introduction
As indicated in the statement of significance in the previous chapter, the work reported in this thesis is of a cross-disciplinary nature. As such the prerequisite background knowledge is drawn from a wide range of fields. This chapter begins with a presentation of the background material from the basics of human auditory perception through the theory of auditory scene analysis and finally on to the relationships between the physical and perceptual characteristics of an audio signal. Having laid this basic theoretical foundation, existing works in the areas of audio signal analysis (including auditory source separation), audio coding and audio information management are then critically reviewed. Thus both the theoretical basis and the motivation for the current work are presented.

2.2. Auditory Perception

2.2.1. Introduction
As indicated by the title of this thesis, the auditory source separation algorithms presented are based on principles of auditory perception. Accordingly, this section provides the relevant background details of the human audio perceptual process beginning with the anatomy of the auditory system moving onto the physiology and psychology of audition and concluding with a description of the postulated mid-level mental auditory representation. This mid-level auditory representation forms the basis of the structured audio representation reported in this thesis.

2.2.2. The Anatomy of Audition
The peripheral auditory system consists of three parts: the outer, middle and inner ear. The pinna and auditory canal make up the outer ear. Their primary function is simply to channel sounds into the later stages, however, the pinna also performs some filtering of the incoming signal and is partly responsible for localisation of sounds. The middle ear, consisting of three bones known collectively as the ossicles, effects an impedance match between the air filled auditory canal and the fluid filled cochlea. The cochlea is responsible for transduction of the mechanical audio stimulus into electrical nerve impulses. These impulses are transmitted to the cochlear nucleus in the brainstem by the

---

1 The primary source for material in this section is [8]
auditory nerve. The function of the cochlea is of most relevance to the work described in this thesis.

The cochlea is filled with almost incompressible fluid, has rigid bony walls and is divided along its length by two membranes: Reissner’s membrane and the Basilar Membrane. The motion of the stapes against the oval window causes a corresponding pressure wave in the fluid of the cochlea that in turn induces oscillations along the Basilar Membrane. Because of the physical characteristics of the Basilar Membrane, a travelling wave is set up that has maxima determined by the frequency content of the exciting pressure wave. The motion of the Basilar Membrane causes a corresponding motion in the hair cells which, in turn, excite the Spiral Ganglion cells resulting in the conversion of mechanical energy into electrical impulses.
2.2.3. **The Physiology of Audition**

*Signal detection and loudness*

The outer ear and Basilar Membrane have a considerable filtering effect on the incoming audio signal while the hair cells in the cochlea more or less evenly distributed. As a result, the minimum detectable sound pressure level\(^1\) (SPL) is dependent upon frequency. A plot of the minimum detectable SPL for pure tones is known as the absolute threshold of hearing. Although absolute thresholds vary from person to person and performance of individuals deteriorates with age, curves derived from large sample populations do exist and are in general agreement in terms of their basic form. An example of an absolute threshold curve is shown in Figure 2-3.

![Figure 2-3: Absolute threshold of hearing](Based on ISO 1961 from table 14.1 in [9].)

A related phenomenon is that pure tones of different frequencies need to be at different intensities in order to be judged equally as loud as a reference tone of 1000 kHz at a given SPL. Plots derived using this technique are known as equal loudness contours. The unit of loudness used is the ‘phon,’ which is equal to the SPL (in dB) of the 1000 kHz test tone used for that particular equal loudness contour. An example of equal loudness contours is given in Figure 2-4.

---

\(^1\) For a definition of sound pressure level see Appendix A
Yet another factor influencing the absolute threshold of hearing is signal duration. As a signal increases in duration the minimum audible SPL decreases, reaching a lower limit around 200ms duration. The obvious conclusion is that there is a temporal integration effect at some point in the auditory system with a window length of approximately 200ms. However, since the peripheral auditory system contains no storage mechanism, this must be due to some higher-level neural action [9].

**Just noticeable differences and Weber’s law**

Psychoacoustic data on the sensitivity of the auditory system are often stated in terms of the smallest detectable variation of a single physical characteristic (e.g., frequency, intensity or duration). These values are known as just noticeable differences (JND). For most signal characteristics, the JND is a constant proportion of the magnitude [9]:

\[ \Delta x = kx \]  

where \( x \) is the magnitude of the characteristic and \( k \) is a constant known as the Weber fraction. This relationship holds under many circumstances and even when it does not (intensity discrimination of pure tones improves at high intensities, for example) the deviation is only slight. For example, the value of the Weber fraction for intensity is [9]:

\[ k = -3.344 \, \text{dB} - 0.072 \times \text{SL} \]  

where SL is the sensation level\(^1\) of the tone.

---

\(^1\) The definition of sensation level appears in Appendix A.
Auditory filters, the critical band and frequency resolution

The action of the Basilar Membrane is responsible for many of the effects of interest to the work proposed here. Firstly, the position of maximum displacement is dependent on the frequency of the stimulus:

\[
W(g) = (1 + \rho g) e^{\rho g}
\]  \hspace{1cm} (2-3)

where \( g \) is the normalised frequency given by:

\[
g = \frac{|f - f_c|}{f_c}
\]  \hspace{1cm} (2-4)

where \( f_c \) is the centre frequency of the particular filter under consideration. Several variants of the \( roex(p) \) function have been suggested to account for results obtained from various experimental procedures.

The parameter, \( \rho \), in equation 2-3 determines the roll-off in the stop bands of the auditory filter and is determined experimentally. Depending on the experimental procedure used to estimate the auditory filter shape of a group of subjects, the value of this parameter may be equal for both the low and high frequency stop bands or it may be different so that the shape of the auditory filter response is asymmetrical. \( \rho \) is related to the equivalent rectangular bandwidth (ERB) of the filter by the relationship:
Glasberg and Moore [13] found that a good approximation for the ERB of each of the auditory filters is given by:

\[
\text{ERB} = 24.7 + 1.08 f_c
\]

(2 — 6)

where \(f_c\) is the centre frequency of the filter in Hz. Given equations 2 — 3 to 2 — 6, the response of each auditory filter may be calculated. Several of these responses are plotted in Figure 2-6. It may be noted that, consistent with the claim that the Basilar Membrane acts as a band of constant-Q band-pass filters, the bandwidth of each auditory filter increases with increasing centre frequency.

As a result of the action of the Basilar Membrane, the frequency resolution of the auditory system approximates Weber’s law: at low frequencies, the frequency resolution is fine while, at high frequencies, the frequency resolution is coarse. This is shown in the following relationship[9]:

\[
\log(\Delta F) = a \sqrt{F} + k + \left( \frac{m}{\text{SL}} \right)
\]

(2 — 7)

where \(a, m,\) and \(k\) are constants, \(F\) is the frequency of the tone and \(\text{SL}\) is the sensation level of the tone.
The critical band theory has resulted in the development of a perceptual frequency scale, known as the bark scale. One bark is equal to the width of one critical band. Several researchers have proposed mappings between the physical frequency scale (measured in Hz) and the bark scale, one example is that defined by Zwicker[14]:

\[ z = \frac{26.81f}{(1960 + f)} - 0.53 \]  

(2-8)

where \( f \) is the physical frequency in Hz and \( z \) is the bark scale frequency in barks.

**Simultaneous masking**
The response of the Basilar Membrane causes an effect known as simultaneous masking. Masking refers to the inaudibility of one signal in the presence of another (masking) signal. If two tones of differing frequency and intensity are presented simultaneously and one tone (signal) is at a lower sound pressure level than the other (masker), the lower level tone may be rendered inaudible. The condition for the signal to be inaudible is that it must fall within the masking pattern of the masker. The basic shape of a masking pattern is shown in Figure 2-7. Note that in general a tone will be better at masking tones whose frequencies are higher than itself.

![Masking pattern for a tone](image)

**Figure 2-7: Masking pattern for a tone**
The masking pattern is superimposed over a tone such that 0 bark corresponds to the frequency of the tone and 0 dB corresponds to the magnitude of the tone. All neighbouring tones that fall below the curve are rendered inaudible (masked).

The mechanism for masking can be described in terms of swamping. The output of the auditory filters effectively drives the transduction mechanism. Since the auditory filters
are bandpass and overlapping, the masking signal will not only pass through the filter with centre frequency corresponding to its frequency but attenuated versions of the masker will also pass through the neighbouring filters. If the neural activity induced by the masking signal in the band corresponding to the test signal is greater than that induced by the test signal itself, the detecting neurones in this region are said to be swamped and the test tone will remain undetected. Hence, given the response of the auditory filters (see equation 2 – 3), one method for deriving the masking pattern of a signal at a given frequency is to determine the responses of each of the auditory filters to the tone. The magnitude of the response of each filter at the frequency of interest is transposed back to the centre frequency of that filter. The values thus obtained may be interpolated to generate the masking pattern. This procedure is illustrated in Figure 2-8. In this example the auditory filters are assumed to be symmetrical. Nevertheless, it is apparent that the masking pattern is still asymmetrical. This can be attributed to the increasing bandwidth of the auditory filters.

![Figure 2-8: Masking pattern calculation for a tone at 1 kHz](image)

The dashed lines show the auditory filter responses in the region about 1kHz. Translating the response of each filter at 1 kHz to the centre frequency of that filter gives the auditory nerve excitation pattern induced by the 1kHz tone. Any tones within the frequency range of the masking curve must produce a greater excitation level to be perceived. Thus the excitation pattern shown can be interpreted as the masking curve for the 1 kHz tone. (After [12].)

In practice the derivation of masking curves and filter responses happens in the reverse order to that just described. Masking curves are determined experimentally, by presenting subjects with a series of test signals and maskers. The general technique
involves the subject adjusting the volume of the test signal to the point where it just “disappears” below the masking signal. Both the masker and test signal can be either narrowband noise or a pure tone with slightly different results depending on the combination selected. Other factors that influence the auditory filter shapes derived from such experiments include the presentation level of the masking signal and the health of subjects tested.

2.2.4. **Mid-level auditory signal representation**
The overall effect of the Basilar Membrane can be viewed as transforming the temporal waveform of the stimulus into a 3-dimensional representation. The dimensions of this representation are time, frequency and amplitude. Such a representation is often referred to as a time frequency distribution (TFD). The most important factors to note are that the three axes have non-linear scales determined by the resolution principles mentioned in the above sections and that the amplitude peaks in this space are of the greatest importance since they correspond to increased rates of neural firings. Further, it has been postulated that the perceptual system achieves its organisation of the incoming data by tracking the peaks of this time-frequency distribution through time, frequency and amplitude [1].

![Spectrogram of a mixed source signal](image)

**Figure 2-9:** Spectrogram of a mixed source signal

A ‘track’ is a time-frequency-amplitude contour traced over the ‘adjacent’ peaks in the TFD. For example, in Figure 2-9 the red dashed lines indicate two possible track trajectories through time and frequency. The intensity of the grey gives a rough guide to the amplitude contour of the tracks.
Although the basic track is sufficient to describe the spectrogram shown in Figure 2-9, there is an obvious lack of support for noise-like or percussive sounds. In support of this observation, the work of Suga [15] suggests that the description should be expanded to include three primitive elements. Suga noted that the basic elements in the communication sounds of advanced animals (including human speech sounds) are tone bursts, noise bursts and frequency sweeps. It follows naturally that the auditory systems of these creatures would be adapted to detect the same basic signals. He referred to these elements as information bearing elements (IBE) and described them in terms of information bearing parameters (IBP). Further, it was noted that individual classes of neurones exist that respond to each of these elements. Hence we may postulate a mid-level auditory representation as shown in Figure 2-10.

![Figure 2-10: Mid-level auditory signal representation](image)

### 2.3. Auditory Scene Analysis

#### 2.3.1. Introduction

Given the set of basic elements that comprise the mid-level auditory representation described in the previous section, the auditory system attempts to make sense of the pressure waves that originally struck the tympanic membrane. It does this with amazing accuracy despite the fact that most of the time it is presented with a cacophony of diverse sounds. In his seminal work, Al Bregman succinctly stated the scope of this problem [1]:

> Two narrow channels are dug up from the side of a lake. Each is a few feet long and a few inches wide and they are spaced a few feet apart. A handkerchief is stretched across the channel halfway up each one. Solely from the motions of the handkerchiefs you are to answer a series of questions: How many boats are there on the lake and where are they? Which is the most powerful one? Which is the closer? Is the wind blowing? Has any large object been dropped suddenly into the lake? The task seems impossible yet it is a precise analogy for what the auditory system performs so well that we are rarely, if ever, conscious of the effort involved.
2.3.2. **Definition of Auditory Stream Segregation**

According to Bregman, the solution of the above problem involves reducing the auditory stream into a series of auditory streams. Bregman defines a stream as a “perceptual unit that represents a single happening [1].” Figure 2-11 gives a very elementary example of stream segregation.

![Figure 2-11: Elementary example of auditory stream segregation](image)

From Figure 2-11 it is evident that a stream corresponds to a perceptually derived group of partials. Given that the stream is simply a perceptually imposed organisation of the partials it is not necessary that each stream encapsulate discrete sounds: the individual notes of the music melody fuse into a single stream, for example. Thus it would appear that a perceptual stream corresponds to a single acoustical event.

Notwithstanding the considerations in the previous paragraph, it is obvious that a system seeking to mimic the process of stream segregation must first break the incoming audio into a series of partials. A perceptually motivated organisation must then be imposed upon these partials. While the formation of composite streams, that is those that consist of several discrete events, will require higher level knowledge and multiple feedback paths, as is the case in the human auditory system, it is the aim of this thesis to show that the formation of the discrete units can, to a large extent, be achieved automatically.

2.3.3. **Principles Governing Auditory Stream Formation**

The previous section stated that that stream formation involves decomposing the incoming signal into its constituent partials and then grouping them into a number of streams. These streams are formed using a parallel process with many feed forward and feed back paths that select the best organisation from a number of competing possibilities. There are a number of general principles guiding this process including:

- **Belongingness:** All partials must be assigned to a stream.

- **Exclusive allocation:** All partials can be assigned to only one stream. This principle exploits the fact that in a natural acoustical environment, it is unlikely that one sound will end at precisely the same time as another starts. Therefore,
once a partial has been assigned to a stream, the presence of that partial is assumed to be due to that stream as long as the stream is in existence.

However, there is evidence that this principle does not apply to primitive grouping, that is, the process being modelled in the work described in this thesis. Liberman [16] reports that when a synthetic three formant syllable is presented such that the first two formants are presented to one ear and the third is played simultaneously in the other ear, the subjects hear the syllable in the first ear as though all three partials were presented to that ear. However, they also hear a chirp in the second ear, corresponding to the third formant that was presented to that ear. Thus the principle of exclusive allocation is broken.

There is also a valid computational reason not to insist on this principle being strictly applied. As has already been stated, it is common for some portions of an audio scene to be partially masked by others and which object constitutes the foreground depends largely on attention. In this situation, the principle of exclusive allocation makes sense. However, in seeking to decompose the scene into a series of auditory objects that are candidates for nodes in a hyperaudio system, each object must be complete and self-contained. Thus, those partials that would normally be grouped differently based on the attention of the user, must be allocated to every possible group to which they could belong.

- **Continuity**: Partialis are grouped to favour gradual changes rather than abrupt changes along the stream.

- **Closure**: Gaps in partials tend to be filled when there is evidence that they have been masked. The presence of a potential mask is vital in causing this effect. As with the principle of continuity, faced with the choice of two distinct acoustic events that end and begin abruptly or a single continuous, masked event, the auditory system will favour the latter. The same principle holds in the visual system so the graphical example in Figure 2-12 provides a good illustration.

![Figure 2-12: The principle of closure](image)
• **Proximity:** Partials in close temporal or spectral proximity tend to be fused into the same stream. A stream will be formed along the dimension that holds the strongest influence in the particular situation. In the illusion of polyphony mentioned earlier, for example, the individual events (notes) are happening so closely together that the wide frequency separation between successive events and relatively shorter temporal separation between alternating events is sufficient to induce the perception of two streams. It is possible to slow down such a sequence sufficiently to cause two streams to fuse into one. If the sequence is slowed further still, the stream will break up into a series of isolated acoustic events.

• **Similarity:** Partials that are similar in some aspect tend to be fused into a single stream. The similar aspect is generally amplitude or frequency modulation. Again, it is unlikely in a natural environment that two distinct sources will give rise to partials that display the same modulation characteristics.

• **Common fate:** partials that begin and end at the same time and vary in the same manner tend to be fused into a single stream.

![Figure 2-13:](image)

**Figure 2-13:** Effect of rate of presentation and frequency separation on streaming of two alternating tones

The duration of each tone is 40 msec. Streaming will always occur in the “always segregated” region and will never occur in the “always coherent” region. Whether streaming occurs in the ambiguous region primarily depends on attention. That is, it is possible to consciously switch between the perception of a single coherent stream or two discrete streams. (From [17].)
The influence that each of these principles has over the selection of the best grouping depends on a number of contextual factors. These include the rate of presentation, the degree of similarity between successive partials and the previously assigned groupings. As an example, the two instruments from one illusion mentioned earlier is highly dependent on the speed at which the passage is played and the separation of the two alternating melody lines. The slower the rate of presentation, the greater the frequency separation required for the illusion to occur. There are thresholds for both frequency separation and temporal separation below which it is virtually impossible to induce this type of streaming. Figure 2-13 illustrates these thresholds.

2.3.4. **Summary**

The process of auditory scene analysis is one of organisation with the eventual result being that a signal representing a cacophony of noises is transformed into a percept of clearly distinct auditory events. The most important aspect of this process is that, to a large extent, it is possible for the auditory system to do this without necessarily recognising what each of the objects is. For example, it is not an uncommon experience to hear an unusual sound and wonder what the source of this sound is. This factor leads to the conclusion that an automatic process, free of any semantic analysis, should be able to mimic the process of auditory scene analysis with a mixed recording being transformed into a description of the auditory scene in terms of the number of objects that appear. The next section details how one might add depth to this description.

2.4. **Audio’s Perceptual Characteristics**

While being able to decompose a mixed recording into a collection of individual auditory objects is a noteworthy achievement, of itself this decomposition is of little use from an information management or retrieval perspective. For that we need to have some way of describing the individual objects. One way would be to perform a semantic analysis, either manually or automatically, of each of the individual objects and compiling an index accordingly. However, as was mentioned in the first chapter of this thesis, there are many shortcomings to this approach, not the least of which is tedium or computational expense. Another method entails performing an automatic analysis of the raw physical characteristics of each object. Once again, the disadvantages of this approach, including limited utility for unskilled users, been already been discussed. By far the preferred approach is to describe the objects in familiar terms that describe how the objects are perceived.
This section aims to answer the question: What, if any, relationships exist between the physical characteristics of an audio signal and the percept they induce? This is a significant question because although these relationships must exist for audio perception to be possible, they are often complicated and, in some cases, not very well understood. For example, the same tone complex may produce different pitch sensations when presented under different conditions (see section 2.4.2). The question is also significant from the perspective of the work presented here in that it gives insight into the essential features of audio that should be readily apparent in an audio representation if it is to provide direct support for content-based retrieval using perceptually relevant index keys.

Before these issues can be discussed, three terms must be defined. The physical characteristics of an audio signal are those intrinsic characteristics that can be readily measured using electromechanical means and objective scales. Examples of physical characteristics include the frequency content at a particular instant in time or the amplitude envelope of the signal. Perceptual characteristics are the fundamental, subjective properties judged by a human listener in response to an audio stimulus. Audio signals possess three perceptual characteristics: loudness, pitch and timbre. Adding subjective interpretation and temporal variation to these perceptual characteristics gives rise to perceptual attributes. Examples of perceptual attributes include melody and rhythm.

In this section a definition of each perceptual characteristic as well as some perceptual attributes will be given. This will be followed by a discussion of the correlation between the characteristic or attribute and relevant physical and/or perceptual characteristics. Also, a review of existing techniques for extracting information about the perceptual characteristic or attribute from a physical audio signal will be given when appropriate.

### 2.4.1. Definitions

Three perceptual characteristics and three perceptual attributes of audio will be discussed. The definitions of each of these attributes is provided here:

- **Loudness** is the subjective intensity of a sound [18].
- **Pitch** is the apparent predominant frequency of a sound [19].
- **Melody** is the temporal variation of pitch.
- **Rhythm** implies a repeating pattern in time or space. In the case of sound, such patterns occur in time and are constructed by varying the relative duration of a series of acoustic events.
• **Timbre** is a vaguely defined quality. Bregman, for example, gives the following definition:

> Timbre tends to be the psychoacoustician’s multidimensional wastebasket category for everything that cannot be labelled as pitch or loudness...[17]

Essentially, timbre is that aspect of an audio signal that allows us to distinguish between two different sounds that have the same pitch and loudness. For example, it’s what makes the individual instruments of the orchestra sound different from one another. Timbre is often referred to as tone colour or tone quality.

• **Sentics** is an analytical study of human emotion first introduced by Clynes [20]. It can be used to describe the emotive information conveyed by music.

### 2.4.2. Loudness

Loudness is related to the physical intensity and frequency content of a sound as well as the level of background noise. Duration also effects loudness for very short sounds (less than 200ms) because of temporal integration effects in the auditory system. However, there is some controversy on the matter of finding an absolute loudness scale that relates physical intensity directly to loudness [21]. Stephens [22] suggested a power function relationship:

\[
L = kI^x \text{ sones}
\]  

(2 — 9)

where \( L \) is the loudness, \( k \) is a constant depending on the experimental conditions as well as the units of intensity used and \( I \) is the physical intensity of the sound. The unit of loudness thus defined is the sone. One sone is arbitrarily defined to be the loudness of a 1000 Hz tone presented at 40 dB SPL. There is some debate as to the value of the power, \( x \), and modifications are required to fit the curve to the empirical data at low levels of loudness. Also, this power relationship holds only for pure steady state tones of duration greater than 200ms. More complicated models are required to calculate the loudness of complex stimuli [8].

Besides the technical problems of finding a single model to reliably fit experimental data for loudness, there are a number of theoretical objections to finding such a scale [8]. One objection is that an experimental subject’s evaluation of a sound as being ‘twice as loud’ as another is based on a personal scale that is not guaranteed to be linear. Another objection is that, in everyday experience, we tend to judge the loudness of sounds based
on the context in which they are heard in relation to what, if anything, is known about the source of the sound.

**Techniques for evaluating loudness**

One method of obtaining a numerical value for loudness is based on an addition of the perceptually weighted sound intensity of each frequency component in the test signal. The weighting approximates the equal loudness contours shown in Figure 2-4. Such a method would probably be best suited for loudness comparisons for retrieval. However, as is evident from the previous discussion, numerical loudness values are not likely to be of much use as query possibilities. Even statements along the lines of ‘this sound is x times louder than that sound’ have little chance of meaning the same thing to two different individuals. Therefore, the only realistic query terms are ‘louder than’ and ‘softer than’; hence, the calculation and comparison of loudness need only be very approximate.

2.4.3. **Pitch**

Frequency content is the most directly related physical characteristic to pitch. However, the relationship between frequency and pitch is not entirely straightforward. In the case of pure tones, intensity can also play a role in pitch perception. Also, the perceived pitch may not always correspond to any of the actual frequencies present in the signal for complex sounds.

The pitch of a pure tone is determined almost entirely by its frequency, however, intensity also plays a minor role. The exact magnitude and direction of the pitch shift can vary considerably amongst individuals, although the general trend indicates that pitch decreases with increasing level for tones below 2 kHz. That is, a given tone at a fixed frequency will be perceived to have a pitch that decreases as its presentation level is increased. The opposite relationship holds for tones above 4 kHz [8].

An interesting phenomenon in the perception of complex sounds is that of the missing fundamental. If a harmonic tone complex (1800 Hz, 2100 Hz and 2400 Hz, for example) is presented the actual pitch perceived may be that of the missing fundamental (300 Hz). This pitch was termed residue pitch by Schouten [23] and is also known as periodicity pitch, virtual pitch and low pitch [8]. Yet another interesting phenomenon occurs when inharmonic complexes are presented. To illustrate, imagine a 2030 Hz carrier tone that is amplitude modulated by a 200 Hz tone. The result is a complex tone containing the frequencies 1830 Hz, 2030 Hz and 2230 Hz. The perceived pitch will be close to 203 Hz [24].
Models explaining perceived pitch are based on two basic premises: a place theory and a temporal theory. The place theory suggests that perceived pitch is due to the overall pattern of vibration of the Basilar Membrane in response to the stimulus. The temporal theory postulates that perceived pitch is a result of the pattern of induced neuron firings in the auditory nerve. According to the place theory, it is the overall envelope of the stimulus that is most important in pitch perception while the temporal theory asserts that the fine structure of the stimulus waveform is most significant. Neither theory is adequate to explain all observed phenomena in pitch perception.

**Techniques for pitch detection**

Many algorithms exist for pitch detection of speech signals because many speech coding techniques require accurate pitch estimation [25]. Many of these algorithms are sufficiently general to be readily applied to general audio signals. A brief review of some of the more important techniques will be presented here.

Classical pitch detection algorithms operate in one of three domains: time, frequency or time-frequency. The simplest time domain method involves analysis of the distance between peaks of the waveform [26]. This method assumes that the signal is periodic and hence has limited application to tonal, or in the case of speech, voiced, sounds. This technique can be very error prone since, even over short intervals, a speech signal, and almost any other naturally occurring sound for that matter, is not exactly periodic. Gold and Rabiner developed an improved method by selecting the best estimate from six such pitch estimators running in parallel, each driven by a different peak distance measured from the signal [27].

A more robust time domain pitch detection method is autocorrelation over a short signal window:

\[
X(m) = \sum_{n=0}^{N-1} x(n) x(m - n), \forall \ m = 0, 1, 2, ..., (2N - 1)
\]  

(2 — 10)

This function will display a peak at the delay, \(m\), which represents the signal period. For aperiodic signals, \(m = 0\). While this method is more accurate it displays fairly high computational complexity. Ross et al developed a less complex variation of this technique. The authors asserted that the convolution only needed to be performed for 8-9ms length windows every 20ms. They also proposed a function to simplify the convolution. This was based on amplitude differences that only required additions in place of the multiplication in equation (2 — 10) [28].
Frequency domain pitch extraction methods use the STFT of the signal as the basis for pitch determination. Some of the simplest methods in this domain involve looking at the peaks in the frequency domain and attempting to find a harmonic relationship between them. One technique for finding the harmonic relationship is to perform a least squares fit of a harmonic series to the peak frequencies as suggested by McAulay and Quatieri in their pitch detection algorithm based on the sinusoidal transform [29].

Another very simple method is spectral compression [30]. This technique involves generating a histogram of the transform and fractions of it. The basic algorithm begins by initialising the histogram with bin width determined by the minimum desired resolution of the spectral compression. For the purposes of illustration, assume that we wish to iterate the algorithm 5 times and that the FFT window size is 1024 giving 512 unique coefficients. The histogram will require $2^5 \times 512 = 16384$ bins corresponding to a resolution of $f_{\text{max}} \frac{2^5}{2^5}$ Hz where $f_{\text{max}}$ is the bandwidth of the transform. The transform coefficients are entered into the appropriate histogram bins (every 32nd bin). Both transform coefficients and the corresponding frequency values are then divided by 2. The scaled transform coefficients are then added to the appropriate histogram bins (every 16th bin). The transform coefficients are scaled and added to the histogram in this manner for as many iterations as required to achieve the desired resolution (5 times for this example). The result should be a predominant peak in the histogram at the bin centre frequency corresponding to the pitch of the signal. Variations to this basic technique include using the product of the transform coefficients in place of the sum to populate the histogram and using only the local peaks of the spectrum. The major limitation of this technique is that it tends to result in pitch estimates that are some fraction of the true pitch. These estimates are often very low in frequency so the problem may be eliminated by setting a minimum allowable frequency threshold.

Another commonly employed method of frequency domain pitch detection in speech coding is cepstral analysis [31]. The cepstrum is the Fourier transform of the logarithm of the power spectrum of the signal. The idea is based on the speech production model. Assume that the vocal tract may be modelled as a linear system with impulse response $h(n)$. Voiced speech is the result of exciting this system with a periodic excitation signal, $s(n)$, as illustrated in Figure 2-14.
From Figure 2-14 it should be obvious that the voiced speech signal, \(v(n)\), is simply the convolution of \(h(n)\) with \(s(n)\):

\[
v(n) = \sum_{m=-\infty}^{\infty} s(m)h(n-m)
\]  

(2—11)

Since convolution in the time domain becomes multiplication in the frequency domain, transforming \(2—11\) to the frequency domain results in:

\[
V(k) = S(k) \times H(k)
\]  

(2—12)

Taking the logarithm converts the multiplication into addition:

\[
\log(V(k)) = \log(S(k) \times H(k)) = \log(S(k)) + \log(H(k))
\]  

(2—13)

thus \(v(n)\) has effectively been separated into \(h(n)\) and \(s(n)\). Taking the Fourier transform of equation \(2—13\) results in a spectrum with two predominant peaks. The low frequency peak will be due to the frequency response of \(h(n)\) and will correspond to the lowest formant position while the high frequency peak will be due to the spectrum of \(s(n)\) and will correspond to the period of the signal and, thus, to the pitch.

Hybrid techniques use a combination of frequency and time domain techniques. Generally, modification is performed in the frequency domain to improve the performance of a simple time-domain technique. One technique is the use of spectral flattening before simple peak detection or autocorrelation analysis [32]. Spectral flattening smooths the spectrum of the signal thus accentuating periodicity in the time domain signal. A similar technique is simplified inverse filter tracking (SIFT)[33]. SIFT uses cepstral analysis to design a filter that removes the vocal tract effects from the signal, leaving only the periodic excitation signal which is subjected to standard autocorrelation analysis.
All the algorithms discussed thus far determine the pitch using techniques that rely on characteristics of the signal or the speech production mechanism. A different approach involves exploiting a perceptual model as in the approach of [34]. The correlogram of the incoming audio signal is calculated and then vertical bands in the correlogram image, which correspond to perceived pitch, are found. The advantage of this technique is that it is able to model many of the phenomena of pitch perception, including situations involving a missing fundamental and ambiguous pitch, however, generating a perceptually accurate correlogram is computationally expensive.

A less complex perceptually based algorithm is proposed by Terhardt et al [35]. This algorithm isolates the tonal components of a STFT and then performs several transforms on these to eventually estimate perceived pitch. The main stages account for simultaneous masking, sound intensity and pitch shifts due to interaction of spectral components. Finally, two pitch measures are calculated based on the transformed tonal components and a perceptually driven decision is made to select the correct one.

The major shortcoming of most of the techniques described thus far is that they return only a single pitch estimate. For most of the algorithms, the assumption is that only a single source is present. While the Terhardt algorithm does not make this assumption, it seeks only to return the predominant pitch percept evoked by complex signals. However, for the purposes of harmonic grouping, and source separation, an algorithm that accounts for multiple simultaneous pitches is required.

This fact was recognised by Klapuri, who with various co-authors, has been developing a multipitch estimation algorithm over the past few years [36][37][38]. The algorithm has the basic architecture shown in Figure 2-15.

![Figure 2-15: Basic multipitch estimation algorithm developed by Klapuri et al (Adapted from [38].)](image)

The predominant pitch estimation algorithm operates on a single resolution STFT of the incoming signal. The transform coefficients are pre-processed to improved noise
robustness by taking the logarithm of the magnitude spectrum then passing it through a high-pass filter. The resultant “enhanced” spectrum is then subdivided into logarithmically spaced, overlapping frequency bands, which are individually analysed. For each band, a vector representing the likelihood of each frequency being the fundamental is calculated. Finally, the vectors from each band are combined to determine the overall most likely fundamental frequency. In addition to determining the most likely fundamental frequency, the predominant pitch estimation algorithm also returns an estimate of the harmonicity error, or “inharmonicity factor,” $\beta$, which is a measure of how far each of the measured harmonic frequencies are from the expected harmonics for the estimated fundamental frequency. This factor is required since many naturally occurring sounds, such as musical instruments, have mistuned harmonics [39].

Other methods for multipitch detection have been based on the pitch perception model developed by Meddis and O’Mard [40]. This model attempts to accurately represent each of the components of the auditory periphery. It consists of a perceptually tuned filter-bank of band-pass filters, followed by a half-wave rectifier and low-pass filter for each band-pass channel. The output of each low-pass filter is then subject to an autocorrelation to determine whether any periodicity exists. The autocorrelation functions from all bands are then summed to determine the overall periodicity of the input signal. The chief disadvantage of this approach is that it is computationally expensive with up to 120 channels being required.

Karjalainen and Tolonen developed a simplified version of this model that required only two analysis channels [41]. The method was thus much less computationally intensive but reported to give results very similar to the Meddis-O’Mard model. The pitch estimates determined were used to tune adaptive comb filters to separate out the individual sources in a mixture. From the point of view of the work reported in this thesis, the major disadvantage of this technique is that it only returns the pitch estimates and that source separation is performed in a totally separate process.

Wu et al developed yet another perceptually based model to track the pitch of speech in noisy recordings [42]. The model comprises four stages. The first is a 128-channel fourth-order gammatone filter-bank followed by a normalised autocorrelation on windows of 16 ms duration. The second stage of the model is a channel and peak selection phase. The role of this stage is to remove from consideration all bands that have a low signal to noise ratio and to keep only those channels that contain predominant peaks in the autocorrelation analysis. These two selections are made to ensure that only the channels
are the most likely to contain accurate periodicity information are used to estimate the pitches present. The third stage combines the information from all the selected channels to determine an overall pitch estimate. The fourth stage consists of a hidden Markov model (HMM) to track the pitches between successive frames. The major disadvantage of this model is its computational complexity, nevertheless, it does represent an interesting alternative method to that employed in the work described in this thesis, however, it is too complex for a formal comparison between the two to have been performed.

Bregman postulated a theory of pitch perception that heavily influenced the work described here [1]. Bregman noted that humans are capable of perceiving a pitch that does not technically exist in two circumstances. The first occurs when a harmonic complex is presented in which the fundamental frequency is missing. Regardless of the partial being absent from the signal, the fundamental frequency will be reported to be the pitch of the complex. The second stimulus that can evoke an unexpected pitch percept is that of an inharmonic series that has been constructed by shifting the partials of some harmonic series by a small fixed value. An example of such a series is a set of tones at 105 Hz, 205 Hz, 305 Hz and 405 Hz. The pitch of this complex will be perceived to be slightly higher than 100 Hz. In an attempt to account for these phenomena, Bregman proposed that, especially in higher frequency regions, pitch perception might be influenced by the interference between partials to a greater extent than by the partial frequencies themselves. The mechanism for this is relatively straightforward. If two partials fall within the bandwidth of an auditory filter, they will interact to produce beats at a frequency equal to the difference of their individual frequencies. For a harmonic series, it should be obvious that the frequency difference between adjacent partials will be equal to the fundamental frequency of the series. If only one source is present, and the analysis is relatively noise free, the fundamental frequency could safely be assumed to be the minimum such difference found. However, when multiple sources are present, it is necessary to generate a histogram of partial frequency differences. The fundamental frequencies are then the local maxima in the histogram.

2.4.4. **Melody**

Melodies are formed by varying the pitch of notes according to scales and rules specified by the particular musical style and culture. It has been shown that melodic contour is abstracted from absolute interval size and pitch in listeners [43][44]. This accounts for the fact that the same tune can be played in a number of different keys and still be recognised as the one tune. Further, stream segregation can influence the number of simultaneous melodies heard. Very rapid sequences of alternating high and low notes may be
perceived as two distinct, simultaneous melodies: a device that was often used by Baroque composers to give the impression that a solo instrument is playing a duet.

**Techniques for melody retrieval and comparison**

Most melody retrieval schemes rely on developing a transcription of the melody. The query-by-humming systems discussed in section 7.2.2 are typical examples. However, since the melodic contour is most important, there really is no need for deriving the exact score of the melody. Indeed, requiring exact melody transcription, and the difficulties associated with obtaining these, restricts the systems described in [45] and [46] to only retrieving melodies that have been encoded in MIDI format.

One consequence of the fact that melodic contour is abstracted during the perceptual process is that it is possible to recognise melodic variants. The variation may be in the form of rhythm, tempo, key or the addition of ornamental notes. Identifying such variations using exact transcriptions requires the use of approximate string matching techniques [47][48]. If the melodic contour is abstracted separately, and comparisons made on its overall shape, the problems of approximate string matching may be avoided.

**2.4.5. Rhythm**

The conscious perception of rhythm appears in part to be the result of a high level mental process of organisation rather than an intrinsic property of the stimulus. Evidence for this is the experimental finding that if a series of physically identical stimuli are presented uniformly in time, subjects will generally perceive the series as a repeating group, the first member of which is slightly accented [49]. One possible reason for this organisational tendency is that the formation of such groups reduces the amount of information that must be memorised [50].

In order to perceive a rhythm, it must be possible to perceive the stimulus as a series of discrete elements. This separation may be furnished by pitch variation or by turning the stimulus on and off, for example. Even when the variation is continuous, such as the amplitude modulation of a pair of pure tones with small frequency separation, two discrete levels can be perceived resulting in a repeating pattern.

Perceived rhythm primarily depends on three physical audio characteristics: accent or relative intensity of individual elements, absolute and relative duration of individual elements and temporal spacing of the elements [49]. Accenting, or increasing the relative intensity of, an element at regularly spaced intervals (e.g., every fourth element) has a “group-beginning effect”. That is, the rhythmic group will be perceived to begin with the
accented element. Similarly, increasing the length of one element in a group has a “group-ending effect” [49]. The rate of presentation of elements and the overall duration of a repeated group also influences rhythm perception. Repetition rates below 0.1Hz cause the elements to be perceived as discrete occurrences in time while rates above 8Hz cause the elements of a group to fuse into a single, continuous sensation [51]. Also, the formation of a rhythmic group is contingent upon the total duration of the group falling below an upper limit termed the “specious or psychological present”. This is generally between 5-10 seconds [52].

Techniques for rhythm detection

Compared to pitch detection, rhythm detection is a relatively new area of research. This is because the major application of rhythm detection is in music analysis: a study that has only become practical since the development of the required technologies to store and reproduce wide-band audio data. As a result, few rhythm detection methods exist.

The two principal methods used for rhythm detection involve detecting sudden changes in amplitude or frequency content. Both these methods are applied in the melody transcription system in [53] to determine note durations. The methods used in this system are suitable only for sung input and operate best when there is a definite separation between successive notes. Also, the note values (quarter note, half note etc) can only be determined if the user inputs the value of the shortest note they intend to sing.

Goto and Muraoka [54][55] have proposed more sophisticated algorithms that are based on the same ideas of amplitude change and chord change detection. The main extensions involve investigating the nature of onsets and using a sophisticated hypothesis testing procedure to decide between several possible beat locations. The rhythm is built up by first looking for beats at the quarter note level and then grouping beats at the half note and finally bar level. The quarter note level of beat detection was developed for music with a drum beat [54] and the higher level grouping procedure was developed for rhythm detection in music without a definite drumbeat [55]. The main drawback of these systems is that the first one is only applicable to modern music with a definite drumbeat and both systems assume a time signature of 4/4.

2.4.6. Timbre

The traditional view is that perceived timbre is completely dependent on the steady state magnitude spectrum of a sound [56] with the formant positions having the most significant influence [24]. In the case of steady state tones, this is in fact true. However,
for more natural sounds such as instrument tones, temporal factors also play an important role. This was established in relatively early experiments by Stumpf who showed that the recognition of musical instruments is made difficult by removing the initial segments of the notes played on these [56].

Unlike pitch and intensity, no single physical characteristic can be said to be predominantly responsible for timbre. Even the timbre of a single instrument is influenced by different factors across its loudness and pitch ranges [44]. There are, however, several general relationships between physical characteristics and timbral descriptions that can be made:

- **Formant structure.** A constant formant structure tends to indicate constancy in timbre [56];

- **Spectral envelope position.** The position of the spectral envelop along the frequency scale is related to brightness. The higher on the frequency scale the envelope is situated, the brighter the sound [56];

- **Attack and decay characteristics.** Many features of attack and decay characteristics influence the perceived timbre of a sound. These include the speed and shape of the attack and decay transients as well as the overall smoothness of the temporal variation. For example, adding a fast attack and a slow decay to any sound will cause it to sound ‘plucked’ while a smoothly varying temporal envelope is characteristic of woodwinds [44];

- **Frequency and amplitude modulation.** The characteristics of the frequency and/or amplitude modulation can affect the perceived timbre of a sound. The modulation rate, envelope and depth can all influence perceived timbre. For example, the sound of a violin can be characterised by a triangular waveform modified by a steady but complex spectral response and a flutelike timbre is, in part, due to the pattern of amplitude modulation [44]; and

- **Relative intensity of partials.** The overall spectral shape can also influence perceived timbre. A predominant fundamental is characteristic of a flute while increasing the intensity of higher frequency partials results in a brasslike quality [44].

The extent to which each of the factors stated above influence timbre may be context dependent. For example, the influence of attack and decay characteristics on trumpet tones is more pronounced in long sustained notes than short notes and the acoustics of
the room have a significant affect on the audibility of transients; the sound of an organ depend greatly on the characteristics of the room in which it is situated [56].

Context has yet another interesting effect on timbre that is the result of stream segregation. This is because the timbre of an acoustical event depends upon which partials are deemed by the perceptual system to belong to that event as stated by Bregman [17]:

\[ \text{Timbre is a perceived property of a stream organisation rather than the direct result of a particular waveform.} \]

Thus, the perceived timbre depends on the results of stream segregation. As was indicated earlier, this process of segregation is greatly influenced by context. Hence, the timbre of a given sound depends on the acoustical environment in which it is presented [57].

**Techniques for timbral analysis**

Techniques for timbral analysis have concentrated on attempting to find a structural representation of the similarities and differences between timbres, much like the musical scale is used to organise pitch relations. Because of the complex nature of timbre, such a scale would necessarily be multi-dimensional; hence it is more correctly termed a timbral space. A good timbral space should place perceptually similar sounds close together and perceptually dissimilar sounds far apart. Ideally, this space should be continuous so that it is possible to investigate the effect on the timbre of varying a single variable over its entire range.

Risset and Wessel [56] describe the generation of subjective timbral spaces in which human subjects are asked to rate pairs of sounds on a number of aspects. Statistical analysis is then used to derive geometrical models of the data. Such a method is effective in displaying the relationship between the timbre of different sounds, however, it is an extremely time consuming and laborious process. Also, further analysis is required to determine the physical characteristics responsible for the variation in each dimension and because of “the intervention of cognitive facets, such as familiarity and recognition... a fully continuous timbre space may not be obtainable.” [56] A solution to these shortcomings is an automatic method to generate the timbre space that does not depend on recognition.

Sandell and Martens [58] propose a technique for generating a timbre space that is fully automatic. They use principle component analysis of the time frequency distribution of
sounds to derive a timbral space. The main disadvantage of this technique is that there is no fixed correspondence between individual dimensions in the space and components of timbre. For example, it is not possible to hear timbres varying only in brightness by tracing a path through the space that is parallel to one axis.

The ideal is thus a method that is automatic, does not depend on recognition and results in a timbre space whose dimensions explicitly correspond to individual timbral qualities. Given the sensitivity of timbre to context, this last point may be an unachievable aim. However, the general trends stated in the previous section indicate that at least an approximation of this ideal is possible. The only existing attempt at creating such a space is extremely limited [59]. Only steady state spectral features are used and hence the only timbral dimensions available are brightness and roughness.

The system proposed in [59] does not use a timbre space explicitly. The timbral characteristics are simply used as possible search criteria in an audio database. Similarly, in the work proposed here, there is no need to explicitly derive a timbral space. The physical audio characteristics that determine timbre can simply be used as a basis for retrieval for a ‘sounds like’ query. However, the representation developed in this thesis would allow the generation of a timbral space, which is a potential direction for future research.

2.4.7. Sentics
It is evident from common experience that music can convey information about emotions; musical passages are often described in terms of emotive adjectives such as “bright and happy” or “morbid”, for example. Many authors have noted that not only does music convey this information, it can actively change the emotional state of the listener [20][60][61]. Although limited work exists in the area it is known that this effect is measurable and culturally invariant. Besides the experimental results in [20], two pieces of evidence exist for this cultural invariance. Firstly, the motivation to listen to musical forms exists across cultures and secondly, there are elements common to the musics of all cultures (the most notable being rhythm).

Given that this effect is culturally invariant, it is likely that, as with the audio characteristics described in this section, there is a correlation between the physical signal and the effect that it induces. Work substantiating this has been performed by Konecni and, more extensively, by Clynes. Konecni showed a correlation between melodic complexity and volume on the likelihood and strength of reaction to situations intended to induce anger [61]. Clynes has reported effects due to melodic pitch contour, amplitude
contour and rhythmic structure on the emotional state of listeners (see for example [62]). Since the work described here proposes to develop a representation based on a perceptually significant structure from which these features can be readily extracted, it will likely be a useful tool for research in this area. It is also conceivable that induced emotional state could become a query key in a future retrieval system or may be used as a method for an automated rating system. These are interesting areas for further research.

2.5. **Audio Signal Analysis Techniques**

Although perceptually congruent descriptions of the auditory scene will be the most generally useful, there may be occasions where (statistical) analysis of the raw physical characteristics is appropriate. Indeed, as the previous section has shown, determining these physical characteristics is usually the first step in deriving a perceptual description of the data. Another example where raw physical analyses may be appropriate is in content-based retrieval via onomatopoeic comparisons where both the query sample and the database records have been subjected to the same analysis. For these reasons, this section provides a review of the various audio signal analysis techniques in existence.

The section begins with the various purely statistical techniques that operate in one of three domains: time, frequency or time-frequency. It then moves onto techniques that have been influenced by what is know of the perceptual process but nevertheless offer only statistical descriptions of the data. Finally, statistical approaches to the auditory source separation problem are reviewed and critically compared with the perceptually based computational auditory scene analysis approaches, a variation of which has been adopted in this thesis.

### 2.5.1. **Time Domain Techniques**

Time domain techniques derive information about the signal’s physical characteristics by examining the waveform directly. The information that may be obtained in this domain includes the intensity of the encoded signal, whether the signal is periodic or noise-like, and the period (frequency) of the signal. There are many time domain analysis techniques including:

- **Zero crossing rates.** Zero crossing rates are used in speech analysis to determine areas of voiced and unvoiced speech. An unvoiced section of speech is noisy while a voiced section is periodic, hence the unvoiced section will have a higher zero crossing rate [63];
- **RMS power.** The RMS power over a fixed window length gives a very rough indication of the intensity of the signal. The technique is used in some systems as a possible basis for comparison between sounds [59];

- **Peak analysis.** Analysis of the distance between peaks in a time-domain waveform has been used to determine the pitch of a speech signal [27];

- **Autocorrelation analysis** of short windows of the signal can be used to measure periodicity in the signal (and hence pitch). Autocorrelation analysis has also been used to determine whether a section of speech is voiced or unvoiced [63]; and

- **Energy separation.** The signal is analysed using a time domain operator, the Teager energy operator, that separates the AM and FM components of the signal. This can then be used to determine the instantaneous amplitude and frequency of the signal provided that the signal bandwidth is small [64].

### 2.5.2. Frequency Domain Techniques

Frequency domain techniques investigate various aspects of the frequency content of a signal. The first step in these techniques is to subject the signal to some frequency transformation in order to obtain a spectral estimate. The most commonly used frequency transform is the Fourier transform, however, spectral estimates can also be obtained from other mathematical transforms. These include the cosine transform [65], sine transform and the Hartley transform [66]. The advantages of the latter transforms are that, unlike the Fourier transform, they are purely real and thus reduce the overall computational complexity of the analysis. In the case of the cosine and sine transforms, it is difficult to separate phase and magnitude information, however, the Hartley transform allows this information to be derived as simply as from the Fourier transform.

The information that can be obtained in the frequency domain includes the bandwidth, pitch, and some low-level information about the quality of the sound. Much of this information is gained by performing statistical analyses of the Fourier transform. Some specific analyses that have been used include:

- **Centroid of the magnitude spectrum** has been used as a measure of brightness. See, for example, Wold et al [7][59];

- **Signal bandwidth.** One method used to determine this is to calculate the average difference between the centroid and other spectral components [59]. Ideally a single frequency tone should have a bandwidth of zero. Bandwidth has been suggested as a means to determine whether a signal contains
predominantly speech since the spectrum of speech is concentrated over a fairly narrow bandwidth [59]. The disadvantage of this technique is that it cannot operate over a band-limited channel such as a telephone line;

- **Harmonicity** can be computed by examining how closely the test signal’s spectrum fits a perfectly harmonic spectrum with fundamental frequency at the same frequency as the lowest predominant frequency in the spectrum under investigation [59]. A significant disadvantage of this technique is that it assumes that only one monophonic source is present in the test signal. The technique would fail even for a piano playing a chord, for example; and

- **Pitch** can be calculated from the spectrum using a variety of methods ranging from very simple spectral compression to highly sophisticated, but computationally intensive, algorithms. Some of these were discussed in 2.4.3.

### 2.5.3. Time-Frequency Analysis Techniques

Thus far different types of analyses that can be performed on a signal in a single domain have been reviewed. In the temporal domain it is possible to obtain information about when changes in the signal occur but it is very difficult, if not impossible, to tell which frequencies are responsible for this change. On the other hand, in the spectral domain a detailed picture of the frequency content is available, however, no information exists about which frequencies were present at what time. The solution is to combine the two techniques to form a new set of techniques called time-frequency analysis.

The most common, and earliest, method of time-frequency analysis is the short time Fourier transform (STFT) [67]. The STFT is the basis of the spectrogram, a speech analysis tool developed in the 1940’s [68]. The basic idea of the STFT is that short time windows of the signal are taken and analysed using a Fourier transform. The result is a picture of the frequency content of the signal at individual instants in time and the variation of frequencies over time. Figure 2-16 gives a simplified illustration of a STFT.

![Figure 2-16: A simplified short time Fourier transform](image)
Selecting the length of the temporal window used to generate the STFT is not a trivial matter, especially if the temporal variation of the signal is rapid as is the case in speech. In order to have an accurate description of the signal’s temporal variation a short window is required. However, since the frequency resolution of the Fourier transform is inversely proportional to the window length used to calculate it, an accurate frequency analysis requires a long window. Hence, there is a time frequency trade-off in generation of a STFT.

Several methods have been developed to overcome the resolution limitations of the standard STFT. One of the simplest methods is to select a different transformation function (or kernel) (such as a modified Discrete Cosine Transform [69]) to increase the frequency resolution for a given window length. Another simple method involves overlapping the analysis windows to increase the temporal resolution. However, these do not offer much gain and the most common methods exploit properties of human perception or of the signal being analysed or a combination of these to perform a better analysis. In general, the result is an analysis with varying time-frequency resolution.

One way to achieve this varying time-frequency resolution is to set up a bank of bandpass filters with varying bandwidth. This is known as sub-band analysis. Since the FFT can be interpreted as a series of bandpass filters with bandwidth equal to the resolution of the FFT, STFT can be viewed as a special case of sub-band analysis where the filters are evenly spaced and have identical bandwidths. A better perceptual tuning is achieved by varying the window length of the FFT at different frequencies as in the constant-Q transform[70]. The most perceptually accurate sub-band analysis would require many filters and would thus be extremely complex. To overcome this, most sub-band analysis techniques involve a combination of a filter bank followed by transform such as an FFT or a DCT. In some cases the transform used is derived from the signal itself. This is the case in Wigner-Ville and the Choi-Williams distributions for example [71].

The aim of all the techniques discussed in this section is to give a picture of the variation of an audio signal both in time and in frequency. There are many uses for these analyses. In some cases, the actual time-frequency representation is not explicit but is nevertheless present. This is the case in frequency domain pitch extraction algorithms, which effectively use a STFT as their basis [35]. In other cases, such as speech recognition, the use of the representation is more explicit.
2.5.4. **Perceptually based techniques**

In the previous section, methods of audio signal analysis exploiting the time-frequency resolution of the human ear were discussed. This section describes some other signal analysis techniques that model the perceptual process much more closely. These techniques mainly find application in psychoacoustics for the study of human audio perception and in computational auditory scene analysis.

A basic extension of the constant-Q analysis described in the previous section is the sinusoidal track representation proposed by Ellis [72] that was based on the sinusoidal transform of McAulay and Quatieri [73]. The motivation behind this representation was the development of a system that could perform source segregation in a manner similar to human perception. The significant feature of this technique was the tracking of the amplitude peaks through time and frequency. The basic idea behind the representation was not new, having been used by McAulay and Quatieri as well as others for speech coding previously, however, the application and interpretation was novel.

Further dimensions can be added to the analysis to provide still further information about the signal. One such analysis technique is the correlogram. In this technique a third dimension is added to the traditional TFD. The third dimension is the lag time of the short-time autocorrelation function of the energy envelope for each frequency band. In a similar extension to his sinusoidal tracks in the TFD domain, Ellis proposed an extension to correlogram analysis called wefts [74]. The weft is a one-dimensional contour that tracks peaks in the two dimensional frequency versus autocorrelation lag surface. Once again the aim was to discover an analysis that could assist in source segregation.

2.5.5. **Computational Methods for Auditory Source Separation**

It was mentioned in the introduction to this thesis that two major approaches have traditionally been employed in solving the auditory source separation problem. The first of these is a purely statistical technique known as blind source separation (BSS) while the second is a perceptually based approach referred to as computational auditory scene analysis (CASA). For reasons that will be made clear, the latter approach is the one adopted in this thesis. This section reviews the major works in BSS and some newer hybrid approaches. While discussion of the details of CASA systems appears in Chapter 4, the advantages of these systems over BSS are detailed here.
Blind Source Separation Techniques

The various blind source separation techniques have found applications in fields as diverse as medicine and radar signal processing. The basic premise of these techniques is that given N observations of a mixed source signal arising from M ≤ N sources, it is possible to extract all M sources given only the assumption that the sources are statistically independent. In practice, many, especially of the earlier, algorithms impose the further restriction that the communications channel is a linear system. When the conditions are met, blind source separation techniques have been found to be useful for improving the performance of automatic speech recognition systems in noisy environments [75] and separating percussive musical sounds (drums). [76]

BSS approaches the problem of signal separation as follows. Assume that N observations, \( y(n) \), are made of a mixture arising from M source signals, \( s(n) \):

\[
S(t) = \begin{bmatrix} s_1(t), s_2(t), \ldots, s_M(t) \end{bmatrix}^T \quad (2-14)
\]

\[
Y(t) = \begin{bmatrix} y_1(t), y_2(t), \ldots, y_N(t) \end{bmatrix}^T \quad (2-15)
\]

In the case of auditory source separation this amounts to using N microphones to detect M sources. The most basic approaches, which are really just an application of the method to signal enhancement, use two microphones assuming only two sources, the target speaker and background noise.

Given the signal vectors defined by equations 2 — 14 and 2 — 15, the relationship between the original source signals and the observed signal can be represented as:

\[
Y(t) = H[S(t)] \quad (2-16)
\]

In equation 2 — 16, \( H[\cdot] \) represents the action of the transmission channel on the source signals. In the general case, \( H \) is a non-linear, time varying matrix-valued function. There is currently no solution to the general case; however, there are some specialised non-linear cases for which solutions have been proposed [77].

To facilitate the solution of equation 2 — 16, the majority of BSS systems assume that \( H \) is a linear process. In this case equation 2 — 16 can be rewritten as:

\[
y_j(t) = \sum_{i=1}^{M} h_{ji} \ast s_i(t) \quad \forall \ j \in [1, N] \quad (2-17)
\]
where $*$ represents convolution. The process of signal separation then involves finding the coefficients $h_{ji}$ such that equation 2—17 can be solved to yield the source signals, $s_i(t)$.

The manner in which these coefficients are estimated is where the various BSS techniques differ. In order to avoid the cumbersome convolution, it is customary to transform equation 2—17 into a domain where the convolution can be replaced by multiplication. Two such transforms are the z-transform and the Fourier transform. Assuming one of these transforms has been applied, equation 2—17 becomes:

\[ Y(t) = H \times S(t) \]  
\[ (2—18) \]

The process of BSS then seeks to find the matrix $H$, knowing only the value of $Y$ and assuming that $H$ is invertible. The necessity for $H$ to be invertible should be obvious, since, to obtain the separated signals given $Y$ and $H$, we must solve the matrix equation 2—18 to yield:

\[ S(t) = H^{-1} \times Y(t) \]  
\[ (2—19) \]

For this to return a unique solution, the following conditions are usually imposed:

1. Each of the sources, $s_i(t)$, are statistically independent of one another. However, as will be discussed below, there has been a method suggested to overcome this restriction for the case of musical signals.

2. $H$ is invertible and $M \leq N$. However, some systems have recently been developed that have used other properties of the data to relax the requirement that the number of observations must exceed or equal the number of sources [78][79].

3. None, or in some cases, at most one, of the sources are Gaussian.

By imposing the above conditions, various statistical techniques have been proposed to estimate the mixing matrix, $H$. The notion of independence is heavily used to solve the BSS problem. Statistical independence implies that the cumulative probability density function (pdf) of the combined sources is simply the multiplication of the individual pdfs:

\[ P(S) = \prod_{m=1}^{M} P(s_m) \]  
\[ (2—20) \]

Each of the proposed techniques operates under one of two different conditions: stationary (or batch) where $H$ is assumed to remain constant over all the data and
adaptive (or online) where $H$ is assumed to vary with time. One important note is that many of the methods will return scaled permutations of the sources, which has implications for matching the output between sample windows. Following is a very brief description of some of these techniques [80].

**Contrast Functions**
Contrast function methods aim to optimise functions of the statistical properties of the output that take on maximum or minimum values when the vector $B$ in equation 2 — 21 is a separating vector. A separating vector is simply a row of the matrix $H$ in equation 2 — 18.

$$y = Bs$$

(2 — 21)

**Maximum Likelihood Methods**
These methods are applicable when it can be assumed that the pdf of the source is differentiable and that a number of independent observations, $T$, are available. These factors are then used to minimise the divergence between the observed pdf and the hypothesised pdf.

**Orthogonal Contrasts**
These methods operate under the condition that the input to the separating function is white noise. Thus the data is first passed through a whitening filter and one of the following conditions is optimised to find the separating function:

- **Maximum likelihood** similar to other maximum likelihood methods except now the separating matrix will simply be a rotation matrix;
- **Minimum kurtosis** methods minimise the difference between kurtosis of estimated signals and the whitened observation data; or
- **Orthogonal cumulant matching** methods minimise the difference between the fourth order cumulants of the estimated signals and the observed data.

It can be seen from the above discussion that BSS methods generally involve minimising (or maximising) some function that relates the statistics of the estimated sources to those of the observed data. The approaches adopted to perform this minimisation operation have ranged from the purely algebraic solution statistical conditions such as those mentioned above through to heuristic solutions such as that employed by [81].

One view of the latter approach that has received some attention is the use of neural networks to achieve source separation. Although often posed in a different light, such as “waveform modelling” [82], the principal difference between the neural network
approach and classical BSS approaches is that the separating matrix is trained to the sources that are expected in the mixture. Some of these methods bear the claim that the only condition that must be satisfied in order for them to be applied is that the number of sources is known [83]. However, there is one major drawback with this method in that it must be trained. This might not always be achievable in practice and thus constrains the applicability of these systems.

Given that BSS techniques rely on the statistical properties of the signals and the communications channel, they are particularly sensitive to the form that the probability functions of the sources assume. This fact was noted by Douglas et al [84] who proposed a technique to improve performance in the case where the mixed source signal contained data from both sub- and super-Gaussian sources or the form of the distributions of the sources was unknown a priori. Their technique was founded on an alternative view of the BSS problem, which states that the communications channel is a series of time-variant filters. Given this formulation of the problem they estimated the coefficients of the separation matrix by optimising the stability criteria for each of these filters.

BSS techniques typically achieve separation in the time domain, resulting in a series of raw signals with each (hopefully) corresponding to a single source separated from the original mixture. Recently, however, techniques have emerged that operate in the time-frequency domain. The resulting systems effectively separate the various bins of the TFD to individual sources. However, there is a chief difficulty that these approaches must address: because the estimation procedures do not guarantee to return the sources in any particular order, there must be some mechanism to correctly line up the assignments from one time frame to the next. One technique that has been proposed in order to solve this problem is to perform a cross correlation of the output from successive analysis frames [85].

The statistical independence requirement greatly limits the utility of BSS methods to generic audio applications. One area where this restriction is particularly infeasible is in the case of source separation in music. The melodic and rhythmic structure of western music ensures that the various sources are usually highly correlated both in time and frequency. This shortcoming was recognised by Viste and Evangelista [86] who proposed a method for overcoming this by exploiting the phase differences between the left and right channels of stereo recordings. Their technique has the added advantage that the number of sources may exceed the number of observations (which is two), however, the number of sources must be known a priori.
Abrard et al [87] proposed another method to relax the independence requirement for general signals as well as allowing source separation from monaural data. The basic premise of this method was that, for any given mixture, there is a high probability that at least some limited regions in the time-frequency plane will represent only one source. Being able to identify these regions allows a closed form solution for the coefficients of the separation matrix. The principle disadvantage of this technique is that it assumes that there are only two sources of interest, usually denoted as the foreground and the background source. Hence, it is essentially only a “structured noise” cancellation procedure.

In summary, it is generally agreed that the somewhat artificial problem of blind source separation from a steady state mixture with linearly mixed, statistically independent sources is well understood and essentially solved [77]. On the other hand, there is much remaining to be done for any situation where these constraints are not met. These include, non-linear mixtures, statistically dependent and convolutive mixtures. These situations are common in the audio collections that would typically be of interest in content-based retrieval and coding applications.

Despite the obvious paradox, this brief review has shown that researchers have considered BSS techniques for the sorts of auditory signal separation tasks that would benefit both coding and information management applications. From the point of view of these applications, the most important characteristics of BSS techniques are:

- **Conditions for application**
  - The sources must be statistically independent;
  - The number of observations (microphones) must generally be greater than the number of sources OR the number of sources must be known *a priori*;
  - The mixed source signal must be a linear combination of the individual sources; and
  - The statistical properties of the sources must generally be known *a priori*. At the very least, no more than one of the sources can display a Gaussian distribution.

- **Output**
  - *Time-domain methods*: a raw audio signal for each source separated out of the mixture; or
• **Time-frequency domain methods**: a TFD of each signal separated from the TFD of the mixture. This is usually inverted to produce the raw audio signal.

**Hybrid Approaches**

A few recent proposals have suggested combining both CASA and BSS approaches. Perhaps the most obvious approach to combining the two methodologies is to use a classical CASA front end to decompose the raw audio signal into a series of bandpass channels and then to perform BSS on each of the channel outputs using any one of the traditional techniques. One advantage of this approach over classical BSS is that it affords some robustness to band-limited noise signals. This approach was adopted by Rutkowski *et al* [88][89] whose main aim was to enhance a target speaker’s voice in a multi-channel recording from a reverberant room with several interfering sources, which could include other speakers. The limited results reported indicate that some success was achieved with the technique, however, no indication of its performance relative to standard BSS or CASA techniques is provided. In [89] the authors conclude that the principle advantage of their technique over standard BSS is that it “is closer to human like processing of sounds” implying that improved performance over classical BSS techniques was achieved and that this was attributed to the CASA influence on the model.

Okuno *et al* [90] propose a method of combining a binaural CASA solution, which two of the authors had previously developed, named BiHBSS [91] with a 2-channel BSS technique that allows the separation of multiple sources. Their work was motivated by what they determined to be shortcomings in the models they chose to combine. Specifically, the CASA technique relied on interaural delay differences to perform the source separation, thus it was highly sensitive to the placement of microphones relative to the sound sources. The BSS technique on the other hand, could only separate two sources since only two observations were available. The technique they proposed essentially used the CASA solution to decompose the scene into collections of two simultaneous sources and then used the BSS technique to complete the separation. They justify this design with the assertion that BSS is better than the CASA technique that they employed for mixtures of two sounds. They evaluated the system by determining the improvement in ASR rates for each of the separated speakers from a mix of three simultaneous speakers. Their combined technique did result in some improvement over the CASA system alone. It was not possible to compare the performance to the BSS separation technique because it was not applicable to the evaluation task.
In a completely non-traditional approach, Wang proposed a neural network model for source separation [92]. The neural network consisted of a two dimensional (representing time and frequency) array of oscillating nodes:

Partial tracking and grouping was achieved because groups of nodes that belonged to the same tracks/groups oscillated in synchrony while those from different groups were asynchronous. Details of the transform/filterbank that decomposed the incoming signal into frequency bands were not supplied. Wang reported that the model was performed similarly to human subjects on several standard psychoacoustic streaming experiments.

**Evaluating Source Separation Strategies**

The review of BSS separation techniques revealed that these methods impose a number of restrictions on the data and the manner in which it is collected that may not be achievable in practice. In particular, the number of observations (microphones) must generally equal or exceed the number of sources that are required to be separated, or, at the very least, the number of sources must be known *a priori*. Further, the statistical properties of the sources must be known *a priori*; the minimum requirement being that at most one of the sources is Gaussian. Also, it is generally assumed that the mixing process is linear and non-reverberant. While some of these restrictions are being relaxed in the most recent attempts, this generally comes at increased computational expense. Yet another disadvantage of this class of technique, from the point of view of the work reported in this thesis, is that BSS techniques generally operate in the time domain, returning a time-domain representation of the separated signals. However, the chief aim of the current work is to develop a representation that will be useful for compact data...
storage and will provide at least some low-level content information about the data that is thus encoded. While it would be a simple matter to apply any standard coding technique to the output of a BSS algorithm and thus achieve the former of these aims, providing content information would require a further stage of processing. In addition, the only content analyses that would naturally apply to such a representation would be the low level statistical and incomplete semantic labels whose disadvantages were discussed in section 2.7.2.

CASA techniques, on the other hand, only present restrictions that are psychologically and practically feasible. There are no implied restrictions on \textit{a priori} knowledge of the number or nature of the sources neither is there generally any requirement for a multi-channel recording, although some systems do benefit from stereo recordings. Certainly there are a number of challenges that must be overcome. These include the problems of co-incident or nearly co-incident and crossed partials as well as selecting the appropriate transformation resolution. From the computational perspective, one must decide whether to favour physical accuracy or computational convenience. However, there have been a number of suggested solutions to these challenges and further research may reveal still more feasible and accurate solutions. One factor that is clearly very important is that, while pitch-based streaming might prove sufficient for most mixtures containing only speech or music or both, general computational auditory scene analysis requires mechanisms for dealing with noisy and impulsive sounds. Hence, when designing a CASA system, a model to deal with such signals should be incorporated, or, at the very least the facility to extend the model in this direction should be provided. Another feature that pitch-based streaming systems lack is the multiple competing hypotheses that are thought to exist in the perceptual system [1]. Hence, a ‘good’, general CASA system should seek to incorporate as many grouping cues as possible as well as a suitable mechanism for combining or selecting between these.

There are a number of advantages with CASA based systems that have a particular bearing on the work reported here. The first is that the ‘modular’ nature of CASA representations provides a natural set of building blocks for hyperaudio. The low level tracks afford useful within-node anchor points while the higher level streams are clearly candidate nodes. Secondly, these representations allow relatively easy extraction of perceptually relevant index features such as pitch contours. Indeed, because of the granularity of the representation, it is possible to identify features, such as onset characteristics, that are important for timbre perception. While the raw descriptions of these are likely to be of limited relevance to the general user, these attributes would be
useful in onomatopoeic comparisons or in timber classification applications. Finally, given that only the perceptually relevant detail is retained, representations that result from CASA lend themselves to compact data storage without obfuscating the underlying data’s inherent structure nor impeding access to the various indexing mechanisms aforementioned.

It should be obvious from the short review of the hybrid CASA and BSS approaches, which have appeared recently, that these have generally sought to overcome shortcomings with specific algorithms. In most cases, they aim to relax some of the restrictions imposed by BSS by resorting to CASA technology. One case [90] claimed that while CASA was useful to allow their stereo BSS system to separate more than two sources, the BSS system achieved superior separation for mixtures of two sources and was therefore still required. This finding may have been due to two main factors. The first is that the evaluation metric used was the improvement in error rates of an automatic speech recogniser. This system would have benefited from the raw audio representation that is the output of the BSS technique. Although CASA-based representation can also be inverted, more care is usually required to ensure that the reconstruction is of a high perceptual quality. Perhaps the most influential reason for the apparently poor performance of the CASA system is that it did not use a majority of the grouping cues that are typically recognised as perceptually important. Rather, it concentrated on interaural time differences. While this is indeed a valid separation cue, it is by no means the dominant one since even subjects with only one functioning ear can easily perceive acoustic streams and, to a surprising extent, are even adept at sound localisation [1]. As with most hybrid systems, the work of [90] would best be described as a BSS technique with CASA influences.

2.6. Audio Coding Methods
In order to fulfil the requirements of the hyperaudio data model, we must achieve three broad aims. The first two, decomposing the mixed audio signal into its constituent objects and describing each one of these in a perceptually significant manner, have been the focus of the discussion thus far in this chapter. The third requirement is for compact, randomly accessible, structured storage. This section will explore the various existing mechanisms for compressed audio data storage with respect to their suitability to the task at hand.

The term ‘speech coding’ (and by implication audio coding) is generally used in the literature to mean data compression in order to distinguish this from ‘speech
compression’: a term commonly used to refer to time-scale reduction [93]. Traditionally audio coding is divided into three broad categories: speech coding, wide-band speech coding and audio coding. The distinction between categories lies in the signal bandwidth, the required data rate and the expected reconstruction quality. Speech coding generally aims to produce toll (standard telephone) quality speech at a bandwidth of approximately 4 kHz while in wide-band speech and audio coding the expectation is that quality will be much higher with wider bandwidths (approx. 7 kHz for speech and 32-40 kHz for audio) being tolerated.

2.6.1. **Waveform coding**

Waveform coding techniques operate on the time-domain waveform. The simplest technique is pulse code modulation (PCM). Strictly speaking PCM is the raw digital representation of a signal, although compression of a sort may be achieved by restricting the bit rate, sample rate and number of channels used to represent the signal. Reducing the data rate of PCM has a very pronounced effect on signal quality.

An audio waveform varies slowly between samples and thus exhibits much redundancy. This redundancy is exploited by a number of techniques that are simple extensions of PCM. These include differential pulse code modulation (DPCM), adaptive differential pulse code modulation (ADPCM) and adaptive delta modulation (ADM). The basic idea behind all these methods is to code only the value of the first sample and the difference between successive samples. For a given sample resolution, this will generally require fewer bits than coding the actual sample values themselves since audio waveforms typically vary slowly with time.

A slightly more sophisticated method of compression in the time-domain is to exploit the auditory system’s non-linear intensity resolution. This technique, known as companding, allocates more resolution to low amplitude samples and less to high amplitude samples. Two common examples are a-law and µ-law quantisation used for telephony in Europe and North America respectively [94]. Both schemes take a 12 or 16 bit sample, \( x \), and map it to 8 bits using the appropriate formula below:

\[
y = \frac{\ln (1 + \mu x)}{\ln (1 + \mu)} \text{ where } \mu = 255 \tag{2-22}
\]

\[
y = \begin{cases} 
\frac{Ax}{1 + \ln(A)} & \text{for } 0 \leq x \leq \frac{1}{A} \\
\frac{1 + \ln(Ax)}{1 + \ln(A)} & \text{for } \frac{1}{A} \leq x \leq 1
\end{cases} \tag{2-23}
\]
2.6.2. **Perceptual Audio Coding**

Waveform coding methods generally produce audio signals with a perceptual quality roughly proportional to the mean square error (MSE) of the output signal relative to the input signal. However, it has been observed that in the general sense, perceived audio quality is not directly related to the MSE of the signal [95]. Rather, perceived audio quality depends greatly on what portions of the signal spectrum have been distorted by the coding scheme (and in what manner). This is obviously a direct result of the characteristics of the human auditory perceptual system described earlier in this chapter. Exploiting these characteristics to achieve high coding gains while maintaining high perceived signal quality has given rise to a class of audio coding methods known as perceptual audio coding. Indeed, as will be made evident in the succeeding sections, most modern day audio codecs in common use can be described as perceptual codecs.

The general form of a perceptual audio codec is shown in Figure 2-18.

![Figure 2-18: General form of a perceptual audio codec. After [96]](image)

Figure 2-18 shows that in a perceptual codec the incoming audio is subjected to a time-frequency analysis from which a set of parameters are derived. This time-frequency analysis may be influenced by a psychoacoustic analysis of the incoming audio data or the psychoacoustic model may govern the parameters of the time-frequency analysis or both. The output of the time-frequency analysis (and the psychoacoustic analysis, if present) is then quantised and encoded before being multiplexed into a single data stream to be transmitted or stored as appropriate.

The following sections will briefly describe the function of each of the blocks in Figure 2-18 with a review of relevant work in the area. From the point of view of the work presented in this thesis, the Time-Frequency analysis and the resulting parameter extraction are of most relevance and will hence receive the most attention.

2.6.3. **Quantisation and encoding**

In most perceptual codecs the psychoacoustic model is used to govern the bit allocation in the quantiser to achieve better quality for a given compression gain. Such codecs use...
various algorithms to quantise the parameter values depending on the intended application. MPEG-1 layer III [97], for example, uses an algorithm with nested loops that employs both non-uniform quantisation and Huffman coding. The inner loop iteratively adjusts the quantisation step size until the total number of bits required to encode the transform components within the block fall below the “bit budget” determined by the psychoacoustic analysis. The outer loop then performs an analysis-by-synthesis evaluation to compare the quantisation noise level with the JND threshold values determined by the psychoacoustic model. [96]

It may also be possible to exploit inherent statistical redundancy in the parameters using traditional coding schemes such as DPCM or ADPCM. The outputs of the quantisation and bit allocation blocks are usually then encoded using a lossless scheme such as run length encoding (RLE) or an entropy based scheme (such as Huffman, Arithmetic or Lempel, Ziv and Welch). Because of the influence of the perceptual model on the quantisation process, most perceptual codecs result in variable rate signals. A constant bit rate requirement can be met by introducing buffer feedback mechanisms; however, this comes at the expense of increased encoding delays [96].

2.6.4. Psychoacoustic Models Used in Perceptual Audio Codecs

Psychoacoustic models influence perceptual codecs in a number of ways. The first is in the resolution of the time-frequency analysis stage. No matter which method is used to perform the time-frequency analysis, an inherent time-frequency resolution trade-off will exist. Intended input signal type and output signal quality will determine how accurately the model employed represents the actual analysis resolution of the cochlear (cf. section 2.5.4). One example of a resolution model that is a fairly close match to that of the Basilar Membrane is that used by the MPEG-1 layer III standard. Another that is commonly used is to assume that the Basilar Membrane consists of a set of 25 discrete band-pass filters as shown in Table 2-1 [96].
Another phenomenon of auditory perception that influences perceptual codec design is that of auditory masking. Auditory masking models may be used in two different ways by perceptual codecs. The first is in deciding which data from the time-frequency analysis are perceptually redundant. Perceptually redundant information will be those signal components that are masked by others. This redundant data can obviously be discarded with little, if any, effect on the perceptual quality of the encoded signal. One example of this use of masking is in the codec developed by Mahieux and Petit [98].

The second way that masking models are often employed in perceptual codecs is to perform what is known as noise shaping. The dominant (or perceptually relevant) signal components will not only mask other components in the original signal, but they will also mask any quantisation noise that fall within their corresponding masking patterns. Hence, once the overall masking pattern for a signal has been determined, it can be used as an upper bound below which any quantisation noise will not be audible. Thus the bit allocation procedure aims to produce a signal that will have a quantisation noise pattern that falls below the masking pattern. Johnston’s [99] perceptual codec is an example of this technique.

The final use of perceptual models in audio codecs is to vary the quantisation resolution of transform coefficients according to the amplitude of the spectral component. This
exploits the auditory system’s intensity resolution characteristics. This is very similar to companding although it now operates on individual frequency components rather than time domain samples.

2.6.5. Time-Frequency Analysis in Perceptual Coding Schemes

The time-frequency analysis block in Figure 2-18 may be any one of the following [96]:

- unitary transform;
- critically sampled, uniform, or nonuniform bandpass filter bank that is either time-invariant or signal-adaptive;
- sinusoidal or harmonic analysis;
- source-model analysis (LPC/multipulse excitation);
- hybrid transform or any two or more of the above.

Each of these schemes can, and indeed has, been used as a signal representation scheme without any regard to perceptual factors. Some, such as LPC, will produce significant compression even under these circumstances while others, such as a unitary transform, perform poorly from a coding perspective without significant use of a perceptual model. This section will discuss each of these methods in the context of perceptual coding with particular emphasis being placed on sinusoidal analysis for its relevance to the task of auditory source separation.

Unitary Transform Coding

As the name suggests, unitary transform coding schemes use a fixed resolution transform to perform the time-frequency analysis. The main advantage of using a transform technique is that, if the transform is selected carefully, a significant portion of the signal can be represented by much fewer parameters than are required to represent the entire signal.

The simplest and most straightforward transform coding technique involves using the STFT. However, because of the time-frequency trade-off involved (cf section 2.5.2), there is not much gain in terms of reduced data rate [26]. Methods to overcome this include changing the transform used and exploiting characteristics of human auditory perception.
One method for reducing the data rate in transform coding is to use a more efficient spectral transform. One such transform is the modified discrete cosine transform (MDCT) developed by Princen and Bradley [69]. The MDCT uses windows that are overlapped by 50% to increase time resolution but only half of the coefficients need to be recorded since the transform is perfectly symmetrical. Also the MDCT is purely real, thus resulting in even further savings over the FFT. However, perhaps the most efficient means for increasing the coding gain in a unitary transform is to fully exploit the masking model to remove redundant data and carefully shape quantisation noise as discussed in section 2.6.3.

**Sub-band coding techniques**
To further reduce the redundancy apparent in transform coding schemes, sub-band analysis can be employed. The relationship between sub-band coding schemes and transform coding schemes is similar to the relationship between the STFT and sub-band analysis techniques mentioned in section 2.5.3. In particular, transform coding can be viewed as a special case of sub-band coding where the bands are equally spaced [100].

There are two main advantages of using sub-band coding schemes with unevenly spaced bands. The first is that the frequency resolution of the auditory system can be exploited to generate fewer coefficients overall in the transform. Secondly, if the filter bank resolution accurately reflects the resolution of the auditory system, noise masking and elimination of masked coefficients can be performed more efficiently and with more controlled quality degradation. The desired quality, complexity and bit rate of the coder will drive the decision as to how closely the filter bank will match perceptual models.

Most existing sub-band coding schemes combine the computational efficiency of fixed-resolution transform methods with the benefits of perceptually tuned sub-band techniques. They tend to be implemented using a coarse perceptually tuned filter bank followed by a fixed-resolution frequency transformation (Fourier or cosine). These schemes are able to achieve high quality, wide band audio coding at relatively low data rates and at lower levels of complexity than would be required to achieve similar resolution using only a filter bank.

**Analysis-by-Synthesis Coding Techniques**
Analysis-by-synthesis methods aim to generate a parametric description of the waveform using an iterative procedure that reduces the error between the original signal and one generated using the last set of derived parameters. In essence, analysis-by-synthesis implies an active process that may be applied to signals that are produced by a generator.
whose basic properties are known [101]. This is the case in speech where vocal tract models have existed for some time, however, there is no general production model for other audio signals. Thus, analysis-by-synthesis techniques are generally applied only to speech signals.

One such technique is linear prediction coding (LPC). The basic aim of LPC is to derive a set of coefficients for a linear predictor of the form [102]:

\[
\hat{x}(n) = - \sum_{k=1}^{p} a_p(k)x(n-k)
\]

(2 — 24)

where \( \hat{x}(n) \) is the predicted value, \( a_p(k) \) are the predictor coefficients and \( p \) is the order of the predictor used. The predictor coefficients are derived by minimising the error, in a mean squared sense, between the predicted values, \( \hat{x}(n) \), and the actual signal values, \( x(n) \). That is:

\[
e(n) = x(n) - \hat{x}(n)
\]

(2 — 25)

The predictor coefficients are transmitted along with a residual that, roughly speaking, is the error signal, \( e(n) \), or a coded version of it. There are many variations of LPC with the most popular variants attempting to minimise the size of the residual. Code excited linear prediction (CELP) is one such scheme that uses vector quantisation to code the residual.

**Sinusoidal and Harmonic Coding Techniques**

Sinusoidal coding schemes for speech have their origin in the work of McAulay and Quatieri [73] who proposed a technique based on a model that represents the perceptually relevant portions of a unitary transform analysis as a series of sinusoidal tracks with time-variant frequency, amplitude and phase:

\[
s(n) = \sum_{l=1}^{L} A_l \cos(\omega_l n + \phi_l)
\]

(2 — 26)

where \( s(n) \) is the current analysis frame of the signal, \( L \) is the number of tracks and \( A_l, \omega_l, \) and \( \phi_l \) are the amplitude, frequency and phase respectively of the \( l \)th track in the current analysis frame. The tracks are composed from amplitude peaks in the time-
frequency distribution of the audio signal. Thus the only information required to represent the TFD are the parameters (amplitude, frequency and phase) describing each track.

Thus the basic principle behind sinusoidal coding is apparently elegant and straightforward. One particularly attractive feature of sinusoidal coding schemes is that, like source-model coding schemes, the parameterisation of the signal offers the potential for large coding gains with the added advantage that no source model is assumed. Thus these schemes are more generally applicable, and were used successfully to represent music, even before they were applied to speech signals [103].

However, there are a number of shortcomings and problems with the basic model described above. These include the resolution of the underlying transform, the method for extracting the required parameters and the inability of the model to adequately represent impulsive sounds. This section will review the evolution of sinusoidal codecs giving examples of the various methods that have been employed to address these shortcomings and challenges.

The most obvious improvement that can be made to the basic model is to change the resolution of the underlying time-frequency distribution either to more accurately reflect that of the auditory system or to more closely match the requirements of the input data. Anderson’s [104] implementation, for example, achieved a better model of the resolution of the human ear by using a quadrature mirror filter bank to generate the TFD. In their later work, McAulay and Quatieri used a signal-adaptive analysis window to more efficiently represent the data [29] [105]. The former technique, and its kin, is more appropriate when the representation is to be used for a task that emphasises the perception of the audio (such as information retrieval) while the latter is particularly suited to very low-rate coding applications where coder complexity is not an important consideration.

A simple enhancement to the basic sinusoidal model that can achieve a large coding gain is harmonic coding. This method was introduced by Almeida and Tribolet [106]. The basic idea is that each frame of the STFT is represented by a series of harmonic sinusoids. Thus the only required parameters are the amplitudes of the sinusoids and the fundamental frequency of the harmonic group. This method was applied to code the residual of an LPC scheme.
Parameter extraction is perhaps the most vital part of a sinusoidal codec. The initial model proposed by McAulay and Quatieri required that the time varying amplitudes, frequencies and phases be extracted from the underlying TFD. Essentially the algorithm described by McAulay and Quatieri in [105] involved selecting the local maxima in each frame of the FFT as candidate sinusoids and then simply taking the corresponding amplitude, frequency and phase information directly from the FFT as the instantaneous values for the sinusoidal component. To avoid having an excessive number of sinusoids, due to either transform or signal noise or the nature of the signal, an upper limit on the number of sinusoids was specified as a threshold value; the sinusoids that were retained were those that had the largest magnitudes. The McAulay and Quatieri parameter extraction method is more fully described in section 3.7.1.

The lack of a noise model is perhaps the most significant omission from the basic sinusoidal model described in [105]. Researchers have suggested several methods to overcome this. The most common basis of all the approaches involves using the extracted sinusoidal track parameters to synthesise an intermediate signal and then subtract this from the input signal to leave the noise portion, or residual. The concept is illustrated in Figure 2-19.

The most common point of difference among the various noise models employed in sinusoidal codecs is the manner in which the noise data is represented. Serra [107], for example, represented the noise portion as the output of a time-varying filter driven by a white noise source. This is similar to the way in which the residual is coded in basic LPC.

One problem that Serra noted with this approach is that it is insufficient to represent musical sounds that fall somewhere between noise and tonal, such as can be found in the high frequency partials of string instruments played with vibrato. Serra termed these problem components ‘noisy partials’ [107].
A more perceptually motivated noise model is that proposed by Goodwin [108]. In this model, the residual signal is transformed into the frequency domain by means of a FFT which is then subdivided into critical bands using the equivalent rectangular bandwidth (ERB) model of the cochlea (cf. section 2.2.3). The energy in each band is calculated and these values form the parameters of the noise model. While being more perceptually motivated, it still lacks the ability to represent the noisy partials.

A recent improvement that addresses the problem posed by noisy partials is the sines + transients + noise (STN) model introduced by Levine and Smith [109]. The original motivation for this representation was to achieve independent time and pitch scale modification of audio signals, a task where an accurate model of all the audio components was particularly important. The transient portions are found using a combination of two separate methods. The first involves looking for rising edges in the energy of the signal computed over short time-frames, while the second involves comparing the short-time energies of the residual and original signal. Once a transient section has been detected it is encoded using a simplified version of the MPEG-AAC (Advanced Audio Coding) system.

Painter and Spanias [110] suggested an improvement to the basic STN codec that involves a more perceptually motivated approach to selecting the sinusoids. As in the basic sinusoidal model, the STN codec of [109] selected the sinusoids based purely on the relative magnitude of local maxima in each transform frame. Painter and Spanias’ method is based on a process called excitation similarity weighting (ESW) that maximises the match between the excitation pattern generated by the original signal and the corresponding pattern generated by the signal reconstructed from the sinusoids. It is therefore an analysis-by-synthesis approach and achieves much better perceptual quality than the traditional method.

Parametric coding schemes have found application in the MPEG4 audio coding standard. The Harmonic and Individual Lines plus Noise (HILN) [111] coder is essentially a hybrid between the sinusoidal model of McAulay and Quatieri, a harmonic model similar to that of Almeida and Tríbolet and a noise model for the residual signal. After the data have undergone sinusoidal analysis, the tracks are analysed to extract harmonic groups, which are encoded as for the harmonic model. The remaining tracks are then encoded individually as for the sinusoidal model. Finally, any residual is encoded using the noise model.
2.7. Audio Information Management Techniques

2.7.1. Introduction
Despite increasing interest in multimedia information management, the audio domain has traditionally been relegated to the “too hard basket.” While many researchers focussing on general multimedia have avoided the peculiar challenges of audio retrieval by mentioning the audio domain as an area of future research, a few recent systems have focused on audio retrieval. This section reviews the audio information management systems that do exist as well as the various issues that have been raised both by the general multimedia and the audio information management communities.

2.7.2. General Audio Content-based Retrieval Systems
The architecture of existing multimedia information systems tends to follow that shown in Figure 2-20.

![Figure 2-20: Multimedia archival architecture](After [59].)

The situation depicted in Figure 2-20 applies if “content analysis” is assumed to take any one of the forms described in section 1.2.6 (ranging from human interpretation to statistical analysis). It is evident that the index and data are stored separately. This is an undesirable situation since it creates processing and storage overheads to data that are by nature voluminous.

Other drawbacks of the architecture shown in Figure 2-20 were mentioned in the previous section. These include the limitations on retrieval scope imposed by the content analysis and lack of structure in the data. Although the index gives some measure of random accessibility, it is still impossible to access individual segments of the data that do not fulfil any of the index criteria. Using the text retrieval analogy, this situation is akin to attempting to locate a particular, partially memorised, quote in a book using only its table of contents. It may even be possible that the index would not be useful since the recalled portion may not contain any significant ‘topic words’. Obviously, the most ideal
way to find it would be via a string search. This example highlights the importance of some content information being directly visible in the stored data.

Recognising that success in audio content-based retrieval will most likely be achieved by modelling the human perceptual process, Klapuri [112] suggested an architecture for a “blackboard” system that allows for several perceptually motivated analysis algorithms to compete. This concept is similar to what has been postulated to occur in the human auditory system where multiple parallel hypotheses compete before a final decision is made [1]. The architecture of Klapuri’s system is shown in Figure 2-21.

![Figure 2-21: Overview of Klapuri’s blackboard architecture (After [112])]({"width":650,"height":420})

The system depicted in Figure 2-21 is interesting in that its modular construction allows multiple algorithms to be tested or to contribute to the overall identification of the audio object. Although the system as depicted Figure 2-21 would appear to only seek to give semantic labels to isolated audio objects, the blackboard architecture is intended to serve for any audio content analysis problem, including separating a mixed source signal into individual audio objects. The work reported here also recognised that concept of allowing multiple hypotheses to compete was an important factor and an initial attempt to model this process was made. Using the various algorithms that have been developed as knowledge sources in Klapuri’s blackboard architecture would be an interesting field of future study. In any case, Klapuri’s architecture is clearly superior to the traditional one of Figure 2-20 in that it is more general and, if combined with the work reported here, would allow for a “self-indexing” representation to be developed, thereby alleviating the need for a separate index.
2.7.3. **Content-based Browsing of Audio and Hyperaudio**

It was made clear in the introduction that Bush’s hypermedia model provides a powerful tool for information management. Yet because of a number of difficulties with the medium, little research has been performed in the area. Ridgeway and De Roure suggest three of these difficulties [113]:

**The lack of technology.** The first generation of hypermedia systems did not have the computer technology to manipulate audio data. Over time this technology has improved.

**The HCI (Human-Computer Interaction) challenges,** such as browsing audio information. For example, audio does not have a unique visual representation and therefore developing intuitive graphical interfaces, to manipulate audio, is quite difficult.

**The problems of the audio file formats.** Currently there are several file formats which can be used to store audio data. However with each format the file size increases as the quality of the audio recording increases. As a result, high quality recordings can rapidly consume large amounts of disk space. Working in the compressed domain raises some new challenges.

Having identified these issues, Ridgeway and De Roure then focus primarily on extending a multimedia streaming protocol, Real-time Transport Protocol (RTP) to allow audio data to fit the Fundamental Open Hypermedia Model (FOHM). FOHM is simply a protocol for the description of the link structure in accordance with the model presented in the introduction to this thesis (c.f. Figure 1-1). Hence, essentially only the HCI issue has been addressed. The remaining two issues will now be investigated.

Assuming that “computer technology to manipulate audio data” means the technology to present the data to the user (i.e. audibly playback) then Ridgeway and De Roure are correct in identifying this issue as essentially having been solved. The third issue, on the other hand, has already been shown to be a significant issue awaiting a satisfactory solution (c.f. Section 1.2). In a manner analogous to much of the research in multimedia content-based retrieval, this work essentially presented a framework for hyperaudio without any real discussion of how it might be achieved.

An important issue that Ridgeway and De Roure neglected to mention is that a major reason that hypermedia research has not been generally extended to the audio domain is that the problem of temporally co-located nodes is unsolved. In this the FOHM model
and the MPEG-7 standard, which aims to “provide a rich set of standardised tools to describe multimedia content,” [114] share common ground. They both implicitly rely on the data to be subdivided into objects yet neither describes any mechanism to achieve this. Thus, the importance of source separation to hyperaudio research is paramount.

Arons proposed a system for speech browsing called Hyperspeech [115]. It is apparently the only true hyperaudio scheme in existence. The system allows a user to navigate through recorded speech in a non-linear fashion using voice commands. The non-linear organisation is created at design time by defining speech nodes (spoken phrases or sentences) with either the aid of a manually generated transcript or a tightly constrained interview technique at record time. Thus the system is not very general or practical for large collections. However, the interface and presentation issues discussed are of relevance to the development of a general hyperaudio system.

2.8. Summary
This chapter has provided a background and context for the work reported in this thesis. Since the work is based on perceptual principles, the chapter opened with a brief introduction to the human auditory perceptual system. The most important factor of this system, from the point of view of the work presented here, is the postulated mid-level mental auditory representation that concluded the section on auditory perception (c.f. section 2.2.4). This representation consists of three basic primitive elements that can be derived from a 3-dimensional (time, frequency and amplitude) description of an audio signal: tone burst, sweep and noise burst.

Section 2.3 was devoted to explaining what is known of how the brain organises these primitive elements to make sense of one’s acoustical environment. It was stated that the field of study concerned with this investigation was named auditory scene analysis by Bregman drawing an analogy to the visual domain. The section reviewed the principles that Bregman postulated govern stream segregation and evaluated how each could be exploited to achieve the aims of the current work.

Having laid the foundations of low to mid-level auditory perception and the early perceptual stages in the decomposition of a cacophony into a series of sound objects, the next section, 2.4, described how the physical characteristics of a given audio signal are related to the percept that it induces. To the human listener, this study is most relevant in the context of individual sound objects. To give an extreme example, one naturally speaks of the pitch of a given note played by a single instrument rather than the “average
pitch" of the ambient noise. The review considered the principal audio characteristics of loudness, pitch, melody, rhythm, timbre and SenticS, noting directions for future research and relating how the current work may be extended to support the description of audio objects in terms of these characteristics.

The remainder of the chapter was devoted to laying the analytical foundations for the current work and pointing forward to applications of the same. This discussion began with a review (in section 2.5) of audio signal analysis techniques beginning at the simplest time- and frequency-domain methods, moving onto the more useful but more complex time-frequency methods through to perceptually motivated extensions of the latter. The section concluded with a review of one of the two principal methodologies for decomposing an audio signal into its separate sources: blind source separation. The various BSS solutions to this problem were reviewed and the fundamental limitations of the approach were explored and contrasted with the advantages of the computational auditory scene analysis approach. Since this latter approach was the one adopted in the current work, a review and discussion of the various CASA techniques is found in Chapter 4 where it serves as an introduction to the techniques developed.

Having provided the motivation for the specific approach that has been adopted in this thesis, the final section critically reviews the existing work in audio information management with a view to how the current work may address some of the shortcomings that have been identified. Two chief areas of concern were identified: 1) providing for efficient storage and transmission of audio data at the same time as allowing for meaningful and efficient access presents something of a paradox: improving one tends to negatively impact the other and 2) the ability to organise audio data in a structured manner, such that it might fit the hypermedia model, for example, is well beyond the scope of current technology. Chapter 1 identified that the principal cause of both of these problems is the traditional reliance on unstructured audio data representations. The foundation of this theses’ hypothesis is that a perceptually based, structured audio representation will address the abovementioned shortcomings. The remainder of this thesis concerns itself with the description of this representation and its application to the initial stages in audio information management.
Melih, K. *Audio Source Separation...for Content-based Coding and Management*
Chapter 3

The Data Representation: A Basis for Auditory Source Separation

3.1. Introduction
The source separation technique described here relies on an audio representation that is similar to the sinusoidal representations presented in section 2.6. The first stage in deriving this representation is to generate a time-frequency distribution of the incoming audio signal and to process this distribution to extract the amplitude peaks from it. Having obtained a series of peaks, these are used to generate a preliminary series of candidate tracks, which serve as the input to the grouping algorithms described in the next chapter. This chapter details the development of the underlying TFD and describes the algorithms developed to extract the preliminary tracks.

3.2. Basic Architecture for Auditory Source Separation
In most CASA systems, auditory source separation involves extracting a set of low-level partials from the output of the peripheral auditory model and then collecting these partials into groups, which form the basis of auditory objects or streams. A number of different methodologies have been proposed to achieve this, however, they can all be generalised to require three basic steps:

1. Identify important regions in the output of the peripheral model;
2. Organise these regions into time-frequency localised partials; and
3. Organise the partials into groups/objects/streams

Although the above steps are implicitly required, not all systems perform them explicitly nor do they all perform them in the sequence indicated above. The most common departure from the above sequence is in employing a feedback mechanism whereby the results of step 3 are fed back to influence the operation of step 2. The various architectures adopted by existing CASA systems are summarised by Figure 3-1.

The chief differences between the existing computational models lie in the details of each of the modules depicted in Figure 3-1 as well as which, if any, of the feedback mechanisms depicted is/are present. One area of variation is the similarity of the peripheral processing module to the physical system it seeks to model, that is, the ear and cochlea. Possibilities range from models that attempt to model each component of the
auditory periphery and cochlea as accurately as current knowledge of these systems allows [116] through to systems that simply approximate the function of these systems based on the relationship between the input and output signal [117]. Another difference is in whether the partial extraction is influenced by the perceptual grouping procedure via feedback mechanisms [118] or without feedback [118].

The approach that has been adopted in this thesis is illustrated in Figure 3-2. It is essentially similar to that of Figure 3-1 where only the inner feedback loop between the mid-level grouping and low-level feature extraction processes has been retained. However, in the traditional systems represented by Figure 3-1, the more commonly adopted feedback paths are that from the streaming stage to the low-level feature extraction and mid-level grouping stages. The high-level streaming stage actually represents the mid stages of auditory cognition whereas in the system shown in Figure 3-2, the feedback represents the lower level-level signal detection and the very earliest stages of cognition. The higher level processes typically rely on learned responses and may imply semantic analysis while the lower level processes appear to be governed by innate processes. The advantage of adopting the approach illustrated in Figure 3-2 is that in doing so, the resultant representation is not shaped by any semantic process, which would be likely to limit the generality of the representation.
3.3. Models of the Auditory Periphery

In modelling the auditory periphery, the chief aim is to produce a time-frequency distribution such as that described in section 2.2.4. Some systems begin by accounting for the filtering effect of the pinna and middle ear by applying a pre-emphasis filter. For example, Grossberg [119] used the following difference equation to pre-emphasise the frequencies between 100 Hz and 5000 Hz:

\[
y(t) = x(t) - 0.95 x(t - 1.25 \times 10^{-4})
\]  

(3 – 1)

The next important stage in the auditory periphery is the Basilar Membrane. As was detailed earlier, the primary role of the Basilar Membrane is to transform the incoming signal into the time-frequency domain. To model this time-frequency decomposition, three main options are available:

1. Band-pass filterbank as in [118] [120] [121];
2. Short-time Fourier transform (or other transform approach) as in [117] [122] [123]; or
3. Hybrid approach – not currently used in CASA systems but common in audio coding systems [124].

The Gammatone Filterbank

The gammatone filter is the most commonly employed in filterbanks for CASA systems. A gammatone filter aims to model the auditory nerve firing response as a result of Basilar Membrane activity. The gammatone function is defined as follows [118]:

\[
g(t) = t^{n-1} e^{-bt} \cos(\omega t) u(t)
\]  

(3 – 2)

where \( n \) is the filter order, \( b \) is related to the desired bandwidth of the filter, \( \omega \) is the centre frequency of the filter in radians and \( u(t) \) is the unit step function:

\[
u(t) = \begin{cases} 
1 & \text{if } t \geq 0 \\
0 & \text{if } t < 0
\end{cases}
\]  

(3 – 3)

Patterson et al showed that a fourth order gammatone filterbank is able to accurately model the \( \text{roex}(p) \) function over a 60 dB range [125]. Hence, fourth order gammatone filterbanks are common [119][118][126].

Solbach proposed an interesting variation on the gammatone filterbank based on the lowpass version of the gammatone filter after [121]:
Solbach multiplied this lowpass prototype by $2e^{j2\pi f_0 t}$ to produce a wavelet-style transform:

$$g(t) = 2t^{n-1}e^{-bt}u(t)$$  \hspace{1cm} (3 — 5)\]

The justification for this variation was that it offered “a simplified interpretability of the filter outputs in terms of instantaneous amplitude, phase and frequency.” This might appear to represent either a transform or hybrid approach, however, it is simply an alternative implementation of a classical filterbank.

The principle disadvantage of filterbank approaches is the computational complexity. To accurately model the cochlea resolution requires approximately 100 filters for a bandwidth of only 6500 Hz [118]. This is obviously an expensive operation. In cases where psychoacoustic accuracy is required, such as in developing models to aid the psychoacousticians in their research, there is little alternative other than optimising the implementation of the filterbank. However, when a close approximation will suffice, transform methods offer a more efficient alternative.

**Time-Frequency Transform-based Models of the Auditory Periphery**

The McAulay and Quatieri [73] sinusoidal representation of audio was recognised as a potential basis for CASA by several researchers including Maher [117] and Ellis [72]. As is the case in speech coding applications, the chief concern with the TFD upon which the model is based is the time-frequency resolution trade-off. One common solution to this resolution problem employed by CASA researchers is to adopt a constant-Q transform [127][128].

The basic idea of constant-Q analysis is that the FFT is viewed as a series of bandpass filters with a constant Q factor (i.e., the ratio of centre frequency to bandwidth). A constant Q transform is realised by performing a series of FFTs on various length windows and only retaining the outputs from each transform that comply with the constant Q requirement. Calculating many unused coefficients to a large extent negates any computational advantage that the transform techniques have over the filterbank implementation.

Another solution to the time-frequency resolution problem that has been offered is to employ a different transform. One such alternative is the Wigner distribution, which
does not assume that the signal is stationary over each analysis frame. Wells and Murphy propose a partial tracking algorithm (the second stage in the generalised CASA system shown in Figure 3-1) based on a Wigner distribution [129]. However, a major drawback with such distributions is that their non-linear analysis results in cross-terms appearing in the distribution, which are likely to be mistaken as actual signal components.

Hybrid Approaches
In order to overcome the computational expense of an accurate filterbank implementation and the inadequate resolution of a fixed resolution transform approach, a hybrid approach may be used. The basic idea is to cascade a coarse filterbank analysis with a fixed-resolution transform at the output of each filter. While this technique will not achieve the perceptual accuracy of a full filterbank implementation, it provides a sufficient compromise for many applications such as in perceptual audio coding [124]. Hence, it was postulated, that for the work described in this thesis, such a representation would be sufficient.

3.4. Time-Frequency Distribution

3.4.1. Selection of Approach
From the early design stages it was recognised that the main challenge in designing the time-frequency distribution was the inherent time-frequency resolution problem described in section 2.5.3. The available alternatives were:

- **A fixed rate short-time transform.** As already mentioned, a fixed rate transform presents a problem in that it is impossible to have good temporal resolution simultaneous with good frequency resolution. This is also a poor model of the auditory system’s frequency dependent resolution; hence, it comes of little surprise that few, if any, CASA systems have adopted this approach. Quatieri and Danisweicz [130] did propose a signal enhancement technique based on a fixed rate transform and the system of Wells and Murphy is based on a STFT, however, because of the inherent resolution problems, this is cascaded with a Wigner-Ville transform [123]. Given its perceptual incongruence, a constant resolution transform was deemed unsuitable for the work presented here.

- **A perceptually tuned band-pass filterbank.** An absolute minimum of 25 filters would be required to simply account for each critical band in the normal hearing range. Many more would be required to provide a good
approximation to the resolution of the Basilar Membrane. While arguably providing the best model of the actual auditory system, and often used by CASA researchers [118] [120] [121], this technique is obviously computationally intensive. Given that one of the aims of the work described here was to develop the first stages of a system with applications to content-based coding and information management, optimising computational complexity was an important concern. Hence, this technique was rejected on the basis that it would be infeasible in practice.

- **A constant-Q transform or filterbank.** A constant-Q transform is a special case of a bandpass filterbank where Q factor, or ratio of centre frequency to bandwidth, remains constant. This gives a rough approximation of the time-frequency resolution of the perceptual system while slightly reducing the complexity of the filterbank. The system initially proposed by Ellis adopted this approach [72]. A constant-Q filterbank can also be approximated by taking a series of fixed-rate transforms and retaining only those coefficients that are of interest in each resolution band. The basic concept is illustrated in Figure 3-3. The main drawback with this approach from the point of view of the algorithms presented later in this thesis is that, while it provides a good approximation of the auditory system’s resolution, it discards data that can be useful in algorithms where statistical redundancy is exploited.

![Figure 3-3: Basic mapping between transform window size and resolution of a constant-Q transform](image)

- **A hybrid filter-bank and short-time transform technique.** A classic method, commonly employed in audio coding, of resolving the time-frequency resolution problem is to initially apply a coarse filtering to the incoming data followed by applying a fixed resolution transformation to the output of each filter. This provides a good compromise between computational complexity and approximation of the time-frequency resolution of the auditory system.
By employing a filterbank with overlapping filters the statistically redundant data can be retained for use by later grouping algorithms. Another advantage of using overlapping filters that was recognised is that the redundancy in the data can be exploited to effectively improve the overall resolution of the transform across the entire range, as illustrated in Figure 3-4. Despite these advantages, as far as the author is aware, no previous CASA system has adopted such a transform.

![Figure 3-4: Resolution of hybrid low-pass filter-bank followed by a fixed resolution transform in each band](image)

Unlike the constant-Q filter bank, the bands overlap at low frequencies.

Given the foregoing discussion of alternatives, it should be obvious that the compromise represented by the hybrid approach is in agreement with the basic aims stated in the introduction to this thesis of developing a system that is perceptually motivated while remaining computationally feasible for practical application to audio information management and coding tasks.

### 3.4.2. Filter-bank resolution

The initial filter-bank design attempted to model the critical band scale as closely as possible. The band distribution and its relation to the critical band scale under this design is summarised in Table 3-1.

Two obvious points arising from the data in Table 3-1 are the apparently arbitrary assignment of band locations and the still relatively large number of filters required (15). The latter of these points may only be addressed by changing the design criteria and will be discussed later in this section. The former is easily accounted for by noting the design parameters employed.
Table 3-1: Initial filter-band band positions relative to critical band scale

<table>
<thead>
<tr>
<th>Critical Band Number</th>
<th>Lower Edge (Hz)</th>
<th>Upper Edge (Hz)</th>
<th>Analysis Band Number</th>
<th>Critical Band Number</th>
<th>Lower Edge (Hz)</th>
<th>Upper Edge (Hz)</th>
<th>Analysis Band Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>100</td>
<td>1</td>
<td>14</td>
<td>2000</td>
<td>2320</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>200</td>
<td>2</td>
<td>15</td>
<td>2320</td>
<td>2700</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>300</td>
<td>3</td>
<td>16</td>
<td>2700</td>
<td>3150</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>400</td>
<td>4</td>
<td>17</td>
<td>3150</td>
<td>3700</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>510</td>
<td>5</td>
<td>18</td>
<td>3700</td>
<td>4400</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>510</td>
<td>630</td>
<td>5</td>
<td>19</td>
<td>4400</td>
<td>5300</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>630</td>
<td>770</td>
<td>6</td>
<td>20</td>
<td>5300</td>
<td>6400</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>770</td>
<td>920</td>
<td>7</td>
<td>21</td>
<td>6400</td>
<td>7700</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>920</td>
<td>1080</td>
<td>7</td>
<td>22</td>
<td>7700</td>
<td>9500</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>1080</td>
<td>1270</td>
<td>8</td>
<td>23</td>
<td>9500</td>
<td>12000</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>1270</td>
<td>1480</td>
<td>8</td>
<td>24</td>
<td>12000</td>
<td>15500</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>1480</td>
<td>1720</td>
<td>8</td>
<td>25</td>
<td>15500</td>
<td>16000</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>1720</td>
<td>2000</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first step in determining the band allocation shown in Table 3-1 was to determine the JND for frequency at the lower edge of each critical band using the formula:

$$\log(\Delta F) = a\sqrt{F} + k + \frac{m}{SL}$$

(3-6)

with parameter values $a = 0.0214$, $k = -0.15$, and $m = 5.056$ as given in [9] and the sensation level selected to be $SL = 60$ dB, representing a mid-range value. The ideal window length for the transform to be applied to the output of each filter could then be calculated as $\frac{1}{\Delta f}$. The Nyquist rate for each critical band was also determined.

Given the values calculated as detailed in the previous paragraph, the following constraints were applied to determine the final band allocation:

i) The sample rate for every band should be a factor of the original 32kHz sample rate, to simplify the down-sampling operation;

ii) The transform window length must be a factor of two to allow for fast implementation of the transform; and

iii) The actual frequency resolution in an analysis band may be coarser than the lowest JND value but by no more than 10%. For example, if a given band is proposed to encompass two critical bands with JND at the lower
edge of, say, 2 Hz and 4 Hz respectively, the minimum allowable window length would be $\frac{1}{2^2} = 0.45$ seconds.

Table 3-2 summarises the foregoing calculations.

Yet another issue, contrary to the aim of producing an efficient system, which is apparent in Table 3-2, is the diverse window sizes of the transforms employed. Additionally, it will be noted that in many instances the resolution achieved by the above band allocation...
is far in excess of that required by the JND for that critical band. In practice, natural sounds will not contain significant components that are any closer than several times the JND in a band, so such an accurate analysis is not particularly necessary and therefore, unjustified given the various computational issues mentioned.

Hence, a different band allocation was sought with the following requirements:

- The band allocation should seek to optimise the number of filters required such that an adequate approximation to the critical band scale is achieved with the minimum number of filters;

- The sample rate in each band should be a factor of the original 32kHz sample rate to simplify down-sampling. A simple relationship between the sample rates in each band was also a desirable characteristic that would further simplify the computation of the overall TFD;

- A single window length (in samples) for the transform should be used across all bands, to simplify buffering requirements; and

- The minimum temporal resolution should be at least ½ of the 30 ms resolution in onset/offset detection [1].

With these considerations in mind, it was noted that a 16 ms window would give a frequency resolution of 62.5 Hz which is far better than that required in the highest critical band. Although 16 ms is slightly longer than the required 15 ms, the temporal resolution of the transform is easily increased by overlapping the windows by 50%. It was also noted that a 16 ms window at 32 kHz is 512 samples long. Further, if each subsequent band is down-sampled by a factor of 2, the same window length of 512 samples produces a frequency resolution that is half that of the previous band. This results in a band allocation shown in Table 3-3.

<table>
<thead>
<tr>
<th>Band Number</th>
<th>Bandwidth (kHz)</th>
<th>Ideal Frequency Resolution (Hz)</th>
<th>Frequency Resolution (Hz)</th>
<th>Temporal Resolution (ms) (after 50 % overlap)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>70.5</td>
<td>62.5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>19.4</td>
<td>31.25</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7.8</td>
<td>15.625</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4.1</td>
<td>7.8125</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2.6</td>
<td>3.90625</td>
<td>128</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.9</td>
<td>1.953125</td>
<td>256</td>
</tr>
</tbody>
</table>
While the critical band boundaries are not strictly observed, the frequency resolution in each band shown in Table 3-3 is less than twice the JND for the lowest edge of each band. Given the many computational advantages of this approach it was decided that it should be used in preference to the previous allocation.

3.4.3. Filter Design

The design of the filter was governed by the following set of requirements:

- Sharp cut-off to avoid incorrect peak magnitude estimation at band boundaries;
- Highest achievable stop-band attenuation to avoid peak tracking problems caused by aliasing; and
- Linear phase so that phase information was not adversely affected as it was postulated that phase would be required for a good quality reconstruction and especially to preserve timbral information.

It will be noted that the above requirements are relatively strict. This may seem at variance with the stated aim of minimising system complexity. However, this aim was subordinate to developing a robust proof of concept system. Therefore, since the order of the filter did not have as significant an impact on overall system performance as did the number of bands, it was elected to use the relatively high order filter that resulted from satisfying these strict design criteria. Having developed and validated the algorithms, optimisations across the entire system are left as an area for future work.

The low-pass filter is a 177-order FIR equi-ripple filter designed using MATLAB. The impulse, magnitude and phase responses of the filter are shown in Figure 3-5 and Figure 3-6.

The implementation of the filter is a simple direct form implementation of the convolution sum:

\[
\hat{s}(n) = \sum_{m=0}^{M-1} s(n-m) h(m) \tag{3-7}
\]

where \(s(\bullet)\) is the original signal, \(h(\bullet)\) are the filter coefficients and \(M\) is the filter length (177).
3.4.4. Choice of Transform

There are many transforms that could have been employed in the time-frequency analysis. During system development, the following three were closely considered:

1. DCT (or MDCT);
2. DHT; or
3. FFT.
Modified Discrete Cosine Transform (MDCT)

The MDCT of a signal \( s(n) \) is given by [69]:

\[
X(k) = \sum_{n=0}^{2N-1} x(n) w(n) \cos \left( \frac{\pi}{2N} \left( 2n + 1 + \frac{N}{2} \right) \left( 2k + 1 \right) \right)
\]

\( k = 0, 1, 2, \ldots, \frac{N}{2} - 1 \) \hspace{1cm} (3 — 8)

It is evident from equation 3 — 8 that, for a window length of 2N samples, only N coefficients need to be calculated. This in contrast with a FFT where the same number of coefficients as input samples must be computed despite only 50% of these providing unique information. Further, the transform coefficients of the MDCT are real. Hence, when compared to the FFT, there is a 75% reduction in the number of values that need to be calculated (complex numbers count as two values each) and a 50% reduction in the amount of data that needs to be stored. In early design of the system, these considerations were thought to be advantageous from data compression and processing complexity perspectives.

However, it will be noted that, being a real valued transform, the MDCT does not provide any explicit phase information. This was not initially considered a problem, as it is generally accepted in the audio perception research community that the auditory system is insensitive to phase. However, initial experiments with an implementation very similar to the McAulay and Quatieri system [73] where the FFT was replaced with the MDCT produced disappointing results with very poor perceptual quality in the inverted signal.

In an attempt to account for this poor performance, a closer investigation of the peaks that were being derived from the transform was performed. This investigation involved analysing a 175 Hz tone using both the MDCT and FFT with the filter-bank arrangement described in section 3.4.6. The 175 Hz tone was selected as being a frequency that is not a harmonic of any of the analyses basis functions. This represented a “worst case scenario” to ensure that any problems, which might occur with either of the transforms, would be clearly visible. Figure 3-7 and Figure 3-8 show the results obtained. Note that both figures are drawn to the same scale.
From Figure 3-7 and Figure 3-8, it is manifest that the frequency “drift” between one frame and the next for the MDCT-based analysis is far greater than that for the FFT-based analysis. This can be accounted for by realising that the phase information in the signal cannot simply disappear and it is not entirely unexpected that it would result in an “error” in the frequency judgement given the close coupling between frequency and phase.
One thing that might seem unusual about the FFT plot in particular is that, despite the 175 Hz tone being theoretically irresolvable by the FFT analysis, the frequency of the tone was clearly correctly identified within very small error bounds. This is because the peaks shown in Figure 3-7 and Figure 3-8 represent those generated by the interpolation procedure detailed in section 3.6.2 thus these results also incidentally demonstrate that the interpolation procedure used in conjunction with the FFT is itself effective.

**Discrete Hartley Transform**

The second transform to be considered was the Discrete Hartley Transform (DHT) [66]:

\[
X(k) = \sum_{n=0}^{N-1} x(n) \left( \cos \left( \frac{2\pi nk}{N} \right) + \sin \left( \frac{2\pi nk}{N} \right) \right), \quad k = 0, 1, 2, \ldots, N-1
\]  

Equation 3—9 shows that, like the MDCT, the DHT is a purely real transform. It thus offers similar advantages in terms of computational complexity and data storage requirements. In their raw form, the DHT coefficients have no frequency domain interpretation, however, a simple relationship between DHT and the FFT does exist [66]:

\[
\Re \{ X_f(k) \} = \frac{X_h(k) + X_h(-k)}{2}
\]  

\[
\Im \{ X_f(k) \} = \frac{X_h(k) - X_h(-k)}{2}
\]

where \( X_f(k) \) represents the Fourier transform coefficients and \( X_h(k) \) represents the Hartley transform coefficients.

Given the relationship described by equations 3—10 and 3—11, we can use the DHT to generate the required short-time spectrum of the data. However, since we are now left with a standard FFT, there is no storage advantage to be gained. There is arguably a gain in reduced computational complexity by avoiding complex arithmetic. Nevertheless, as these are purely implementation issues, this option was not pursued.

**Fast Fourier Transform**

Given the considerations in the proceeding two sections, it was finally decided to use a standard FFT to achieve the frequency decomposition. The magnitude coefficients for the TFD are given by the relationship [124]:

\[
X(k, m) = PN + 10 \log_{10} \left| \sum_{n=0}^{N-1} w(n) x(n, m) e^{-\frac{2\pi kn}{N}} \right| \text{ dB SPL}
\]  

\[ (3—12) \]
where $m$ is the analysis frame number; $n$ is the time index; $N$ is the window length (512); $k$ is the frequency index; $PN$ (97) is a power normalisation factor; $w(\bullet)$ is an analysis window; and $x(n, m)$ is a scaled version of the input signal [124]:

$$x(n, m) = \frac{s(n, m)}{2^{b-1}}$$  \hspace{1cm} (3—13)

where $b$ is the sample resolution in bits.

The power normalisation factor, $PN$, in equation 3 — 12 is used to scale the FFT coefficients so that they can be interpreted in terms of dB SPL. Scaling of the amplitudes in this manner is required for calculation of masking thresholds and also provides a more perceptually congruent view of the peaks, which is useful during the track formation and grouping stages. It must be noted that the SPL value thus calculated can only give a conservative estimate of the actual percept induced by the signal since the actual value will depend on the presentation conditions (i.e. play back volume, loudspeaker type etc.) which are entirely at the control of the user and thus cannot be determined \textit{a priori}. The value of $PN = 97$ was selected on the basis that a full amplitude tone of 1kHz is assigned a SPL of 85 dB SPL. This approach is consistent with general practice in audio coding applications [124].

3.4.5. \textit{Windowing and Buffering}

The window function, $w(\bullet)$, in equation 3 — 12 is used to reduce the Gibbs phenomena, which is manifested by ‘ringing’ in the frequency domain. There are several standard windows to select from, and the effect that a majority of these have on the FFT analysis of a 1 kHz tone is shown in Figure 3-11. Given the plots in Figure 3-11, it was decided that the Hann (also known as the Hanning) and Blackmann windows present the best compromise between main-lobe width and side-lobe attenuation. Equations 3—14 and 3—15 define the Hann and Blackmann windows respectively. The Hann window was selected since it requires only approximately half the number of operations to calculate than the Blackmann window.

$$w(n) = 0.5 \left[1 - \cos\left(\frac{2\pi n}{N}\right)\right]$$  \hspace{1cm} (3—14)

$$w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N}\right) + 0.8 \cos\left(\frac{4\pi n}{N}\right)$$  \hspace{1cm} (3—15)

The windowing operation is also influenced by the group delay of the filter, which is 88 samples. Since the first band does not need to be filtered, it will be 88 samples in advance
of the second band. Further, as the signal used to generate the third analysis band has passed through the filter a second time, this signal is delayed 88 samples with respect to the second analysis band. However, since the signal was downsampled at the second stage of processing, it has now been delayed by \(88 + 2 \times 88 = 264\) samples with respect to the first analysis band. This pattern continues for the remaining bands in the series.

To compensate for these varying delays, the first window of samples in each band is advanced by 45 samples \(\left(\frac{\text{group delay}}{2} + 1\right)\). This represents a delay with respect to the bottom (or final) band. In order to allow for proper window alignment, the first window in band 0 is also delayed by 3 samples. Further to improve temporal resolution and ensure that the windowing operation has no nett effect on the amplitude of the time-domain signal, adjacent windows are overlapped by 50% which gives a temporal resolution of each band of 256 samples.

Given the above considerations, each window of samples in a given band can be expressed as:

\[
s_b(n, m) = \begin{cases} 
  s_b(3 + 256m + n) & \text{if } b = 0 \\
  \frac{s_{b-1}(2(3 + 256m + n))}{s_{b-1}} & \text{if } b > 0
\end{cases} \tag{3—16}
\]

where \(s(\bullet)\) is the original signal; \(s_b(\bullet)\) is the signal in the \(b^{th}\) band; \(\hat{s}_b(\bullet)\) is the filtered signal in band \(b\); \(n\) is the sample number in the window and \(m\) is the window (frame) number. The factor of two in the second equation accounts for the down-sampling operation. Equation 3—16 is illustrated in Figure 3-9 and the apparent window alignment that results is shown in Figure 3-10.
Figure 3-9: Window alignment to take compensate for filter delay
Figure 3-10: Apparent window alignment
Figure 3-11: FFT analysis of a 1 kHz tone with commonly used windows applied. 
Note that on each plot the abscissa represents the FFT sample number and the ordinate represents the magnitude values.
3.4.6. **TFD implementation**

Given the design considerations detailed in the previous sections, the final architecture of the TFD generation algorithm is as shown in Figure 3-12.

![Figure 3-12: Architecture of TFD generation system](image)

Figure 3-12 makes it obvious that TFD generation is essentially a parallel process. The incoming raw audio signal is buffered until a window’s worth of samples have been received. This window of signal is then passed through a delay, a Hann windowing function and finally an FFT. At the same time, the same window of signal samples is split off and passes through a filter and sub-sampler into a second buffer. When this second buffer is filled, the samples pass through a delay, Hann window and FFT as for band 0. This process proceeds down through the bands. The result of this process is a series of 5 time-frequency distributions of varying temporal-spectral resolutions that are time-aligned.

3.5. **Masking Thresholds**

It has been noted that simply in the process of peak picking a level of simultaneous masking is implicitly applied [104]. Also, while developing the tracking algorithms, it was noted that a small amount of noise in the data is desirable as the techniques used employ a probability summation paradigm. Further, while in a mixed source collection, some of the harmonics of one source may mask out those of another, for the purposes of source separation it would be advantageous to retain all these harmonics. This is because without the absence of a masker, the missing harmonics will be conspicuous, much the
same as the illusion of continuity is not as strong without the maskers in Figure 2-12. However, there is no advantage in keeping components below the threshold of hearing, but rather by removing such components, a large proportion of analysis noise (due to the Gibb’s phenomena and other effects) may be removed. Hence, it was decided to only apply absolute thresholds to the data.

The absolute masking threshold is applied by setting all coefficients whose amplitudes fall below the threshold to zero. The absolute threshold in quiet has been determined empirically by a number of researchers. The values used here are according to the formula given by Terhardt [35]:

\[
Th_{Abs} = 3.64 \left( \frac{f}{1000} \right)^{-0.8} - 6.5e^{-0.6\left( \frac{f}{1000} - 3.3 \right)} + 10^{-3} \left( \frac{f}{1000} \right)^4
\]  

(3-17)

Figure 3-13 shows a graph of the absolute threshold function over the frequency range 0-20 kHz.

![Figure 3-13: Absolute threshold of hearing curve used to remove inaudible noise from the TFD](image)

3.6. Identifying Important Regions in the TFD: Peak Picking

Physically, ‘important regions’ in the peripheral auditory representation are those of maximal neural activity. In the filterbank models, this corresponds to filter channels
where the output is at a higher level than the surrounding channels at a given instant in time. In the case of the transform representation, the important regions are simply those where the function displays local maxima. In his filterbank-based model, Cooke referred to these important regions as “place groups” which he identified as regions where adjacent filter outputs contained similar dominant frequencies [118]. In the case of transform-based models, the McAulay and Quatieri (M&Q) [73] method of “peak picking” is commonly employed, although other, more noise resilient algorithms exist, such as the one proposed by Terhardt et al [131]. This section explores the various peak picking strategies in existence and presents a new one based on observations of the performance of these.

### 3.6.1. Peak Detection Algorithm

The peak-picking algorithm of McAulay and Quatieri [73] (M&Q) is very straightforward and involves finding, for each frame, $m$, all $k$ such that:

$$|X(k-1, m)| < |X(k, m)| > |X(k+1, m)|$$

(3—18)

However, it was found that this simple procedure produced too many, obviously erroneous, peaks. Figure 3-14 shows an example of the peaks detected by for a single frame of voiced speech: clearly, not all of the peaks detected are strong spectral peaks.

![Figure 3-14: Example of voiced speech spectrum before and after masking has been applied](image-url)

The McAulay & Quatieri algorithm detects peaks as per all the circles on the plot. The filled circles show the actual perceptually relevant peaks. Clearly, the M&Q algorithm has introduced some noise.
To overcome these problems, an algorithm suggested by Terhardt et al [131] was investigated. This algorithm may be defined as searching for all $k$ such that:

$$|X(k-1)| < |X(k)| \geq |X(k+1)| \text{ AND } |X(k) - X(k+j)|$$

$$j = \{-3, -2, +2, +3\}$$

(3—19)

However, it was discovered that, for the data and analysis resolution employed, this algorithm is too strict. Given the same example data set used with M&Q algorithm, the Terhardt algorithm returned the peaks as shown in Figure 3-15. Note that as well as not detecting some key harmonics of the sound, two of the high frequency, “irrelevant” peaks were incorrectly detected.

![Figure 3-15: Results of the original Terhardt algorithm](image)

Reducing the threshold in the second relation of equation 3—19 to as low as 0.5dB does improve performance as shown in Figure 3-16, however, there is obviously still an error in the peak detection at high frequency.

In general, because the later algorithms rely on probability summation, it is preferable to have a small level of noise in the data rather than not detecting important spectral peaks. Although not so apparent in the above examples, across all analysis bands and test data used, the Terhardt algorithm performs particularly poorly, returning only a very small subset of the actual peaks. Thus it would appear that the M&Q algorithm is the superior
of the two. However, during development of the later algorithms, it was determined that the level of noise present in the peaks detected by the M&Q algorithm was too high for optimum performance.

![Figure 3-16: Peak-picking result of the Terhardt algorithm with threshold reduced to 0.5dB](image)

Hence, it was obvious that an alternative algorithm, representing a compromise between the M&Q and Terhardt, was required. The algorithm described by equation 3—20 was devised and tested. The requirement that the two samples about the peak value be non-zero was imposed to facilitate the interpolation procedure that is described in the next section. The results of this algorithm for the same data as used above are shown in Figure 3-17.

\[
\begin{align*}
|X(k - 1, m)| &< |X(k, m)| > |X(k + 1, m)| \\
\text{AND} \\
|X(k - 1, m)| > 0 &\text{ AND } |X(k + 1, m)| > 0 \\
\text{AND} \\
|X(k - 2, m)| &< |X(k - 1, m)| \text{ AND } |X(k + 1, m)| > |X(k + 2, m)|
\end{align*}
\tag{3—20}
\]
Figure 3-17: Peaks detected by the compromise algorithm

Figure 3-17 shows that the compromise algorithm detects all the actual peaks correctly and only one incorrect peak is included in the set. As the incorrectly detected peak is supported by several samples above the absolute threshold of hearing it would be considered a “border-line” case and hence there is a strong argument to include it in the set of “aurally relevant” peaks.

3.6.2. Peak Interpolation

Once the peaks have been located, quadratic interpolation is used to obtain a better estimate of the actual peak frequency and magnitude values. This procedure is in keeping with the work of Terhardt et al [131] who assert that such interpolation can provide a frequency estimate for the peaks with an error of approximately ±1 Hz. As noted in section 3.4.4 the results in Figure 3-8 confirm this assertion. This interpolation procedure is defined by the following equations.

Magnitude of the $l^{th}$ peak in the $m^{th}$ frame:

$$X(l, m) = \frac{-b}{2a} \quad (3-21)$$

Frequency of the $l^{th}$ peak in the $m^{th}$ frame:

$$f(l, m) = aX(l, m)^2 + bX(l, m) + c \quad (3-22)$$
Where:

\[
a = \frac{1}{X(k+1,m) - X(k,m)} \left( \frac{f(k+1,m) - f(k, m)}{X(k+1,m) - X(k-1,m)} \right) - \frac{f(k,m) - f(k-1,m)}{X(k,m) + X(k-1,m)} \quad (3-23)
\]

\[
b = \frac{f(k,m) - f(k-1,m)}{X(k,m) - X(k-1,m)} - a \left( X(k,m) + X(k-1,m) \right) \quad (3-24)
\]

\[
c = f(k-1,m) - bX(k-1,m) - aX^2(k-1,m) \quad (3-25)
\]

And \( f(i) \) represents the analogue equivalent frequency for the frequency sample number \( i \), that is:

\[
f(i) = 2\pi i \frac{F_s}{N} \quad (3-26)
\]

where \( F_s \) is the sample rate of the analysis band from which the data originate.

### 3.7. Partial Extraction: Peak Tracking

A number of algorithms have been suggested to form partials from the discrete “place groups” or “peaks” derived from the peripheral auditory representation. The simplest has its origin with the McAulay and Quatieri (M&Q) sinusoidal transform [73]. The details of the M&Q algorithm are provided in section 3.7.1 but it may be summarised as follows. Peaks in adjacent time frames that are in close frequency proximity are considered to belong to a single “track” (or partial). For a given pair of analysis frames, any peaks in the earlier frame that do not find matches in the later frame are considered to terminate the track to which they belong while peaks in the later frame that cannot be attached to any existing tracks are considered to begin new tracks. Ellis adopted this simple approach in his early work [72].

Inspired by the work of Ciocca and Bregman [132], Cooke [118] adopted a similar approach for combining his place groups into “synchrony strands. However, Cooke employed an extension beyond the simple proximity-based M&Q algorithm. The synchrony strands were constructed such that the first derivative of the synchrony trajectories was also smooth. This was in keeping with the observation in [1] that trajectory continuation may play a role in partial formation. Once the place groups were collected into synchrony strands, Cooke approximated the frequency and derivative of these strands with a weighted least-squares line, \( y = at + b \) fitted to the place groups [118] where the coefficients \( a \) and \( b \) are given by equations 3 — 27 and 3 — 28.
In the above equations, the weights, $w_i$, are equal to the dominance estimate for each place group. The dominance estimate is a measure of how many frequency bins the place group was derived from.

Various autoregressive prediction methods for partial tracking have also been proposed in the literature. Solbach proposed one such tracker to operate on the output of his gammatone filterbank [121]:

$$f_0(i) \cdot t_p(i) = \frac{(n - 1)}{2 \pi \sqrt{2n - 3} \cdot Q^{-1}}$$  \hspace{1cm} (3 — 29)

where $n = 3; Q = \frac{f_0}{\Delta f} = 0.05$ (the Q factor of the filterbank) and:

$$t_p(i) = \frac{2(n - 1)}{\sqrt{2n - 1}} \Delta t(i)$$  \hspace{1cm} (3 — 30)

Another variation on the partial tracking procedure is to use a feedback mechanism as illustrated in Figure 3-1. Ellis’ later attempt at “prediction-driven CASA” [120] is an example of such a strategy where the feedback occurs from the streaming stage. Grossberg’s neural network-based system also uses feedback in his “spectral” stream layer, which roughly corresponds to the partial tracking stage [119]. In this case pitches from the “pitch stream” layer influence the tracking. Both of these are perceptually justified designs as McAdams and Bregman have noted that streaming and partial formation are interrelated processes [17].

Given these considerations, coupled with the aims of maintaining generality and computational efficiency, it was decided to adopt the relatively simple McAulay and Quatieri approach as an initial starting point for the investigation. However, because of the multi-band nature of the TFD employed, as well as the observed results of the basic M&Q technique, it was necessary to significantly adapt this basic scheme.
3.7.1. The McAulay and Quatieri Algorithm

The method of McAulay and Quatieri can be summarised as follows:

1. Assume that a set of tracks is already in existence. (The initial set of tracks is simply the first frame of peaks)
2. For each track, attempt to find all peaks in the current analysis frame that fall within $\Delta$ of the frequency of the last peak in the track.
3. IF more than one possible match is found, select the closest in frequency.
4. IF a match has been found,
   (a) Check whether the peak is a better match for another track
   (b) IF no better match is found, assign the peak to the track
   (c) ELSE IF another potential peak exists, assign it to the track
5. ELSE track dies
6. IF any peaks remain in the current analysis frame, set them as new tracks

The algorithm of McAulay and Quatieri described above effectively performs zero order linear prediction. It is thus prone to displaying a horizontal bias. This will be a particular problem when tracking multiple sources or when there are closely spaced harmonics that display frequency contours with a sharp gradient. An illustration of the tracks that will tend to form in the latter case is given in Figure 3-18. Figure 3-19 shows the results of applying the M&Q algorithm to the peaks derived for band 1 of a mixed source file. The tracking errors that have occurred as a result of the horizontal bias have been indicated on the figure.

![Figure 3-18: Horizontal bias of McAulay and Quatieri peak tracking algorithm]
3.7.2. Horizontal Bias: A Challenge for CASA Systems

The problems posed by the horizontal bias of the M&Q tracking algorithm highlight a particularly difficult challenge for CASA techniques; that is, crossing partials. An example is shown in Figure 3-20.
Figure 3-20 shows a particularly challenging example of the crossed partial problem. Two possible sets of trajectories are shown. The first, represented by the solid lines, is that which most closely represents the original source data. Such a signal might arise from mixing two artificially generated sweeps of the form shown, or even by playing a musical scale in the directions shown, either as a duet or by mixing two recordings. The same effect is even achieved by presenting each signal to a separate ear. At moderate presentation rates, the actual percept invoked, even when the signals are presented to opposite ears, is that shown by the dashed lines [1]. This is an interesting finding, but how should it influence the design of partial trackers in CASA systems?

A system such as that of M&Q, which relies wholly on the proximity of peaks in forming the partials, might select either the actual or perceived trajectories, depending on the time-frequency resolution and peak discrimination accuracy in the region of the crossover. On the other hand, a system that also takes into account the derivative of the trajectory (or imposes smooth trajectory transition in some other manner) will almost certainly select the actual trajectories and may even break each of these in two at the centre to form two descending and two ascending chirps. On the surface, the first of these behaviours is the more psychologically congruent. However, the simple proximity rule that would (arguably correctly) result in the perceived trajectory illustrated in Figure 3-20 would also be likely to result in the obviously incorrect tracking behaviour in Figure 3-21.

Clearly, the tracking methodology chosen for a CASA system should be able to perform correctly in a majority of cases. If top-down feedback mechanisms are available, then it may be possible that both of the situations depicted in Figure 3-20 and Figure 3-21 could be catered for using appropriate rules. However, for a low-level “data driven” approach as adopted in this thesis, it is necessary to make a decision based on the pragmatic
requirement that the majority of situations be correctly handled. As such, a smooth trajectory based approach is considered the most appropriate.

### 3.7.3. Extension to the McAulay and Quatieri Algorithm

In order to overcome the problem of the horizontal bias, it was decided to increase the order of the predictor by 1. The motivation for selecting a first order, rather than a higher order, predictor was the observation that, in general, the frequency contour of tracks varies relatively slowly in time. This means that over short segments, the tracks are basically linear. Further, the accuracy of the peak frequency estimation in a given band is heavily dependent on the match between the properties of the signal (i.e. modulation rate and fundamental frequency) and the time-frequency resolution of the band. Hence, in some bands the estimated peak frequency values will oscillate about the true value. The higher the order of predictor, the more prone it is to similar oscillatory behaviour, which, in those bands where frequency estimates are inaccurate, would result in many misdirected tracks. Hence, the first order predictor represents a compromise between the two undesirable characteristics of horizontal bias and a tendency to oscillate.

The only modification to the M&Q algorithm as stated in the previous section required to accommodate first order prediction is a change to step 2 as follows:

2. For each track, attempt to find all peaks in the current analysis frame that fall within $\Delta$ of the predicted frequency for the next peak in the track.

The predicted value is given by:

$$\hat{f}_i = \begin{cases} 
2f_{i-2} - f_{i-1} & i > 1 \\
f_{i-1} & i = 1 \\
f_i & i = 0 
\end{cases}$$  \hspace{1cm} (3—31)

In equation 3—31 above, $\hat{f}_i$ is the predicted peak frequency in the current frame, $f_{i-1}$ and $f_{i-2}$ are the frequencies of the previous two peaks in the track and $f_0$ is the frequency of the initial ‘seed’ peak for the track.

Equation 3—31 clearly shows that in the first iteration of this step for each track, when only one previous peak value will be available, the predictor reverts the zeroth order.
Figure 3-22 shows the results of applying this algorithm to the same peaks as in the previous example. The red ellipses on the Figure indicate the positions where tracking has been improved over the standard M&Q algorithm.

3.8. Track Consolidation

3.8.1. The Track Consolidation Problem Defined

The result of the procedure described in the previous section is a set of tracks for each resolution band. The accuracy of tracking in any given band is determined by the precision of the match between the signal modulation rate and the time-frequency resolution of the band. Steady state tonal signals whose components fall within the range of the lower frequency bands will be tracked much more accurately in these bands as they have a better frequency resolution than the higher bands, while a signal that displays a high rate of frequency modulation will be tracked more accurately in the higher bands that have a better temporal resolution.

Figure 3-23 shows the tracks generated by the modified M&Q algorithm described in section 3.7.2 for a frequency modulated tone with modulation rate and index typical of those that would be found in music. Note that the smooth variation in frequency is most apparent in bands 0 and 1, which have the best temporal resolution, while band 4 does a
poor job of tracking the signal’s frequency variation. On the other hand, Figure 3-24 shows the tracks generated for a pure tone of 525 Hz. Note that now the most accurate track is that from band 4 while the tracks in bands 0 and 1 oscillate about the true frequency. Hence, the ‘best’ analysis band is determined by the modulation rate of the signal itself. The band collapsing procedure aims to combine the data from all bands in a manner that produces the best compromise between time and frequency resolution thereby producing the most accurate tracks over all possible data sets.

Figure 3-23: Multi-band tracking for a signal displaying FM typical of musical vibrato

Figure 3-24: Multi-band tracking for purely tonal signal
One important conclusion that may be drawn after consideration of Figure 3-23 and Figure 3-24 is the advantages of the apparently redundant overlapped band approach that has been taken in the TF analysis stage as described in section 3.4.2. Given that the most appropriate resolution for the time-frequency analysis is signal dependent, it is obvious that a time-frequency analysis that has only a single resolution available in each band, regardless of how the bands are distributed, cannot be optimum across the entire range of signal possibilities.

3.8.2. Overview of Possible Solutions to the Track Consolidation Problem

Several approaches to solving the problem of generating a single set of tracks from the output of the multi-band tracking stage were developed and tested. The first of these represented a completely different system structure to that depicted in Figure 3-2 and is described in detail in section 3.8.3. Briefly, it assumed that each of the main steps in the algorithm could be performed in isolation without any feedback between the stages whatsoever. As might be expected, this total lack of feedback produced rather disappointing results, which led to the development of the approach illustrated in Figure 3-2. Section 3.8.3 details the original approach to track consolidation while sections 3.8.5 and 3.8.6 describe the two different methods for performing the first step in the alternate approach: spline approximation of track shapes.

3.8.3. Initial Approach

The initial approach to band consolidation was very simple. In essence all tracks from the various bands that lay within a threshold of each other were considered to be a single track. To determine the frequency, magnitude and phase values for each peak along the resultant track, the values of all co-incident peaks from the set of candidate tracks were averaged. This approach was sufficient for data that contained relatively low levels of noise, however, performance degraded rapidly when only minor errors in the initial tracking stage were present. The other disadvantage of this technique was that it performed poorly under conditions where the transform resolution produced a poor tracking in one or more of the bands as illustrated in Figure 3-23 and Figure 3-24.
The results of this algorithm for the two test cases illustrated above are shown in the following figures:

**Figure 3-26: Results for band collapsing for the FM signal depicted in Figure 3-23**

**Figure 3-27: Results of band collapsing for tonal signal depicted in Figure 3-24**

The most obvious shortcoming of the simple track averaging procedure visible in Figure 3-26 and Figure 3-27 is the tendency to introduce discontinuities in the final signal. The sharp changes may be removed to some extent by smoothing the tracks as shown in the following figures where the tracks have been passed through a 3-point moving average filter.
Chapter 3: The Data Representation: A Basis for Auditory Source Separation

147

Figure 3-28: **Collapsed FM track after smoothing**

Figure 3-29: **Collapsed tonal track after smoothing**

However, while smoothing provides a partial solution to the shortcomings of this algorithm, it will be noted that in the case of the signal shown in Figure 3-29, although most of the ‘noise’ has been eliminated, the FM is still present in the track. These sort of errors in the partials of some sounds might effect the overall timbre, and therefore, perceptual quality of the inverted signal. The other disadvantage of this technique was its inability to detect and account for errors in the initial tracking stage.
Figure 3-30: Multi-band tracks for an example of male speech
The different colours denote the different analysis bands.

Figure 3-31: Results of the band collapsing procedure for the tracks in Figure 3-30
Figure 3-30 shows the results of the multi-band tracking procedure for an example of male speech. The utterance is the letter ‘u’. Given that this is a vowel sound, we expect only harmonically related tracks to be formed. It will be noted that there are three sources of error. The first two are failures of the tracking procedure itself, marked by the red ovals. One of these errors is due to an interaction between the first order prediction and noise in the peak data. The erroneous track within the oval marked A suffers from such an error where a noise peak incidentally was closer to the correct peak because the track should have been changing direction in this region. The lower of the two erroneous tracks within oval B is another example of this. The upper track in oval B was seeded by two noise peaks and then, by virtue of the prediction error, captured peaks that should have belonged to two higher tracks. The track in oval C is due to the resolution problem producing nothing but noise peaks in the wideband analysis thus the tracking procedure simply joined what tracks it could find within range. The magenta tracks in the blue oval, labelled D, also display a temporal resolution that is unsuitable to the signal in this region. All of these errors result in noise artefacts as highlighted by the similar ovals in Figure 3-31.

3.8.4. *Group-based Approach Overview*

The second method achieves noise robustness by performing a probability summation over several frames of data across a significant frequency range. Thus, data from the bands or frequency ranges that were tracked correctly may be used to compensate for tracking errors in those where tracking performance was poor. This process is motivated by the observation that most natural sounds will produce harmonically related tracks with frequency and amplitude contours of the same shape [133]. The basic algorithm is illustrated in Figure 3-32.

![Figure 3-32: Algorithm for consolidating multi-band tracks](image)

The algorithm illustrated in Figure 3-32 shows a marked difference from the straightforward procedure of Figure 3-25. The first two stages of the algorithm are
identical to those in Figure 3-25 and have hence been left off the diagram for reasons of clarity. The multi-band tracking stage is also identical to that used in the previous algorithm, however, only the peaks below 2 kHz are permitted to be included in the tracks. This is because the multi-band tracking algorithm produces more accurate tracks in this range due to the stronger peaks generally found in this region. The only disadvantage to this approach is that any signal that has a fundamental frequency above 2kHz will not be detected. This situation is simply accommodated by removing all peaks that have been accounted for by the final set of tracks from the input data. If a significant number of peaks remain, their frequencies are halved and they are fed back through the algorithm illustrated in Figure 3-32 from the input. This procedure can be iterated as many times as is necessary to reduce the number of peaks below a minimum threshold value.

The second stage in Figure 3-32 generates a spline approximation of each of the multi-band tracks. Two methods were considered for achieving this spline-based track shape approximation. The first of these methods employed an approach based on the Hough transform that promised good results. However, the required resolution and dimensionality of the transform rendered it impractical because of extraordinary computational load. Also, the data space was, in general, not large enough for the implicit statistical analysis performed by the Hough transform to be conclusive. The Hough-based approach is detailed in section 3.8.5. The second method simply employed a least-squares method to fit a third order curve to each of the multi-band tracks. Despite its simplicity, this method was found to perform at least as well as the Hough based technique with a considerable reduction in computational complexity. The details of the design and results achieved using of this method are presented in section 3.8.6.

Once a set of approximating splines has been generated for the frequency range of 0-2kHz, these are ‘refined’ with reference to the original peak data. This step is required because the approximating splines are not capable of representing the fine detail of most tracks, such as onset transience or vibrato. In effect, at the output of this stage, the band consolidation is complete for frequencies below 2kHz. However, in order to correctly handle crossed or co-incident partials only the tracks produced after the feedback loop has been completed are retained.

The feedback loop generates a set of more robust track shape estimates by grouping the refined spline-based estimates according to the principles expounded in section 2.3.3. A model track is then determined for each group, which is assumed to be the fundamental
track of the group. From this it is a simple matter to estimate all the tracks for the group over the entire signal bandwidth by assuming that they are harmonics of the model. However, since the characteristics of natural sounds are generally such that the harmonics are rarely exact, it is necessary to repeat the track shape refinement procedure using the model-based track estimates as the input to refinement stage.

### 3.8.5. **Hough-based Spline Fitting**

The first stage in the band consolidation procedure illustrated in Figure 3-32 is a third order polynomial based track shape approximation. The reasoning for selecting a third order approximation was that the overall shape of most tracks was observed to be adequately represented with at most a third order polynomial. Ignoring onset transients and effects such as vibrato, the most suitable approximating function for each of the expected track classes is summarised in Table 3-4.

<table>
<thead>
<tr>
<th>Track Class</th>
<th>Optimum Approximating Polynomial Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tone</td>
<td>0</td>
</tr>
<tr>
<td>Chirp (natural or artificial)</td>
<td>1</td>
</tr>
<tr>
<td>Speech Formant</td>
<td>2 or 3</td>
</tr>
</tbody>
</table>

Considering musical data in particular, the approximating functions given in Table 3-4 may seem insufficient in that it is impossible to represent a track that displays the frequency modulation typical of vibrato using a third order polynomial. Manifestly, this technique will yield an approximation of such a track that is basically a tone at the “carrier” frequency of the track. Fortunately, this is sufficient for the next phase of the algorithm that locates the peaks closest to the spline approximation and uses these to generate a better approximation of the actual track shape.

In fitting the polynomial approximations to the multi-band track data that was produced by the modified M&Q algorithm described in 3.7.3, the chief concern was to address the resolution considerations mentioned in section 3.8.1. Hence, the data from all bands contributed to the approximation and did so in a voting fashion to ensure that the hypothesis with the greatest support was selected. Secondly, it was noted that the problem was akin to shape extraction from an image where partial shapes had already been extracted. Taking these two considerations into account led to the conclusion that
employing a modified Hough transform might be a useful method for performing the spline approximation.

A Hough transform is usually used to extract line shapes from images by transforming the x-y co-ordinate space into a parameter space of the order required to represent the particular line shape that is required. If one is searching for straight lines, for example, the parameter space would have two dimensions representing the gradient and y-intercept of the lines. This parameter space acts as a histogram whose bins are populated such that every pixel in the image contributes to all bins that represent lines of which it could be a member. Peaks will form in bins that represent lines along which significant pixels lie. The only difference for higher order curves is that the parameter space increased in dimensionality.

Given this description it might seem that the Hough transform would be an ideal alternative to the modified M&Q algorithm employed, however, there are two main reasons against this. The first is that the peak data by themselves are generally too dense for a Hough transform to be practical, since many possible trajectories could be traced. One way to overcome this might have been to restrict the trajectories both in order, direction and location. However, the other, somewhat related, problem that precluded this is that the range of allowable shapes (c.f. Table 3-4) meant that such a restriction would have to be so lax as to provide little benefit. However, one way that the allowable trajectories could be restricted without a priori knowledge of the optimum parameter ranges was to use the initial tracking estimates to reduce the number of “votes” that each peak had in the Hough space. This was achieved by using short track segments, rather than points, to populate the Hough space.

Because the shape of a polynomial in the explicit form given in equation 3—32 is highly sensitive to its parameter values \((a, b, c, d)\), it would be very difficult to determine the best parameter value ranges and resolution of a Hough space for polynomials in this form.

\[
y = ax^3 + bx^2 + cx + d
\]  

(3—32)

Splines, on the other hand, provide a very good correlation between parameter values and curve shape. For this reason, it was assumed that the shape of each track could be approximated using a Beziér spline. A Beziér spline approximates a curve using four control points, two of which are the end-points of the curve. The other two points are located such that the gradient between each end-point and the nearest interior control point is equal to the gradient of the curve at the endpoint. Figure 3-33 illustrates this concept.
Chapter 3: The Data Representation: A Basis for Auditory Source Separation

\[ \frac{\beta_1 - \beta_0}{\alpha_1 - \alpha_0} = f'(n_0) \]

\[ f(n) = an^3 + bn^2 + cn + d; \quad 0 \leq n \leq N \]

\[ \frac{\beta_3 - \beta_2}{\alpha_3 - \alpha_2} = f'(n_{N-1}) \]

**Figure 3-33: Illustration of a Beziér spline**

The parametric description of a Beziér spline is given by the equations:

\[ n(t) = \alpha_0 (1 - t)^3 + 3 \alpha_1 t (1 - t)^2 + 3 \alpha_2 t^2 (1 - t) + \alpha_3 t^3 \quad (3-33) \]

\[ f(t) = \beta_0 (1 - t)^3 + 3 \beta_1 t (1 - t)^2 + 3 \beta_2 t^2 (1 - t) + \beta_3 t^3 \quad (3-34) \]

where \( n \) and \( f \) are the frame number frequency of each peak respectively. \( t \) is a parameter that has range 0, 1. The \( \alpha_i \) and \( \beta_i \) represent the co-ordinates of the spline control points. This would require an 8 dimensional Hough space, which was considered computationally impractical.

To reduce the dimensionality of the required Hough space, it was assumed that all the approximating splines begin and end at the same points, which are located at the first and last frame of those being analysed. Thus the values of \( \alpha_0 \) and \( \alpha_3 \) were fixed at 0 and \( N-1 \) respectively, where \( N \) was the number of frames being analysed. A further reduction in dimensionality was achieved by assuming that the remaining two control points were equally spaced in the analysis interval. That is, \( \alpha_1 = \frac{N}{3} \) and \( \alpha_2 = \frac{2N}{3} \).

With the assumptions stated in the previous paragraph, the problem has been reduced to one of finding the \( \beta_i \). This was achieved according to the following procedure. Firstly the tracks were broken into segments of 4 peaks. For each of these segments, the gradient of the least squares line approximating the section was determined. The line segment then had a vote in all cells in the Hough space for which the corresponding \( \beta_k \) gave splines with a similar gradient to the approximating line segment in the time-frequency region of the peaks represented by the segment. The idea is illustrated in Figure 3-34 with only a few of candidate splines shown.
While in theory this method should provide a very good approximation to the track shape for tracks that are purely tonal or ‘formant’ shaped, it suffers a number of disadvantages. The first is that it performs poorly for any track whose shape cannot be described by a cubic function. This is particularly problematic for music that contains vibrato or highly transient attacks. Secondly, the resolution of the space required to achieve a good approximation of even the low order track shapes is such that several approximating curves are generated for each track requiring an additional data reduction step that is non-trivial due to the density of the data. Finally, the minimum resolution of the space required for a reasonable approximation was found to be such that the dimension of the Hough space was $71^4 = 25,411,681$ elements. This proved to be an extraordinarily heavy processing load.

### 3.8.6. Least-squares Spline Fitting

In order to overcome the problems associated with the dimensionality of the Hough space and its inability to cater for higher order curves, a second method has been developed and evaluated. In this method it was assumed that each track could be described by a third order polynomial in its explicit form:

$$f(n) = a_0n^3 + a_1n^2 + a_2n + a_3$$  \hspace{1cm} (3 — 35)

In order to determine the values of the polynomial coefficients, a least squares fitting procedure was employed which essentially involved solving the matrix equation:

$$X'Xa = Xf$$  \hspace{1cm} (3 — 36)
where: $X = \{x_{i,j}\} \quad \forall \quad 0 \leq i < N \text{ and } 0 \leq j \leq 3; \quad x_{i,j} = n_i^{3-j}; \quad n_i$ is the frame number of the $i^{th}$ peak in the track, $f = \{f_i\}$ are the peak frequency values and $a = \{a_i\}$ are the polynomial coefficients.

The result of this technique is a spline approximation for each track from the multi-band analysis. Hence, there may be several spline approximations (hereafter referred to as either splines or tracks, depending on context) for each true track as shown in Figure 3-36 and Figure 3-36. It is therefore necessary to employ a data reduction stage. However, the data reduction problem is much simpler than that in the Hough-transform based technique because the number of approximations is considerably reduced and the similarity of corresponding splines is greater than those produced because of the necessarily fine resolution of the Hough space.

![Figure 3-35: Results of spline fitting procedure for a speech signal overlaid (in black) on the original multi-band track data](image)
Figure 3-36: Results of spline fitting procedure for a speech signal

Obviously, the splines in Figure 3-36 are only “rough” approximations of the track shape. In the case of formant tracks, the approximation is good, since the tracks themselves are fairly close to 3rd order curves. In contrast, the approximation for a signal displaying FM might be considered quite poor. Figure 3-37 and Figure 3-38 show two cases.

Figure 3-37: Splines fitted to an artificial FM tone
Because of the spline fitting procedure’s somewhat poor approximation to any tracks displaying variations that cannot be described by a third order curve, a “track shape refinement” procedure was developed. This involved collecting all peaks, from all bands, that were within a threshold of each spline and calculating the average peak for each time frame. In order to avoid similar errors to those encountered with the simple track averaging procedure, the peaks were collected independently of their track assignment. Once all peaks within the threshold range were collected, the mean and standard deviation of the frequencies were counted. A final average peak frequency value was calculated after all peaks falling outside one standard deviation of the mean value were excluded. Data from the second and third highest frequency bands were linearly interpolated to improve the effectiveness of this operation by providing more data for the statistical analysis. The results of applying this track shape refinement procedure are shown in Figure 3-39, Figure 3-40, and Figure 3-41.
Figure 3-39: Results of track shape refinement on the splines of Figure 3-36

Figure 3-40: Result of applying track shape refinement procedure to tracks from Figure 3-37
Although it is not immediately apparent from Figure 3-39, Figure 3-40, and Figure 3-41, most of the tracks are represented by multiple identical, or nearly identical, splines. This is because the refinement procedure was applied to every spline, lest any tracks be inadvertently neglected. Secondly, while the approximations are already superior to those achieved using the simple track averaging procedure, there are a few “noisy” sections observed for some tracks. Hence the final stage is a “track reduction stage” that collects together identical, or nearly identical, tracks, calculates the average of these and applies a simple three-point moving average filter to each average track:

\[
\hat{f}(i) = \frac{1}{3} \sum_{j=0}^{3} f(i - j) \quad 0 \leq i < N
\]

\[
f(-1) = f(i)
\]

(3—37)

where \(f(\bullet)\) are the original track frequencies, \(\hat{f}(\bullet)\) are the filtered track frequencies and \(N\) is the number of peaks in the track.

The final results of the track reduction and smoothing stage are shown in Figure 3-42, Figure 3-43 and Figure 3-44.
Figure 3-42: Result of the track reduction procedure for the data in Figure 3-39

Figure 3-43: Result of track reduction procedure for data in Figure 3-40
This section has shown that the results of the spline-based approximation procedure are superior to those achieved by the simple track averaging procedure, while the algorithm is still relatively computationally efficient. By way of a final comparison with the simple averaging procedure, the results of applying the spline-based technique to the tone in Figure 3-24 are shown in Figure 3-45. It will be noted that there is no FM introduced, as was the case with the simple averaging procedure as shown in Figure 3-29.

3.8.7. Model Track Calculation
The result of the procedure described in the previous section is a set of relatively robust tracks for the 0-2 kHz frequency range. These tracks can then be grouped according to the procedures described in Chapters 4 to 6. Once the tracks have been grouped, the
group model (or fundamental) track can be used as a kind of harmonic sieve to extract all the harmonic group members from the initial peak data. In order to do this, however, we must obviously have a method for calculating the model track.

Given that the harmonic number of each track was estimated during the grouping procedure, it was decided that a simple averaging procedure was sufficient for this task. The technique proceeded as follows:

1. Scale each track to its fundamental frequency
2. Calculate the average values of frequency, amplitude and phase of all the (scaled) peaks present in each time frame for which the group exists.

The concept is further illustrated in Figure 3-46.

![Figure 3-46: Illustration of model track determination procedure](image)

The results of applying this model calculation procedure to two typical test files are shown in Table 3-5. The first data file consisted of a musical tone displaying vibrato while the second was a segment of voiced speech.
Table 3-5: Results of model track calculation for two test files

The model shapes are highlighted as red, dashed lines. The number alongside each track indicates the harmonic number estimated for that track.

<table>
<thead>
<tr>
<th></th>
<th>Musical Vibrato</th>
<th>Speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models with</td>
<td>![Musical Vibrato graphs]</td>
<td>![Speech graphs]</td>
</tr>
<tr>
<td>tracks at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>original</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scale.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Harmonic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numbers shown</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alongside</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tracks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close-up of</td>
<td>![Close-up of</td>
<td>![Close-up of</td>
</tr>
<tr>
<td>models and</td>
<td>models and</td>
<td>models and</td>
</tr>
<tr>
<td>tracks scaled</td>
<td>tracks scaled</td>
<td>tracks scaled</td>
</tr>
<tr>
<td>to fundamental</td>
<td>to fundamental</td>
<td>to fundamen-</td>
</tr>
<tr>
<td>Smoothed</td>
<td>![Smoothed</td>
<td>![Smoothed</td>
</tr>
<tr>
<td>Models</td>
<td>models</td>
<td>models</td>
</tr>
</tbody>
</table>

The results illustrated in Table 3-5 indicate that the averaging procedure works adequately. However, it will be noted from the second row of the table, that the endpoints of the model track for the speech signal display sharp transitions due to a single track (the fundamental) being longer than the others and displaying significantly
different onset and offset behaviour. This is compensated for by performing the simple smoothing procedure described by equation 3 – 37, with the results illustrated in the third row of the table above.

3.9. Final Track Extraction
As was stated earlier, once the spline-based track approximations have been grouped and a model track calculated for each group, these model tracks are used to extract the actual tracks from the original peak data. In the discussion of techniques that follows two test cases will be considered. The first case is a single source data file consisting of a short section of male speech. The original multi-band track file from which the peaks for the track extraction procedure are derived, is shown in Figure 3-47 while the model tracks calculated for the file are shown in Figure 3-48. The second test file contained two sources and appears in Figure 3-49 and Figure 3-50. The short model track in Figure 3-50 represents the same source as for the single source file while the other model is from a group derived for a short tone played on a trumpet.

![Figure 3-47: Multi-band track data for single source file](image)
The colours represent the different analysis bands. The crosses on the figure represent the individual peaks used by the track extraction procedure reported here.
Chapter 3: The Data Representation: A Basis for Auditory Source Separation

Figure 3-48: Model tracks derived for single source file

Figure 3-49: Multi-band track (and peak) data for mixed source file
Two methods were devised for track extraction. The first was relatively simple being based on the concept of a harmonic sieve. When this algorithm was found to perform somewhat poorly however, a second more robust algorithm was devised. These two algorithms are detailed in the succeeding two sections.

3.9.1. Initial Approach
The first attempt at achieving this was a simple harmonic sieve concept where the model track was multiplied by successive harmonic numbers and all peaks within a threshold range of the scaled model were used to construct the corresponding harmonic track. If more than one peak lay within the threshold of the scaled model at a particular time frame, then the average of all such peaks was used. To avoid extreme outliers, due to noise, effecting this average, the collection of peaks for a particular time frame were subjected to a median filtering operation where any peaks further than one standard deviation from the mean peak value were discarded and a new average peak calculated.

Since the model track reflects the longest track in the group, it may not be possible to find peaks for the entire length of every estimated harmonic. Hence, a track is considered to start at the time frame in which the first peak within range of the expected harmonic is found and terminated at the last frame for which a peak is found. However, in order to account for ‘gaps’ in the peak data due to crossed or co-incident harmonics or some other...
interference, it was decided to allow a track to continue for several “empty” frames before it is terminated. The results of this simple technique for two test cases are illustrated in Figure 3-51 and Figure 3-52 below:

Figure 3-51 shows the results of applying the simple track extraction procedure to the single source data file. Tracking in the lower frequency regions is good, with even some gaps being correctly “filled in”. However, from approximately 2kHz and above errors become apparent. Two major errors may be identified. The first is that the evidence for filling in some tracks seems limited, examples of these are marked with blue ellipses on the figure. The second, and more serious, form of error is the tendency to join peaks very distant in frequency at higher frequencies. These are fairly obvious, particularly in the cluster of ‘tracks’ above 4300 Hz. The first error is due to the localised nature of peak collection. Since a decision to grow a track is based purely on the proximity of peak data to the model track at the current frame only, no decision as to the accuracy of this continuation is possible. The second, and more serious error is due to the proportional nature of the threshold used, which was a fixed percentage of the current model frequency for which matching peaks were sought.

Figure 3-52 shows similar results for the mixed source data file. While the resolution of the figure does not reveal any examples of the first incorrect gap-filling problem, the second more serious problem is clearly evident for frequencies above approximately 2800 Hz.
3.9.2. **Extended Technique**

In an attempt to alleviate the above problems, a second technique was developed and tested that involved fitting peaks to sections of the model that were 10 frames in length. The mean and standard deviation of the error between the data peaks and the corresponding model were calculated and all peaks further then 1 standard deviation away from the model value were discarded. This procedure was repeated until the mean and standard deviation values fell below the threshold. It was hoped that having more data for the statistical analysis would reveal a trend thus mitigating the gap-filling problem. The second problem was accounted for by setting a fixed frequency threshold. The results of this technique for each of the test cases are shown in Figure 3-53 and Figure 3-54.

Figure 3-53 shows a marked improvement over the results of the previous algorithm shown in Figure 3-51. The dubious “in fills” have been removed leaving gaps in the tracks and the “noise” tracks at high frequencies have been entirely eliminated. While the gaps in the tracks in the 2-3 kHz range may appear to be a concern, it was found that they had little noticeable effect on reconstruction quality.
Figure 3-53: Results of extended track consolidation algorithm for single source file

Figure 3-54 shows a similar improvement over the previous results. The only matter of mild concern is the very slight FM on the second harmonic of the group belonging to the trumpet tone (blue tracks), which is probably due to interference between the tone and the co-incident speech track. Given the number of tracks in the group this mild level of FM has no detectable effect on signal quality.

Figure 3-54: Results of extended track consolidation algorithm for mixed source file
3.10. Inversion

Given that the main aim of this thesis is to present a representation that can be used as the basis for content-based audio coding and information management, it is necessary that the representation be invertible. Inverting the representation presented in this chapter is a relatively straightforward exercise and can be easily and efficiently implemented. The basic algorithm is illustrated in Figure 3-55 below.

A conspicuous omission in the discussion of the representation thus far is a model for the transient and noisy sections of the data. The chief reason for this is that affecting source separation for noise-like signals is an extremely difficult problem in its own right and has been left as an area for future work. Nevertheless, assuming that it was desired to reconstruct the entire (mixed source) signal, the perceptual quality of the reconstruction can be greatly improved if a residual is recorded at encoding time. This residual may be calculated using the system illustrated in Figure 3-56.

In terms of the perceptual quality of the reconstructed signal, the tracks-only reconstruction yielded signals that ranged from perfect for artificial signals through very good and virtually indistinguishable from the original for some spoken vowel sounds and wind instrument tones to intelligible and recognisable for similar signals. However, as would be expected, quality dramatically decreases when a significant proportion of the
signal appears noise-like. Preliminary tests of signal reconstruction with a residual provided near perfect signal quality, however, it should be noted that this was achieved with the raw residual. Some degradation in signal quality might occur if the residual is compressed, as would normally be the case. The level of degradation would depend heavily on the compression scheme and extent.

### 3.11. Summary

This chapter has presented the design of the audio representation that forms the basis for the source separation techniques that form the main theme of the research reported here. The representation belongs to the class of parametric representations first introduced by McAulay and Quatieri [73]. It is therefore natural that the initial stages of its generation bear much resemblance to the work of McAulay and Quatieri, however, modifications were made to cater for various challenges that were the result of the wider bandwidth required of general audio applications (as opposed to the speech only signals of interest to McAulay and Quatieri).

Despite the manifest correspondence between sinusoidal representations and the low-mid level auditory signal representation, it will be noted in the next chapter that there exists a bias against applying sinusoidal representations to CASA. The most often cited argument for this bias is the apparent inability to correctly handle co-incident and missing partials. However, an architecture involving feedback at what corresponds to the low-mid levels of perception was presented that overcomes these problems by exploiting similarities within groups to reconstruct the tracks that would otherwise be obfuscated. Results presented in this chapter have substantiated this claim.

A particularly unique feature of the system described in this chapter is the three pass tracking procedure that is achieved via a feedback mechanism. The first pass generates a set of initial tracks for the data that appears below 2 kHz in each analysis band. A single set of spline approximations of the tracks are then derived from these. These approximations are used to derive a series of spline-based track shape estimates from the original peak data. These track estimates form the basic input to the grouping algorithms which ultimately output a single of model track for each group. The final stage of track extraction uses the model tracks to generate model-based track shape estimates which are used in conjunction with the original peak data to extract the final series of grouped tracks that encompass the entire bandwidth of the input data. This chapter has dealt with all aspects of the above process that explicitly relate to track generation. The fourth
phase, group formation, being the central aim of the work reported in this thesis, is the subject of the next three chapters.
Chapter 4

Frequency and Amplitude Contour-based Grouping

4.1. Introduction

Having decomposed the audio signal into a set of partials some grouping criteria must be applied to form them into groups. The earliest and most frequently employed criterion is that of harmonicity which will be the focus of the next chapter. Other commonly recognised mechanisms responsible for partial fusion in human audition that are employed by some CASA systems include common amplitude or frequency modulation and common onset/offset times. Many CASA systems employ only a single one of these criteria (usually either harmonicity – “pitch-based” – or common onset – “rhythm-based”) while some use a combination. Systems are distinguished by how the similarity measures are calculated and, for those that employ multiple metrics, how they are combined. Perhaps the single most challenging problem for CASA research, which has had a significant impact on the design of CASA systems, is that of co-incident harmonics. As such, a brief description of this problem will be provided as an introduction to a discussion of the various architectures that have been employed for general auditory grouping. Having provided a context for the discussion of grouping, the remainder of this chapter and the next analyses specific grouping methods that have been tried in the past and compares their performance with various new ones that have been developed in the course of the work that is reported here.

As previously mentioned, a chief problem faced by CASA systems is the phenomenon of co-incident harmonics. A simplified example of this is shown in Figure 4-1.

![Figure 4-1: Illustration of the problem of shared harmonics](image_url)
The situation depicted in Figure 4-1 occurs commonly in many natural sounds of interest to CASA researchers, and especially in speech and music. It is also a well-recognised problem and many researchers at least address it as a problem requiring further research [134]. This is also a very good argument against the application of the principle of exclusive allocation in computational models of the auditory periphery. The situation can be accounted for in one of two ways, firstly, in a manner analogous to that adopted by Cooke [118], the principle of exclusive allocation can be completely ignored and all tracks be allowed to be members of as many groups as they are candidates for. Secondly, a post-processing stage can examine the grouped tracks for ‘missing’ harmonics and search the original data for evidence of their presence, inserting these tracks if the evidence is found. The latter method has perceptual support in the continuity illusion discussed in section 2.3.3.

An associated problem that occurs particularly in musical data is that harmonics that are very close together in frequency (within a few auditory filter bandwidths) create interference patterns in the auditory representation that are heard as beating. This beating is due to the non-linearities of the analysis system and appears in both the physical system (i.e. the ear) and its model (e.g. a perceptually tuned TFD). When the two sounds are played simultaneously, this beating is familiar and simply adds to the timbre of the chord, however, when the sounds have been separated, the residual amplitude modulation is annoying to listeners possibly to the point of preventing source identification.

Cooke [118] was perhaps the first to describe a system that used a number of different metrics to facilitate partial grouping. In his system, any number of grouping principles could be applied to the synchrony strand representation. Examples include harmonicity, common amplitude modulation, common onset or common pitch relations. Each of these grouping metrics was applied independently and the results were then combined based on the similarity of their pitch contours. The initial grouping is governed by three heuristic principles. Variations of these heuristics were adopted in the work described in this thesis.

1. **Explain all the evidence.** Grouping is terminated only when all the partials have been assigned to a group (singleton groups are allowed). This principle ensures that the grouping process discards none of the data.

2. **Same group, different starting point.** The relationship between group members should be such that, regardless of which of the group members is used as a “seed” for the group, the same grouping will result. Specifically, any three
elements in the group should have a transitive relationship. That is, given three elements, A, B and C, if A is related to B and B is related to C then A is also related to C. Even for algorithms that do not begin with a group “seed”, this principle is a useful measure of the reliability of the grouping mechanism employed.

3. Start from “dominant” objects. Cooke asserts that the more dominant an object is (he measured dominance by the estimated level of neural activity associated with an object), the less likely that the object represents noise or insignificant data. Two variations of this principle were applied in the grouping mechanisms tested as part of the work reported here. The first involved using the longest ungrouped track to seed the next group while the second variation involved disallowing tracks below a threshold length from having any influence in the grouping decision.

An important feature of Cooke’s work is that he purposely disregarded the principle of exclusive allocation. The main reason for this was to deal with the problem of co-incident partials. The current work bears similarity to Cooke’s system in that it allows for a hierarchical approach to grouping, although only the lower levels of grouping are considered with a view to finding improved algorithms to perform this grouping.

In his initial work, Ellis [74] proposed grouping based on four metrics: common onset, harmonicity, proximity and common modulation. The four metrics were to be combined “by some kind of iterative approximation technique such as the k-means algorithm. Unlike Cooke, Ellis did allow the principle of exclusive allocation. The inability of such a technique to adequately deal with co-incident harmonics was cited as one of the motivations for Ellis abandoning this approach in favour of his “prediction driven” approach.

Ellis’ later work on “prediction driven computational auditory scene analysis” (PDCASA) departs significantly from the traditional CASA architecture, which Ellis refers to as data driven. The first significant difference between traditional CASA systems and the PDCASA approach, illustrated in Figure 4-2, is that the signal features that are output by the auditory front end are not the traditional time-frequency partials but rather three different features said to be the minimum required to describe any generic sound: noise clouds, impulses and “wefts”. The grouping of these components is based on a blackboard architecture that is governed by a world model, which consists of a series of hypotheses that describe likely valid groupings. An analogous idea is a system
that predicts a word that is being typed from the first few letters based on legal combinations of letters in the language. The world model is based on heuristics, is highly context dependent and may, as a result, be tuned to a particular environment. For example, in an automatic speech recognition application, the world model itself would apply speech recognition techniques to perform the grouping. Hence, the system looses an element of generality, which is considered a disadvantage from the perspective of the work proposed here.

Since the work of Ellis many CASA systems have focussed on the correlogram representation from which his wefts were derived. This representation is effectively a wideband analysis technique that seeks to study the interferences produced in the filterbank channels due to unresolved harmonics. In this model harmonicity is derived as peaks in the autocorrelation function of the channel outputs, particularly in the higher frequency channels. The concept of a frequency or amplitude contour is now somewhat obfuscated although frequency and amplitude modulation rates are sometimes used as grouping cues.

However, despite the bias against sinusoidal-based approaches Klapuri and colleagues have recently been developing a CASA-based source separation technique for musical instrument sounds that is based on a series of extensions and refinements to the basic sinusoidal model [38][135][136][141]. They identify a mechanism for dealing with the problem of co-incident partials that involves using information in regions where no interference occurs to moderate track formation in regions where co-incident harmonics may be a problem. Since all the tracks in a group should have correlated amplitude and frequency contours, it is relatively simple, and perceptually justified, to replace either of
these contours for a particular track with one derived from the other tracks in the group. Determining which tracks will likely interfere with each other is relatively trivial once the pitch of the two groups has been determined. If the pitches of the two groups, A and B are given by $f_{0A}$ and $f_{0B}$, then the harmonics that will interfere with each other will be all those for which the following relationship holds:

\[ \frac{n_A}{n_B} = \frac{f_{0B}}{f_{0A}} \]  

In equation 4—1, $n_X$ represents the harmonic number of the track from group X.

The ability to exploit statistical redundancy in a sinusoidal representation to account for not only coincident partials, but also tracking errors due to weak partials in the presence of noise, was independently recognised by the author. This provided a significant motivation for continuing to explore a CASA system based on a somewhat out of favour sinusoidal model. One advantage of this approach has been the ability to exploit, in a more explicit manner, the grouping cues that are somewhat obfuscated by the correlogram approaches, that is frequency and amplitude contour.

The general approach adopted in the current work was to individually develop and evaluate metrics for grouping the partials based on harmonicity, frequency contour and amplitude contour similarity. For reasons that will be discussed later in this chapter, common onset was only loosely implied as a similarity measure. Having evaluated a number of different approaches for grouping the tracks along each of the individual dimensions, the best performing in each dimension was then used to evaluate two different combined approaches for track grouping.

### 4.2. Evaluating Grouping Performance

#### 4.2.1. Introduction

Before proceeding with the description of the various grouping strategies that have been developed and evaluated, it is prudent to explain the evaluation procedures employed. There are a number of ways to evaluate grouping performance at a global level. Generally they involve inverting the representation used to perform the separation and evaluating the waveform produced for each group via either a subjective or an objective measure to determine the quality of the separation. However, given that during algorithm development many hundreds of tests were to be performed, neither of these techniques was deemed to be practical. Further, the global nature of these tests means that it is impossible to have an accurate indication of the relative performance of each
grouping technique with respect to the others. Hence, a different method of evaluation was required.

Since we were interested in determining the accuracy of the track grouping algorithms, it was decided that the evaluation procedure should operate on the grouped tracks themselves, rather than on the inverted signal. The procedure that was developed involved selecting a subset of the test files used and manually grouping the tracks in these files to act as a reference against which to compare the automatically grouped tracks. The files in the subset were selected because they contained tracks that were relatively easy to group manually. All such files were selected to ensure that the subset was as representative of the entire data set as possible.

In some ways, this track-based evaluation procedure is similar to that employed by Cooke [118] who also performed an evaluation based on the group allocation of his synchrony strands. However, Cooke chose to have a completely automatic procedure which exploited the fact that his test cases consisted of artificially mixed sources for which he had the original single source recordings. In deciding whether the synchrony strands were correctly grouped he searched for a correspondence between the groups extracted from the mixed source signal and the synchrony strands formed for the corresponding single source files. The evaluation metrics were the percentage of synchrony strands that were present in a group with respect to the number expected for that group as well as the percentage of synchrony strands whose presence in the group was unexpected again with respect to the expected size of the group. This evaluation procedure needed to take the following correspondence relationships into account (A and B represent the single source signals and AB represents the mixed source signal) [118]:

- A single element of AB corresponds to a single element of A;
- A single element of AB corresponds to more than one element of A;
- A single element of AB has no corresponding element in A;
- Several elements of AB correspond to a single element of A; and
- Several elements of AB correspond to several elements of A.

Cooke reduces this set of considerations by taking each element in AB separately, thereby effectively collapsing the last two considerations into the first two cases. This method suffers two chief disadvantages. The first is that the correspondence will rarely be absolute, hence, a level of flexibility is required in the matching algorithm, which is a potential source of error, particularly in an automated process. The second disadvantage
is that it relies on having access to the original single source data, hence, the technique
could not be employed to evaluate the performance of a system using data that were
recorded in a more natural setting. This is a significant consideration since Kouwe et al
[137] found that the relative performance of CASA and BSS systems is dependent on
whether the sounds were artificially mixed or naturally recorded. Nevertheless, an
automated procedure has some merit for testing large data sets and might also be useful
in removing the restriction on test file selection that requires it to be easy to group
manually. Hence, while the technique was deemed unsuitable for initial testing and
evaluation, it may be a useful supplementary evaluation method for future research
when the number of test cases is increased.

4.2.2. Evaluation Criteria
To compare the automatically generated grouping for a given file with the manually
generated grouping, a series of statistical and analytic tests were performed. Table 4-1
defines each of these tests and gives the ideal value of each.

<table>
<thead>
<tr>
<th>Test Description</th>
<th>Test Formulation (See below for definition of symbols)</th>
<th>Ideal Value</th>
<th>Value Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Number Ratio</td>
<td>[ GR = \frac{M_a}{M_r} ]</td>
<td>1</td>
<td>( \left[ 0, \frac{N}{M_r} \right] )</td>
</tr>
</tbody>
</table>
| Hausdorff Distance Manual \( \Rightarrow \) Automatic | \[ h(S_r, S_a) = \text{MAX}\{h(G_r)\} \quad \forall \ G_r \in S_r \]
and \( h(G_r) = \text{MIN}\{h(G_r, G_a)\} \quad \forall \ G_a \in S_a \)
where \( h(G_r, G_a) = \text{MAX}\{h_i\} \quad \forall \ 0 \leq i < N_r \)
where \( h_i = \text{MIN}\{|T_{ri} - T_{aj}| \quad \forall \ 0 \leq j < N_a \}
where |T_{ri} - T_{aj}| = \sqrt{(t_{ric} - t_{ajc})^2 + (f_{ric} - f_{ajc})^2}
and \( T_{ri} \in G_{ri} \) AND \( T_{aj} \in G_{aj} \)
and \( (t_c, f_c) = \sum_{l=0}^{L-1} (t_l, f_l) \) | 0 | \[ \left[ 0, \frac{M_r}{M_a} \right] \] |
| Hausdorff Distance Automatic \( \Rightarrow \) Manual | \[ h(S_a, S_r) = \text{MAX}\{h(G_a)\} \quad \forall \ G_a \in S_a \]
and \( h(G_a) = \text{MIN}\{h(G_a, G_r)\} \quad \forall \ G_r \in S_r \)
where \( h(G_a, G_r) = \text{MAX}\{h_i\} \quad \forall \ 0 \leq i < N_a \)
where \( h_i = \text{MIN}\{|T_{ai} - T_{rj}| \quad \forall \ 0 \leq j < N_r \}
where |T_{ai} - T_{rj}| = \sqrt{(t_{aic} - t_{rjc})^2 + (f_{aic} - f_{rjc})^2}
and \( T_{ai} \in G_{ai} \) AND \( T_{rj} \in G_{rj} \)
and \( (t_c, f_c) = \sum_{l=0}^{L-1} (t_l, f_l) \) | 0 | \[ \left[ 0, \frac{M_a}{M_r} \right] \] |
| Complete Groups Found | \[ CG = \frac{\text{COUNT}\{G_{ai} \in S_a\}}{M_r} \] | 1 | [0, 1] |
Partial Groups Found

Partial Groups Found

\[ PG = \frac{\text{COUNT}\{ G_{ai} \in G_{rj} \}}{M_a} \]
where \( G_{ai} \neq G_{rj} \) AND \( \text{SIZE}\{ G_{ai} \} > 1 \)

Ideal Value: \( 1 - CG \)

Value Range: \([0, 1]\)

Incorrect Groups Found

Incorrect Groups Found

\[ IG = \frac{\text{COUNT}\{ G_{ai} \notin S_r \AND G_{ai} \notin G_{rj} \forall G_{rj} \in S_r \}}{M_a} \]

Ideal Value: 0

Value Range: \([0, 1]\)

Mis-grouped Tracks

Mis-grouped Tracks

\[ MT = \frac{1}{M_a} \sum_{i=0}^{M_a-1} MT_{ai} \]
where \( MT_{ai} = \frac{(2C_{\text{over}} + C_{\text{under}} + C_{\text{wrong}})}{N_{ai}} \)

\[ C_{\text{over}} = \begin{cases} N_{ai} - N_{rj} & N_{ai} > N_{rj} \\ 0 & \text{otherwise} \end{cases} \]

\[ C_{\text{under}} = \begin{cases} N_{rj} - N_{ai} & N_{rj} > N_{ai} \\ 0 & \text{otherwise} \end{cases} \]

\[ C_{\text{wrong}} = \frac{N_{ai} - C_{\text{over}} - \text{SIZE}\{ G_{ai} \cap G_{rm} \}}{N_{ai}} \]

where \( m \in \text{SIZE}\{ G_{rm} \cap G_{ai} \} = \text{MAX}\{ \text{SIZE}\{ G_{rj} \cap G_{ai} \} \} \forall 0 \leq j < N_{rj} \)

Ideal Value: 0

Value Range: \([0, \infty]\)

Table 4-2: Definition of Symbols Used in Table 4-1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscript ( a )</td>
<td>Relating to automatically grouped data</td>
</tr>
<tr>
<td>Subscript ( r )</td>
<td>Relating to manually grouped (reference) data</td>
</tr>
<tr>
<td>S</td>
<td>Entire data set (grouped tracks)</td>
</tr>
<tr>
<td>G</td>
<td>A single group</td>
</tr>
<tr>
<td>T</td>
<td>A single track</td>
</tr>
<tr>
<td>((t, f))</td>
<td>Time and frequency (of a single peak)</td>
</tr>
<tr>
<td>N</td>
<td>Number of tracks in a group (with subscript) or set (without subscript—both sets have the same number of tracks since no tracks are added/removed during grouping)</td>
</tr>
<tr>
<td>M</td>
<td>Number of groups in a set</td>
</tr>
<tr>
<td>L</td>
<td>Number of peaks in a track</td>
</tr>
</tbody>
</table>

4.2.3. Evaluating the Evaluation Criteria

To determine the utility of the evaluation criteria defined in Table 4-1, the analysis was performed on a small set of data to qualitatively determine the correlation between the values and the accuracy of grouping achieved by the automatic process. It was found that the best single measure for evaluating grouping performance was the number of mis-grouped tracks present, \( MT \), which is effectively the same metric employed by Cooke.
in his scheme. The second best measure was the total number of correct and complete correct groups found, CG. A discussion of the performance of each of the measures appears below. All example graphs are derived from an analysis of the grouping accuracy of the simplest MSE-frequency contour-based grouping strategy.

- **Group Number Ratio:** This measure is obviously a very crude measure of grouping accuracy. It was included in the test to provide a quick test that might be useful in a combination of measures. However, given the problems with some of the other measures that are detailed later in this list, the option of combining measures was not further explored. Figure 4-3 shows an example of the group number ratio grouping accuracy metric. It will be noted that, according to this metric, the example algorithm performed best when an override threshold of approximately 50 was applied.

![Figure 4-3: Group number ratio for example case](image)

- **Hausdorff Distance:** The Hausdorff distance is a measure of the similarity between two sets that is commonly used in image processing [138]. As it is not commutative, it is necessary to calculate it in both directions. In the traditional image processing application, the Hausdorff distance is calculated between a set of points that represent some reference image and another set of points that represent some test image. In order to conform the track data set to this concept, the distance measure needed to be calculated in two stages. The first stage assumed that each track was a point (represented by its centre of gravity) and calculated the Hausdorff distance between individual groups. The second stage calculated the Hausdorff distance between the two data sets, assuming that each group in a set was a point. The distance between groups required to calculate
this secondary Hausdorff distance was simply the Hausdorff distance as calculated in the first stage.

While this procedure initially promised to be a viable quantitative measure of grouping performance, a significant problem with the space was discovered. In the traditional image processing approach an Hausdorff distance of 0 is most likely to indicate that the test image is a portion of the reference. In the track domain, however, there are two factors that make the Hausdorff distance a somewhat misleading accuracy metric. Firstly, unlike in the image processing approach, we are performing a global comparison between several groups simultaneously. Secondly, the initial set of “points” (i.e. tracks) is identical for both files. Because of these two factors there are many situations where an incorrect grouping can result in an Hausdorff distance of 0. For example, if the automatic grouping procedure fails entirely and no groups are formed, each track is a perfect subset of one of the reference groups. In the situation where there is only one reference group, the overall Hausdorff distance in the direction of automatic ⇒ reference will be zero. Similarly, if the automatic grouping procedure fails such that all the tracks are grouped to form a single group, when several groups are present, the Hausdorff distance in the direction reference ⇒ automatic will be zero.

With these deficiencies in mind, another error measure, sum of the two Hausdorff distances, was considered. It should be evident from the previous discussion that the Hausdorff distances in both directions will be zero when the reference and test groupings are identical. Hence, plotting the total of the two Hausdorff distances can give a rough indication of grouping accuracy.

![Figure 4-4: Hausdorff distance grouping accuracy metric](image-url)
Chapter 4: Frequency and Amplitude Contour-based Grouping

- **Number of Complete Groups Found:** This is a very strict measure of grouping performance. Only those groups in the automatically generated set that are complete matches for some group in the reference set are counted. This is perhaps the ideal measure for grouping performance, however, given that there will unavoidably be noise in the data, it would be naïve to expect that many test cases would achieve a perfect or even very high score on this measure. This was indeed found to be the case in both initial trials and exhaustive testing of each algorithm. Further, as the track extraction algorithm only needs to have a model track to proceed, each group need only contain enough correct tracks to form such a model. An example of the results obtained for this measure is provided in Figure 4-5.

- **Number of Partial Groups Found:** This measure is a count of the number of groups in the test set that are a perfect subset of a group within the reference set. There is, however, a potential problem with this approach and that is that a file in which the grouping has failed completely can generate a perfect score, in much the same way as the Hausdorff distance does. To mitigate the effects of this, single track groups are excluded from the count. Since this measure strictly counts only those groups that are partial subsets, it is still a relatively difficult condition to satisfy. The ideal measure, therefore is to either minimise the number of incorrect groups, as detailed in the next point, or to use the combined total of completely and partially correct groups as the grouping performance measure. An example of the results obtained for this measure is provided in Figure 4-5.

- **Number of Incorrect Groups Found:** This is a count of all groups that do not fall into one of the previous two categories. Thus another way to calculate this measure would be to subtract the sum of the previous two counts from the total number of groups in the test set. This is once again a fairly strict measure since it only takes one track to be incorrectly assigned to a group that would otherwise form a subset of a valid group for the group to be considered an incorrect group. An example of the results obtained for this measure is provided in Figure 4-5.

- **Number of Mis-grouped tracks:** This measure counts all tracks that have been assigned to the incorrect group. It is possible that an automatically generated group contains tracks from two or more different groups. Hence, it is necessary to first decide which group in the reference set the current test group most closely resembles. This is done by determining the group that has the greatest number of
tracks in common with the test group. Should there be two or more groups for which the test group is an equally likely match, then all are considered to be corresponding groups. The number of mis-grouped tracks in a particular group is then the number of tracks that the test group and its corresponding reference group/s do not have in common. It should be noted that this number is not an absolute value, given that it is dependent on the grouping itself. The metric indicates the proportion of tracks that have been incorrectly grouped relative to the number of groups that have been found. Thus, it was elected to represent this value as a percentage in the reported results. This precipitated the apparently odd situation in which some of the percentage values were greater than 100%. This is not an error, given that the total number of tracks is always greater than or equal to the number of groups found for a particular data set. Nevertheless, this measure proved to be a very indicative measure of grouping accuracy as it allows one or two tracks to be incorrectly grouped without discounting the entire group.

Some values of particular note are apparent in Figure 4-5. Most notably, all the values plateau after a minimum override threshold value of approximately 70. The value of each of the evaluation parameters beyond this point represents a constant value that occurs when, for each of the test files, all overlapping tracks are grouped together into a single group. This will occur at a different threshold for each algorithm but will definitely occur at some point. The plateau value of each parameter was determined using Table 4-3.
Given that Table 4-3 represents the statistics for each file when the tracks were overgrouped, it might be surprising to note that the number of groups found is not always 1 as might be expected (cf the entries for file 9, 18 and 22). This is because these files contain tracks that are temporally remote from one another. One such example is shown in Figure 4-6.

<table>
<thead>
<tr>
<th>File #</th>
<th># Tracks</th>
<th># Groups Found (% of expected)</th>
<th>Hausdorff Distance (found-&gt; expected)</th>
<th>Hausdorff Distance (expected -&gt; found)</th>
<th>% Complete Groups</th>
<th>% Partial Groups</th>
<th>% Incorrect Groups</th>
<th>Mis-grouped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>2</td>
<td>50.0</td>
<td>129</td>
<td>0</td>
<td>129</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>4</td>
<td>25.0</td>
<td>134</td>
<td>0</td>
<td>134</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>3</td>
<td>33.3</td>
<td>103</td>
<td>0</td>
<td>103</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>4</td>
<td>25.0</td>
<td>136</td>
<td>0</td>
<td>136</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>5</td>
<td>20.0</td>
<td>964</td>
<td>0</td>
<td>964</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>4</td>
<td>50.0</td>
<td>304</td>
<td>85</td>
<td>389</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>1</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>6</td>
<td>16.7</td>
<td>87</td>
<td>0</td>
<td>87</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>3</td>
<td>66.7</td>
<td>19</td>
<td>0</td>
<td>19</td>
<td>33.3</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>6</td>
<td>16.7</td>
<td>410</td>
<td>0</td>
<td>410</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>1</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>2</td>
<td>50.0</td>
<td>181</td>
<td>0</td>
<td>181</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>2</td>
<td>50.0</td>
<td>320</td>
<td>0</td>
<td>320</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>2</td>
<td>50.0</td>
<td>396</td>
<td>0</td>
<td>396</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>2</td>
<td>50.0</td>
<td>155</td>
<td>0</td>
<td>155</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>1</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>6</td>
<td>16.7</td>
<td>181</td>
<td>0</td>
<td>181</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>18</td>
<td>7</td>
<td>2</td>
<td>150.0</td>
<td>0</td>
<td>114</td>
<td>114</td>
<td>50.0</td>
<td>33.3</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td>5</td>
<td>20.0</td>
<td>200</td>
<td>0</td>
<td>200</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>1</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>21</td>
<td>13</td>
<td>4</td>
<td>25.0</td>
<td>386</td>
<td>0</td>
<td>386</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>22</td>
<td>7</td>
<td>5</td>
<td>40.0</td>
<td>268</td>
<td>0</td>
<td>268</td>
<td>20.0</td>
<td>0.0</td>
</tr>
<tr>
<td>23</td>
<td>20</td>
<td>3</td>
<td>33.3</td>
<td>128</td>
<td>0</td>
<td>128</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>1</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>25</td>
<td>15</td>
<td>8</td>
<td>12.5</td>
<td>243</td>
<td>0</td>
<td>243</td>
<td>0.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Avg 11.76 3.32 52.0% 189.76 7.96 2 24.1% 1.3% 73.3% 83.2%

Table 4-3: Table used to calculate plateau values of evaluation metrics
4.2.4. Measure for pair-wise comparison of techniques

Although the mis-grouped tracks measure was found to be a good absolute measure of grouping accuracy, it was decided to define a metric that would be useful for comparing pairs of algorithms with respect to each other. The method defined involves a simple histogram based approach, which is applicable to the comparison of any number of algorithms. For each of the test files, the best value achieved for the grouping accuracy metric is determined (over all combinations of algorithm and the algorithm parameters that were varied for the test). Then each combination of algorithm and parameters for which the same value was obtained is given a vote of 1 in the histogram while all other combinations are given a vote of 0. The histogram consists of the totals across all files for each combination of algorithm and algorithm parameters. The concept is illustrated below in Figure 4-7 where it is assumed two algorithms are being compared with respect to the minimum threshold value across 4 files. It should be obvious that this method clearly supports comparison among any number of algorithms as long as only one parameter value (in this case minimum threshold) is varied for any given trial.

Figure 4-6: Example of a file containing tracks that are temporally remote from one another
In this case a single track, indicated by the red ellipse, does not overlap with any of the other tracks in the file.
4.2.5. Summary of Grouping Evaluation Techniques

Given the forgoing discussions, and the need to have a method of comparing techniques that gives accurate results in terms of correlation between the accuracy metric and the observed groupings while remaining efficient in the sense of minimising the data that
needs to be interpreted, it was decided to use only two of the comparison metrics discussed in this section. The first was the absolute measure of mis-grouped tracks and the second was the relative performance histogram. In most of the results presented throughout the remainder of this thesis, both of these metrics are plotted on the same graph for comparison purposes.

4.2.6. Grouping Accuracy Chance Level

When performing statistical analyses of the grouping results, nothing can be said of the performance of any one technique in an absolute sense without reference to the grouping accuracy that could be achieved by chance. That is, for comparison purposes, we need to determine what would be the expected average grouping accuracy should the grouping be performed by simply selecting both the number of groups and the group assignment for each track at random.

However, given the nature of both the test data and the grouping accuracy evaluation metric that was selected (that is, the “mis-grouped tracks” metric), determining the value of this measure for a random grouping is a non-trivial matter. Since the mis-grouped tracks metric requires knowledge of the contents of both the reference grouping and the test grouping, it would be necessary to determine all possible groupings for all test files in order to determine the chance value of this metric. In the case of the files containing only a few tracks, this is a relatively straightforward problem since, for files containing 5 tracks or less the number of possible grouping permutations is 47 or less. However, this number quickly grows with the number of grouping permutations possible in a file with 25 tracks (the maximum number in the test data set) being $6.31746 \times 10^{22}$. Hence it was decided that determining the chance value of this metric empirically would be impractical.

A more practical approach is to estimate a lower bound for the chance level empirically. In order to do this, the following assumptions were made:

- The only possible groupings are:
  - Entirely correct;
  - All tracks unassigned to any group (i.e. each track is a single track group);
  - All overlapping tracks assigned to a single group; or
  - Any grouping that gives the maximum possible value for the grouping accuracy measure.
Each of these events noted in the previous point is equally likely. This is an extremely conservative estimate, since, for all but the files containing less than 4 tracks, the last case would be the most likely to occur as the other three can only be satisfied by a single permutation while there are several that would satisfy the last.

To calculate a conservative estimate for the lower bound of the accuracy measure, the accuracy measure for each of the cases listed under the first assumption was calculated for each of the test files. Then an average value for the accuracy measure was calculated such that one quarter of the files contributed their accuracy measure for the first case, one quarter for the second case and so forth. The selection as to which files would contribute which measure was made such that those files for which perfect grouping was most likely contributed this measure. After this, the measures for the remaining files were selected such that the average value would be minimised. The value thus calculated was 69.9.

It will be noted that the lower bound for the accuracy measure stated in the previous paragraph is below the average value of this metric for the case where all files are overgrouped and slightly above that for when all files are completely undergrouped. However, as was mentioned in the foregoing discussion, this value represents a conservative estimate, and it is likely that the chance level would be higher than the value for the overgrouped case since the maximum value of the measure is more likely to occur and is generally much higher than the value for the overgrouped case. Indeed, the average maximum value of the accuracy metric across all files is 226.9. To give a more realistic average value for the chance level, the selection procedure described in the previous paragraph was modified as follows: the first quarter of files with the fewest tracks contributed a 0 value (perfectly grouped) the next quarter of files with the greatest number of tracks contributed their maximum possible value for the accuracy metric and the remaining half were divided equally such that the values contributed for the overgrouped and undergrouped cases minimised the average. Under these conditions the chance level of the accuracy metric is 96.65%.

Another statistic of interest in evaluating grouping performance is the probability of achieving a perfect grouping by chance. For files with more than 5 tracks this is less than 1% while for those with less than 5 tracks the probability ranges from 6.7% to 50%. The probability that all five files containing less than 5 tracks be grouped perfectly by chance is 0.014%. Similar probabilities for the results achieved with each of the algorithms under test will be reported as appropriate.
4.3. Techniques for Contour-based Grouping

Given the bias against sinusoidal representations in modern CASA systems, and the classical preference for harmonicity as a grouping cue, little work has been performed in the area of contour-based grouping. However, it was recognised that comparing two tracks by their frequency and amplitude contours is essentially similar to scale independent shape comparison, which is a common problem in image processing. Hence, the starting point in determining a suitable similarity measure for frequency and amplitude contours was to consider some common image processing techniques for achieving the same goal. This section reports on the limited work on frequency and amplitude contour grouping that has appeared in the literature and will then discuss the various image processing techniques that could be applied to the problem.

4.3.1. Existing Metrics for Contour-based Grouping

Asserting that frequency modulation rate did not have as much psychoacoustic experimental support as did amplitude modulation for being a grouping cue, Cooke [118] only presented details of an amplitude modulation similarity metric. However, since the two problems are essentially very similar, there should be no reason why the same technique could not be applied to frequency contour matching. Cooke suggested that common amplitude modulation could be seen at two different time scales: common envelope repetition rate and common amplitude fluctuations. The former would be the result of some periodic amplitude variation as might be the result of vibrato, for example, while the latter represents the characteristic, slowly changing trajectories that all signals display in the time-frequency domain. Common envelope repetition rate is considered a higher-level grouping cue by this work while common amplitude fluctuation corresponds to “contour-based” grouping in the current work.

Cooke’s measure for the similarity across envelope repetition rates is given by [118]:

\[
sim(x, y) = e^{-\frac{1}{2}\left(\frac{x-y}{\sigma}\right)^2}
\]

where \(\sigma\) is set to 25Hz and \(x\) and \(y\) are the amplitude modulation rates of the two groups given by measuring the distance between successive peaks in the instantaneous amplitude at the output of the complex gammatone filter. However, given that searching for peaks in such a noisy signal is a troublesome exercise, Cooke adopted the approach of low-pass filtering the output and then measuring the distance between successive downward going zero crossings.
In order to measure the similarity between two synchrony strands based on common amplitude fluctuations Cooke recognised that the contours must first be scaled to a common level to avoid problems associated with varying modulation depth and absolute amplitude across the tracks in a group. Cooke scaled his synchrony strands by subtracting the mean and then rescaling the amplitudes such that the variance along the track was unity [118]. The similarity metric was then a simple cross-correlation between the overlapping amplitudes of the synchrony strands:

\[
sim = \sum_{t=t_1}^{t_2} \hat{a}_{\text{strand}}(t) \hat{a}_{\text{seed}}(t)
\]

In equation 4—3, \( \hat{a}_{\text{strand}}(t) \) represents the instantaneous scaled amplitude of the candidate strand for the group under consideration and \( \hat{a}_{\text{seed}}(t) \) represents the scaled amplitude of the ‘seed’ for the group. This measure is very similar to the mean-squared error metric that was employed in the work reported here.

The initial proposal of Ellis [72] suggested common modulation as “a powerful cue to the fusion of the relevant components” of a frequency-modulated periodic source. He suggests two methods for affecting this grouping; the first is to perform a pair-wise comparison between candidate tracks while the second is to have “some modulation feature detector run over the tracks and flag moments of modulation perhaps above some threshold.” However, having abandoned as was mentioned previously, the track-based grouping approach in his later work, no details of either of these methods were provided.

Virtanen and Klapuri [141] used the mean-squared error between two tracks as their contour similarity measure with the track frequencies and amplitudes first scaled by their average values, \( f_i \) and \( f_j \):

\[
d(t, f) = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} \left( \frac{f_i(t)}{f_i} - \frac{f_j(t)}{f_j} \right)^2
\]

The mean-squared error is a natural choice for a similarity measure when formulating the problem in terms of image processing and is thus one of the techniques that was considered in the work reported here.
4.3.2. Image Processing Techniques Applicable to Contour-based Grouping

Formulating the problem of contour-based grouping as one of scale invariant shape classification, suggested a number of algorithms and two broad approaches that could be applied. The two broad approaches were: model based and distance-vector based classification. Model based classification involves matching the shapes to a small, fixed set of model shapes. One disadvantage of model-based approaches is that they can fail to find a classification if the data set contains an element that does not fit any of the models, hence they are not applicable to all cases. Nevertheless, it was decided to investigate whether model-based classification would be appropriate for the track grouping task.

Distance-vector based classification involves calculating a distance (or similarity) metric between pair-wise combinations of elements and then grouping together those for which the similarity metric falls below a given threshold. The methods of Cooke [118] and Virtanen and Klapuri [141] are examples of this approach. A number of algorithms exist to govern how the clustering (or grouping) proceeds.

The simplest of these, employed by Cooke [118], involves arbitrarily selecting a “seed” element from the data set, perhaps according to some dominance measure, and then simply calculating the distance between this seed and all other elements in the data set in order to find all those elements that are within the threshold of the seed. If any elements remain in the data set after this search, a new seed is selected from among the remaining elements and the grouping procedure is repeated. If no element is found in the original data set that is within the threshold of the seed, the seed is considered to belong to a singleton group. The entire process is repeated until no elements remain in the data set.

The method of Virtanen and Klapuri [141] is a slight variation to this approach where the initial seeds are not individual tracks but rather small groups of tracks that have been formed by matching onset times. The distance between the seed group and each of the remaining tracks in the data set is then defined as the average distance between each of the tracks in the seed and the track under consideration. Virtanen and Klapuri adopted this method as a means for computational complexity reduction, over performing an exhaustive search to minimise the distance between tracks within each group:

\[
\min \left\{ \frac{1}{|S_1|} \sum_{j \in S_1} d_{all}(i, j) + \frac{1}{|S_2|} \sum_{l \in S_2} d_{all}(k, l) \right\}, \quad S_1 \cup S_2, S_1 \cap S_2 = \emptyset, (4-5)
\]

where \( S_1 \) and \( S_2 \) are the sets of tracks representing two sounds, \( S \) is the set of all tracks and \( |S| \) is the cardinality of a set. However a major drawback of the approach is that...
the initial grouping based on onset times is not very robust. It would fail, for example, for sounds that begin at the same time, as is often the case in music.

A more robust variation of the simplest technique is the k-means or pair-wise nearest neighbour matching algorithm. This was the technique suggested by Ellis in his initial proposal [72]. This is an iterative technique in which every iteration results in the pair of elements that are closest together being grouped. Once a pair has been grouped, their average value is used as the representative element for the group. The most important factors in the success of this algorithm are the selection of the distance metric and the stopping condition.

The selection of the stopping condition is often determined empirically, however, this can result in a loss of generality, and can involve tedious, or even impractical, experimentation. A more generic method involves optimising the statistical distribution of tracks among groups. This might involve preserving some relationship between the mean distance within a group with respect to that between groups or the entire set.

Selection of the distance metric presents several alternatives. The simplest is a simple mean squared error between the pair of elements under consideration. If, as is the case here, the shape comparison must be independent of scale, the elements must first be scaled in some manner. This is often selected such that the centre of gravity or average value of the elements is common. However, a more robust approach when one element is a subset of the shape of the other is to perform shape normalisation. This involves scaling the elements such that the distance between them is a minimum. As was already mentioned, Virtanen and Klapuri [141] employed the mean-squared error as a distance metric where the tracks were scaled to have a common average (frequency or amplitude) value of unity.

Yet another metric that has been employed for shape matching is the Fourier shape descriptor. The Fourier shape descriptor for a given line is calculated in the polar coordinate space calculating the distance between each point along the line and its centre of gravity. A Fourier transform is then taken of the resultant vector. While this is a scale invariant technique that showed initial promise, it suffered the problem that varying track lengths presented a different scale problem.
4.4. Model-based Frequency Contour Grouping

Although model-based classification does suffer a lack of generality, it was decided to investigate its applicability to the task of track grouping [139]. The main motivation for this investigation was the concept expounded by Suga that only a small set of elements is sufficient to describe all sounds. Suga’s original set of elements, which, if we exclude the noise burst, contains only two elements: the tone and the sweep, was expanded to the “alphabet” shown in Figure 4-8. To allow for the most common variation in track shape in musical signals, a special class of sweep, the periodic, was also introduced. Further, the class of formant was added to account for the most commonly occurring track shapes due to speech.

![Figure 4-8: Alphabet of possible track shapes](image)

Given this set of track shapes, each of the track approximations may be classified into one of the above categories. If the shape of a track does not correspond to any of the above shapes, it is assumed that the unusual track shape is the result of two or more different tracks that were incorrectly joined by the tracking algorithm and it is thus broken up into the minimum number of pieces required such that each piece fits into a category. An example of the concept is illustrated in Figure 4-9.

![Figure 4-9: Classification of track shapes](image)
Given the classification procedure illustrated in Figure 4-9 as well as the fact that the classified tracks will form the basis of groups, and we would like each of these groups to form a semantically significant unit, the reason for expanding the alphabet of fundamental track shapes becomes apparent. While both the periodic sub-class and the formant class may be composed of simple combinations of tones and sweeps, the groups corresponding to tracks of these shapes would be far from semantically relevant if decomposed into these very primitive elements. For example, using an alphabet containing only pure tones and sweeps with the procedure of breaking up a track into classifiable pieces would ensure that a track corresponding to a single note played using vibrato would be broken into a series of very short alternating upward and downward sweeps which from a semantic perspective would not be particularly significant.

Having classified, each of the tracks, the grouping procedure employed in this method is very straightforward and relies most heavily on the harmonicity principle, that is, tracks that have a similar frequency contour and harmonically related frequencies are considered to belong to a single group. The algorithm may be described as follows:

1. Find the lowest track that is longer than the minimum allowed length. Set this to be the fundamental of the current group.
2. Make a copy of the track’s frequency contour.
3. FOR all multiples of the frequency contour < the maximum frequency in the data

Find all tracks of the same type as the current fundamental that lie within an allowable error bounds of the current multiple of the frequency contour and assign these to the current group.
4. Repeat steps 1-3 while ungrouped tracks remain.

The results of applying the above algorithm to three separate files are shown in Figure 4-10. The three files consisted of two single source files, one of them a female uttering the digit ‘one’ and the other a flute playing the note G2 without vibrato, and a mixed source file which was an artificial mix of the two single source files. One of the main problems with this procedure that is apparent in these results is that there is a tendency to leave ‘holes’ in the data when tracks from sources overlap. This is particularly the case when the overlapping region includes the onset of one or both of the sources. Hence, it was determined that this algorithm does not adequately address the challenge stated in section 4.1.
Figure 4-10: Results of classification based grouping procedure for single and mixed source files
Top to bottom: mixed source file, speech extracted from mixed source file, speech from single source file, music extracted from mixed source file, music from single source file.
Yet another problem with this procedure that became apparent with further testing is that the alphabet of track shape classes is still insufficient to describe all cases even restricting the audio domain to speech and music. Two examples of track shapes that would be incorrectly classified by the above procedure are shown in Figure 4-11 and Figure 4-13.

**Figure 4-11:** Tracks derived for a male speaker saying "g"

**Figure 4-12:** Close-up of fundamental track in Figure 4-11

The red markings indicate the pieces into which the track would be broken up using the template matching approach.
Figure 4-11 shows the tracks generated for a section of human speech. It is apparent that the lower frequency tracks display a concavity in both the upward and downward directions, and would probably best be described using a cubic curve rather than the second order formant shapes available in the alphabet of Figure 4-8. Figure 4-12 shows a close-up of the fundamental track from this data set that highlights the problem. Depending on the thresholds applied during track shape classification, this track would most likely be broken up into three formant sections and a sweep section as shown in the figure. Given that the tracks in Figure 4-11 were derived from the utterance of a single word, breaking up the track in this manner is contrary to the aim of generating track groups that represent semantically significant units.

Figure 4-13 shows yet another example of a track that would be divided incorrectly. This time the track is derived from a piano note. The most obvious problem here is the onset shape. At best this would cause a misclassification of this track as formant but more likely a part of the onset would be broken off from the rest of the track.

The two examples shown in Figure 4-11 and Figure 4-13 are only two in a large number of exceptions that were discovered upon observation of the results from the preliminary tracking algorithm. It would obviously be impractical to do an exhaustive investigation of all sound sources to generate an exhaustive fixed alphabet of fundamental shapes. Even if this were to be attempted, the necessarily large set of classes would result in the differences between classes being significantly reduced. One very likely outcome of this is that the probability of an incorrectly formed “composite” track (cf. Figure 4-9) being
misclassified as a legitimate shape in its entirety would be increased. This would obviously have a significant impact on the quality of source separation.

4.5. Distance Vector-based Grouping

4.5.1. Basic Grouping Algorithm
Section 4.3 presented a number of techniques for performing distance-vector based grouping. Given the considerations mentioned in that section, it was decided that the pair-wise nearest neighbour approach would be the best compromise between computational complexity and robust operation. The algorithm proceeds as follows:

1. Calculate distance between each pair-wise combination of tracks (or group model tracks)
2. Find the pair of tracks/groups that are the closest to each other
3. IF the closest pair are < the threshold apart
   a. Join the two tracks/groups to form a new group
   ELSE
   b. set the distance between this pair of groups to the maximum value
      (to exclude this combination from further consideration)
   END IF
4. REPEAT step 3 until a group has been found or no more valid pairs remain
5. REPEAT steps 1 – 4 until no more groups can be formed.

As was mentioned in section 4.3, the above algorithm depends heavily on the calculation of two important values, the distance between each pair-wise combination of tracks and the threshold at which to stop grouping.

Since the calculation of the distance metric has the most significant effect on grouping accuracy, it was the focus of the majority of research and experimental work. A number of different error metrics to effect this distance calculation were devised and tested. The metrics employed and their evaluation appear in section 4.2.2. The second important parameter, the cut-off threshold at which to stop grouping nearest neighbours, is the focus of the next section.

4.5.2. Threshold Calculation
The simplest approach to providing a threshold at which to stop grouping is to select a single, fixed threshold, the value of which is empirically determined. However, the
major disadvantage of this approach, discovered by empirical study, is that the optimum threshold value is heavily dependent on the nature of the track data. Hence, a threshold value that ensures proper grouping performance for one file will result in an incorrect grouping in another file. Further, for a mixed source file, such as speech and a musical tone, it is often impossible to find a single optimum threshold even for this one file. This was resolved by using statistical analysis of the data to determine a threshold value for each track in the file. The basic idea for a single track is as follows:

1. Calculate the distance between the track and every other track that it overlaps with in time.

2. Determine the mean, $\bar{x}$, and standard deviation, $\sigma$, of these distance values.

3. Set the threshold to be: $\text{MAX}\{\bar{x}, \bar{x} - \sigma\}$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{thresholdCalculation.png}
\caption{Illustration of threshold calculation}
\end{figure}

Given the tracks in Figure 4-14 the threshold calculation for track A, $Th_A$, proceeds:

\begin{align*}
\bar{x}_A &= \frac{(d_{AB} + d_{AC} + d_{AD} + d_{AE})}{4} \\
\sigma_A &= \sqrt{\frac{4 (d_{AB}^2 + d_{AC}^2 + d_{AD}^2 + d_{AE}^2) - (d_{AB} + d_{AC} + d_{AD} + d_{AE})^2}{4 (4 - 1)}} \\
Th_A &= \text{MAX}\{\bar{x}_A, \bar{x}_A - \sigma_A\}
\end{align*}

Note: that since tracks A and F do not overlap, the distance $d_{AF}$ is not included in the calculation of $Th_A$.

This procedure gives a threshold for every single track in the data set. Given a pair of groups, $\{G_1, G_2\}$, each with its own threshold value, $\{Th_1, Th_2\}$, the threshold that is
used at step 3 in the grouping algorithm is given by: $Th = \text{MAX}\{Th_1, Th_2\}$. This threshold value is also assigned to the resultant group once the pair has been combined.

However, one shortcoming of this approach is that it assumes that there is more than one group present in the data set which will not necessarily always be the case. This was easily resolved by introducing a fixed minimum allowable threshold value, $Th_{\text{min}}$. With this addition, the threshold calculation at step 3 of the algorithm becomes: $Th = \text{MAX}\{Th_1, Th_2, Th_{\text{min}}\}$.

### 4.6. Classification-Based Grouping Approaches

#### 4.6.1. Mean-Square-Error

Perhaps the simplest shape comparison metric available is the mean-square error (MSE). For this reason, it was the first technique employed as a measure of similarity along the frequency and amplitude contours. This section will detail the methods developed and used to generate an MSE-based distance metric for the frequency contour. Since exactly the same considerations apply for the amplitude contour, with only variations in some key values, no separate explanation for the amplitude contour will be given but rather these minor variations will be noted as appropriate.

The mean-square error between two functions, $F(\bullet)$ and $G(\bullet)$ is defined as:

$$\text{MSE} = \frac{1}{N} \sum_{n=0}^{N-1} \left( F(t_n) - G(t_n) \right)^2$$  \hspace{1cm} (4-9)

Since we were interested in comparing the shape of the frequency (or amplitude) contours independent of their exact frequency location, it is obvious that the tracks must first be scaled to some common level. The first approach to such scaling investigated involved scaling each track to a common average frequency of 2 kHz (or common average amplitude of 90dB SPL). This scaling may be expressed as follows:

$$\hat{f}_i = \alpha f_i$$  \hspace{1cm} (4-10)

$$\alpha = \frac{2000}{f_{\text{avg}}}$$  \hspace{1cm} (4-11)

where, $f_{\text{avg}} = \frac{1}{N} \sum_{n=0}^{N-1} f_n$  \hspace{1cm} (4-12)

The reason for selecting what might appear to be a high value for the common average is that, for most of the data, this represents a scaling up. The result of this is that any
differences in shape between tracks are amplified. This is a desirable characteristic when comparing shape since we want to be able to distinguish between tracks that may be similar but are indeed different. The concept is illustrated in Figure 4-15 that shows a pair of tracks scaled first to an average frequency of 200 Hz and then to an average frequency of 2kHz.

It is evident from the figure that when the pair is scaled to the lower frequency, it is uncertain whether the tracks are indeed a different shape or whether the slight differences are due to noise in the data. However, when the same pair of tracks is scaled to an average frequency of 2kHz, it is now very obvious that the tracks are indeed different shapes. As one would expect, the MSE error values for the pair scaled to the high frequency is 100 times greater than for the pair scaled to the low frequency (47584.6 and 475.8 respectively). Although this would appear to be easily accounted for by scaling the grouping thresholds accordingly, scaling the frequency values up before calculating the MSE has many advantages in that it avoids problems that might be caused by floating point resolution and allows the threshold values to be less sensitive to noise due to the increased dynamic range.

Having stated the case for scaling the frequencies up, one may ask, “why stop at 2kHz and not scale the frequencies even higher to, say, 20kHz?” There main reason for this is that the range of frequencies over which tracks representing sweeps and formants would
be extended considerably (by an order of magnitude). This would then magnify any “errors” in the track shapes caused by noise or a failure of the tracking routine to properly resolve closely located tracks. The second reason, which will be made clear shortly, is that the distances between tracks of similar shapes would be greatly amplified. Thus, in selecting a scaling factor, a compromise has to be reached between the errors that can occur at low frequencies due to precision problems and those that can occur at very high frequencies.

The results of scaling all the tracks from a mixed source signal to an average frequency of 2kHz are shown in Figure 4-16. The mixed source file in this case was a female speaker pronouncing the letter ‘o’ artificially mixed with a trumpet playing a passage from Handel’s “The Trumpet’s Loud Clamour” in the background. This file was selected for the illustration because the combination of tonal and formant shaped tracks clearly illustrates the strengths and weaknesses of the technique.

Figure 4-16 shows a fundamental problem with scaling all the tracks such that they have a common average frequency. Namely, because of their varying lengths, the formant shaped tracks are still fairly spread out in frequency even after scaling. This results in a large MSE between the highest and lowest of the tracks that should belong in the one group. The actual MSE between the two extreme tracks in the formant shaped group shown in Figure 4-16 is 6186.56 which is very close to the MSE between some of the formant shaped tracks and the tonal tracks (i.e. between groups) which has a minimum
of approximately 7000. Hence, with this scaling it would be very difficult to distinguish between the tonal and formant-shaped groups.

The reason that the formant tracks do not cluster as close together after scaling as might be expected is that, because of the signal characteristics and the interaction between the two sources, not all of the tracks have the same onset and offset. Since the formant shaped tracks vary greatly over frequency, the length of the track plays a significant role in its average frequency. The concept is illustrated in Figure 4-17.

![Figure 4-17: Scaling tracks to a common frequency](image)

4.6.2. **Shape Normalisation**

In order to overcome the scaling problem mentioned in the previous section, a new “relative” scaling technique was developed that is similar to shape normalisation that is sometimes employed in image processing. In this technique, the first track is scaled to an average frequency of 2 kHz while each of the subsequent tracks is scaled relative to the previously scaled track that it is most similar in shape to. The method proceeds as follows:

1. Sort all tracks by decreasing length and increasing average frequency.
2. Scale the first track such that it has an average frequency of 2-kHz
3. FOR all remaining tracks, T₁
   a. FOR all tracks that have already been scaled, T₂ with which T₁ overlaps
      i. Scale T₁ relative to T₂ recording the scale factor (SF).
      ii. Calculate and record the MSE between the scaled version of T₁ and T₂
   END FOR
   b. Find the scale factor corresponding to the minimum MSE, SF_{MIN}
   c. Scale T₁ by SF_{MIN}
   END FOR
The reason for sorting the tracks by decreasing length is to increase the probability that, once the first track has been scaled, subsequent tracks will overlap with at least this track. However, it is possible that no overlapping track is found among the previously scaled tracks. In this case, the track is simply scaled to a mean of 2-kHz.

**Scale one track, \( T_1 \), relative to another, \( T_2 \)**

The scaling procedure operates over the overlapping portion of the two tracks which is defined to be the interval \([t_0, t_{N-1}]\) such that:

\[
t_0 = \text{MAX}\{t_{1,0}, t_{2,0}\}
\]

\[
t_{N-1} = \text{MAX}\{t_{1,N_1-1}, t_{2,N_2-1}\}
\]

(4—13) (4—14)

To scale one track, \( F(t_1, f_1) \), relative to another, \( G(t_2, f_2) \), the ratio between each corresponding pair of frequencies is first determined:

\[
r_k = \frac{f_{1i}}{f_{2j}} \quad 0 \leq k < N \quad \forall \quad i, j \ni f_{1i} = f_{2j}, \{ f_{1i}, f_{2j} \in [f_0; f_{N-1}] \}
\]

where \( N \) is the number of peaks that the two tracks have in common:

\[
N = t_{N-1} - t_0 + 1
\]

(4—15) (4—16)

The first track is then scaled by each \( r_k \) and the MSE between the resultant contour and \( G \) is determined:

\[
e_k = \frac{1}{N} \sum (r_k f_{1j} - f_{2j})^2
\]

(4—17)

The scale factor \( R \) is then defined as the value of \( r_k \) for which the value of \( e_k \) is a minimum:

\[
R = r_{k_{\text{min}}} \ni e_{k_{\text{min}}} = \text{MIN}\{r_k\} \quad \forall \quad 0 \leq k \leq N - 1
\]

(4—18)

Finally, the track, \( F \), is scaled by the scale factor, \( R \):

\[
\hat{F} = R \times F
\]

\[
\Rightarrow \hat{f}_{1i} = R f_{1i} \quad \forall \ ni \ni f_{1i} \in [f_{1,0}, f_{1,N-1}]
\]

(4—19)

### 4.6.3. Results of Shape Normalised Scaling Procedure

The results of applying the relative scaling procedure to the mixed source file from the previous example are shown in Figure 4-18.
It is immediately obvious that the tracks, particularly those of formant shape, are clustered much closer together in frequency than was the case with the previous method. The MSE between the two extreme tracks of the group in this case is 310.77, which represents a reduction by nearly 95% from that achieved with the fixed average frequency scaling. Thus, this method provides a much more accurate basis for grouping to proceed.

**4.6.4. Accounting for onset and offset transients**

One problem that influences both shape normalisation and track grouping in general is the onset and offset transients of tracks. For the case of higher harmonics, the transience is often due to tracking errors, while in most cases the fundamental and low order harmonics display transients that are characteristic of the particular sound represented by the tracks. The reason that these ‘valid’ transients appear in the low order harmonics and not, in general, in high order harmonics, is that the signal power is usually concentrated in the low order harmonics, making the transients easier to track at this level. The significance of this to grouping is that the low order harmonics of a group may appear radically different in shape to the higher order harmonics based on almost any shape comparison metric. The matter is only exacerbated by the apparent transience introduced by tracking errors. An example of a grouping error caused by transience is shown in Figure 4-19.
Figure 4-19: Grouping error due to transience in the fundamental

Figure 4-19 shows a case where the fundamental track was not grouped with the rest of the tracks to which it obviously belongs due to onset transience. This is more obvious in Figure 4-20 which illustrates the tracks after shape normalisation.

Figure 4-20 clearly shows that the fundamental track displays a significant onset transient that can have an adverse affect on grouping. The transience also influences shape normalisation as this track is obviously slightly separated from the main group of tracks.

It will be noted that the fundamental also displays a significant offset transient, however,
in this case it is unlikely to greatly influence the grouping since only one track overlaps with the relevant portion of the fundamental. In order to counteract the influence of these transients on grouping, the tracks were trimmed by 15\% of their length at both the start and end. The result of this trimming procedure for the current example is shown in Figure 4-21.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure4-21.png}
\caption{Results of track trimming}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure4-22.png}
\caption{Result of applying shape normalisation to trimmed tracks}
\end{figure}

Figure 4-22 shows the result of applying the shape normalisation procedure to the trimmed tracks. Although the fundamental track is much more likely to be (correctly) grouped with the rest of tracks, the colouring has been maintained for the purposes of illustration. It is obvious that the normalisation procedure has resulted in the fundamental track being scaled much closer to the other tracks in the group as required.
4.6.5. Results for MSE based frequency and amplitude contour grouping

Frequency Contour Grouping Results

Having applied the relative scaling to the tracks, it is now possible to use the MSE between tracks as a distance measure to group the tracks according to their frequency or amplitude contour. The results of frequency contour based grouping using the MSE are shown in Figure 4-23 for various threshold values. By way of comparison, Figure 4-23 also shows the results where the tracks have been scaled to a common average frequency of 2kHz.

![Figure 4-23: Summary of results for frequency contour MSE-based grouping](image)

When reading Figure 4-23 it should be recalled (c.f. section 4.2.5) that the bar chart is a histogram of the number of files that performed best when comparing that particular algorithm and threshold combination with all others on the graph, while the line graphs are a plot of the average mis-grouped tracks measure across all files for the noted algorithm and threshold combination. Given these considerations, Figure 4-23 shows that the MSE based grouping of frequency contours performs best when the tracks are shape normalised. This result is as would be expected from the discussion in sections 4.6.1 to 4.6.3. The best performance overall is achieved when the override minimum threshold is set to a value of 40. Under these conditions, the “mis-grouped tracks” measure is equal to 40.25. The plateau in both sets of graphs after a minimum threshold of 75 indicates that all files were overgrouped from this point on.
Figure 4-23 also indicates that the performance of both versions of MSE-based grouping of the frequency contours is better than even the most conservatively estimated chance value for the mis-grouped track measure of 69.9% for threshold values below 60. To verify this finding, the number of files for which perfect grouping performance was achieved is plotted in Figure 4-24.

![Figure 4-24: Number of files for which grouping performance was perfect](image)

Figure 4-24 shows that at threshold values of 35 and 40, the mean normalised version of the frequency-contour-MSE-based grouping results in 7 files attaining a perfect grouping. The probability of this occurring by chance has an upper bound of $3.74 \times 10^{-8}\%$, representing the combined probability assuming that the seven files are those for which perfect grouping is most likely. Discounting those files for which an overgrouped condition corresponds to a perfect grouping, the probability of this occurring is still only 1.7%, at best.

**Amplitude Contour Grouping Results**

Figure 4-25 shows the grouping results using the MSE between amplitude contours as the distance metric. The relative count of best performance is at a maximum for the shape normalised tracks at threshold values of 25-35, indicating that optimum MSE-based amplitude contour grouping occurs under these conditions. However, the more indicative mis-grouped tracks measure displays a minimum of 63.7 at a threshold value of 25 when the tracks were normalised to a common mean amplitude, indicating that mean normalisation gives a slightly better performance than shape normalisation.
This somewhat unexpected result can be accounted for in two ways. The first is that the difference between the two mis-grouped tracks curves is very slight with less than 5% difference between any two corresponding values on the curves. The second consideration is that the amplitude contours, in general, display a relatively large degree of noise and variation across a group. As a result, the mean normalisation often gives better ‘separation’ of the amplitude shapes than does the shape normalisation procedure. An example is shown in Figure 4-26.

A final point of note is that the results above indicate that MSE-based amplitude contour grouping performs only slightly better than the conservative estimate of the chance value for mis-grouped tracks of 69.9. As was mentioned in the previous paragraph, the mis-grouped tracks measure for optimum amplitude contour grouping was 63.7, which
represents an 8.9% improvement over chance. To further investigate the performance of MSE-based amplitude contour grouping with respect to chance, the number of perfectly grouped tracks with respect to threshold values was plotted in Figure 4-27.

Figure 4-27 indicates that the number of files for which perfect grouping occurs remains below 5 for threshold values below 45. The 5 files that were grouped perfectly above this value represent files for which the “overgrouped” case corresponds with perfect grouping. It was noted earlier that the mis-grouped tracks metric indicated optimum performance at a threshold value of 25; only 2 files were correctly grouped at this threshold value. The probability of achieving this perfect grouping performance is 12.5%. These poor results are to be expected in light of the considerations noted about the noisy nature of the amplitude contours. Nevertheless, it was hoped that these could be improved upon with a different algorithm as will be discussed in section 4.6.6.

Comparison of Frequency and Amplitude Contour Grouping

Figure 4-28 compares the results achieved for MSE-based frequency and amplitude contour grouping. In keeping with the results reported above for each version, the frequency contour results are for tracks that were shape normalised while the amplitude contour results relate to tracks that were mean normalised. The count of files with best performance is relative to these two data sets only. Once again, the line graphs represent the mis-grouped tracks metric, which should be minimised for optimum performance, while the bar charts represent a histogram of relative performance, with a higher value indicating a better performance. Incidentally, at every threshold value, at least one file was grouped correctly. Hence the bar charts also indicate the number of files with perfect grouping performance with respect to threshold.
The results in Figure 4-28 clearly indicate that, for the MSE-based algorithm, frequency contour grouping performs significantly better than amplitude contour grouping. The optimum value of the mis-grouped tracks metric for frequency contour grouping is approximately 20 less than that for amplitude contour grouping, indicating an improvement of 33%. Excluding the “overgrouped” region of the graph, the relative performance histogram indicates an improvement of 50% with the best frequency contour count being 8 while amplitude contour grouping achieved a maximum count of only 4. The main reason for the relatively poor performance of amplitude contour grouping is that, as has already been mentioned, the amplitude contours display a relatively significant level of noise. The frequency contours, on the other hand display much less noise. Figure 4-29 gives a few examples comparing the frequency and amplitude contours of test files.

It is clear from Figure 4-29 that the frequency contours display less noise and much less variation within groups than do the amplitude contours. Hence, the conclusion, drawn from the results in Figure 4-28, that MSE-based grouping performed better for frequency contour grouping than amplitude contour grouping is not surprising. However, careful examination of the amplitude contours in Figure 4-29 reveals that while the amplitude contours belonging to a single group vary considerably on a local level, the global shape remains relatively constant. This observation motivated the development of a more global shape comparison metric in the hope of improving amplitude contour grouping, which is the subject of the next section.
Figure 4-29: Comparison of scaled amplitude contours and frequency contours

The left hand pane in each pair represents the scaled amplitude contour of the tracks while the right hand pane shows the frequency contour. The colouring of the tracks reflects the expected grouping. Notice that, in general, the amplitude contours display a greater level of noise. Notice also, that if ‘global’ shape is considered, the grouping of the amplitude contours is more obvious.
4.6.6. Shape Parameters

While the above results indicated that the MSE based algorithms were able to achieve grouping results far in excess of the chance level, it was also noted that the MSE is, in general, highly sensitive to noise which, as discussed, is of particular concern in the amplitude contours. Hence, it was felt that a parametric description of the track contours might provide better grouping performance. To this end, it was decided to determine a set of parameters that would sufficiently describe the shape of a track along both the frequency and amplitude contours for it to be correctly grouped while remaining general enough to avoid problems caused by noise and onset transients. From observation of the frequency and amplitude contours, the following list of characteristics were determined to be important or useful to group formation.

- Tones that are essentially steady state after a brief onset transient can be difficult to group in the presence of tones displaying vibrato. This is because there is often a large error introduced in the transient region that results in a larger MSE between tracks belonging to the “steady state” group than between some of the steady state tracks and the tracks displaying vibrato.

Figure 4-30 gives an example of this problem. Assume that the tracks illustrated were derived from a file containing two sources, one a piano tone and the other a wind instrument tone played with vibrato. Tracks A and C would be due to the former while track B would result from the latter. Hence, we would like A and C to be grouped and track B to remain independent (or to group with any other like tracks in the file). However, the figure also illustrates the common occurrence whereby some tracks do not include onset transients even though they are present. This will usually occur at high frequencies where the gradient of the time/frequency contour is too high for the initial tracking algorithm. If the onset transience displayed by track A is sufficiently large, and the frequency modulation index of track B is sufficiently small, the MSE between tracks A and C may be less than the MSE between tracks B and C. This is obviously a problem for frequency contour based grouping that must be accounted for by the parametric description.
Formant shaped tracks can usually be adequately described by a third order curve. As such they will have at most two local extrema. This is in contrast with a tone that is played with vibrato, which may have any number of local maxima and minima. This is a characteristic that may be exploited to distinguish between formant shaped tracks and all types of tonal tracks (both steady state and vibrato).

Simple sweeps can be difficult to distinguish from some formant shaped tracks. Some formant shaped tracks display a very shallow concavity, which can make them difficult to distinguish from a noisy track corresponding to simple sweep. This is yet another problem that must be accounted for in the shape parameters.

Given the above considerations, the following set of parameters was proposed to describe the shape of the tracks in both frequency and amplitude.

- Gradient of least-squares line through frequency contour of entire track;
- MSE between least-squares line and frequency contour;
- Gradient of least-squares line through left half of the frequency contour;
- MSE between left hand least-squares line and frequency contour;
- Gradient of least-squares line through right half of the frequency contour; and
- MSE between the right-hand least-squares line and frequency contour.
These shape parameters account for each of the considerations given above in the following ways:

- The MSE and gradient difference values for the right hand portion of the tracks are guaranteed to be much smaller for the A, C pair than the B, C pair as required;
- Having broken the track into two halves, a formant shape track can be much more closely represented by the two straight-line segments. This means that the MSE between each of these line segments and the formant will be much smaller than the MSE between the corresponding line segments and an overlapping section of vibrato; and
- Sweeps will be distinguishable from formants because the gradients of the two half lines that are fitted to a sweep will vary less than the two that are fitted to even a shallow formant.

Figure 4-31 illustrates the shape parameter calculation for several tracks that should belong to the same group.

![Figure 4-31: Shape parameters for some typical track shapes](image)

Figure 4-31 shows a situation that is typical of real data where portions of tracks can often be missing due to tracking errors or interaction between the harmonics of various sources. However, if shape parameters are only calculated for each track independent of the others, as illustrated in Figure 4-31, each one of these tracks would obviously be considered to be a different shape and hence not grouped. To overcome this problem the parameters were “time-normalised” or calculated for each track over the portion that overlaps with each other track in the data set. In other words, for each track, $i$, a set of parameters, $p_{i,j}$, are calculated where $0 \leq j < M$, and $M$ is the total number of tracks.
When the vector $p_{i,j}$ is being calculated, only those peaks of track $i$ such that $m_i = m_j$ are used in the calculation. When $i = j$, we have the parameters for the entire track, however, this information is of little use to us, since we already know that a track is a perfect match for itself. This concept of normalised shape parameters is illustrated in Figure 4-32 below.

**Figure 4-32: Time-normalised shape parameters**

**Algorithm for calculating the shape parameters**

Given the frequency contours of two tracks $f_i(m_i)$ and $f_j(m_j)$,

1. Determine the range of time frames over which the two contours overlap:

   $$m_{ij,0} = \text{MAX}\{ m_{i,0}, m_{j,0} \}$$
   $$m_{ij,end} = \text{MIN}\{ m_{i,end}, m_{j,end} \}$$

   \hspace{1cm} (4—20)

2. IF $m_{ij,0} < m_{ij,end}$ tracks overlap, proceed with parameter calculation

   ELSE tracks don’t overlap, disregard parameter calculation for this pair

3. Determine the portion of track $i$ that the overlapping region represents:

   $$OL_{ij} = \frac{M_{ij}}{M_i} \times 100\%$$

   \hspace{1cm} (4—21)

   where $M_{ij}$ is the number of frames in the overlapping region and $M_i$ is the number of frames in track $i$.

4. Determine the centre-point of the overlapping region:

   $$m_{ij,mid} = \frac{(m_{ij,0} + m_{ij,end})}{2}$$

   \hspace{1cm} (4—22)

5. Scale the overlapping portion of track $i$ to a common average frequency. 2 kHz has been selected as the common average frequency, since for most tracks, this represents a ‘scaling up’. This has the effect of accentuating
differences in shape which is favourable for shape grouping. The scale factor for either frequency contour is given by:

\[ sf = \frac{1}{M_{ij}} \sum_{m_{ij}} \frac{2000}{f_i(m_{ij})} \]  

(4 – 23)

The frequency contour then becomes:

\[ f_{is}(\bullet) = sf \times f_i(\bullet) \]  

(4 – 24)

6. Determine the parameters \((a, b)\) of the line of best fit \(f = am + b\) through the entire overlapping portion of track \(i\) using a least-squares criterion:

\[
a = \frac{M_{ij} \sum_{m_{ij}} m_{ij} f_{is}(m_{ij}) - \sum_{m_{ij}} m_{ij} \sum_{m_{ij}} f_{is}(m_{ij})}{M_{ij} \sum_{m_{ij}} (m_{ij})^2 - \sum_{m_{ij}} m_{ij} \sum_{m_{ij}} m_{ij}}
\]  

(4 – 25)

\[
b = \frac{\sum_{m_{ij}} f_{is}(m_{ij}) - a \sum_{m_{ij}} m_{ij}}{M_{ij}}
\]  

(4 – 26)

7. The gradient for the track portion is taken as the angle between the least squares line and the horizontal (time) axis:

\[
G_{ij} = \left( \frac{180 \tan^{-1}(a)}{\pi} \right) ^\circ
\]  

(4 – 27)

8. The MSE for the track portion is calculated as follows:

\[
\text{MSE}_{ij} = \frac{1}{M_{ij}} \sum_{m_{ij}} \left( f_{is}(m_{ij}) - (am_{ij} + b) \right)^2
\]  

(4 – 28)

9. The “left-hand” track portion is defined over the time frames \(m_{ij,0} \leq m_{ij} \leq m_{ij,\text{mid}}\) and the “right-hand” track portion is defined over, \(m_{ij,\text{mid}} \leq m_{ij} \leq m_{ij,\text{end}}\). The MSE and gradient for these two halves are determined as for the entire overlapping portion in steps 4-6.

The final parameter vector \(p_{i,j}\) is given by:

\[
p_{i,j} = \{ \text{MSE}_{ij}, G_{ij}, \text{MSE}_{ij,\text{LEFT}}, G_{ij,\text{LEFT}}, \text{MSE}_{ij,\text{RIGHT}}, G_{ij,\text{RIGHT}}, \text{OL}_{ij} \}
\]  

(4 – 29)
Parameter-based Grouping Results

The results of using the shape parameter-based grouping technique along the frequency contour dimension are shown in Figure 4-33 which also shows the results of MSE-based frequency contour grouping for shape normalised tracks for comparison purposes. The best performance using shape parameters was achieved when the minimum override threshold was set to 30. The plots of the mis-grouped tracks measure indicate that the optimum performance of both algorithms is fairly similar with the best value for the shape parameter being 41.08 compared to 40.26 for the MSE-based approach. Thus, according to the mis-grouped tracks metric, the shape parameter algorithm performs approximately 2% worse than the MSE-based approach, which is not a significant difference. The histogram of relative performance, however, gives a clearer indication that, with one exception at a threshold value of 65, the MSE based approach consistently performs as well as or better than the shape parameter approach. Hence, the shape parameter technique did not provide any improvement for frequency contour grouping. One reason for this may be that a lot of the noise that would interfere with MSE-based grouping was removed by the trimming procedure described in section 4.6.4, under which conditions the MSE-based approach is sufficient to achieve the optimal grouping.

Figure 4-33: Results for Frequency Contour Shape Parameter-Based Grouping

Figure 4-34 shows the shape parameter-based amplitude contour grouping results as well as those for the MSE-based technique with the amplitude contours normalised to a common mean. It is clear from the graph that the shape-parameter-based amplitude contour grouping methods provide a noticeable improvement over the MSE-based algorithms. The mis-grouped tracks plots indicate an improvement of approximately 4% while the relative performance histogram indicates an improvement of approximately 50%. Given that the amplitude contours represent a particular challenge with respect to
noise, and that the shape-parameter-based algorithm was intended to be more robust to noise, this improvement was as hoped. However, the mis-grouped tracks value is still relatively high (60.4) suggesting that there was simply too much noise in the amplitude data for even the shape parameter based approach to overcome.

![Figure 4-34: Amplitude Contour Parameter-based Grouping Results](image)

### 4.6.7. Fourier Shape Descriptors

Given that there appeared to be little difference between the MSE and shape-parameter-based distance techniques, it was decided to try one other contour-based grouping method to see if any further improvement was possible. Once again, since the problem was essentially one of scale invariant shape comparison, an image processing approach was considered appropriate. This time the technique selected was the Fourier shape descriptor [140]. While the technique would normally be applied to images and thus be described relative to pixels, the following explanation is given relative to track peaks. Further, while the following description is relative to the frequency contour only, exactly the same process applies to the amplitude contour.

The concept of the Fourier shape descriptor is relatively straightforward. Essentially it involves deriving a Fourier domain description for the track shape under consideration. This is then used in place of the actual track shape when performing a comparison (via a simple Euclidean distance) between two tracks. The first step in determining the Fourier shape descriptor for a track is to transform the peak values (time and frequency) into polar coordinates assuming that the origin of the coordinate space is the centre of gravity
of the track. Assuming that the original peak values are represented by \( (t_i, f_i) \) and the polar coordinates are represented by \( (r_i, \phi_i) \) the transformation proceeds as follows:

\[
(t_c, f_c) = \frac{1}{N} \sum_i (t_i, f_i)
\]

\[
r_i = \sqrt{(t_i - t_c)^2 + (f_i - f_c)^2}
\]

\[
\phi_i = \tan^{-1} \left( \frac{f_i - f_c}{t_i - t_c} \right)
\]

It is then a simple matter of taking the complex valued FFT of the polar track function. The Fourier shape descriptor usually consists of the first three coefficients of the transformed track. However, because of the properties of the track shapes, it was determined empirically that it would be best to use the 2nd, 3rd and 4th coefficients.

**Results of Fourier shape descriptor-based grouping**

Figure 4-35 shows the results of grouping the tracks by frequency contour using the Fourier shape descriptor method along with those for MSE-based frequency contour grouping.

Once again, it will be noted that the optimum performance of both algorithms is fairly similar, with the MSE approach achieving a slightly better result than the Fourier shape descriptor algorithm (40.2 versus 41.3). The relative performance histogram gives the clearer indication that the MSE-based approach performed the best. The main reason for
this result would be that the global and local variations between frequency contours are of similar magnitudes.

Figure 4-36 shows the results of Fourier shape descriptor-based amplitude contour grouping compared to the results obtained for shape-parameter-based amplitude contour grouping. The Fourier shape descriptor-based amplitude contour grouping achieved its best mis-grouped tracks performance of 62.9 when the minimum threshold was set to 15; slightly worse than the best shape-parameter-based result of 60.4. The relative performance histogram also indicates that the shape-parameter approach is slightly better than the Fourier shape descriptor method. In the case of the amplitude contours, the noise dramatically effects the overall shape of the contours, accounting for the poorer performance of the Fourier shape descriptor-based technique relative to the shape parameter-based approach, which is less sensitive to these differences.

![Figure 4-36: Results of Fourier shape descriptor-based amplitude contour grouping compared to the shape parameter-based method](image)

### 4.7. Summary

#### 4.7.1. Summary of Results

**Frequency Contour Grouping**

Figure 4-37 summarises the results for the frequency contour grouping experiments. The plots of mis-grouped tracks values indicate that all three algorithms achieve a very similar optimum (minimum) performance of approximately 40. However, the MSE-based algorithm maintains a fairly steady performance over a much wider range of threshold values than the other two algorithms. This would suggest that the MSE-based
algorithm is less sensitive to the minimum threshold value and is therefore most likely to be the most robust of the algorithms. The relative performance histogram also gives the same indication, however, it is also evident that the Fourier shape descriptor-based approach does perform slightly better that the MSE-based approach for threshold values between 15 and 25 inclusive. As has already been implied in the discussion of the results of individual algorithms, it is believed that the MSE-based approach performs best for frequency contour grouping because the pre-processing applied to the tracks removes most of the noise that would otherwise degrade the performance of the MSE-base algorithm. In addition to this, the MSE-based approach appears to represent the best compromise between global and local analysis with respect to the general frequency contour characteristics.

![Figure 4-37: Summary of results for frequency contour grouping](image)

**Amplitude Contour Grouping**

Figure 4-38 summarises the results for amplitude contour grouping. The mis-grouped tracks plots indicate that the best algorithm is the shape-parameter-based grouping followed by Fourier shape descriptors and finally by MSE. The relative performance histogram indicates that there is little difference between the former two algorithms while they both perform significantly better than the MSE-based approach. The reason for these results is that, as noted previously, the amplitude contours display a significant level of localised noise, which the MSE-based approach is sensitive to, while the other two approaches are somewhat immune to localised noise. In light of this consideration, it is not surprising that the shape-parameter based grouping performs best overall, albeit slightly, since the shape parameters provide the coarsest description of track shape.
Comparison of Frequency and Amplitude Contour Grouping

The above results indicate that the optimum algorithm for frequency contour grouping was the MSE-based approach with a threshold value of 40 and the best amplitude contour grouping method was the shape parameter approach with a threshold value of 25. Figure 4-37 and Figure 4-38 also reveal that grouping by frequency contour is generally more effective than by amplitude contour as expected given the relative levels of noise. Figure 4-39 compares the optimum frequency and amplitude contour grouping methods. Full grouping results for these methods appear in Appendix B.

Figure 4-38: Summary of results for amplitude contour grouping

Figure 4-39: Comparison of results of the optimum frequency and amplitude contour-based grouping algorithms
Final Discussion of Contour-based Grouping Results
Various techniques have been presented for frequency and amplitude contour grouping. Contrary to the assertion of Cooke [118], it was found that, in every case tested, the frequency contour based approaches produced superior grouping performance with respect to corresponding algorithms applied to the amplitude contour. The principal reason for this has been identified as the fact that the amplitude contours in general display a much greater level of noise than do the frequency contours. While there was little variation between the three techniques tested, it was found that the best method for frequency contour grouping was the MSE-based approach where the tracks were normalised with respect to their shape. The best performing algorithm for grouping along the amplitude contour was found to be the shape-parameter-based approach where the tracks were normalised with respect to their mean frequency.

4.7.2. Concluding Remarks
Being the first relating to track group formation, this chapter began with a critical review of existing work in CASA to provide the background theory as well as a motivation for the current work. The main conclusion to be drawn from this critical review is that there is an apparent bias against sinusoidal representations on the part of researchers in the CASA field. Nevertheless, some work in musical source separation has suggested the viability of this approach. Further, the previous chapter noted that the architecture adopted for the current work overcomes the major objection to applying sinusoidal representations to the CASA problem.

In keeping with the somewhat unique approach to the CASA problem, a new method of evaluation was also developed. While bearing some similarity to the evaluation technique employed by Cooke [118], it has several advantages over it. The principal advantage is that it does not rely on artificially mixed source files. The evaluation method developed is described in section 4.2 and is used throughout this and the succeeding two chapters to evaluate the various grouping algorithms developed.

Using a sinusoidal representation has facilitated the relatively straightforward exploitation of three auditory grouping cues to achieve source separation. The two explored in this chapter, frequency and amplitude modulation, are the least frequently cited in the literature. Nevertheless, it has been shown that the frequency contour especially is an effective grouping cue. The approach adopted by the current work has also suggested new algorithms for harmonicity-based grouping which is the theme of the next chapter.
Harmonicity-Based Grouping

5.1. Introduction

By far the most commonly employed principle to achieve partial grouping in CASA systems is that of harmonicity. The earliest attempts at CASA-style auditory source separation employed pitch contours, or harmonicity, as the sole cue for group formation [134]. Harmonicity based grouping is often termed “pitch-based” grouping because, in general, the methods involve estimating the pitch/es of the sound/s present and then matching each of the tracks to one or more of the pitches found.

Grossberg [119] states, “perhaps the most important cue for perceptual streaming is the pitch of a sound.” As such, it is not surprising that his system, the SPINET, employs a pitch-based grouping mechanism. Roughly speaking, having found the peaks in the output from a gammatone filterbank, the SPINET system estimates the pitches that are present and then employs a harmonic sieve to form each of the groups. The chief disadvantage of this model is that it cannot be extended to include impulsive sounds.

Klapuri and colleagues propose yet another series of pitch-based grouping models. Their primary interest is in separating musical signals where pitch does indeed play an important role. The initial model was based on a M&Q style sinusoidal representation [141] and it gradually evolved to a more complicated model of the overtone series to account for mistuned harmonics [142]. Because the system assumes that the sources are purely harmonic, it has limited application for general audio. Nevertheless, it is a good starting point since even general audio is dominated by harmonic relationships.

Harmonicity was also the main grouping mechanism employed by Cooke [118]. However, in a departure from traditional pitch-based approaches, Cooke’s system did not rely on an explicit estimate of pitch. Rather each of the seed elements was used to produce a harmonic sieve by assuming that each might represent the fundamental, first, second, third or later harmonic of a group. The effect was to generate a histogram of candidate pitches with the peaks in this histogram occurring at bin locations corresponding to the pitch of each group and the tracks falling within these bins representing the members of the group. The number of harmonics that a particular track was assumed to be a candidate for was restricted such that the fundamental frequency

--- 227 ---
estimates fell in the range of 70-400 Hz. The technique is effectively a subset of one of the techniques that was devised for the work reported here.

Another technique that does not rely on pitch estimation was that proposed by Virtanen and Klapuri [141]. Indeed this technique does not calculate the pitch at all, but rather involves calculating a distance metric that measures the likelihood that two candidate tracks are harmonics of a common fundamental track. The error measure exploits the fact that if two frequencies are harmonically related, their ratio will be equivalent to the ratio between two small positive integers [141]:

\[ \frac{f_i}{f_j} = \frac{a}{b} \]  

(5—1)

The range of allowable values for the integers \( a \) and \( b \) are restricted such that the fundamental frequency cannot be below the minimum frequency in the data set. Parenthetically, there is a disadvantage in this choice of minimum fundamental frequency in that it is possible that the perceived fundamental frequency may not actually be present in the data (c.f. section 2.4.3). The distance metric was simply the difference between the frequency ratio and the \( \frac{a}{b} \) ratio that was the closest in value to the frequency ratio. To avoid having to be careful about the order in which the division was performed the logarithm of ratios was taken. The distance metric was then defined as [141]:

\[ d_h(i, j) = \min \left| \log \left( \frac{f_i}{f_j} \right) \right| \]  

(5—2)

This technique is very similar to one that was independently devised for the work reported here with only two differences. The first is that the ratio was always calculated such that \( f_i \geq f_j \) hence there was no need to take logarithms and the ratio in equation 5—2 returned to being a difference. The primary reason for this difference was obviously a reduction in computational complexity. The second difference was that the distance over the overlapping region between the pair of tracks was not taken as a minimum as in equation 5—2, but rather the mean square error between the track frequency ratios and the harmonic number ratios was calculated. The main reason for this last difference is explained in the next section.

5.2. Problem definition

While frequency and amplitude contour similarity measures compare the shape of the tracks independent of their location in frequency, harmonicity is a measure of the
position of tracks relative to one another along the frequency axis. Given this consideration it may seem that it would be sufficient to use a single key frequency, such as the mean, max or median frequency of the track, to represent the location of each track along the frequency axis and then use this key frequency to determine whether a pair of tracks are harmonically related. However, it was found that there are two principle reasons that render this technique unsuitable.

Figure 5-1 illustrates the first of the problems associated with using a single key frequency to calculate the harmonicity metric. For this example the key frequency was chosen to be the mean frequency. Note that because the onset and offset of Track 2 have not been detected by the initial tracking stage, the average frequency of the track is higher than that expected given that the track is an exact match for the first harmonic of Track 1. Regardless of the method used to determine the key frequency, a similar situation would arise for at least some track shapes, if not all. This problem can easily be overcome by applying the same principle as was used to account for it in the shape parameter calculation, that is by only calculating the key frequency over the portion of the tracks that overlap. However, a second problem remains.

![Figure 5-1: Illustration of problems associated with using a key frequency for harmonicity metric calculation](image)

In explanation of the second problem associated with using a single key frequency, consider the examples shown in Figure 5-2. Each pair of tracks would be considered to belong to the same fundamental based on the method (or methods) for calculating the
key frequency indicated, however, it should be obvious that none of the track pairs are actually harmonically related as they have radically different frequency contours.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>median</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>median</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>min</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-2: Problems with using a key frequency as a harmonicity measure

Hence it is obvious that the harmonicity measure must take into account the entire overlapping portion of the two tracks. Two possibilities were investigated. The first involved making a harmonicity decision for each peak along the track then either calculating an average error value over all the peaks or counting the number of peaks (as a percentage of the total) that gave a positive result while the second method involved using an algorithm that considered the entire track as a single unit.

5.3. Overview of Harmonicity Grouping Solutions Evaluated

From the discussion in section 5.1 it is evident that there are two basic approaches available for harmonicity-based grouping. The first involves either explicitly or implicitly estimating the fundamental frequency of each of the sources present in the signal (without reference to the number of sources). Having obtained this estimate a
A harmonic sieve may be constructed for each source. The concept of a harmonic sieve is illustrated in the simplified example of Figure 5-3.

Figure 5-3 shows that a “harmonic sieve” is simply a method for isolating the expected harmonics of the calculated fundamental tracks, allowing for noise in the data. Two “tracks” are generated in parallel to the fundamental track, one slightly above and the other slightly below the fundamental in frequency. These tracks are then multiplied by successive harmonic numbers in search of the harmonics of the fundamental. This method is applicable even when only the fundamental track has been formed and the peaks at higher frequencies remain isolated. While this method has not been applied in track grouping, it is used to refine the tracks with reference to the original peak data after grouping has been completed.

An alternative implementation of the harmonic sieve that is applicable only to fully tracked peak data is to estimate the harmonic number of each track then to divide it down by this value and determine which, if any, fundamental it is the closest match for. This is the method that was devised for and employed by the harmonicity grouping reported here.

Yet another variation to the basic harmonic sieve idea that, to the author’s knowledge, is unique to the work reported here, is the notion of estimating the fundamental track as the basic unit, rather than estimating the fundamental frequencies on a frame-by-frame basis. The motivation for this technique was the ability to exploit correlation along the track to improve the noise performance of the fundamental estimation algorithm.
The second approach to harmonicity-based grouping that was described in section 5.1 is the notion of a harmonicity error based on the ratio between track frequencies. One proposed by Virtanen and Klapuri [141] was described and a variation of this that was devised for the work in this thesis was also summarised. Another variation of this same idea is to estimate the values of harmonic numbers in the same manner but to then use the mean squared between the tracks scaled to the estimated fundamental frequency as the actual distance metric.

Having expanded the first approach into two, three basic approaches to harmonicity grouping resulted. A number of variations to these were also investigated:

1. **Frame-by-Frame Harmonic Number Estimation**
   a. Analytic fundamental frequency estimation using a variant of the well known Euclid’s Algorithm to determine the highest common factor of the two track frequencies.
   b. Perceptually motivated pitch detection approach based on a theory expounded by Bregman.

2. **Track-Based Harmonic Number Estimation** – Generate a histogram of potential fundamental tracks based on a modified version of the Bregman-inspired algorithm from the previous approach using the entire track as though it were a single peak.

3. **Frequency Ratio Approach** – Determine how harmonically related two tracks are on a peak-by-peak basis by calculating the error between the ratio of the two peak frequencies and the closest ratio of allowed harmonic numbers. The method for determining the harmonic numbers whose ratio is closest to that of the two peak frequencies may be either:
   a. Continuing fractions; or
   b. Table lookup.

### 5.4. Evaluating Harmonicity Metrics

In evaluating the above algorithms, two evaluation strategies were employed. Given that the first two categories of algorithm produced a harmonic number estimate for each track, this suggested an obvious method for the evaluation of these. For the same set of test data used to evaluate the contour-based grouping, the correct harmonic number was manually assigned to each track. These reference files were then used to compare against the results of each of the algorithms where a harmonic number estimate could be obtained. The accuracy of the algorithm was measured as the average percentage of
tracks per file for which the harmonic number estimate was correct. The other method of evaluating the performance of the harmonicity-based grouping algorithms was the same as that used to evaluate the performance of the frequency and amplitude contour-based algorithms as described in section 4.2.

5.5. Frame-by-Frame Harmonic Number Estimation

Two basic algorithms were considered for frame-based harmonic number estimation. The first began with an analytic approach while the second began with a perceptual model of pitch perception. Both algorithms, however, employed a specialised histogram to achieve the actual fundamental frequency estimation. The essential difference in the two methods is in how this histogram is populated with data. This section describes the unique properties of this histogram and how it is generated from an arbitrary data set. It then describes the two different techniques that were investigated for populating the histogram. Finally, the manner in which the histogram was used to obtain an estimate of the harmonic number of each track is detailed.

5.5.1. Proportionally Spaced Histogram

The histogram used consisted of bins whose width was proportional to the centre value of the bin. The bin centres themselves were determined by the actual data values recorded in the bin and the bin width was a proportion of the centre frequency that was empirically determined such that the performance of the harmonic number estimation algorithms based on this histogram was optimised (cf. Figure 5-10). An illustration of the important parameters of a histogram bin is provided in Figure 5-4.

![Illustration of important histogram bin characteristics](image-url)

The method employed to generate such a histogram was as follows:

1. Place each data value in its own zero-width bin.
2. Sort the bins in ascending order of bin centre.
3. Set \( n \) to 0
4. \textbf{WHILE} \( n < N \), where \( N \) is the number of bins
   a. IF \( \text{bin}[n] < \text{bin}[n+1] \cdot (1 - Th_n) \) where \( \text{bin}[n] \) is the centre of bin \( n \)
      \[
      \text{bin}[n] = \frac{\text{bin}[n] \times \text{count}[n] + \text{bin}[n+1]}{\text{count}[n] + 1} 
      \]
      increment \( \text{count}[n] \)
   ELSE increment \( n \)
\textbf{END WHILE}

The reason for employing such a histogram is that, as stated in section 2.2.3, the resolution of the human ear is not fixed. Thus, it would be impossible to find a single frequency offset that would appropriately cluster frequencies for fundamental frequency determination. Further, given that there is no restriction on the fundamental frequency of real data to be an integer, or any other fixed value, it was not appropriate to fix the centre frequencies of the bins. Yet another advantage of this approach is that it lends itself to storage efficient implementation since space is only allocated for bins that actually contain data whereas, with a conventional histogram, the sparse data sets arising from both of the algorithms employed would leave many bins empty.

5.5.2. \textit{Harmonic Ratio Error}
Having defined the histogram that was used, it is now necessary to detail the methods that were investigated for populating this histogram. The first of these algorithms was analytical in nature. As such, it relied on the formal definition of harmonicity. Two frequencies, \( f_1 \) and \( f_2 \), are harmonically related if, and only if, they are integer multiples of a common fundamental frequency, \( f_0 \):

\[
\begin{align*}
  f_1 &= mf_0 & m \in \{1, 2, 3, \ldots\} \\
  f_2 &= nf_0 & n \in \{1, 2, 3, \ldots\}
\end{align*}
\]

(5 — 3)

Assuming that the fundamental frequency is an integer value, determining the fundamental frequency is simply a matter of finding the highest common factor of the frequencies \( f_1 \) and \( f_2 \) which, given our assumption, will also be integer values. If the highest common factor is greater than one, we can assume that the two frequencies are harmonically related and that this factor is their fundamental frequency. A well known algorithm for determining the highest common factor of two integer values is Euclid’s algorithm. Given two numbers, A and B, such that A is greater than B, the highest common factor is found using the following procedure:
1. let \( R := A \mod B \)

2. WHILE \( R \) is greater than 0
   a. let \( A := B \) and \( B := R \)
   b. let \( R := A \mod B \)
END WHILE

3. \( B \) is the highest common factor

The situation with real data is unfortunately more complicated. Firstly, since most of the data represent real, naturally occurring sounds, there is no restriction on the fundamental frequency (or any other frequency, for that matter) being an integer value. Secondly, because the analysis procedure has finite accuracy, limited by the resolution of the TFD, there is an appreciable level of noise in the data. These two factors mean that the probability of finding an exact harmonic relationship between two frequencies is very low. Nevertheless, it was decided to attempt to develop an approximation of Euclid’s method to perform harmonic grouping.

The algorithm thus developed employed the following variations to Euclid’s algorithm as defined above. Firstly, a real number version of the modulus operation was defined as follows:

\[
A \mod B = A - B \left\lfloor \frac{A}{B} \right\rfloor \quad \text{if} \quad A > B \quad \forall \{A, B \in \mathbb{R}\}
\]

It should be obvious that equation 5 — 4 is essentially the modulus operation as defined for integer values with the exception that the operands are allowed to be elements from the set of real numbers rather than being restricted to the set of integers. Consequently, the result may also be a real value. The second variation from the classical approach is that the stopping condition in step 2 of the above algorithm was changed to be the minimum allowable fundamental frequency value, which was 50Hz, instead of zero. Finally, each of the \( B \) values obtained during progress of the algorithm were used to populate the histogram. This modified Euclid’s algorithm was applied to every pair-wise combination of peaks in the frame to populate the histogram.

5.5.3. Bregman-based Fundamental Calculation

The perceptually motivated approach to populating the histogram is based on the observation of Bregman [1] that the differences between frequencies play an important role in pitch perception. In the simplest case, where we have a single harmonic series of frequencies, \( f_1 = f_0, f_2 = 2f_0, f_3 = 3f_0, f_4 = 4f_0, \ldots \), it should be obvious that the
difference between each pair of adjacent frequencies is equal to the fundamental frequency, \( f_0 \), itself. That is, if there are \( N \) harmonics in the series:

\[
(f_2 - f_1) = (f_3 - f_2) = (f_4 - f_3) = \ldots = (f_N - f_{N-1}) = f_0
\]  \( (5 - 5) \)

Further, the difference between any two non-adjacent frequencies will be some integer multiple of the fundamental. For example,

\[
f_3 - f_1 = 3f_0 - f_0 = 2f_0.
\]  \( (5 - 6) \)

If the data were perfectly noise free, and we only ever dealt with a single source, determining the fundamental frequency would thus be a very simple matter of determining the difference between any two adjacent frequencies in the frame. In practice, however, to deal with both noisy data and multiple source separation, we must use a histogram to record the difference between all pair-wise combinations of peak frequencies available. Given the proportional histogram described earlier, this will obviously lead to a peak at each of the fundamental frequencies present. From equation 5—6, it is apparent that further weight can be given to the peaks by also recording fractions of the difference between each pair-wise combination of peak frequencies. Both of these variants were investigated.

Having obtained the histogram using one of the two methods described above, the most likely harmonic number for each peak frequency, \( f_n \) in the current frame is determined thus:

1. Let \( f_{0i} \) be the \( i^{th} \) candidate fundamental for the time frame. Find \( e_{\text{min}} \) such that:

\[
e_{\text{min}} = \text{MIN} \left\{ \text{ROUND} \left[ \frac{f_n}{f_{0i}} \right] - \frac{f_n}{f_{0i}} \right\}
\]  \( (5 - 7) \)

where \( N \) is the number of frequencies and \( M \) is the number of candidate fundamentals in the current time frame.

2. The harmonic number is then:

\[
h_n = \text{ROUND} \left[ \frac{f_n}{f_{0i_{\text{min}}}} \right] \quad 0 \leq n < N
\]  \( (5 - 8) \)

where \( i_{\text{min}} \) is the index corresponding to \( e_{\text{min}} \) in 5—7.
For every track we now have a harmonic number estimation for each of its peaks. The harmonic number estimate for each track is then obtained by simply selecting the harmonic number that was estimated for the majority of peaks along the track.

5.5.4. Comparison of Techniques
The accuracy of these two harmonic number estimation procedures is shown in Figure 5-5. The chance level, or the number of tracks that would have the correct harmonic number assigned if this was done entirely at random, can be determined thus by assuming that the maximum harmonic number allowed by the algorithm is 25. Thus, if the harmonic number is selected at random, each track has a 1 in 25 chance of having the correct harmonic number assigned which represents a probability of 0.04. Hence, the chance level for harmonic number estimation is 4%, that is, 4% of tracks would be expected to have their harmonic number selected correctly if this was done at random.

Figure 5-5 compares the harmonic number estimation accuracy of the two basic approaches to harmonic number estimation described in this section. There are three plots because the two variants of the Bregman-based approach are included. It is clear that the simplest Bregman based approach (BREG) is by far the best performer, with an average of just under 85% of tracks in a file having their harmonic number correctly estimated. This is followed by the Bregman based approach where the fractional differences between frequencies are also recorded in the histogram (BREG_IN). The performance of this version of the algorithm is clearly less consistent than either of the other two algorithms. This might seem surprising given that equation 5 – 6 suggests that including fractional differences in the histogram should increase the weighting of the correct fundamental bin. However, including fractional differences also increases the weighting for (incorrect) fractions of the true fundamental. Because of the harmonic structure of natural sounds, this has a particularly adverse effect on harmonic number estimation for signals resulting in many tracks even from only a single source.

It is also apparent from Figure 5-5 that the Euclid based algorithm performs consistently over a relatively wide range of threshold values. However, this performance is significantly less (approximately 30%) than that achieved by the Bregman based approach. The plateau that are apparent at the tail of all graphs (beyond a threshold of 0.2 for BREG; 0.35 for EUCLID and 0.48 for BREG_IN) indicate the point at which each of the algorithms degenerates into estimating the harmonic number of all tracks to be 1. Because this will be the correct estimate for at least one track in all of the test files, the average number of correct harmonic number estimates has been calculated to be 30.3% under this condition which agrees with the approximate plateau values.
Figure 5-5: Accuracy of harmonic number estimation for each algorithm

"EUCLID" is the real-valued variation on Euclid’s algorithm. "BREG" is the Bregman inspired perceptual approach using only the frequency differences to populate the histogram and "BREG INC" is the Bregman inspired perceptual approach using both the frequency differences and fractions of them to populate the histogram.

Figure 5-6 shows the grouping results as a function of the minimum override grouping threshold (c.f. section 4.5.2) for the algorithms discussed in this section. The misgrouped tracks measure displays an overall minimum value of 35.99 at a threshold of 40 for the simplest Bregman based algorithm (BREG). The other Bregman based algorithm at a threshold of 60 follows this with a value of 41.90 and finally the Euclid based algorithm also at a threshold value of 60 achieved a minimum value of 50.53. These results agree with those obtained for harmonic number estimation indicating that harmonicity grouping performance is directly related to harmonic number estimation accuracy for the basic algorithm described in this section.

Figure 5-6: Grouping results for Bregman and Euclid based harmonicity grouping algorithms
The optimum grouping performance achieved by the Bregman based algorithm of 35.99 is significantly better than that achieved for frequency contour grouping which was 41.08. This represents an improvement for harmonicity grouping of approximately 12% over frequency contour grouping. This would suggest a strong reason why most researchers concentrate on harmonicity based grouping techniques. The significant improvement might also have excused further investigation of harmonicity based grouping algorithms. Nevertheless, since the Bregman based algorithm suggested an obvious alternative approach, it was decided to determine whether the harmonicity grouping performance could be improved by adopting this approach.

5.6. Track-Based Harmonic Number Estimation

It was recognised that Bregman’s approach can be readily extended to consider an entire track as though it were a single peak. The histogram was now defined such that each bin centre represented a frequency contour rather than a single frequency and was populated with track differences defined as:

\[ T_{\text{diff}}(t_k, f_k) = (t_i, f_{1,i} - f_{2,i}) \quad \forall \ i, j, k \ \ni \ t_i = t_j = t_k \]  

(5-9)

As with the frame-by-frame approach, once the histogram has been filled with differences for each pair-wise combination of tracks, the local maxima of the histogram are found and the corresponding bin centres assumed to be the fundamental tracks present. Each track is then compared to each potential fundamental and its harmonic number determined relative to the fundamental. Next, the track is scaled by this harmonic number and the MSE between the scaled track and candidate fundamental is calculated. Having performed this calculation for all candidate fundamentals, the harmonic number for which the MSE is a minimum is deemed to be the estimated harmonic number of the track.

The harmonic number estimation accuracy achieved by using this track-wise approach to the Bregman-based algorithm is shown in Figure 5-7. The grouping results with respect to the threshold for the harmonic number estimator with a fixed main override threshold value of 42.8 are shown on the same figure. The fixed value was determined by first selecting the threshold that resulted in the best performance of the harmonic number estimator (8). Once an initial graph of results for grouping versus the main grouping threshold was obtained (similar to Figure 5-8), the threshold value giving the best performance (42.8, found by fitting a least squares parabola to the data) was used to generate the data plotted in Figure 5-7. The mis-grouped tracks plot of grouping performance displays two strong local minima: one at a threshold value of 5 and the
other at a threshold of between 45 and 50. These correspond with two local maxima in the plot of harmonic number estimation accuracy. However, the relative magnitudes of the extrema are the opposites of each other. This indicates that the grouping performance for the track-based Bregman inspired algorithm was not strictly correlated with harmonic number estimation accuracy. The most likely explanation for this finding is that the profile of tracks whose harmonic number was estimated incorrectly for the higher threshold was more conducive to an accurate grouping than was the case for the lower threshold value. Such a situation might arise, for example, if all the tracks that should belong to a group had their harmonic number incorrectly estimated by a constant factor, such as twice the expected value. That is, the first harmonic was assigned a harmonic number of 2, the second 4 and so forth.

![Figure 5-7: Results for track-based harmonic number determination and grouping with respect to the harmonic number estimator threshold](image)

The harmonicity grouping with respect to the main threshold was then investigated by fixing the harmonic number estimator threshold to the optimum value of 45 and sweeping the main threshold through its range. Figure 5-8 shows the results of this experiment as well as a parabolic curve fitted to the mis-grouped tracks result to illustrate how the optimum value for the main threshold was calculated. Both the mis-grouped tracks plot and the relative performance histogram display optimum values (minimum and maximum respectively) at a threshold value of approximately 40. The trend line was fitted using the method of least squares, which resulted in the equation

$$y = 0.00029637 x^2 -0.02372 x +0.79398 .$$

Solving the differential equation \( \frac{dy}{dx} = 0 \) for the location of the minima yields a threshold value for optimum performance of 40.02.
The minimum mis-grouped tracks value of 29.2 in Figure 5-8 also reveals that the track-based version of the Bregman inspired algorithm (BREG_TRACK) performs better than the peak-based version (BREG) which gave a minimum value of 35.99. The performance of the two algorithms with respect to the threshold value is compared in Figure 5-9.

Figure 5-9 shows that while the mis-grouped tracks measure indicates that the track-based version of the algorithm has a better overall performance, the peak-based version performs best at low threshold values. Also, the relative performance histogram indicates that there is very little difference between the two algorithms over all.
Nevertheless, for the purposes of further experimentation, the track-based version of the algorithm was considered to be the best performing of the two.

5.7. Frequency Ratio Approach
The frequency ratio approach is another algorithm that performs an initial analysis on a peak-by-peak basis and then combines the individual values to determine a single error measure that may be used to group the tracks. It is also an analytic approach arising from the formal definition of harmonicity. However, unlike the previous approaches discussed, the frequency ratio approach does not explicitly estimate the fundamental frequencies present nor does it estimate the harmonic number of each track distinct from the grouping procedure. With this approach, it is only possible to obtain an accurate estimate of the fundamental frequency of each track after the grouping procedure has been completed. Thus, while the accuracy of harmonic estimation based on this technique is reported in this section as a basis for comparison with the previously mentioned techniques, it is less of an indicator of the algorithm’s performance than for the other techniques.

5.7.1. Basis of Approach
The frequency ratio approach is based on the observation that, according to the definition given in 5 — 3, the ratio of the two frequencies $f_1$ and $f_2$ will be a simple fraction of two integers:

$$\frac{f_1}{f_2} = \frac{mf_0}{nf_0}$$
$$\Rightarrow \frac{f_1}{f_2} = \frac{m}{n} \quad (5—10)$$

Thus the problem now is to determine how close the ratio $\frac{f_1}{f_2}$ is to being a ratio of two integers. Two methods are available to achieve this task. The first uses a look-up table with the row indexes representing the numerator of the fraction and the column indexes representing the denominator of the fraction. It is then a simple matter of searching through the table to find the value closest to the fraction of the two frequencies. The second method relies on the well-known mathematical technique of continuing fractions.

5.7.2. Continuing Fractions
The method of continuing fractions is used to estimate a real number with a ratio of integers. The method proceeds as follows. Given a real number, $x$, with integer and decimal portions:
\[ x = A + B \quad \Rightarrow A \in \{\ldots -3, -2, -1, 0, 1, 2, 3, \ldots\} \]
\[ = A + \frac{1}{B} \quad \text{and} \quad 0 < B < 1 \]
\[ = A + \frac{1}{C + D} \]
\[ = A + \frac{1}{C + \frac{1}{D}} \]
\[ = A + \frac{1}{C + \frac{1}{E + F}} \]
\[ = A + \frac{1}{C + \frac{1}{E + \frac{1}{F}}} \]
\[ = \text{etc…} \]

The stopping condition on the above continuing fraction occurs when the value of the fraction falls within the desired error bounds of the real number being converted. In the experiments reported here, the harmonicity threshold set this condition, being the number of decimal places to which the continuing fraction and the frequency ratio agreed.

5.7.3. Harmonic Number Estimation Based on Frequency Ratio Error

As a simple test of these two methods, and to serve as a comparison with the previous methods, the following algorithm was devised to use these methods to estimate harmonic numbers for the tracks in the test data files.

1. FOR each track, \( T_1 \), in the file, \([T_1 = F(t_1, f_1)]\)
   a. FOR every track, \( T_2 \), in the file excluding \( T_1 \), \([T_2 = F(t_2, f_2)]\)
      FOR every corresponding peak pair, \((t_{1,i}, f_{1,i})\); \((t_{2,j}, f_{2,j})\):
         \[ \text{Calculate the fraction} \frac{f_{1,i}}{f_{2,j}} \]
- Use either the table lookup or continuing fractions method to determine the ratio $\frac{m_i}{n_j}$ that best approximates this fraction.

END FOR

END FOR

b. Set the harmonic number of $T_1$ to be the most commonly occurring $m$ value over all pairs and time frames.

END FOR

Figure 5-10 shows the results for this algorithm.

The method of continuing fractions requires a threshold, which is the accuracy of the approximation required stated as the number of decimal places to which the fractions $\frac{f_{1,i}}{f_{2,j}}$ and $\frac{m_i}{n_j}$ must agree. The lookup table method does not require a threshold, however, for the purposes of comparison, its value has been plotted against each of the threshold values tested for the continuing fractions algorithm. The figure clearly shows that the optimum performance achievable with the method of continuing fractions is better than that of the lookup table method. The average number of tracks for which the harmonic number is estimated correctly is approximately 75% for the method of continuing fractions when a threshold value of 1 is applied while the lookup table method achieves an accuracy of 71%. In terms of the relative performance, the continuing fractions
method also shows itself to be superior with twice as many files achieving the best performance (of 100% success rate) as for the table lookup method.

The results shown in Figure 5-10 may be explained in terms of the precision of the two methods. The first important factor to note is that the optimum threshold for the continuing fractions technique is 1. This means that the fractional estimate of the frequency ratio should be correct to 1 decimal place for the numerator and denominator of this fraction to accurately estimate the relative harmonic number of the original frequencies. It may be surprising that such a low level of precision achieves optimal results, however, it is easily understood when one realises that the continuing fractions technique will generally select very successively larger values for the numerator and denominator in an attempt to improve the estimate of the fraction. This quickly results in the estimated values for the harmonic numbers being much larger than their correct values. A second factor contributing to the relatively low value for the optimum level of precision is that both noise in the data and the inherent characteristics of natural sounds mean that the ratio of frequencies only roughly approximates the ratio of their harmonic numbers.

These two techniques will give an estimate for the relative harmonic numbers of each pair of peaks (or tracks) regardless of whether they are actually harmonically related or not. In order to decide whether they are indeed harmonically related as well as what degree of confidence may be placed on this decision, a distance metric must be employed.

5.7.4. distance metrics
The distance metric that was employed simply used the error between the ratio of frequencies and the ratio of estimated harmonic numbers. This is similar to that used by Virtanen and Klapuri [141].

\[
e_{i,j} = \left| \frac{f_{1,i}}{f_{2,j}} - \frac{m_i}{n_j} \right| \quad \exists m_i, n_j \in \{1, 2, 3, \ldots\}
\]

(5 — 11)

The initial attempt at deriving a single metric based on these errors was to simply average the values obtained using equation 5 — 11 for each pair of corresponding peaks between the two tracks. This technique, however, was found to give very poor results. The reason for this was determined to be that the average computed in this way did not take any account of the relative track shapes and was, in effect, equivalent to the key frequency approach. The alternative that was developed involved calculating the \( \frac{m_i}{n_j} \) fraction for each pair of corresponding pairs between the two tracks and then selecting
the one that occurred most frequently. The ratio was then the average of those used to calculate the corresponding $e_{i,j}$.

5.7.5. Results
Both the table lookup and continuing fractions version of the frequency ratio based harmonicity grouping were tested over a range of threshold values. The results of these experiments appear in Figure 5-11. The most obvious feature of this graph is that it displays considerably more noise than the equivalent graphs for the other harmonicity-based grouping strategies. Also, the minimum mis-grouped tracks, at 54.49, is considerably higher than the best value obtained for the Bregman-based algorithm of 29.24, and the number of files with best performance has a maximum value of only 5 over the entire range of threshold values.

Figure 5-11: Harmonicity-based grouping results for frequency ratio approach using the method of continuing fractions and table lookup

Figure 5-11 shows that the best grouping performance was achieved by both versions of the algorithm when the minimum threshold was set to 15. The table lookup method gave very slightly better results than the continuing fractions approach and gave a slightly more stable performance in the region of threshold values from 15 to 20. Hence, according to these results, given a choice between these two techniques, the table lookup method would be preferred. This might be a surprising result in light of the results shown for the harmonic number estimation. However, it should be noted that the evidence for giving preference to the table lookup method is only slight. Further, the
grouping performance is highly sensitive to the exact nature of the harmonic number estimates. If the errors are such that the harmonic number is consistently selected to be a multiple or fraction of the actual harmonic number, then the grouping performance measure may still give a relatively low value since two correct half groups (as would be the result of such an error) is considered to be preferable than one single group incorporating several tracks incorrectly.

To confirm that the harmonicity threshold for the continuing fractions method was indeed set to the optimum value, an experiment was performed to determine grouping accuracy as a function of the harmonicity threshold. For this purpose, the main minimum grouping threshold was set to the optimum value of 15 indicated by Figure 5-11. The results of this experiment are shown in Figure 5-12, which does indeed confirm that the best threshold value was 1.

![Figure 5-12: Grouping performance versus harmonicity threshold for the continuing fractions technique](image)

### 5.8. Summary

#### 5.8.1. Summary of Results

Figure 5-13 summarises the grouping performance for each of the harmonicity-based grouping methods tested. The minimum mis-grouped tracks value was obviously achieved by the track-wise version Bregman-based algorithm at a threshold value of 40, indicating that this combination of algorithm and threshold value provides optimum performance. The two fractional based algorithms achieved the worst performance. Plots of the actual grouping results across the test data set for both the BREG and BREG_TRACK approaches are provided in Appendix C.
5.8.2. Chapter Summary and Concluding Remarks

This chapter has explored the issue of harmonicity-based grouping. It began with a review of previous work in the area, noting that harmonicity is the most common grouping cue employed by CASA systems. It then presented a formal definition of the problem that emphasised the importance of evaluating harmonicity over an entire track rather than at some key frequency. Then six different algorithms for harmonicity-based grouping were proposed and evaluated. Three of these were derived from a theory of pitch perception postulated by Bregman. The track-based version of these was particularly unique in its truly track-wise calculation of a fundamental track (as opposed to the usual fundamental frequency). The remaining algorithms were all based on the definition of harmonicity. The first used Euclid’s method to determine the highest common factor of pair-wise frequency combinations in an attempt to estimate their common fundamental. The final two determined the likelihood that two frequencies were harmonically related by comparing their ratio with the ratio of their harmonic numbers estimated under the assumption that they were harmonics of a common fundamental. The experimental results found that the best technique for harmonicity grouping was the track-based Bregman inspired approach.
Chapter 6

Combined Grouping Approaches

6.1. Introduction
Given metrics for each of the important dimensions, amplitude contour, frequency contour and harmonicity, two methods of combining these were considered. The first was to group the tracks along each of the dimensions individually and then use simple set theory to determine the intersection of the resultant groupings. The second method used the three error measures to construct a three dimensional vector, the length of which determined the distance between the two tracks. The details of both of these methods and their results are reported in this section.

6.2. Summary of Results for Single Dimension Grouping Experiments
Figure 6-1 summarises the relative performance of each of the distance metrics tested. The values on the figure are those for the mis-grouped tracks metric as defined in section 4.2.2.

![Figure 6-1: Summary of performance of each of the distance metrics tested](image)

The graph shown in Figure 6-1 may be interpreted by bearing in mind that the mis-grouped tracks measure is optimally 0% thus, we require it to be minimised for optimal performance. It should also be noted that to simplify the graph, only one variant of each pair of the contour-based grouping algorithms is represented. For the frequency contour
grouping this is the version where the tracks were shape normalised while for the amplitude contour grouping it is the version that used tracks normalised to a common mean. This is in keeping with the results in 4.6.5. The ovals on the figure indicate the minimum mis-grouped tracks value achieved for each of the dimensions across all of the algorithms represented for that dimension.

Figure 6-1 indicates that the best algorithm for grouping the tracks overall is the track-wise Bregman-based approach to harmonicity-based grouping (Harmonicity BREG_TRACK). This is followed closely by the peak-wise version of the same (Harmonicity BREG). The optimum algorithm along each dimension is:

- **Amplitude contour**: Gradient-based shape parameters with an optimum threshold of 25.
- **Frequency contour**: MSE with optimum threshold of 40.
- **Harmonicity**: Track-wise Bregman-based approach with optimum threshold of 40.

It was decided to investigate the possibility of combining these three grouping strategies to determine whether this would improve the overall grouping performance. Two possible methods were determined. The first was based on the mathematical concept of the intersection of sets while the second employed a distance vector approach which simply involved extending the algorithm detailed in section 4.5.1 to use the Euclidean distance between the distance metrics along all three of the metrics rather than only a single one.

### 6.3. Set Intersection Approach to Combined Grouping

The set intersection-based approach to combined grouping along all three dimensions involved making three copies of the tracks to be grouped. The first copy was grouped by frequency contour, the second by amplitude contour and the final one by harmonicity. In keeping with the conclusions of the previous section, amplitude contour grouping was achieved using the gradient-based shape parameters as the distance metric, the frequency contour-based grouping employed the MSE between tracks normalised according to shape and harmonicity grouping was performed using the track-wise Bregman-based approach as the distance measure. The three groupings were then intersected to form the final set of groups using the following procedure:

1. FOR each shape group
a. WHILE tracks remain in the group
   i. Initialise an empty combined group
   ii. Find the frequency group that contains the first track in the current shape group
   iii. Extract this track from both frequency and shape groups. Add it to the combined group.
   iv. FOR all tracks in the current shape group
       1. IF the track is a member of the current frequency group
          Extract if from both frequency and shape groups and add it to the combined group.
          END IF
       END FOR
   END WHILE
END WHILE

However, it was discovered that this algorithm did not perform satisfactorily as it was prone to over segmenting (or under grouping) the tracks. The reason for this was determined to be that, because the nearest neighbour matching algorithm forces grouping decisions, the probability of minor errors in grouping is relatively high. The intersection procedure is very sensitive to even slight errors in the grouping as it only takes one of the three groupings to be in error to force a valid group to be broken up. Thus another algorithm had to be devised.

Since the problem with the previous approach was that errors in grouping individual tracks often resulted in valid groups being broken up, an alternative that performed a very rough grouping along each of the individual dimensions was investigated. In this method, each of the individual tracks was allowed to belong to several groups. The individual grouping was performed as follows:

1. Calculate the distance between each pair-wise combination of tracks (using the appropriate technique as per previous paragraph)
2. FOR each track
   a. Form a group of all tracks within the threshold of this track
END FOR
Having obtained the three sets of groupings (one for each of amplitude contour, frequency contour and harmonicity), they were intersected according to the following algorithm:

1. **Label the three groupings:**
   
   \[ G_A = \{ a_0, a_1, a_2, \ldots, a_N \}; \quad G_B = \{ b_0, b_1, b_2, \ldots, b_N \}; \quad G_C = \{ c_0, c_1, c_2, \ldots, c_N \} \]

   where the \( a_i, b_i, \) and \( c_i \) represent each individual group within the respective data sets and \( N \) is the number of groups in each data set (need not be the same for each).

   (Note: the labels are arbitrary and need not be applied in any particular order to the groupings).

2. **Remove any groups that contain only one track from each of the groupings**

3. **Generate a table (table A_B) that records all groups in set B that have a non-null intersection with each of the groups in set A.** That is:

<table>
<thead>
<tr>
<th>Set A group</th>
<th>Intersecting groups from set B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \forall i \ni \cap b_i \neq \emptyset )</td>
</tr>
<tr>
<td>1</td>
<td>( \forall i \ni \cap b_i \neq \emptyset )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

4. **Repeat step 3 for set A with C, set B with A, set B with C, set C with A and set C with B.**

5. **Compare the two tables that have A in the first column. Remove any rows from both tables where there is an empty cell in either of them.** For example:
Chapter 6: Combined Grouping Approaches

6. Repeat step 5 for the other two pairs of tables.

Now using tables $A_B$ and $B_C$:

7. FOR each row in table $A_B$, use the B values for that row as indexes into table $B_C$. Form the intersection between each A, B, C group trio thus obtained. Add this to the list of output groups.

8. Remove any duplicate groupings that may have been generated in the list of output groups.

While a more straightforward intersection between each possible three-way combination of groups would have yielded a similar result to the above algorithm, the computational complexity would make it prohibitively slow to execute. Thus, the above algorithm was devised to perform the task of intersection in a more efficient manner.

6.3.1. Results

In order to determine the optimum threshold values to use in the intersection-based grouping, a series of experiments were performed where the threshold value along each of the dimensions (frequency, amplitude and harmonicity) of the space were swept through the range of 0-100 in steps of 5. As with the previous experiments, two measures were used to compare among the various combinations of threshold, the number of files giving optimum performance and the mis-grouped tracks metric. However, since there were three variables, the relative performance histogram requires four dimensions. This histogram is presented as a series of three-dimensional histograms in Appendix D. The maximal region from this histogram is shown in Figure 6-2 below.
Figure 6-2 indicates that, according to the relative performance histogram, the intersection-based approach performs best when $\{Th_F = 60; Th_H = 55; 45 \leq Th_A \leq 100\}$ with the 11 files achieving best performance. Three other regions display similar performance with the relative number of files with best performance equalling 10:

- \(\{60 \leq Th_F \leq 65; 40 \leq Th_H \leq 55; 45 \leq Th_A \leq 100\}\);
- \(\{Th_F = 40; 85 \leq Th_H \leq 100; 45 \leq Th_A \leq 100\}\); and
- \(\{Th_F = 60; Th_H = 70; 45 \leq Th_A \leq 100\}\).

![Figure 6-2: Maximal region from the four-dimensional relative performance histogram showing performance of the intersection-based grouping technique](image)

Figure 6-3 shows the mis-grouped tracks value as a function of the three thresholds for the same optimal region as depicted in Figure 6-2. The minimum value of 41.1, marked on the figure, occurs in a volume of the time frequency-harmonicity-amplitude threshold space described by: $\{Th_F = 65; Th_H = 45; 45 \leq Th_A \leq 100\}$. Four other regions display minima similar (< 45) to this one:

- $\{Th_F = 55; Th_H = 25; Th_A = 40\}$;
- $\{50 \leq Th_F \leq 55; Th_H = 25; 45 \leq Th_A \leq 100\}$;

---

1 $Th_F$: Frequency contour grouping threshold; $Th_H$: Harmonicity grouping threshold; $Th_A$: Amplitude contour grouping threshold
Chapter 6: Combined Grouping Approaches 255

- \( \{70 \leq Th_F \leq 100; Th_H = 45; 45 \leq Th_A \leq 100\} \) and

- \( \{ Th_F = 65; 50 \leq Th_H \leq 55; 45 \leq Th_A \leq 100\} \).

Figure 6-3: Region of best performance from the four-dimensional frequency, amplitude, harmonicity, mis-grouped tracks function

Given the above considerations, the optimal combination of thresholds was deemed to be \( \{ Th_F = 65; Th_H = 45; Th_A = 45 \} \), which is within the region of the absolute minimum of Figure 6-3 and one of the secondary maximal regions in Figure 6-2. Figure 6-4 shows the mis-grouped tracks value as a function of each threshold value when the other two thresholds are held fixed at their optimum value. The figure clearly indicates that the minimum mis-grouped tracks value occurs when \( \{ Th_F = 65; Th_H = 45; 45 \leq Th_A \leq 100\} \).

Figure 6-4: Results for intersection based grouping as a function of the various thresholds
The actual track grouping results for the test data files are shown in Appendix D. Careful inspection of those graphs reveals that there are two main problems with this technique. The first is that the algorithm tends to undergroup tracks and the second is that it is prone to erratic behaviour, grouping tracks that obviously do not belong in the same group. The former problem is the lesser of the two, as the model-based track extraction will work even if some of the tracks are missing from the group. In fact, two (or more) nearly identical are likely to result, which can easily be reduced by combining any groups that are very similar. The latter problem can be much more severe as if the erroneously grouped track/s is/are sufficiently different from the group, the model track generated for the group will be incorrect. This can result in poor separation of the sources in the final tracks. Figure 6-5 shows an example of the former while Figure 6-6 gives an example of the latter case.
Figure 6-5 shows an example where the intersection-based grouping resulted in a group being split in two. The shorter tracks in the centre of the plot should all be part of the same group, however, two have not been included with the rest. Figure 6-6 shows an example of mis-grouping. Obviously the two tonal tracks should have been grouped together while the chirp should have been in a separate group of its own. Both problems stem from a common cause, being that if there is an error in any two of the individual groupings, it will be propagated through to the output by virtue of the intersection operation. Hence, although the modified intersection operation does result in a more robust grouping than a standard set intersection would, it is still prone to apparently random error.

6.4. Distance-Vector-based Grouping
Having observed the results of the intersection-based grouping, it was decided to test a distance-vector based approach. This approach was expected to resolve the problem of erratic grouping behaviour mentioned in the previous section. The distance-vector-based approach is simply the basic algorithm described in section 4.5.1 with the calculate distance step being achieved by adding the distances obtained for frequency contour, amplitude contour and harmonicity. However, because each of these quantities exists over a different range, they must be weighted with respect to one another. Thus, four quantities were required to be optimised: the relative weightings to be applied to the frequency
contour, amplitude contour and harmonicity contour and the overall threshold for the grouping.

In order to determine the optimum values for the threshold and each of the weightings, two series of experiments were run. In the first the threshold was swept through the range 0, 200 in steps of 5, while the weightings applied to each of the distance metrics were kept equal to one another and constant. In the second series of experiments the threshold value remained constant at the value determined to be optimal based on the results of the first series of experiments and the relative weightings applied to each of the distance metrics were each varied from 0 to 100% such that the total weighting remained constant at 100%. That is:

\[ w_f + w_a + w_h = 100 \]  \hspace{2cm} (6 - 1)

where \( w_f, w_a, \) and \( w_h \) are the weightings applied to the frequency contour, amplitude contour and harmonicity distances respectively.

Figure 6-7 shows the results of the first series of experiments. The mis-grouped tracks metric and the relative performance histogram are plotted on the figure.

![Figure 6-7: Results of distance-vector based grouping as a function of the minimum threshold value](image)

The relative weighting of each of the distance metrics remained constant at 33% for each.

It may be observed from Figure 6-7 that there is an obvious trend on the mis-grouped tracks plot between threshold values of 70 and 185. The least squares parabola that was fitted to this data was found to have the equation: \( y = 0.01x^2 - 2.336x + 171.89 \). Solving
\[
\frac{dy}{dx} = 0 \text{ gives the location of the minimum value to be 116.7. This value was rounded up to 117 to be the threshold value selected for the second set of experiments.}
\]

Figure 6-8 shows both two- and three-dimensional representations of the results for the second series of experiments where the relative weightings of each of the distance metrics were varied while the threshold remained constant at 117. Note that since the sum of the three weightings was restricted to equal 100, the amplitude weighting is implicitly defined in the plot and is given by:

\[
w_{\text{amplitude}} = 100 - (w_{\text{frequency}} + w_{\text{harmonicity}}).
\]

---

**Figure 6-8: Results for the distance-vector based combined grouping approach**

The top pane shows the 3D view while the lower pane shows the collapsed 2D view. The legend entries on the bottom pane refer to the weighting applied to the frequency contour distance. Note that the results are implicitly plotted against amplitude weighting as well since \((F_{\text{requency Weight}} + A_{\text{mplitude Weight}} + H_{\text{armonicity Weight}} = 100)\).
Figure 6-8 shows that with the threshold set to 117, the overall best mis-grouped tracks value of 32.29 was achieved when the weightings for the frequency contour, amplitude contour and harmonicity contour were 25%, 35% and 40% respectively. However, a very similar level of performance (mis-grouped tracks = 32.57) was achieved when the weightings were 35%, 35% and 30% respectively. These results indicate that the optimum relative weighting is to have the three weights approximately equal. Further, comparing the best mis-grouped tracks value of 32.29 against that achieved for the intersection-based grouping of 41.1 reveals that the distance vector-based approach achieved a significantly better level of performance as was expected. The actual grouping results for the optimum relative weighting for each of the files in the test data set are shown in Appendix E.

6.5. Summary

This chapter has presented two means for performing grouping along all three dimensions simultaneously. The first involved grouping the tracks along each of the dimensions separately and then performing a modified version of a set intersection to determine the final grouping while the second calculated a combined distance metric based on those obtained for each dimension and then using this in the basic distance-vector grouping algorithm. The technique used to group or calculate the distance metric along each dimension was that which was found to give the best performance from the experiments reported in the previous two chapters.

The experimental results reported revealed that the first of the combined approaches has a tendency to make erratic grouping errors, sometimes failing to complete groups and at other times including tracks that are completely incorrect into a group. This finding motivated the development of the second method, which was shown to have superior grouping performance. The optimum performance for the distance-vector based combined grouping approach was achieved when the relative weightings of the dimensions was approximately equal.
Chapter 7

Application to Information Management: A Preliminary Study

7.1. Introduction
As was implied in the introduction, information management encompasses a number of very different tasks. These include efficient storage and transmission, content-based retrieval and content-based as well as temporal browsing. The first and final of these will be discussed in the conclusion as directions for future work while the second is the focus of this chapter.

7.2. Domain Specific Audio Content-based Retrieval Techniques
The audio domain can be divided into three areas: speech, music and general environmental sounds. Of the existing work in audio retrieval, most systems focus on only one of these three categories with environmental sounds receiving by far the least attention. Those that do deal with heterogeneous audio data generally need the data to first be segmented into sections containing only one type of sound. This section reviews existing systems and techniques with an emphasis on the advantages and disadvantages of each.

7.2.1. Speech
The oldest and most established form of content-based audio retrieval operates in the speech domain and relies on recognition: automatic speech recognition (ASR). However, as was mentioned previously, there are a number of disadvantages of this mode of operation. Firstly, there is as yet no ASR system that can operate in an unconstrained environment. Noise or other interfering background sounds are typically a problem for ASR systems, for example [143]. Secondly, there is much information in a speech signal that cannot be described by a simple transcription. For example, varying the inflection in English can change a statement into a question and in tonal languages, such as Chinese; prosodics carries much of the meaning. Finally, as with most of the systems to be described here, relying on a transcription necessitates the use of separate index files.

Wilcox et al propose an audio indexing scheme that relies on speaker identification [144]. The algorithm segments a recorded conversation into individual speakers. This information can then presumably be presented graphically or alternatively the user could
request all segments containing a particular voice to simplify browsing of the recording. The main disadvantage of this technique is that the speakers need to be known \textit{a priori} in order to be able to perform the recognition and an initial training phase is required. However, this work does highlight the fact that even a simple segmentation, based on speaker transitions, can be useful for browsing audio.

The problems of speech recognition and speaker identification techniques can be overcome by indexing speech using acoustical cues as proposed by Schmandt \textit{et al} in \cite{145}. They propose a system that automatically “structures” a recorded conversation (telephone, meeting etc.) by identifying change of speaker locations as well as potential changes of topic signified by speaker emphasis and long pauses. The results are displayed graphically to the user with the time line of the conversation flagged at speaker/topic transitions. Although, to avoid relying on \textit{a priori} knowledge, speaker identification in the traditional sense is not performed, each unique participant in the conversation is identified and given a label using identification techniques. Hindus \textit{et al} \cite{146} describe a similar system that is slightly more constrained; it can only operate on two person telephone conversations where the problem of speaker identification is not present. These systems highlight the utility of providing structure to audio data for the purposes of browsing.

\subsection{Music}
Music information retrieval is receiving increased attention. Since the year 2000, there has been a conference dedicated to this area of research. However, despite this growing interest, the systems that have been proposed have been highly constrained. Typically, they are applicable only to small collections of monophonic, western music and operate on symbolic representations such as the written score or a MIDI event list. Examples include those by McNab \textit{et al} \cite{45} \cite{46} and Ghias \textit{et al} \cite{6}.

One vital element of most melody retrieval systems is the query interface. Most use a “query-by-example” paradigm where a user hums a melody that is converted into the appropriate search key (that is, the melody contour). Since most of these systems begin with collections of music already in symbolic form, a different technique is usually required to extract the search key to that which was used to produce the index keys. Techniques that have been used to achieve this task have included autocorrelation analysis \cite{6}, and the pitch-detection procedure developed by Gold and Rabiner \cite{27} (used by \cite{46}).
Song et al propose a melody retrieval system that operates on collections of raw audio data. This system performs a time-frequency analysis of the raw audio signal, which is then filtered to enhance the spectral peaks, in a process they term “harmonic enhancement,” according to the following relationship [147]:

$$E_{t}^{EP}(k) = \sum_{i=-W}^{W} A\left(E_{t}(k) - E_{t}(k + i)\right), \quad \forall \ 0 \leq k < N \tag{7-38}$$

where $W$ is the size of the harmonic enhancement analysis window, $E_{t}(\bullet)$ is the FFT of frame $t$ of the incoming audio signal and:

$$A(x) = \begin{cases} x & \forall \ x \geq 0 \\ 0 & \forall \ x < 0 \end{cases} \tag{7-39}$$

They then extract the pitch of each frame of data using the harmonic sum:

$$F(p) = \frac{1}{N} \sum_{m=1}^{\left\lfloor \frac{N}{p} \right\rfloor} E_{t}^{EP}(mp) \tag{7-40}$$

Where $p$ represents the potential pitch values. This amounts to being a histogram of relative pitch strengths. Having performed these calculations, the spectrogram is segmented into note boundaries using the following relationships and the symbolic value of each is determined. A similar analysis is performed on hummed queries. The advantage of this system is obviously that it is applicable to raw audio data.

A large portion of the music retrieval community contends that the most important index attribute for music is pitch with many systems using the pitch (or melody) contour as the sole index attribute [148]. However, this is akin to a transcription-based system in speech information retrieval. The disadvantages of relying on a purely transcription-based approach were already discussed. Nevertheless, in the case of music, a melody-contour based index can only be considered a partial transcription. Music carries information in four dimensions: pitch, rhythm, timbre and dynamics. Of these pitch and rhythm are fully transcribable while dynamics are often indicated subjectively on a musical score and timbre is difficult, if not impossible, to describe using a textual description. According to Byrd and Crawford [148], the proportion that each of these dimensions contributes to musical understanding is likely to be: pitch 50%, rhythm 40% and timbre and dynamics 10%. Thus, it is clear that while melody does indeed contribute a major portion of the meaning, rhythm is also significant and just as simple to represent in a transcription.
However, while rhythm is relatively easy to represent, it is somewhat difficult to extract from raw audio data. Even using a MIDI representation, the perceived meter and rhythmic structure of a piece can be difficult to determine because it is not directly related to note boundaries. While type I MIDI files can encode tempo and key signature values, rhythm researchers ignore these quantities and use the MIDI representations to “side step” the otherwise necessary pre-processing step of locating note onsets [149].

Rauber and Frühwirth, [150] proposed a non-transcription scheme to cluster music according to its signal characteristics. To achieve this they used a self-organising map with feature vectors extracted from STFT of the musical signal. This method has the advantage that it does not rely on melody extraction and thus is equally applicable to monophonic and polyphonic music. However, this same factor proves to be a disadvantage in that the method is only useful for determining which pieces sound similar in their entirety and hence would not be useful for someone wishing to search for a given melody. Further, while the authors claim that a principle advantage of the technique is that the resultant organisation is free of subjective genre classifications, as would be the case in a manually organised collection, the results reported indicate some clustering behaviour that would be far from intuitive. For example, the popular song “Everything I do” by Bryan Adams is clustered together with pieces by Mozart. While the authors note that an orchestra accompanies the popular song, it is unlikely that, at any one time, a single user would be simultaneously interested in a Mozart composition and the popular song. Of course, this statement is in itself subjective, which highlights a major difficulty in music information retrieval, that is, how to evaluate the relevance of results obtained.

7.2.3. General Audio

As has already been mentioned, little work on content-based retrieval from general audio collections exists. Most of the systems that claim a place in this category deal primarily with only speech and music or are simply mechanisms to discriminate among the three domains, speech, music and general sound. Of the limited systems that do exist, the indexing mechanisms tend to be either automated semantic labelling or low-level statistical analysis.

Wold et al use statistical feature vectors as the index keys in a system for content-based retrieval of general sounds [59]. These vectors describe the loudness, pitch, brightness, bandwidth and harmonicity of the signal. The advantage of this technique is that it can be performed automatically. However, the disadvantages include that the analysis can only be performed on sounds that form a “single Gestalt”, that is they can contain only
one type of sound (e.g., music, speech or a door slam). Also, while they do describe the possibility of melody based retrieval, this would require a second level of processing. Hence, the temporal resolution of the search is determined at design time to correspond to the implementer’s decision of what constitutes an individual sound.

Speech recognition techniques are used in [151] to recognise general environmental sounds. Two dimensional cepstral coefficients were used in feature vectors of length that ranged from two to sixteen parameters. Once again, the sounds analysed needed to be single contiguous sounds. Also, being a recognition scheme, lengthy training is required when the database is created as well as whenever a new sound is to be added.

Pfeiffer et al [152][153] describe a system for automatic audio content analysis. They first segment the audio into four categories: speech, music, silence and noise. This segmentation is performed by observing characteristics in both the temporal and frequency domains. Music content is analysed by first determining the rhythm using cues in the amplitude envelope of the time domain signal then extracting the melody line using a simple pitch detection algorithm for each note. This transcription is used as the index keys for the musical sections of the audio database. General environmental sounds are characterised by statistics on the following quantities: mean square amplitude, frequency, pitch, onset, offset and frequency transitions. The two main disadvantages of this system are that, as described earlier, a musical transcription ignores significant audio information and the requirement of storing a separate index file increases storage requirements. An advantage of the system is the general description used to characterise general audio sounds. This information should be easy to derive from a perceptually based audio representation and so is applicable to the work described in this report.

7.2.4. **Perceptually Relevant Cues for Content-based Retrieval**

Content-based searching and retrieval relies on the presence of some form of index or search key. As was discussed in Chapters 1 and 2, the indexing mechanisms currently available for audio data range from automatically generated statistical features through to manually or semi-automatically generated semantic descriptions. The former are generally too low level to be of genuine utility while the latter require tedious extraction processes, are inflexible and tend to be inadequate to describe the acoustic qualities of the data. The most basic form of searching in textual data is the concept of a key word, or string matching search. An analogous method for searching audio data, with the provision of query-by-example would be particularly useful. However, in order to do so, suitable indexing keys must be identified.
It is the contention of the author that the tracks and harmonic groups identified in this thesis form the ideal basis for such queries being equivalent to letters and words in the textual analogy. This hypothesis finds validation in the work of Fineberg and Mammone [133] who proposed a technique for the detection and classification of “multi-component signals.” They define a multi-component signal as one that has a definite, identifiable structure in the time-frequency domain. This is clearly the same underlying assumption that motivated the current work. Hence, the conclusions in [133] are applicable to the representation that has been developed. Concentrating on speech, the authors suggested that despite the fact that two like utterances will never possess identical time-frequency structures (even if spoken by the same speaker in similar circumstances), there will be a simple transformation between the components (in the time-frequency domain) of the two that remains constant across all components. A comparison is performed by calculating the transformation for one component and then attempting to derive the rest of the components in one utterance by applying the transformation to the other utterance. In the context of the current work, this would amount to using the group models to derive the transformation and then determining whether the individual groups of the reference group could be derived from those of the test group. Application of the technique for other naturally occurring sounds was not reported and is an area that warrants still further investigation.

7.3. Audio Segmentation Based on Source Type

Although somewhat coarse, an index based on the ‘type’ of audio present can sometimes be very powerful. For example, a user may wish to locate all musical items in a radio broadcast. Using the concept of model shapes detailed in section 4.4 a preliminary study as to the representation’s suitability for source type classification was performed.

7.3.1. Track type identification

In keeping with the grouping methodology described in section 4.4, the tracks were first classified according to type. Very short tracks were classed as noise, horizontal tracks were defined to be tone bursts, sweeps were tracks whose frequency contour approximated a monotonic increasing or decreasing function and formants were those tracks that displayed a concavity in either the upward or downward direction. The procedure for determining the track type was as described below.

Given that \( f_i \) is the track frequency at frame \( i \) along the track and \( N \) is the number of peaks in the track, calculate the parameters given by equation 7—1 to 7—6.
\[
\tilde{f}_L = \frac{(f_0 + f_1 + f_2)}{3} \quad (7-1)
\]
\[
\tilde{f}_C = \frac{(f_{N/2-1} + f_{N/2} + f_{N/2+1})}{3} \quad (7-2)
\]
\[
\tilde{f}_R = \frac{(f_{N-3} + f_{N-2} + f_{N-1})}{3} \quad (7-3)
\]
\[
df_L = \frac{(f_2 - f_0)}{2} \quad (7-4)
\]
\[
df_C = \frac{(f_{N/2+1} - f_{N/2-1})}{2} \quad (7-5)
\]
\[
df_R = \frac{(f_{N-3} - f_{N-1})}{2} \quad (7-6)
\]

Then the track type, \( T \), is defined as:

\[
T = \begin{cases} 
\text{NOISE} & N < \text{threshold} \\
\text{TONES} & \frac{(df_L + df_C + df_R)}{3} < \text{threshold} \\
\text{FORMANT} & f_L < f_C > f_R \text{ OR } f_L > f_C < f_R \\
\text{SWEEP} & f_L > f_C > f_R \text{ OR } f_L < f_C < f_R \\
\text{UNKNOWN} & \text{otherwise}
\end{cases} \quad (7-7)
\]

It will be noted that equations 7–1 to 7–3 are simply localised average frequencies at either endpoint and at the centre of the track. The reason that these average frequencies were calculated is that this provided a low complexity method of approximating the ‘shape’ of the track. Another method that may have been useful for identifying formants would have been to determine the concavity of the track by calculating the second derivative of the frequency contour. One potential advantage of this is that formants that are not symmetrical about the centre would be correctly classified (usually incorrectly classified as sweeps with the above procedure). Yet another possibility would be to adapt the shape parameters designed for contour based grouping in Section 4.6.6. However at the time that this preliminary investigation was performed, the former potential improvement was recognised to incur an additional computational expense, which may be unnecessary since, for the purposes of source segmentation, every track need not be identified correctly while the latter improvement was not apparent as the shape parameter-based grouping technique was yet to be developed. A preliminary study of the latter appears at the end of this chapter while the former is left as a direction for future research.
7.3.2. Source Segmentation

The aim of source segmentation is to identify contiguous sections of audio that consist primarily of one of four sound categories: speech, music, silence or noise. Each of these categories exhibits unique characteristics in the track domain. Silence can be identified by the absence of tracks. Music consists primarily of long harmonically related, horizontal tracks (tones). Speech is characterised by the presence of relatively short frequency sweeps, tone and noise bursts interspersed by frequent short periods of silence. The same can be said for many environmental sounds (classified as noise) however speech displays a further, unique characteristic: the presence of concave formants [154]. Finally, sections consisting entirely of noise bursts or relatively short randomly distributed tracks are classed as noise. This preliminary investigation was performed before track grouping had been implemented and, as such, the segmentation was purely on a temporal basis. However, a similar method would be applicable to group-based source type identification.

Segmenting a mixed collection of audio by sound type on a temporal basis involves first identifying the dominant track type on a frame-by-frame basis. Similarly, identifying the most likely source type for a given group simply involves determining the type of the majority of the tracks that comprise the group. For the low-level harmonic groups extracted by the procedures reported in this thesis, this would simply require identification of the type of the fundamental (or model) track.

For the former (temporal) segmentation, determining the dominant source type in a given segment of audio was simply a matter of counting the relative frequency of occurrence of each of the track types. As a result of this analysis, each frame was assigned a label indicating the most likely source type for that frame. The procedure is described by equation 7–8 where $PS$ is the predominant source for the segment, $N_{\text{tone}}$, $N_{\text{sweep}}$, $N_{\text{formant}}$, and $N_{\text{noise}}$ are the number of tone, sweep, formant and noise tracks respectively and $T_1$ and $T_2$ are thresholds.

$$PS = \begin{cases} \text{SILENCE} & N_{\text{tone}} + N_{\text{sweep}} + N_{\text{noise}} \leq T_1 \\ \text{SPEECH} & N_{\text{formant}} > T_2 \text{ AND } N_{\text{formant}} \geq N_{\text{tone}} \\ \text{MUSIC} & N_{\text{tone}} > N_{\text{sweep}} \\ \text{NOISE} & \text{otherwise} \end{cases}$$

Once this frame-by-frame analysis was performed, adjacent frames that were labelled identically were grouped together. To account for errors due to noise, any very short sections were combined with the previous adjacent section. A refinement of the above
technique that was suggested by the preliminary results was to allow a second pass in which adjacent sections were grouped together if they satisfied one or both of the following conditions:

- Relatively short sections bounded by two larger sections that are identical to each other. This condition indicates that the short section was likely labelled in error.
- The predominant track type might be characteristic of the source type in the adjacent section. That is, a short silence section may be grouped together with longer sections of any of the other sound types while a short section initially labelled as music might be reclassified as noise resulting in it being grouped with an adjacent speech section.

7.3.3. Preliminary Results

As this was a preliminary investigation, exhaustive testing of the algorithm was not undertaken. However, some initial test results were obtained and are presented here indicating the feasibility of the technique and motivating future research. The tests were performed using two files: one containing only speech and the other containing a mix of speech and music.

**Track type identification accuracy**

The accuracy of the track type identification procedure was gauged by counting the number of tracks that were correctly identified as a proportion of the total number of tracks in the file. The “correct” identification for the tracks was determined by manual observation of a plot of the tracks. The results obtained are shown in Table 7-1.

<table>
<thead>
<tr>
<th>File</th>
<th>Tracks correctly identified</th>
<th>Total tracks in file</th>
<th>% correctly identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>speech</td>
<td>332</td>
<td>368</td>
<td>90%</td>
</tr>
<tr>
<td>mix</td>
<td>562</td>
<td>662</td>
<td>85%</td>
</tr>
</tbody>
</table>

**Source segmentation accuracy**

The accuracy of the source segmentation algorithm was determined by counting the number of frames correctly identified as a proportion of the total number of frames in the file. Table 7-2 provides the results of source segmentation.

<table>
<thead>
<tr>
<th>Source</th>
<th>Frames correctly identified</th>
<th>Total frames in file</th>
<th>% correctly identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>speech</td>
<td>471</td>
<td>558</td>
<td>84%</td>
</tr>
<tr>
<td>mix</td>
<td>559</td>
<td>602</td>
<td>93%</td>
</tr>
</tbody>
</table>
The actual source segmentation achieved for the file containing both speech and music is shown in Figure 7-1.

![Figure 7-1: Example of segmentation for a file containing mixed source data](image)

The number of segments (8) is as expected although the location of the segment boundaries and the identification of the source type in each segment is not completely accurate. The errors are as follows:

- Segment 0 was terminated earlier than it should have been (approximately 15 frames too early);
- Segment 2 should have been merged into segment 1; and
- A silence segment should have been identified between segments 4 and 5.

### 7.3.4. Discussion of results

The most common inaccuracy in the track type identification procedure is the incorrect identification of a formant track as a sweep. This is followed by the incorrect identification of a tone as a sweep. The former is a lesser problem than the latter since, as long as some formants are correctly identified in a given section of speech, it will be correctly classified. The latter problem is more significant since, if too many tones are classed as sweeps, a music section will be likely to be incorrectly labelled as speech. The refinements that were mentioned are likely to increase the accuracy of the algorithms. However, given that this was a preliminary study, these results are acceptable and serve to indicate that the approach is valid and worthy of future research.
7.3.5. **Extension: Shape parameter-based track type identification**

As has been mentioned, the shape-parameter-based grouping method suggested a more robust method for track-type identification. This section describes the technique and provides the results of a preliminary investigation of its effectiveness.

**Description of the Algorithm**

The heart of this algorithm is the recognition that each of the track classes possess the following characteristics in the shape-parameter domain:

- **Noise:** short tracks (< 10 peaks) are typically due to noise or tracking errors which should also be counted as noise;
- **Tone:** the gradient of all three least-squares lines is close to zero;
- **Formant:** there is usually a significant change of gradient along the track resulting in a measurable difference between the gradient of the left-hand least-squares line and that of the right;
- **Sweep:** there is little or no change in gradient along the entire track, however, the absolute value of the gradient of all three least-squares lines is significantly greater than zero;
- **Unknown:** any track that cannot be described by one of the previous situation is classed as an unknown track.

These principles are illustrated in Figure 7-2:

The principles expounded above, and illustrated in Figure 7-2, may be described by the following equation for T, the type of a track:
where \( N \) is the number of peaks in the track; \( N_{\text{min}} \) is the threshold length for a track to be classified as noise, which was set to 10 for this preliminary study; \( G_L \) is the gradient of the left-hand least-squares line; \( G_R \) is the gradient of the right-hand least-squares line and \( Th \) is a threshold value, which, in this preliminary study, was set to 5.

In order to compare the performance of this technique against the previous method, a simple experiment was performed on the test data set used in the grouping experiments. Using both the simple and shape parameter-based methods, the type was estimated for each track in the combined test data set (294 tracks in total) and compared with the type that it should be (manually labelled). The results of this experiment appear in Figure 7-3. The graphs show that the shape-parameter based algorithm is much better at identifying formants and tones than the simple technique. This can be accounted for by considering that the shape-parameter approach is much more global in nature than the previous approach. Both algorithms detect noise tracks with the same precision. This is because they both employ the same condition to detect noise (see equations 7—7 and 7—9).

Further, the simple method is slightly more efficient at identifying sweeps. One cause of this has been postulated to be an inadequate threshold value in the third line of equation 7—9. However, the set of sweeps was so small that no conclusions can rightly be drawn.
This short study has served to indicate that the shape-parameters developed for track grouping are also useful in track type identification which is the first step in source type identification. It is believed that still further improvement will be possible by using all the elements of the shape parameters, not just the left and right gradients as has been used in this study. Since it was essentially designed for this purpose, using all the coefficients of the shape parameter vector would allow both purely tonal tracks and tracks derived from music performed with vibrato to be distinguished. Further, sweep identification performance should be improved, as more cues to distinguish sweeps from formants would be available. Nevertheless, this preliminary study has reinforced the applicability of the work reported in this thesis to a simple information management task.

7.4. **Summary**

This chapter has presented the results of a brief study of one means by which the main work reported in this thesis can be applied to a problem in audio information management. Although an apparently coarse level of classification, identifying just the source of an audio object can be a very powerful tool for audio browsing, particularly through a lengthy mixed source recording. The brief study has revealed that relatively simple algorithms can give quite acceptable classification rates. This supports the claim that a properly designed perceptually-based representation will facilitate easy access to indexing mechanisms suitable for content-based retrieval and other audio information management tasks. In particular, it should be obvious that, having decomposed the audio data into elemental, cognitively significant units as well as providing a method whereby the probable source of these units may be identified, a firm foundation has been laid for a hyperaudio data model.
Chapter 8

Conclusions and Future Work

8.1. Introduction

This thesis has presented an approach to auditory source separation intended to support content-based coding and information management tasks for general audio data. The chief focus has been low-level auditory object formation from a sinusoidal representation. This has entailed the extraction of low-level primitive elements from the sinusoidal representation followed by the grouping of these primitive elements that may serve as the basis for a fully structured representation that will support audio object-based coding, organisation and manipulation of audio data. This conclusion presents a summary of the main findings of this study as well as providing directions for future work.

8.2. General Conclusions

The goal of the research reported was to address the impasse that exists in the area of audio information management. On the one hand, there is the need to reduce the space and time required to store and transmit what is a naturally data hungry medium. On the other hand, such a large data repository is useless without some means of organising and facilitating access to the individual records within it. Traditionally, these two problems have been addressed in isolation from one another, resulting, in audio representations that completely obfuscate the structure and semantic content of the underlying data, while at the same time, relying on indexing schemes that are inefficient or of little semantic relevance.

The hypermedia data model provides the ideal solution to this dual problem. However, decomposing a raw audio stream into units that may form the nodes and link anchor points in this model is by no means a trivial undertaking. The nascent field of computational auditory scene analysis (CASA) attempts to address this challenge. Yet, despite its limitations, the findings of CASA are sufficiently advanced to allow an investigation of its potential to solve problems in other domains. With respect to the above paradox, it has been shown that CASA provides the most promising means for generating an object-based audio representation.

Hence, the approach of this thesis has been to present a perceptually based, structured audio representation designed to provide intrinsic support for content based coding and
information management applications. While several audio representations may claim to be structured, and some even allow the concept of an audio object by virtue of description languages, an automatic mechanism for transcoding from a raw signal representation to an object based representation is far from being realised. Thus, the contribution of this thesis has been two fold. Firstly, it has made a contribution to the general body of CASA research by suggesting new methods for source separation, while simultaneously this study has been conducted in the context of unifying the fields of audio coding and audio information management. This reference to three different fields significantly influenced the scope and directions of the work that has been presented.

One consequence of this multidisciplinary approach has been the somewhat unorthodox (from the perspective of the CASA community) choice of a sinusoidal representation as the underlying basis for the source separation algorithms. The motivation for this choice arose largely from the requirements imposed by the aim to support data compression and information management. Whereas most CASA systems are simply required to either produce separated raw audio signals or some form of listing of identified sound events, the stated aim of the work reported here was to develop a representation that supports data compression. Further, the inherent structure of the underlying data was to be visible in the encoded stream. The sinusoidal representation allowed all of these requirements to be met.

The modern trend against sinusoidal representations within the CASA community typically finds its justification in the assertion that it is impossible, or at the least very difficult, to adequately address the problem of coincident partials with such a representation. Thus, it was a key requirement of the current work to establish whether this need be so. As such a significant contribution has been made in that an architecture has been developed that has provided a preliminary solution to the problem of coincident partials by exploiting the relationships between the partials of a single harmonic group.

The use of a sinusoidal representation as the basis for source separation also suggested the obvious cues to use in affecting this separation. The definition of a sinusoidal representation means that the frequency and amplitude contours are directly visible in the representation. Hence these were the first cues for which grouping algorithms were devised. Further, the availability of frequency contours suggested a novel track-based method for harmonicity grouping that was shown to be superior to a ratio of individual peaks based approach, a version of which was employed by previous researchers [141].
In summary, this thesis has presented work that is motivated and influenced by three
diverse fields of study. The top-level architecture presented for low-level auditory object
formation is unique in its use of feedback at a lower perceptual level than appears in
most CASA systems. A further contribution has been made in the development and
testing of several alternative low-level algorithms to perform specific grouping tasks
within the framework of the architecture developed.

As a result of this study it has been shown on a preliminary scale that the problem of
coincident partials can be overcome and that source separation using a sinusoidal
representation is achievable. Further, given that sinusoidal representations have already
been applied to data rate reduction tasks, it is manifest that the aim of compressed data
storage is simultaneously achievable. Preliminary studies, which will be reported in the
next section, suggest that the final aim, support for information management and
retrieval is also achievable given the results of this research.

8.3. Conclusions Drawn from Grouping
Experiments

Chapters 4, 5 and 6 presented three basic grouping strategies along with several
variations for each. Some of these strategies were entirely novel while others were based
on previous work. Chapter 4 presented strategies for frequency and amplitude contour-
based grouping while Chapter 5 presented various strategies for performing
harmonicity-based grouping. As was discussed in Chapter 4, previous researchers have
tended to focus on only one or the other of these dimensions with harmonicity-based
(often referred to as “pitch-based”) grouping apparently receiving the most attention.
Finally, Chapter 6 presented two strategies for performing grouping along all three
dimensions simultaneously. The first of these was an entirely unique approach while the
second was similar to that employed by Virtanen and Klapuri [141].

Three variations were presented for frequency and amplitude contour-based grouping.
The first of these was a simple mean-squared error between pairs of tracks. This was the
metric employed by Virtanen and Klapuri [141] as a distance measure along these two
dimensions. However, it was recognised that the simple scaling to a common mean
frequency, employed by Virtanen and Klapuri, was not ideal and a shape normalisation
procedure was introduced to address this shortcoming. In an attempt to improve
grouping performance further, a new gradient based ‘shape parameter’ was developed.
Noting that only a slight performance improvement was made for amplitude based
grouping, a third technique was investigated: the Fourier shape descriptor. The
application of this image processing technique was entirely new in the audio domain.
The results of these investigations indicated that the optimum technique for grouping along the frequency contour dimension was the simple MSE-based approach where shape normalisation had been applied to the tracks while the best performing technique along the amplitude contour dimension was the shape parameter-based technique where the track amplitudes were scaled to a common mean.

In Chapter 5 it was stated that the most commonly applied grouping strategies in CASA research exploit the notion of harmonicity or pitch. Chapter 5 reviewed the various harmonicity-based strategies in existence noting the strengths and weakness of each. Several new or improved strategies were then suggested and investigated. The harmonicity-based grouping strategies that were developed can be classified into two broad categories. The first involves determining an estimate of the fundamental tracks responsible for all of the tracks in the data then grouping together all tracks that have a common fundamental. The second relies on a ‘harmonicity’ error that indicates the likelihood that any two tracks are harmonics of a common fundamental.

Considering the first class of harmonicity-based grouping strategies, two basic methods were devised with the first allowing two variations. The first was an implementation of a theory of pitch perception expounded by Bregman [1]. While this method was similar to that employed by Cooke [118] in that each track contributed at least one “vote” for the identity of the candidate fundamentals, it was unique in its employment of the differences between track frequencies, which was the essence of Bregman’s theory. The two variations of this technique consisted of a peak-by-peak- and a track-based calculation estimate of the fundamental tracks, which simply involved altering the characteristics of the histogram employed to generate these fundamental track estimates. The second method in this class involved determining the common fundamental of each pair-wise combination of tracks using Euclid’s algorithm to determine the highest common factor of each pair of corresponding frequencies. Once again a histogram was employed to determine a small set of the most likely resultant fundamentals. Experimental results indicate that the track-based variation of the first approach gave the best performance of algorithms in this class.

The second class of harmonicity-based grouping strategies was similar to that employed by Virtanen and Klapuri [141] along this dimension. The approach involved determining the relative likelihood that a given pair of tracks was harmonically related. This was achieved on a peak-by-peak basis by calculating the ratio between the two peak frequencies and then determining the error between this and the ratio of the most likely
Chapter 8: Conclusions and Future Work

Harmonic numbers of the two frequencies assuming that they were harmonics of a common fundamental. Two variations of this technique were presented based on the method employed to determine the most likely harmonic numbers. The first variation, similar to the Virtanen and Klapuri approach, involved finding the minimum error between the ratio of corresponding peak frequencies and the ratio between all pair-wise combinations of allowable harmonic numbers while the second variation employed the method of continuing fractions. The experimental investigation revealed that the continuing fractions variation gave superior performance to the simpler first variation. However, the performance of this technique was inferior to that of the best performing algorithm from the first class of harmonicity-based grouping techniques.

Having determined the optimum techniques in each of the individual grouping dimensions, Chapter 6 presented two strategies for combined grouping along all three dimensions simultaneously. The first method involved grouping the three separate copies along each of the dimensions and then combining these by employing a variation on set intersection that was devised to overcome the observed tendency for simple set intersection to produce groups that were too small. The second method involved calculating the weighted Euclidean distance between vectors consisting of the three error measures calculated along each of the grouping dimensions and using this as the distance in the basic grouping algorithm. Experimental results revealed that the latter of these two strategies gave the best performance.

From the general user’s perspective, the most interesting results are those reported at the end of Chapter 3, since this chapter deals with the audible input and output to the system. This chapter began with a discussion of the basic architecture of the system developed to achieve auditory source separation and to ultimately produce the object-based audio representation, comparing and contrasting it to systems currently in existence. Various experimental findings were then presented to justify the design of each stage of the system except the track grouping stage for the obvious reason that this was the theme of Chapters 4 through 6.

Specifically, the experimental study in chapter 3 began with an investigation of phase effects on peak picking performance and their influence on the choice of the underlying frequency transform, which concluded that an FFT was the most appropriate. It was then recognised that the classic McAulay and Quatieri peak picking algorithm was too lax for the broadband analysis required in general audio applications. Thus, a stricter algorithm...
proposed by Terhardt was tested, which proved to be too strict. Finally, a compromise solution was proposed and found to indeed provide optimal performance.

In a similar manner, the simple McAulay and Quatieri peak tracking procedure was found to be unsuitable for the resolution of the transform required in the current system. The principal problem was identified as being the horizontal bias of what amounts to being a zeroth order predictor. Hence, a modification to the algorithm, based on first order prediction, was suggested and found to improve tracking performance.

A problem was then encountered that was unique to the design of the time-frequency distribution that formed the basis of the representation. This distribution was designed with overlapping frequency bands to provide redundant data in the hope of optimising the time-frequency resolution across the entire analysis bandwidth. However, this resulted in a set of overlapping tracks at different time-frequency resolutions. Initial observation of these tracks validated the choice of an overlapped TFD in that the optimum time-frequency resolution is more dependent on the time-frequency characteristics of the specific signal rather than on which band/s of the time-frequency analysis plane it exists in. Although this represented an advantage on the one hand, it presented a challenge in combining the tracks from the individual bands in a manner that captured the ‘correct’ time-frequency characteristics. Results of the initial very simple approach to the problem were presented and noted to be unsatisfactory. A second more sophisticated solution was then outlined and the results of this were shown to be superior.

The final stage reported in Chapter 3 was that of track inversion. As a detailed study of compression performance was beyond the scope of this thesis, a formal study of reconstruction quality was not conducted. However, informal tests revealed that the separated and inverted signals were, at the least, intelligible with the source recognisable and, at best, of very good quality.

The results obtained and reported herein have served as a proof of concept. That concept being that it is feasible to attempt auditory source separation based on a sinusoidal representation. The significance of this is three fold. For the CASA community it challenges the assumption that sinusoidal representations can only be employed with limited success where overlapping partials exist. To the field of audio coding it manifests the possibility of achieving audio compression without sacrificing direct access to the inherent structure of the underlying data. Indeed, it reveals the structure that would have been hidden even in a raw representation. Finally, it solves one of the most difficult
yet pressing problems facing the audio information management world: how to deal with concurrent audio objects. Based on these results a number of avenues for future research are opened in each of these fields. These will be discussed later in this chapter.

8.4. Comparison with Other Contributions in the Literature

In this section brief concluding summaries of the similarities and differences between the current work and the more important and, from the point of view of the current work, influential contributions to the literature are provided.

8.4.1. McAulay and Quatieri

McAulay and Quatieri [73] devised the sinusoidal representation upon which the work reported here is based. The original M&Q system was designed for speech data with a bandwidth of approximately 5kHz. The underlying transform was of fixed resolution and the peak picking and tracking algorithms were relatively simple and straightforward. Some of the ideas employed by McAulay and Quatieri were applicable to the work reported here while others required modification to be useful.

The first difference between the work of M&Q and that presented here is the expected input data. In contrast to the relatively narrowband speech signals expected by the M&Q system, the current work assumes the input data to consist of general audio at a minimum bandwidth of 16kHz. This necessitates a much wider bandwidth for the time-frequency analysis, which can only be adequately covered with a multi-resolution analysis. The simple peak picking procedure of M&Q was also found to need extending to account for additional noise in the higher resolution analysis employed. Finally, the peak tracking algorithm was found to have a horizontal bias that once again became a problem in the higher resolution analysis employed in this system. This shortfall was easily overcome by increasing the order of prediction from 0 to 1.

8.4.2. Cooke

Cooke [118] presented what was possibly the first major attempt at CASA for a “real world” application: improving the performance of speech recognition systems by separating the voice of interest from the background “noise”. This aim presents the first difference between Cooke’s system and that proposed here. Cooke essentially employed CASA as a sophisticated method for signal enhancement, while the aim of the current work is to organise the entire auditory scene into a collection of audio objects, each of potentially equal importance.
Cooke’s time-frequency decomposition was effected by a perceptually tuned gammatone filterbank. One disadvantage of a filterbank implementation is computational expense. Also, from the perspective of the current work, it was decided that a coarse lowpass, fully overlapped filterbank provided time-frequency resolution and data redundancy benefits that a partially overlapped filterbank cannot achieve.

The basic element in Cooke’s representation is the place group, which corresponds roughly to the peak used here. Cooke’s version of the track was dubbed the synchrony strand. One feature of strand formation that bears resemblance to the work reported here is the assumption of a continuous first derivative. The first order prediction used here implies this also.

The main grouping cues employed by Cooke were harmonicity and amplitude modulation rate with frequency contour similarity only being used to affect a secondary level of grouping. This is in stark contrast to the current system where amplitude contour, frequency contour and harmonicity have been combined to generate a more robust grouping. Cooke’s algorithm also coincidentally bears some resemblance to one of the ones developed here (BREG). However, Cooke’s approach appears to have resulted from a purely pragmatic analysis of the problem, while that which is proposed here was born out of a faithful implementation of Bregman’s theory of pitch perception and exploits a greater level of statistical redundancy to improve robustness.

8.4.3. Ellis

Ellis’ initial attempt at CASA was based on a sinusoidal representation derived from a constant-Q filterbank implementation of a time-frequency distribution [72]. Few details of the source separation results were reported of this attempt before Ellis moved onto the prediction-driven approach (PDCASA). PDCASA is based on the premise that ASA is aided by top-down processes in which recognition of sounds aids in their tracking and separation from the “background.” As such Ellis incorporated a recognition engine within a feedback loop that aided the system to describe the general auditory scene that was presented to it. While Ellis’ basic premise is true to some extent [1] it is also true that a significant level of grouping must occur at the lower levels of perception before cognitive processes come into play. The ability to instinctively react to one’s name faster than would be required to actually recognise it is one example of this.

8.4.4. Solbarch

Solbarch’s [121] system, like that of Cooke’s, was based on a perceptually tuned filterbank. The chief grouping cues employed by Solbarch were pitch (or harmonicity)
and onset synchronisation. Perhaps the most interesting aspect of Solbarch's system from the perspective of the current work is its employment of feedback in the peak tracking phase which is carried out simultaneous to group formation. This feedback mechanism allowed the track onsets to be more accurately determined and a greater ability to separate nearby partials. By Solbarch's estimation the chief difference between his system and those transform based strategies that preceded it was that “it is neither strictly feed-forward nor single-pass [121].” That being the case, the chief difference between Solbarch's strategy and the current work is that, while his system will step back a few frames after a track has started to form to adjust the onset time, the current work revisits the entire data set (over the window being analysed) after all the initial track and group estimates have been extracted in order to refine the entire track group.

Yet another similarity exists between the two works that somewhat validates the resolution selected for the transform employed in the current work. Solbarch notes that there is a need to optimise the resolution in both time and frequency and that high temporal resolution is required to accurately identify onsets (of particular concern to him as they formed a grouping cue) while high temporal resolution is required to accurately separate nearby partials. He uses a combination of a multi-resolution transform and two pass processing to affect this compromise.

8.4.5. Klapuri and Colleagues
Over the past few years Klapuri and colleagues have been developing a source separation system aimed at musical instrument separation [37][38][36][141][142]. As such many of its features have been adapted to be uniquely suited to this domain. The initial system began with a sinusoidal representation as the underlying transform and employed a grouping mechanism similar to the combined distance-vector approach adopted in this thesis [141]. The distance metrics employed by this system were mean squared error comparison of the frequency and amplitude contours as well as a frequency ratio approach for harmonicity similar to the lookup table method of the current work. Unfortunately, it was not possible to perform a direct comparison between the technique employed by Virtanen and Klapuri [141] and the current work because the required constants were not reported in the literature. However, the experimental evaluation of grouping cues revealed that the new shape parameter approach developed out performed MSE for comparison of the amplitude contour and the track-based Bregman inspired harmonicity grouping strategy performed significantly better than the frequency ratio approach.
8.5. Directions for Future Work

8.5.1. Object-based Coding Potential
It has been stated that one of the aims in developing the representation has been to demonstrate the ability to compress audio data without compromising the information management aspects. To this end, a preliminary study into the compression possibilities presented by the representation has been undertaken. However, it was also noted that data compression is an extensive field of research in itself and a complete evaluation of the representation’s compression performance was beyond the scope of the work reported here. Accordingly, this section reports on a preliminary study of the suitability of the representation for compressed data storage as a motive for future research.

The very fact that the representation developed is based on a sinusoidal coding model implies that it is suitable for data compression. However, in the raw state presented in this thesis, the representation may produce either compression or expansion depending on the acoustic complexity of the signal encoded. As an example, the encoding as described in this thesis was applied to two files with an original data rate of 64 kbps. The first, an artificially generated chirp resulted in a single track giving an average data rate of 27 kbps while a sample of recorded male speech resulted in 47 tracks giving an average data rate of 74 kbps. The subjective quality of the reconstructed signals was perfect for the artificial sweep and average-good for the speech signal. The original waveforms, extracted tracks and reconstructed waveform of both signals appear in Figure 8-1.

The above results may not seem particularly encouraging, especially considering that the speech signal actually displayed data expansion. However they must be viewed in light of the fact that no effort was made to optimise the storage of the track parameters (that is, the peak-by-peak time, frequency, magnitude and phase values). There was also a reasonable amount of additional overhead data stored to facilitate manipulation and grouping of the tracks in the experimental setting. Only minimal attention was given to efficient storage of this overhead.

Given the above results, it was decided to perform a preliminary investigation as to suitable methods for optimising the storage requirements of the representation presented here. Compression at two levels was considered; at the track level and at the harmonic group level. At both levels, the need to support random access to self-contained, individually decodable units (tracks and/or groups) was an overriding concern that influenced the selection of compression techniques.
Figure 8-1: Example of coding an artificial chirp and a male speech signal
Track-based Encoding
Four essential parameters must be recorded for a track:

1. Temporal extent;
2. Frequency contour;
3. Amplitude contour; and
4. Phase contour.

In the raw representation, each one of these parameters is recorded for every peak along each track. The first is recorded with 2-byte integer precision while the latter three are recorded using 8-byte floating-point precision. In addition each track contains 11-bytes of overhead data. Thus, there is an absolute minimum of 36-bytes required to store each track (with only one peak). Clearly, achieving compressed data storage involves reducing the amount of data required per track. There are a number of ways in which this might be achieved that include:

- Reducing the precision of each of the parameter values;
- Using a predictive coding scheme to encode the track trajectories;
- Using a parametric representation, such as a polynomial description to encode the track trajectories; and
- Reducing the amount of overhead data for each track (group-based coding).

Of these methods, the first is by far the least preferred as it would tend to degrade quality very quickly for little compression gain. The second method promises good results, given the slowly varying nature of the frequency contours in particular and was the subject of a preliminary investigation reported below. Given the apparently simple set of track trajectories, the third method also holds some promise for compressed data representation, however, being a more complex task than predictive coding, it is beyond the scope of this preliminary investigation. Finally one method to achieve the fourth point is to encode the tracks in their harmonic groups. This reduces the storage requirements to one complete contour (describing the model track) along with a small amount of residual information for each track. Thus on a per track basis the overhead and raw data required is significantly reduced. Further, the more tracks that belong to an individual group, the greater the coding gain.
Predictive coding of track trajectories

The amplitude and frequency contours for the tracks vary slowly in time. Hence, DPCM might be an ideal way to encode them. To further increase the coding gain, variable length codewords can be used.

The first step is to find the contours’ numerical gradients:

\[
d f(t_i) = f(t_{i+1}) - f(t_i)
\]

(8—1)

\[
d A(t_i) = A(t_{i+1}) - A(t_i)
\]

(8 — 2)

where \( f(t_i) \) and \( A(t_i) \) represent the frequency and amplitude values respectively of the model track at time \( t_i \). Huffman coding is used to assign the variable length codewords.

Thus for each model contour the following quantities must be recorded: an initial value, the Huffman codebook and the Huffman encoded values. Table 8-1 summarises the bit allocation for the amplitude and frequency contours.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Number of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial value</td>
<td>7 bits (amp) 10 bits (freq)</td>
</tr>
<tr>
<td>Code book</td>
<td># symbols * 7 + sum of symbol lengths</td>
</tr>
<tr>
<td>Contour (approximately)</td>
<td># of frames * entropy of gradient function</td>
</tr>
</tbody>
</table>

Since the values of the amplitude contour vary from 0 to 90 dB SPL, with a resolution of 1 dB SPL, 7 bits are used for the initial value. Similarly, the frequency range is 0 to 1 kHz with 1 Hz resolution, so 10 bits are used. Each entry in the codebook follows the pattern: value – 4 bits; codeword length – 3 bits followed by the codeword – 7 bits max.

Another option for the amplitude and frequency contour is to employ a three-dimensional chain code. This is an extension of the two dimensional Freeman chain code introduced in [155]. Chain code represents a trajectory in terms of relative transitions in a parameter space. Chain codes are an ideal way to code the tracks since the most significant information contained in the tracks is their contour. By using a three dimensional chain code, both frequency and amplitude contours can be coded at the same time further compacting the final representation. This may seem to come at the expense of a more complicated search since the frequency and amplitude contours are
not explicitly separated as required in some retrieval tasks. However, judicious code assignment can separate these two contours implicitly.

To simplify chain coding, two assumptions are made concerning track nature. The first assumption is that successive codes along a chain correspond to a single step in time, where the size of this step is defined to be the length of the smallest analysis window. The second assumption is that the amplitude and frequency can only vary by one ‘unit’ (the size of which is determined by the previous value) from one frame to the next. The track extraction procedure usually results in tracks that are fully sampled, although it may occasionally result in tracks with one or two missing peaks. In these cases, satisfying the first assumption requires that the tracks be interpolated across these intervals. However, since this would be required at synthesis time, there is no disadvantage in performing this step at an earlier stage. The second assumption is valid since audio signals are well known to vary slowly in time.

Given these two assumptions and setting the current point on a particular track as the origin of a three dimensional space, the next point in the track may occupy one of nine positions:

![Figure 8-2: Allowed track trajectories for 3-D case](image)

From Figure 8-2 it can be seen that nine codes are needed requiring only four bits to store each point. A track termination code is required which can be any one of the unused four bit combinations. A small amount of ‘header’ information is also required for each track: the position (in time) of the first point in the track, and the initial amplitude and frequency values. Further coding gains can be achieved with variable length chain codes.
Group-based Coding

Considerable coding gain may be achieved by exploiting the redundancies within the harmonic groups identified as the main aim of this thesis. Encoding the tracks in groups has the added benefit of making the inherent structure of the underlying audio (c.f. Figure 1-5) more explicit in the encoded data stream. In the case of harmonic groups, the key parameters that must be recorded include:

1. Number of tracks (harmonics);
2. Model Frequency versus time contour;
3. Harmonic number and frequency offset of each track;
4. Model Amplitude versus time contour;
5. Amplitude versus harmonic scale factor for each track;
6. Model Phase versus time contour;
7. Phase versus harmonic phase offset for each track;
8. Start and stop time of group model; and
9. Start and stop of all tracks (relative to group model).

The model contours may be encoded in a similar manner to that described for tracks in the previous section. In addition to these and the parameters noted above, a small amount of residual information may be required to maintain acceptable reconstruction for the harmonic component of the signal.

Transient and Noise Representation

The foregoing discussion has concentrated solely on the harmonic component of the signal. For many natural sounds, particularly speech and music, this makes up a large portion of the sound. However, transient and noise-like portions are also important features of these. From the point of view of simple compressed data storage, where a given audio file is encoded and then reconstructed and played back in its entirety, a simple residual of the sound may be derived and encoded using any form of statistical encoding on a per-frame or per-object basis. However, this method would severely compromise source separation possibilities.

Substituting the residual in only those time frames where no track data was found for separated speech only mixtures is a strategy employed by Okuno et al who argued that this would be a better guess than silence is for any missing segments and that induction effects might mask out the interfering ‘noise’ [156]. This would be a useful starting point for future development of the general representation reported here. Ultimately, however,
a method to separate the noise and transient components will need to be found. This is perhaps the most difficult challenge to be met.

8.5.2. The Beginnings of True Hyperaudio

The track and group-based structure developed in this thesis obviously follows the model shown in Figure 1-5, which implies it would be a highly suitable base for hyperaudio. The only elements that remain to be addressed are the formation of the higher-level auditory objects or streams and the nature of the navigation interface. The former of these issues is an extension of the focus of this thesis and, as such, has been discussed previously, hence, the focus of this section is the latter.

**Design Considerations for a Hyper-audio Interface**

Determining the nature of a hyper-audio interface is by no means a trivial matter since many issues that do not apply to visual data arise. These problems are a direct result of the transitory nature of the medium. For example, an area of an image node may be linked to some other node and this can easily and unobtrusively be indicated by an appropriate change in the display as the cursor is positioned over the area. Also, because of its persistence, the user is free to follow the link at any time. Providing an analogous level of interaction in an unobtrusive, intuitive and user-friendly manner in the audio domain requires careful design of the presentation engine.

Three possibilities exist for the hyper-audio navigation interface. It may be purely auditory, purely visual or a combination of both. Hyperspeech [157] is an example of a purely auditory navigation interface. Such an interface is useful whenever a visual display is unavailable (over a telephone line) or inappropriate (for visually impaired users). However, navigation becomes increasingly cumbersome as the number of links and nodes increases since the user has to either remember the names of the links available from each node or must listen through a spoken list of them.

Related problems with such an interface are that labelling the link source points in an unobtrusive manner is difficult in data where an interruption by a spoken menu would be unacceptable to the user (in musical passages for example). In this case, it is quite probable that the link will have vanished even before the user has decided whether or not to follow it. Further, unlike nodes in visual media, it is impossible to audition an entire recording simultaneously to get a feeling for the overall length and structure of the node or the general location of anchor points within the node. Playing all the highlights of a radio recording or musical piece, for example, would result in a cacophony that would be
neither intelligible nor informative [157]. Possible solutions to this include time-compressed and summary playback.

One audio interface feature that has been used in a small set of applications, that is particularly useful, is that of audio data input facilities. These have taken the form of spoken commands (as in the hyper-speech system), the provision for entering audio data for the purposes of onomatopoeic comparisons in an audio database [59] and hummed melodies as queries in a musical database. While illustrating the feasibility and usefulness of the concept of an auditory input interface, these systems suffer the disadvantage of being operational only in constrained environments.

The idea of a purely visual navigation interface for auditory data may seem somewhat illogical, however, this is precisely what most of the existing audio interaction tools provide. In most audio editing applications, for example, interaction is with either a time-domain waveform or time-frequency domain spectrographic display. The only auditory component of the interface is the ability to audition the results after some manipulation has been applied. Multimedia enhanced hypertext also falls in this category, since the entire navigation up to the point of actual playback is performed using a visual interface.

Despite the apparent paradox, there are valid application domains for a purely visual interface for interaction with audio data. The most obvious being the case of hearing impaired users. However, current display mechanisms fall far short of the requirements. Traditional waveform or spectrographic [71] displays offer very little content information and can only be interpreted by expert users. Time-line displays [158] offer a part solution to this problem, however, those in existence suffer one major disadvantage: they cannot represent simultaneously occurring sounds. Finally, with the possible exception of a hypertext link referring to an entire audio file as its destination, no provision for source and destination link anchor points has been made in any of these systems. Hence, the requirements of a visual interface for hyper-audio applications are:

1. Meaningful content information for each object must be clearly visible or easily accessible. Representation of both objective (type: music/speech) and subjective (loud) qualities is desirable;

2. Must be able to display temporal variation along an object (e.g., a sweeping tone whose pitch varies);

3. A method to present link source and destination anchor points must be provided;
4. It must be possible to view the data’s underlying structure along the two important dimensions of time and object, either individually or in combination;

5. The display of this structure must incorporate the notion of a hierarchical object definition; and

6. The ability to edit the structure by adding, or redefining the links between modifying objects should be provided.

The ideal general hyper-audio interface is audiovisual in nature. Thus the framework proposed possesses such an interface. It combines all of the features of the ideal visual interface described in the previous paragraph with the following auditory interface features:

1. Audio input to allow for query by example retrieval facilities and possible voice command input;

2. Options for summary/time-compressed playback across time and/or object dimensions;

3. The ability to commence playback from any point in the time-object space; and

4. The ability to specify a ‘play-list’ (graphically or textually) of links allowing summary or complete playback along an arbitrary path through the hyper-audio structure.

An Elementary Hyper-audio Interface Suggestion

The number and nature of specifications on the visual interface require extensive use of an interactive GUI environment with several display options available to the user. A basic display that allows simultaneous presentation of most of the elements specified in the previous section is a combination of a graph or tree like display and a time-line display as illustrated in Figure 8-3.

In keeping with the hierarchical definition of hyper-audio several levels of this basic idea are required. Navigating between the levels is a simple matter of “zooming in” on an object in the graph or a section of the time line or “zooming out” to the next higher level.

At the highest (or summary) level, only basic content information is revealed. Slightly more content information can be provided via the time-line by indicating the average relative loudness of each object by the width of the bar used to represent it and average relative pitch by the bar’s placement along the vertical axis. At increasing levels of detail,
relative instantaneous pitch and volume can be displayed by varying the position and width of the bars along their length.

Several display options are available for the organization of the objects displayed in the top panel. They can be grouped by type (as in Figure 8-3), by voice (specific instrument, person or other noise source), or be grouped by the user in the same way elements are grouped in computer graphics applications.

Specifying links is simply a matter of defining bookmark or tag locations in either the time-line or structural display. Specifying an anchor in the structural display results in the anchor referring to the entire object while anchor locations defined in the timeline are instantaneous. Where the anchor location refers to an entire object, audio playback commences at the beginning of the object and continues for whatever length of time is implied by the presence (or absence) of subsequent source anchors within the object, as specified by the referring link itself or for the duration of the object.

Finding the desired location for these anchor points is greatly simplified by both the content information available in the combined display and the structure of the underlying data representation. Indeed, the underlying data model allows for the display to “zoom in” to the level of an individual note or phoneme making it a relatively straightforward task to accurately locate almost any desired portion of a sound event without having to “hunt around” individual sample values in a waveform display.

In terms of the audio interface, providing audio input facilities is obviously a trivial matter. Processing the resultant input is, however, not so trivial. Current ASR technology would allow for only very basic voice commands to be incorporated without subjecting users to the tedium of a training routine. This feature could be expanded as the...
technology advances. Using the input data to facilitate query-by-example access to the data is, however, a viable aim given the underlying audio representation.

The track-based structure at the very lowest level of the underlying data model also makes complete random access and playback very easy to achieve. Each individual track is directly accessible and can be decoded from any point along its extent. The data from individual tracks co-located in time is simply added together. With efficient indexing of the tracks and implementation of the required modulator-adder, this decoding can be performed in near real time. This provides a solid foundation upon which to build each of the playback requirements specified in the previous sub-section.

8.6. Final Remarks

This thesis has presented a new approach to computational auditory scene analysis that has been motivated by the dual problem of compressed storage and information management of audio data. At the foundation of the approach has been a sinusoidal representation with a time-frequency resolution unique to the field of CASA. Based on this representation, a method to extract low-level elements and group these into low-level auditory objects has been presented. The top-level architecture used to affect these steps displayed feedback in a manner that is atypical to traditional CASA systems. It is this feedback mechanism that allowed the use of a sinusoidal representation by resolving the problem of coincident partials.

The result of this study has been the description of the lowest two levels of what can eventually form a fully structured audio representation allowing content-based coding and manipulation of audio data. The low-level objects that have been identified in this thesis would form the ideal basis for link anchor or destination points in a hyperaudio system and provide useful coding primitives. Thus a promising foundation has been laid for future research.

As has been the case in this study, the pursuit of this future research will certainly bring gains to each of the three fields that this study has both sought to influence and been influenced by. Perhaps the most challenging contribution yet to be made to the CASA field is successful separation of transient and noise-like sources. The audio coding domain will benefit from a more sophisticated approach to extracting the harmonic components in the HILN class of codecs while audio information management research will make obvious gains, with perhaps the most apparent starting points for future investigation being the use of the low-level primitives and auditory objects identified as
the basis of a simple hyperaudio system or as index keys in a query-by-example system. Despite its long incubation, Bush’s vision may soon finally become a reality.
Appendix A

Units of Measurement in Acoustics

Intensity [9]

Intensity is related to sound pressure according to:

\[ I = \frac{p^2}{\rho c} \text{ W.m}^{-2} \tag{A–1} \]

where \( p \) is the RMS sound pressure of the signal in pascals, \( \rho \) is the density of the medium and \( c \) is the speed of sound in that medium.

Sound pressure level [9]

Sound pressure level (SPL) is a measure of the intensity of a sound relative the standard reference sound pressure of 20\( \mu \)Pa. The SPL may be obtained using the relationship:

\[ L = 20 \log \left( \frac{p}{20 \mu \text{Pa}} \right) \text{ dB SPL} \tag{A–2} \]

where \( p \) is the RMS sound pressure of the signal in pascals.

Sensation level [9]

Sensation level is the sound pressure level of a sound measured relative to the minimum audible sound pressure level of the tone under test. Sensation level is thus subject dependent.
Appendix B

Full Results for Frequency and Amplitude Contour Grouping

Table B-1: Results for frequency and amplitude contour-based grouping.
Frequency contour-based grouping was performed using the MSE-based approach with shape normalisation applied to the tracks. The amplitude contour-based grouping was achieved using the shape parameter method with the tracks normalised to a common mean.

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency Contour</th>
<th>Amplitude Contour</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>10</td>
<td>101</td>
<td>151</td>
</tr>
<tr>
<td>20</td>
<td>102</td>
<td>152</td>
</tr>
<tr>
<td>30</td>
<td>103</td>
<td>153</td>
</tr>
<tr>
<td>40</td>
<td>104</td>
<td>154</td>
</tr>
<tr>
<td>50</td>
<td>105</td>
<td>155</td>
</tr>
</tbody>
</table>

---

299
Appendix B: Full Results for Frequency and Amplitude Contour Grouping

Time (Frames)

Frequency (Hz)

Time (Frames)

Frequency (Hz)

Time (Frames)

Frequency (Hz)

Time (Frames)

Frequency (Hz)
Appendix B: Full Results for Frequency and Amplitude Contour Grouping

![Graphs showing frequency and amplitude contours over time.](image-url)
### Appendix C

**Full Results for Harmonicity Grouping**

The table below shows the results for the two best harmonicity grouping algorithms, that is, the peak- and track-based Bregman algorithms:

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td>30</td>
<td>400</td>
</tr>
<tr>
<td>40</td>
<td>500</td>
</tr>
</tbody>
</table>

---

![Graph showing time vs. frequency for peak-based (BREG) algorithm]

![Graph showing time vs. frequency for track-based (BREG_TRACK) algorithm]
### Appendix C: Full Results for Harmonicity Grouping

<table>
<thead>
<tr>
<th>Peak-based (BREG)</th>
<th>Track-based (BREG_TRACK)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency (Hz)</strong></td>
<td><strong>Frequency (Hz)</strong></td>
</tr>
<tr>
<td>Time (Frames)</td>
<td>Time (Frames)</td>
</tr>
</tbody>
</table>

#### Peak-based (BREG)

- Frequency: 1245, 1863, 1081, 1000, 1120, 650, 877, 386, 967, 595, 352, 210, 110, 20
- Time (Frames): 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75

#### Track-based (BREG_TRACK)

- Frequency: 1245, 1862, 1081, 1000, 1120, 650, 877, 386, 967, 595, 352, 210, 110
- Time (Frames): 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75

---

[Graphs showing time vs frequency plots for Peak-based (BREG) and Track-based (BREG_TRACK) with different frequency bands and time frames.]
Peak-based (BREG)

Track-based (BREG_TRACK)
### Appendix C: Full Results for Harmonicity Grouping

#### Peak-based (BREG)

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1740</td>
</tr>
<tr>
<td>6</td>
<td>1720</td>
</tr>
<tr>
<td>12</td>
<td>1700</td>
</tr>
<tr>
<td>18</td>
<td>1680</td>
</tr>
<tr>
<td>24</td>
<td>1660</td>
</tr>
<tr>
<td>30</td>
<td>1640</td>
</tr>
<tr>
<td>36</td>
<td>1620</td>
</tr>
<tr>
<td>42</td>
<td>1600</td>
</tr>
<tr>
<td>48</td>
<td>1580</td>
</tr>
<tr>
<td>54</td>
<td>1560</td>
</tr>
<tr>
<td>60</td>
<td>1540</td>
</tr>
<tr>
<td>66</td>
<td>1520</td>
</tr>
<tr>
<td>72</td>
<td>1500</td>
</tr>
<tr>
<td>78</td>
<td>1480</td>
</tr>
<tr>
<td>84</td>
<td>1460</td>
</tr>
<tr>
<td>90</td>
<td>1440</td>
</tr>
<tr>
<td>96</td>
<td>1420</td>
</tr>
<tr>
<td>102</td>
<td>1400</td>
</tr>
<tr>
<td>108</td>
<td>1380</td>
</tr>
<tr>
<td>114</td>
<td>1360</td>
</tr>
<tr>
<td>120</td>
<td>1340</td>
</tr>
</tbody>
</table>

#### Track-based (BREG_TRACK)

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1740</td>
</tr>
<tr>
<td>6</td>
<td>1720</td>
</tr>
<tr>
<td>12</td>
<td>1700</td>
</tr>
<tr>
<td>18</td>
<td>1680</td>
</tr>
<tr>
<td>24</td>
<td>1660</td>
</tr>
<tr>
<td>30</td>
<td>1640</td>
</tr>
<tr>
<td>36</td>
<td>1620</td>
</tr>
<tr>
<td>42</td>
<td>1600</td>
</tr>
<tr>
<td>48</td>
<td>1580</td>
</tr>
<tr>
<td>54</td>
<td>1560</td>
</tr>
<tr>
<td>60</td>
<td>1540</td>
</tr>
<tr>
<td>66</td>
<td>1520</td>
</tr>
<tr>
<td>72</td>
<td>1500</td>
</tr>
<tr>
<td>78</td>
<td>1480</td>
</tr>
<tr>
<td>84</td>
<td>1460</td>
</tr>
<tr>
<td>90</td>
<td>1440</td>
</tr>
<tr>
<td>96</td>
<td>1420</td>
</tr>
<tr>
<td>102</td>
<td>1400</td>
</tr>
<tr>
<td>108</td>
<td>1380</td>
</tr>
<tr>
<td>114</td>
<td>1360</td>
</tr>
<tr>
<td>120</td>
<td>1340</td>
</tr>
</tbody>
</table>
Peak-based (BREG)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1592</td>
<td>0</td>
</tr>
<tr>
<td>1592</td>
<td>10</td>
</tr>
<tr>
<td>1592</td>
<td>20</td>
</tr>
<tr>
<td>1592</td>
<td>30</td>
</tr>
<tr>
<td>1592</td>
<td>40</td>
</tr>
<tr>
<td>1592</td>
<td>50</td>
</tr>
<tr>
<td>1592</td>
<td>60</td>
</tr>
<tr>
<td>1592</td>
<td>70</td>
</tr>
<tr>
<td>1592</td>
<td>80</td>
</tr>
<tr>
<td>1592</td>
<td>90</td>
</tr>
<tr>
<td>1592</td>
<td>100</td>
</tr>
<tr>
<td>1592</td>
<td>110</td>
</tr>
<tr>
<td>1592</td>
<td>120</td>
</tr>
<tr>
<td>1592</td>
<td>130</td>
</tr>
<tr>
<td>1592</td>
<td>140</td>
</tr>
<tr>
<td>1592</td>
<td>150</td>
</tr>
<tr>
<td>1592</td>
<td>160</td>
</tr>
<tr>
<td>1592</td>
<td>170</td>
</tr>
<tr>
<td>1592</td>
<td>180</td>
</tr>
<tr>
<td>1592</td>
<td>190</td>
</tr>
<tr>
<td>1592</td>
<td>200</td>
</tr>
<tr>
<td>1592</td>
<td>210</td>
</tr>
<tr>
<td>1592</td>
<td>220</td>
</tr>
<tr>
<td>1592</td>
<td>230</td>
</tr>
<tr>
<td>1592</td>
<td>240</td>
</tr>
<tr>
<td>1592</td>
<td>250</td>
</tr>
<tr>
<td>1592</td>
<td>260</td>
</tr>
<tr>
<td>1592</td>
<td>270</td>
</tr>
<tr>
<td>1592</td>
<td>280</td>
</tr>
<tr>
<td>1592</td>
<td>290</td>
</tr>
<tr>
<td>1592</td>
<td>300</td>
</tr>
<tr>
<td>1592</td>
<td>310</td>
</tr>
<tr>
<td>1592</td>
<td>320</td>
</tr>
<tr>
<td>1592</td>
<td>330</td>
</tr>
<tr>
<td>1592</td>
<td>340</td>
</tr>
<tr>
<td>1592</td>
<td>350</td>
</tr>
<tr>
<td>1592</td>
<td>360</td>
</tr>
<tr>
<td>1592</td>
<td>370</td>
</tr>
<tr>
<td>1592</td>
<td>380</td>
</tr>
<tr>
<td>1592</td>
<td>390</td>
</tr>
<tr>
<td>1592</td>
<td>400</td>
</tr>
<tr>
<td>1592</td>
<td>410</td>
</tr>
<tr>
<td>1592</td>
<td>420</td>
</tr>
</tbody>
</table>

Track-based (BREG_TRACK)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1590</td>
<td>0</td>
</tr>
<tr>
<td>1590</td>
<td>10</td>
</tr>
<tr>
<td>1590</td>
<td>20</td>
</tr>
<tr>
<td>1590</td>
<td>30</td>
</tr>
<tr>
<td>1590</td>
<td>40</td>
</tr>
<tr>
<td>1590</td>
<td>50</td>
</tr>
<tr>
<td>1590</td>
<td>60</td>
</tr>
<tr>
<td>1590</td>
<td>70</td>
</tr>
<tr>
<td>1590</td>
<td>80</td>
</tr>
<tr>
<td>1590</td>
<td>90</td>
</tr>
<tr>
<td>1590</td>
<td>100</td>
</tr>
<tr>
<td>1590</td>
<td>110</td>
</tr>
<tr>
<td>1590</td>
<td>120</td>
</tr>
<tr>
<td>1590</td>
<td>130</td>
</tr>
<tr>
<td>1590</td>
<td>140</td>
</tr>
<tr>
<td>1590</td>
<td>150</td>
</tr>
<tr>
<td>1590</td>
<td>160</td>
</tr>
<tr>
<td>1590</td>
<td>170</td>
</tr>
<tr>
<td>1590</td>
<td>180</td>
</tr>
<tr>
<td>1590</td>
<td>190</td>
</tr>
<tr>
<td>1590</td>
<td>200</td>
</tr>
<tr>
<td>1590</td>
<td>210</td>
</tr>
<tr>
<td>1590</td>
<td>220</td>
</tr>
<tr>
<td>1590</td>
<td>230</td>
</tr>
<tr>
<td>1590</td>
<td>240</td>
</tr>
<tr>
<td>1590</td>
<td>250</td>
</tr>
<tr>
<td>1590</td>
<td>260</td>
</tr>
<tr>
<td>1590</td>
<td>270</td>
</tr>
<tr>
<td>1590</td>
<td>280</td>
</tr>
<tr>
<td>1590</td>
<td>290</td>
</tr>
<tr>
<td>1590</td>
<td>300</td>
</tr>
<tr>
<td>1590</td>
<td>310</td>
</tr>
<tr>
<td>1590</td>
<td>320</td>
</tr>
<tr>
<td>1590</td>
<td>330</td>
</tr>
<tr>
<td>1590</td>
<td>340</td>
</tr>
<tr>
<td>1590</td>
<td>350</td>
</tr>
<tr>
<td>1590</td>
<td>360</td>
</tr>
<tr>
<td>1590</td>
<td>370</td>
</tr>
<tr>
<td>1590</td>
<td>380</td>
</tr>
<tr>
<td>1590</td>
<td>390</td>
</tr>
<tr>
<td>1590</td>
<td>400</td>
</tr>
<tr>
<td>1590</td>
<td>410</td>
</tr>
<tr>
<td>1590</td>
<td>420</td>
</tr>
</tbody>
</table>
### Appendix C: Full Results for Harmonicity Grouping

The table below presents the results for peak-based and track-based methods in terms of time (frames) and frequency (Hz).

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Peak-based (BREG)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Track-based (BREG_TRACK)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagrams illustrate the frequency vs. time plots for each method, showing the peaks and tracks over time.
Audio Source Separation… for Content-based Coding and Management
Appendix D

Full Results for Intersection-based Grouping

Table D-1 shows the 4-dimensional histogram used to compare the relative performance of the intersection-based grouping algorithm for all the combinations of amplitude, frequency and harmonicity threshold values tested. Table D-2 shows the 4-D plot of the mis-grouped tracks measure over the same ranges.

Table D-1: Histogram of relative performance for all combinations of Frequency, Amplitude and Harmonicity thresholds in the Intersection-based grouping method.
Amplitude = 25

Amplitude = 30

Amplitude = 35
Appendix D: Full Results for Intersection-based Grouping

Amplitude = 40

Amplitude = 45 - 100
Table D-2: Mis-grouped tracks as a function of frequency, harmonicity and amplitude grouping threshold for the intersection-based grouping algorithm.
The figures below show the actual grouping results corresponding to the best threshold combination of \( Th_F = 65; Th_H = 45; Th_A = 45 \).
Appendix D: Full Results for Intersection-based Grouping 321

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

![Graphs showing time and frequency data](image-url)
Melih, K. *Audio Source Separation…for Content-based Coding and Management*
Appendix E

Full Results for Distance-Vector-based Grouping

The table below presents the results of applying the distance-vector grouping method to all of the test files used in the experimental work. The two optimum sets of weighting \((w_{freq}, w_{amp}, w_{harm})\) are shown for comparison purposes.

<table>
<thead>
<tr>
<th>(25, 35, 40)</th>
<th>(35, 35, 30)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48</td>
<td>0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80</td>
</tr>
</tbody>
</table>

---

![Graphs showing frequency over time for different weighting sets](image-url)
Melih, K. Audio Source Separation…for Content-based Coding and Management

(25, 35, 40)

(35, 35, 30)

Time (Frames)

Frequency (Hz)
Appendix E: Full Results for Distance-Vector-based Grouping

(25, 35, 40) vs. (35, 35, 30)

Frequency vs. Time (Frames)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200-1400</td>
<td>0-10</td>
</tr>
<tr>
<td>1400-1600</td>
<td>10-20</td>
</tr>
<tr>
<td>1600-1800</td>
<td>20-30</td>
</tr>
<tr>
<td>1800-2000</td>
<td>30-40</td>
</tr>
<tr>
<td>2000-2200</td>
<td>40-50</td>
</tr>
</tbody>
</table>

Frequency vs. Time (Frames)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200-1400</td>
<td>0-10</td>
</tr>
<tr>
<td>1400-1600</td>
<td>10-20</td>
</tr>
<tr>
<td>1600-1800</td>
<td>20-30</td>
</tr>
<tr>
<td>1800-2000</td>
<td>30-40</td>
</tr>
<tr>
<td>2000-2200</td>
<td>40-50</td>
</tr>
</tbody>
</table>

Frequency vs. Time (Frames)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200-1400</td>
<td>0-10</td>
</tr>
<tr>
<td>1400-1600</td>
<td>10-20</td>
</tr>
<tr>
<td>1600-1800</td>
<td>20-30</td>
</tr>
<tr>
<td>1800-2000</td>
<td>30-40</td>
</tr>
<tr>
<td>2000-2200</td>
<td>40-50</td>
</tr>
</tbody>
</table>

Frequency vs. Time (Frames)

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Time (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200-1400</td>
<td>0-10</td>
</tr>
<tr>
<td>1400-1600</td>
<td>10-20</td>
</tr>
<tr>
<td>1600-1800</td>
<td>20-30</td>
</tr>
<tr>
<td>1800-2000</td>
<td>30-40</td>
</tr>
<tr>
<td>2000-2200</td>
<td>40-50</td>
</tr>
</tbody>
</table>
References


