COMBINED COLOUR AND CONTOUR REPRESENTATION USING ANTI-ALIASED HISTOGRAMS

Jolon Faichney and Ruben Gonzalez

School of Information Technology, Griffith University
PMB 50 Gold Coast Mail Centre, QLD 9726, Australia
{J.Faichney,R.Gonzalez}@gu.edu.au

ABSTRACT

Image databases require accurate colour image retrieval, minimal storage requirements, and fast query execution. Histograms have been used to compactly represent colour distributions providing low storage requirements and fast query execution. We propose the novel idea to also use histograms to represent contour distributions. Our contour distribution approach is shown to perform as well as an exhaustive contour comparison approach. In addition, *anti-aliased histograms* also known as *fuzzy histograms* are applied to reduce the storage requirements and query times by a factor of four. Colour and contour similarities are combined to provide more accurate query results than either individual technique.

1. INTRODUCTION

The ultimate goal of an image database is to satisfy a user's query through feature extraction and query execution. Feature extraction is often an off-line process and involves processing an image for useful features to query by. These features are indexed and stored in a feature database. Queries on the otherhand are performed on-line and involve the user interacting with the image database looking for images which satisfy their queries. Queries therefore must provide quick and accurate responses, whilst feature extraction, often being an off-line process, can be more complex and time consuming.

Researchers have identified many different types of features which can be used for image comparison such as colour, texture, shape, objects, and motion. Colour is often represented by its distribution using histograms, making it one of the simplest features to query by because of the low storage requirements and fast query execution. In this paper we propose the novel idea to also represent image contours by their distribution using histograms making them feasible in both storage requirements and query execution times. We also propose to apply fuzzy histograms to further reduce the required size of histograms.

Distribution representations are discussed in Section 2, experiments involving colour and contour histograms are presented in Sections 4 and 5, and further work and conclusions are discussed in Section 6.

2. DISTRIBUTION REPRESENTATIONS

Distribution representations have most commonly been used in image databases to represent colour. A distribution may be represented a number of different ways. In its simplest form the distribution can be represented by the mean or average. The mean, however, can be non-descriptive. For example, an image with blue sky, red sun, and green ground, might have an average colour of grey. Other statistical methods are usually used in combination with mean, such as standard deviation and kurtosis. These statistical descriptions can be generalised using moments where the first moment is the mean, the second standard deviation and so on. Stricker and Orengo [1] used moments to create a feature vector with only nine floating point values. Moments have the general form:

$$M_n = \sum \frac{(x - \bar{x})^n}{N} \quad (1)$$

where N is the number of data points and n is the order of the moment.

Instead of using statistical measures such as mean and standard deviation the distribution can be represented directly using frequency histograms. Histograms divide a range into a number of bins. A bin is incremented if for example a pixel falls within the bin's range. Histograms are simple to generate and compare.

Histograms can be treated as feature vectors, or points in a multi-dimensional space, allowing the Euclidean distance to be used to compare two histograms. One benefit of Euclidean distance is the ability to store histograms in multi-dimensional indexes. However, a technique which is more suited to histograms is *histogram intersection* [2] which sums the minimum bin values of all corresponding bins. Histogram intersection was designed to overcome a

Figure 1: (a, d) Two distributions. (b, e) Conventional histograms of distributions. (c, f) Fuzzy histograms and membership functions.

number of problems including background distractions, different viewpoints, occlusion, and varying image resolution.

$$d(I_i, I_j) = \frac{\sum_{k=1}^n \min(H_i(k), H_j(k))}{\sum_{k=1}^n H_j(k)} \quad (2)$$

The two important factors in image databases are feature vector size and query processing time. Both of these factors are affected by the number of dimensions and bins in a histogram. For some features such as colour the dimensions are fixed to three dimensions. The number of bins however is dependent on the resolution of the distribution required. More bins will improve the accuracy of distribution representation but will increase storage and processing requirements.

One problem which can occur with a limited number of bins is shown in Figure 1 where the distributions are offset by only a small amount but enough to place most of the pixels into another bin. Most histogram comparison techniques only compare corresponding bins. So even though the two images may only vary by a small amount in colour, the histogram comparison will report that the images are very different.

In this paper we propose a technique called *anti-aliased histograms* or *fuzzy histograms* which more accurately represent the distribution by distributing a pixel's increment over more than one bin. In signal processing this is known as anti-aliasing. It also can be viewed as a fuzzy membership where a pixel no longer belongs to just one bin but partially belongs to multiple bins. The increment becomes the result of a fuzzy membership function. A fuzzy membership function also needs to know the centre of a bin in addition to its boundaries. The membership function used in this paper

is a linear function which decreases from a bin's centre to adjacent bin centres (Figure 1(c) and (d)). The fuzzy increment can be extended to multiple dimensions by calculating the increment for a bin in each dimension and multiplying the results together for the final bin increment.

The result of applying the fuzzy histogram to the first distribution in Figure 1(a) can be seen in Figure 1(c) and is much more similar to the fuzzy histogram of Figure 1(f). The fuzzy histogram is much more descriptive of the distribution and still compatible with existing histogram comparison techniques resulting in better query results.

In this paper we apply the fuzzy histogram technique to image colour distribution in Section 4. We also present the novel idea of representing contours, which include shape and texture information, in histograms as well, reducing the storage requirements of contours and query processing times. Fuzzy histograms are also applied to the contour distributions and the results are presented in Section 5 along with a combined colour and contour histogram.

3. EXPERIMENTS

Experiments were performed on a real world photo database [3]. The results are shown in Figure 2 and involve a search for a car image. The database contains three photos of the car image taken at different distances and with different camera settings affecting the size of objects, warping, and overall colour distribution. The query provides a good evaluation of colour retrieval performance as well as contour performance. The results are discussed in the next two sections.

4. COLOUR DISTRIBUTION

Colour histograms have been used extensively in image retrieval [1, 2, 4]. One major factor in colour query performance is the selection of colour model. RGB is the simplest because images are often already in this format once loaded into a memory buffer. RGB however is not perceptually uniform. Opponent colour models such as YUV and XYZ are able to separate the luminance component from the chrominance component modelling the signals that leave the retina. Conversion from RGB to an opponent colour model can be relatively simple however opponent colour models have not shown to provide better performance than RGB [2]. The closest colour models to human perception are those that represent colour as hue, luminance, and saturation. One of the most perceptually uniform is the HVC colour model [5], however conversion from RGB to HVC can be complex and non-invertible. A simpler representation is HSV and has been used by many researchers to represent image colour [1, 6].

We chose to use the HSV colour model and histogram intersection to compare image database retrieval performance using different numbers of histogram bins and fuzzy histograms. The results are shown in Figures 2 (a)-(i). The numbers following each description are the number of bins for each dimension. An 'F' indicates that fuzzy histograms were used. The results show that performance improves as the number of bins is increased in each dimension.

The effects of fuzzy histogram generation are most significant when the number of bins are low. For example, Figure 2 (b) gives better performance than Figure 2 (c) even though only half the number of bins are used. The same is true for Figures 2 (h) and (i). Using fuzzy histograms reduces the storage requirements and increases the query performance of colour retrieval by a factor of 2.

5. CONTOUR DISTRIBUTION

Contour information is able to represent both texture and shape information. In image databases texture information is generally represented by edge information [7], frequency decompositions such as FFT [8] or wavelet [9], or statistical dependencies such as SAR [10]. Shape on the other hand is generally represented by fourier descriptors [11] or eigenmodes [12] of extracted contours. Our proposed approach is to use extracted contours to represent both texture and shape.

In this paper we present a contour comparison technique based on accurate contour extraction which provides fast query execution and low storage requirements through the use of histograms. We compare contour histograms with an exhaustive contour comparison technique and also apply fuzzy histograms.

Our contour extraction technique [13] is very accurate involving multi-staged filtering using 12 orientations of Gabor-like filters. The outputs are thinned across neighbouring pixels as well as neighbouring orientations providing single pixel thick contours. Contour following will only connect pixels of neighbouring orientations, ie. no more than 15° deviation, allowing curves to be represented but stopping sharply at corners. The output from this stage is a long list of all of the contours in an image including individual contour points, orientations, and strengths. This information is too large to store in a database (around 2 Mb per image) so it is summarised to only the information required for comparison.

The features selected for summarisation are contour position, length, orientation, and curvature. The contour position is simply the average x and y positions of each contour point. Contour length is the number of contour points and contour orientation is the average orientation of all of the contour points. The curvature is the average absolute deviation of each point's orientation from the average orientation

already calculated.

Even with contour summaries the storage requirements are still large, around 100k per image. An exhaustive contour comparison technique was implemented as a benchmark for comparison with histogram techniques. The exhaustive approach compares every contour in each image and sums the similarities [13]. The technique is slow requiring approximately 1 second to compare two images which would result in very slow query times for large databases. The results of this technique are shown in Figure 2(q).

Our proposed contour representation technique is to represent the distribution of the contours using a histogram rather than the individual contours themselves. The histogram consists of five dimensions: x , y , length, orientation, and curvature. Rather than increment a bin for each contour, bins are incremented by the number of points in a contour.

Colour histograms are easily normalised by the number of pixels in an image. To provide the best results the contour histograms were normalised by the total number of contour points.

The contour experiments evaluate the impact of contour position information and the impact of increased bins for orientation and length. Contour histograms were generated with and without position information and also with 4 and 8 orientation bins. Two bins were for the curvature dimension indicating that a contour is straight or curved. Two and four bins were used for length while two bins were used for each position dimension dividing an image into four quadrants. Figure 2(m) shows that the presence of position information provides more accurate results. The results in Figure 2(o) and (p) show that eight orientation bins and four length bins provide slightly better than four and two. The best combination was 8, 4, 2, 2, and 2 bins for the orientation, length, curvature, x , and y dimensions resulting in 256 bins. As can be seen the histogram technique which uses much less storage space and provides faster query execution provides similar performance to the exhaustive contour comparison method in Figure 2(q).

Applying the fuzzy histogram generation technique the 4, 2, 2, 2, 2 histogram is able to match the 8, 4, 2, 2, 2 technique using four times fewer bins.

When the colour and contour histograms are combined even fewer bins can be used for better results than either individual technique. The 3, 2, 2 fuzzy HSV histogram is combined with the 4, 2, 2 fuzzy contour histogram providing results (Figure 2(r)) as good as the contour histogram in Figure 2 (o) but using only 28 bins instead of 256. The histograms are combined by calculating the similarity for each individual histogram and multiplying the similarities together.

$$S = S_{\text{colour}} \times S_{\text{contour}} \quad (3)$$

Figure 2: Results of queries with the car image on the sample image database.

6. CONCLUSIONS AND FUTURE WORK

Our results show that representing contours as distributions using histograms can perform just as well as an exhaustive contour comparison technique with the benefit of greatly reduced storage requirements and execution times. Fuzzy histograms are also able to reduce colour histogram sizes by a factor of two and contour histograms by a factor of four. However, with larger numbers of bins fuzzy histograms don't provide a significant improvement over conventional histograms. Combining colour and contour allows even less bins to be used providing very accurate results.

Future work will involve applying fuzzy histograms to other features such as regions and motion. We are also developing a cluster-based 3D visualisation environment which takes advantage of the low complexity comparisons made possible with histograms.

7. REFERENCES

- [1] Markus Stricker and Markus Orengo, "Similarity of color images," in *Proceedings of Storage and Retrieval for Image and Video Databases III*, 1995, vol. 2420, pp. 381–392.
- [2] Michael J. Swain and Dana H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [3] <http://www.int.gu.edu.au/~j.faichney/database/>.
- [4] W. Niblack, R. Barber, W. Equitzand, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: Querying image by content using color, texture, and shape," in *SPIE Proceedings Storage and Retrieval for Image and Video Databases*, 1993, vol. 1908, pp. 173–187.
- [5] Makoto Miyahara and Yasuhiro Yoshida, "Mathematical transform of (R,G,B) color data to Munsell (H,V,C) color data," in *SPIE Visual Communications and Image Processing*, 1988, vol. 1001, pp. 650–657.
- [6] John R. Smith and Shih-Fu Chang, "Automated image retrieval using color and texture," Tech. Rep. 414-95-20, Columbia University, July 1995.
- [7] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, June 1978.
- [8] Rosalind W. Picard and Fang Liu, "A new Wold ordering for image similarity," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1994, pp. 129–132.
- [9] John R. Smith and Shih-Fu Chang, "Quad-tree segmentation for texture-based image query," in *ACM Multimedia 94*, 1994, pp. 279–286.
- [10] Jiangchang Mao and Anil K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.
- [11] Klaus Arbter, Wesley E. Snyder, Hans Burkhardt, and Gerd Hirzinger, "Application of affine-invariant fourier descriptors to recognition of 3-D objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 640–647, July 1990.
- [12] Stan Scarloff and Alex P. Pentland, "Modal matching for correspondance and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545–561, June 1995.
- [13] Jolon Faichney, *Content-based Retrieval of Digital Video*, Ph.D. thesis, Griffith University, Australia, 2002.