Techniques for efficient and effective transformed image identification

Author

Awrangjeb, Mohammad, Lu, Guojun

Published

2009

Journal Title

Journal of Visual Communication and Image Representation

Version

Accepted Manuscript (AM)

DOI

10.1016/j.jvcir.2009.07.004

Techniques for Efficient and Effective Transformed Image Identification

Mohammad Awrangjeb[*,a], Guojun Lu[b]

[a] *Cooperative Research Center for Spatial Information, The University of Melbourne*
723 Swanston St, Carlton Vic 3053, Australia
Phone: +61 3 8344 9182, Fax: +61 3 9349 5185
[b] *Gippsland School of IT, Monash University, Churchill Vic 3842, Australia*
Phone: +61 3 5122 6857, Fax: +61 3 5122 6879

Abstract

In many applications, one common problem is to identify images which may have undergone unknown transformations. We define this problem as *transformed image identification* (TII), where the goal is to identify geometrically transformed and signal processed images for a given test image. The TII consists of three main stages – *feature detection*, *feature representation*, and *feature matching*. The TII approach by Lowe [1] is one of the most promising techniques. However, both of its feature detection and matching stages are expensive, because a large number of feature-points are detected in the image scale-space and each feature-point is described using a high dimensional vector. In this paper, we explore the use of different techniques in each of the three TII stages and propose a number of promising TII approaches by combining different techniques of the three stages. Our experimental results reveal that the proposed approaches not only improve the computational efficiency and decrease the storage requirement significantly, but also increase the transformed image identification accuracy (robustness).

Key words:
Feature detection, Feature representation, Feature matching, Transformed image identification

[*]Corresponding author
Email addresses: mawr@unimelb.edu.au (Mohammad Awrangjeb), Guojun.Lu@infotech.monash.edu.au (Guojun Lu)

1. Introduction

In many applications, such as image copyright protection [2], face recognition [3] and object recognition [1], a common problem is to identify images which may have undergone unknown transformations. We define this common problem as the *transformed image identification* (TII), where the goal is to identify the geometric transformed and the signal processed images for a given image. Therefore, the TII is different from conventional *content-based image retrieval* (CBIR) [4], where all images having the same or similar features, e.g., similar colors, are considered relevant to each other. Li et al. [5] proposed a user feedback-based CBIR technique using learning machines for classification [6].

The TII is also called *near-duplicate image identification* in the literature [7, 3, 8, 9, 10] and consists of three main stages [11]: *feature detection*, *feature representation*, and *feature matching*. In the *feature detection* stage, a set of features, e.g., corners, blobs, T-junctions, are detected. The most valuable property of a feature detector is *repeatability*, i.e., whether it reliably finds the same feature points after the image has undergone different transformations. In the *feature representation* stage, each detected feature point is represented by a feature vector calculated possibly from its neighborhood. In the *feature matching* stage, the feature vectors of the test image and the stored images are compared to identify transformed images for the test image. The matching is often based on a distance, e.g., the Euclidean distance [11], between the feature vectors.

Low's approach [1] has demonstrated its superior performance in identifying transformed images over many other approaches [12, 13, 14]. However, both of its detection and matching stages are expensive, because a large number of keypoints are detected in the scale-space using the difference-of-Gaussian (DoG) filter and each keypoint is described using a 128-dimensional vector known as SIFT (scale invariant feature transform) descriptor. Further research has been carried out in the literature for the dimensionality reduction of the SIFT descriptor, but with the expense of loss of robustness [11, 13, 12]. In this paper, we explore the use of different techniques in each of the TII stages. In stage one, we present two possible solutions for feature-point reduction. First is to down scale the image before the DoG keypoint detection and second is to use corners (instead of DoG keypoints) which are visually significant, more robust, and much smaller in number than the DoG keypoints. In stage two, we explore the use of corner curvature as well as to

SIFT descriptors. In stage three, feature point matching techniques based on the geometric point matching technique [15] is used in addition to the nearest-neighbor-distance-ratio based matching technique [1]. Consequently, we propose a number of TII approaches in this paper by combining different techniques at different stages and will discuss them in Section 4. The two feature-point reduction solutions combined with the SIFT descriptors and our previously proposed feature-point matching technique [15] not only improve the computational efficiency and decrease the storage requirement significantly, but also increase the transformed image identification accuracy (robustness).

The organization of this paper is as follows. In Section 2, we briefly review the existing feature detection, representation, and matching techniques and highlight key contributions of the paper. Section 3 compares three existing feature detectors in terms of robustness and detection time. In Section 4, we present the proposed TII approaches. In Section 5, we discuss the performance study and finally, in Section 6 we conclude the paper.

2. Related Work and Contributions of This Paper

In this section, we first briefly present some relevant existing work in three stages of the TII (feature point detection, feature representation and feature matching). Then we briefly discuss existing promising TII approaches and present the contribution of this paper.

2.1. Feature Detection

A large number of corner and interest-point detectors have been proposed in the literature [1, 16, 14, 17, 18, 19, 20, 21, 22, 23]. While corner detectors detect image spatial locations where edge segments make significant angles, interest-point detectors not only detect corners, but also image locations that have large gradients in all directions at a predetermined scale [1].

All corner and interest-point detectors can be broadly classified into two groups: single-scale detectors [22, 21, 23] and multi-scale detectors [1, 16, 14, 17, 18, 19, 20]. Single-scale detectors work well if the image has similar size features, but ineffective otherwise; because either fine or coarse scale feature is poorly detected, but images may contain both kinds of features. To improve the effectiveness of the detection stage, multi-scale detectors have been proposed.

Corner and interest-point detectors can also be categorized into three groups: intensity-based [1, 16, 14, 22], contour-based [17, 18, 19, 20, 21], and model or template-based [23] methods. Intensity-based methods estimate a measure which is intended to indicate the presence of an interest-point directly from the image pixel values. Contour-based methods first obtain planar curves using some edge detector and then search for the curvature maxima along those curves. Model or template-based methods find corners by fitting the image signal into a predefined model.

The main drawback with the model-based detectors is that the corner in natural images cannot be approximated by a model of a perfect corner, as it can take any form of the bidirectional signal change [14]. Moreover, they are computationally too expensive [23] and are not used for general purpose [24]. This paper will focus on the intensity and contour-based detectors only.

2.1.1. Intensity-based Detectors

Probably the most widely used detector is the Harris interest-point detector [22] which is based on the eigen values of the second-moment matrix. However, Harris points are not scale-invariant [11]. Lindeberg [16] introduced the concept of automatic scale selection which allows detecting interest points in an image, each with their own characteristic scale. Mikolajczyk and Schmid [14] refined this technique by creating robust and scale invariant features with high repeatability. They used a scale-adapted Harris measure or the determinant of the Hessian matrix to select the location, and the Laplacian to select the scale. Lowe's [1] approximation of the Laplacian of Gaussian using the DoG filter speeded up the feature detection stage significantly. The recently proposed fast-Hessian detector in the SURF (speeded up robust features) detector-descriptor scheme [11] used a basic approximation of the Hessian matrix and relied on the integral images to reduce the computational cost.

2.1.2. Contour-based Detectors

The CSS (curvature scale-space) detector in [19] is one of the earlier contour-based multi-scale detectors. It detected corners at a high scale and tracked them through multiple lower scales in order to improve localization. Since different curves require different smoothing-scales and there may be different sizes of corners on the same curve, this detector is highly sensitive to the use of a single corner detection scale and a single fixed curvature-threshold. He & Yung [21] improved this detector by using the adaptive

curvature-threshold and the dynamic region-of-support on both sides of each curvature extremum point. Zhang et al. [20] further improved it by introducing the idea of curvature-product. In terms of the curvature product, the strong corners become more distinguishable from the weak and false corners. Awrangjeb et al. [17] proposed another improvement (known as affine resilient CSS - ARCSS - detector [25]) by selecting three corner detection scales based on the curve's affine-length.

Recently Awrangjeb and Lu [18] pointed out the two main problems of the above CSS-based detectors. First, the CSS curvature estimation technique is highly sensitive to the local variation and noise on the curve. Second, the CSS corner detection technique requires appropriate Gaussian smoothing-scale selection which is a difficult task. To overcome these two problems, they proposed a new corner detector [18] based on the *chord-to-point distance accumulation* (CPDA) for the discrete curvature estimation [26], which is less sensitive to the local variation and noise on the curve and does not require appropriate Gaussian smoothing-scale selection.

2.2. Feature Representation

In order to facilitate feature-point matching in any subsequent application, each feature-point must be represented with some of its associated information. The more the representation is distinctive, the less the number of false candidate matches will be obtained in the matching stage. The feature representation is also known as the feature descriptor in the literature.

There are two different types of representations found in the literature: geometric descriptors and local descriptors.

2.2.1. Geometric Descriptors

This type of representation [27, 28] is purely geometric, where each corner or feature-point is represented using its curvature, angle, and distances from neighbor corner-points. Zhou et al. [27] used the angles of Delaunay triangles which are formed among the Harris interest-points [22]. These angles are invariant to image translation, rotation, and uniform scaling. Huttenlocher used [28] distance ratios defined by the quadruples of the feature-points. The distance ratios are invariant to affine transformations.

Awrangjeb and Lu [15] used the curvature descriptors for their proposed *geometric point matching* (GPM) technique. In fact, their proposed ARCSS [25] and CPDA [18] detectors provide various information available for later use. For each corner, these detectors provide its position, absolute curvature

value, angle with its two neighboring corners or with the endpoints when the necessary number of neighbor corners are not found and the affine-lengths between neighboring corners on the same curve. All of the above information can be used collectively as the 'curvature descriptor' to represent the corner.

Though the geometric representation is easy to design and requires quite small amount of storage per feature-point, the representation is not unique. As a consequence, either the true correspondences are missed or a huge number of false correspondences are obtained between two images and the matching procedure becomes prohibitively expensive. This is why, the use of geometric representation is quite limited in the literature [29].

2.2.2. Local Descriptors

This type of representation [1, 13, 11, 12] is based on the pixel values in a specific neighborhood around each feature-point. They are harder to design, but more distinctive than the geometric representations discussed above.

The SIFT-based descriptors, originally proposed by Lowe in [1], have been described as the best among the different types of feature descriptors [12]. The original SIFT descriptor [1] captures a substantial amount of information about the spatial intensity patterns around a feature point. For each descriptor it computes a 3-D histogram of gradient location and orientation, where the location is quantized into a 4×4 location grid and for each location grid (total 16) the gradient angle is quantized into 8 orientations. The resulting descriptor is a 128-dimensional vector (8 orientation bins for each of 4×4 location bins). This vector is robust to small deformations or localization errors. To obtain illumination invariance, the descriptor is normalized by the square root of the sum of squared components.

Further research has been carried out in the literature either to reduce the descriptor's dimension [11, 13] or to make it more distinctive [13, 12]. The PCA-SIFT (principal component analysis-SIFT) [13] descriptor encodes the salient aspects of the image gradient in the feature point's neighborhood; however, instead of using SIFT's smoothed weighted histograms, it applies PCA to the normalized gradient patch to reduce the descriptor's dimension. The SURF descriptor, on the other hand, describes a distribution of Haar-wavelet responses within the interest point neighborhood and uses the integral images to reduce the computational time. Though the lower dimensional PCA-SIFT [13] and SURF [11] are helpful for fast feature matching, PCA-SIFT was proved to be less robust than SIFT under affine transformations [12] and SURF was designed to handle rotation and scale attacks

only as a compromise between feature complexity and robustness [11]. The GLOH (gradient location-oriented histogram) [12] is another variant of SIFT. It considers more spatial regions for the histograms and reduces the higher dimensionality of the descriptor using PCA. Though it was proved to be more distinctive and robust than the original SIFT [12], it is computationally more expensive [11]. Consequently, in spite of the above refinement versions, the original SIFT is still the most popular.

The above local descriptors are invariant to image rotation, scaling, and translations. The affine invariant local descriptors which are based on the iterative affine-region detection around the feature-points are computationally very expensive [14].

2.3. Feature Matching

Mikolajczyk and Schmid [12] evaluated three feature matching techniques. In *threshold*-based matching, two features are matched if the distance between their descriptors is below a predefined threshold. A feature may have several matches in this strategy. In *nearest-neighbor*-based matching, two features F_1 and F_2 with descriptors D_1 and D_2 respectively are matched if the descriptor D_2 is the nearest neighbor of D_1 and their distance is below a threshold. With this approach a feature should have at most one match. In *nearest-neighbor-distance-ratio*-based matching (NNDR matching), which is similar to nearest-neighbor-based matching, the threshold is applied to the distance ratio between the first and second nearest neighbor matches, i.e., two features F_1 and F_2 with descriptors D_1 and D_2 respectively are matched if $|D_1 - D_2|/|D_1 - D_3| < t$, where D_3 is the descriptor of the second nearest neighbor match F_3 of F_1. In this approach, a feature has also at most one match.

The worst case (as well as the average case and the best case) running time of these algorithms is $O(mn)$, where m and n are the numbers of features in two images. Note that using these matching techniques a repeated feature may be missed or there may be some false positive matches.

The GPM (Geometric Point Matching) technique by Awrangjeb and Lu [15, 25] is shown in Fig. 1. Let S_1 and S_2 be two sets of feature-points (corners) from two given images where each feature-point is described with a descriptor. The initial candidate matches are obtained using the *threshold*-based matching discussed above. Then for each combination of three candidate matches, it estimates the affine transformation parameters between the images using an iterative procedure. In each iteration, it transforms all

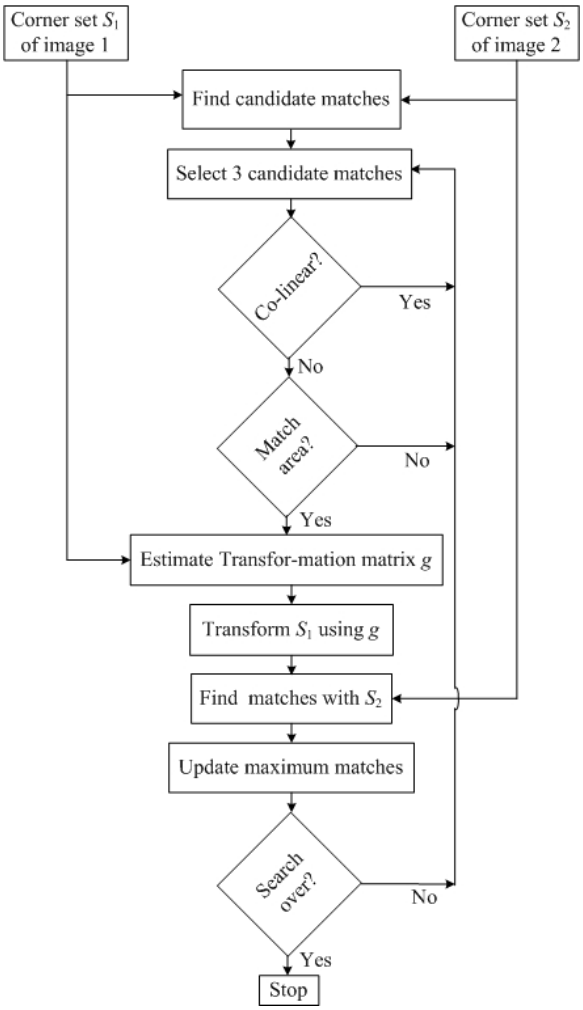

Figure 1: Block diagram for the geometric point matching technique.

the feature-points in one image using the estimated parameters and finds matches in the other image. The algorithm tracks the set of parameters that offer the highest number of matches over all the iterations. Consequently, the GPM technique works in following three steps:

- *Step* 1: Find the candidate corner matches by matching the descriptors in S_1 and S_2.
- *Step* 2: If three candidate matching corners are non-collinear on each image and the ratio of the areas of the corresponding triangles in both

of the images is within a specific range, estimate the transformation matrix g between these triangles.

- *Step* 3: Transform all the corners in S_1 using g and determine the number of corner matches with S_2 allowing a maximum localization error of 3-pixels.

This above algorithm, though offers better matching performance, is more expensive than any of the above three matching techniques. However, in practice, if the feature representation is very distinctive, it costs close to the above techniques, i.e., $O(mn)$ [25, 29].

Note that though the idea of the GPM technique [15] is similar to that of the RANSAC (Random Sample Consensus) [30], the former differs from the later in the following aspects. Firstly, the GPM obtains the candidate matches using the feature descriptors. Since the standard RANSAC does not use such local matching process, it performs poorly when the proportion of the outliers increases above 50% [1]. Secondly, the RANSAC sets a termination condition, e.g., terminate the algorithm once a specific number of iterations are executed. We conjecture that if the feature descriptors are distinctive, then the number of false candidate matches will be low and we do not need to set any termination condition. Finally, the RANSAC requires a careful setup of a number of algorithmic parameters to ensure its good performance. The GPM requires only a few parameter setups based on the range of geometric transformations we want to consider.

2.4. Promising TII Approaches

In this section we briefly present promising TII approaches and their main problem.

Zhang and Chang [9] formulated a stochastic attributed relational graph matching for TII. Corners were considered as vertices and their positions, corresponding pixel values and wavelet features were used to represent (attribute) the vertices. However, the graph matching method involves a complex process of stochastic belief propagation and thus identification speed is slow [8]. Lowe [1] used DoG keypoints, SIFT descriptors and NNDR matching technique. Ke et al. [10] and Zhao et al. [8] used PCA-SIFT descriptors to represent DoG keypoints. While Ke et al. [10] used the RANSAC algorithm to match them; Zhao et al. [8] used a bipartite graph matching algorithm. Xu et al. [7] used SIFT descriptors around DoG keypoints to apply a multi-level (block-by-block) image matching method.

However, because of the large number of interest points in images (possibly exceeding 1000), matching based on interest points is extremely time consuming and inappropriate for many applications.

Since Lowe's TII approach has demonstrated its superior performance [12], we compare it with our proposed TII approaches.

2.5. Contributions of This Paper

All three stages of a TII system are important for its performance. In Lowe's approach [1], DoG keypoints, SIFT feature vector and NNDR are used in the three stages respectively. We propose to use alternative techniques in each of the three stages.

In stage one, since the number of DoG keypoints is high, we propose two solutions to reduce the number of keypoints. In the first solution, the detection stage becomes faster and the number of detected DoG keypoints is reduced largely, which also speeds up the later stages. In the second solution, we propose using corners instead of DoG keypoints. In general, corners offer the following advantages over keypoints:

- Corners are visually distinguishable and more robust than their keypoints counterparts.
- In an image, the number of corners is much lower than the number of keypoints.
- Corners can be ranked based on their strength like the curvature value or the number of corners can be controlled by changing the detection thresholds. Therefore, a particular number of strong corners can be selected based on the application and it gives further reduction in computational cost during matching. In contrast, it is very hard to rank the DoG keypoints.
- Corner detection requires less time than keypoint detection in the scale-space.

In stage two, corners can be represented with the curvature descriptors as well as with the SIFT feature vectors.

In stage three, a matching technique has to be used to obtain all the true matches between images. The NNDR matching technique described in Section 2.3 matches feature descriptors between two images but does not

consider the underlining geometric relations between them. Therefore, it may miss some true matches or find some false matches. On the other hand, the GPM technique not only matches feature descriptors between images but also considers their geometric relations. Consequently, the GPM provides better matching performance than the NNDR matching [29].

The three main contributions of this paper are as follows:

1. We compare corners with DoG and scaled-DoG keypoints in Section 3 and through experimentation we show that corners perform better than their keypoints counterparts in terms of both repeatability and localization error.
2. In Section 4, we propose five different TII approaches by combining different techniques in different stages.
3. Through experimentation and analysis we show that the two feature-point reduction solutions combined with the SIFT descriptors and the GPM technique not only improve the computational efficiency and decrease the storage requirement, but also improve the TII accuracy (see Section 5).

3. Comparison of Feature Detectors

What feature detectors and what type of feature-points to use in the stage one of TII have a significant effect on both of the efficiency and effectiveness of a TII system. In this section, we briefly describe three feature detectors – namely, ARCSS corner detector [17, 25], CPDA corner detector [18], and DoG keypoint detector [1] and present a comparative performance study of these three detectors. We compare the detectors in terms of robustness (repeatability and localization error) and efficiency (detection time). We will use these three detectors in our proposed TII approaches later presented in Section 4.

3.1. ARCSS Detector

The ARCSS corner detector [17, 25] extracts edges in the gray-scale image using the Canny edge detector [31]. It then parameterizes each curve (edge) with the affine-length. In order to eliminate noise, it convolves each parameterized curve using the Gaussian kernel in one of three medium scales based on the curve's affine-length. Thereafter, it calculates absolute curvature value on each point of the smoothed curves and considers curvature

maxima points as candidate corners. Both weak and false corners are removed using the appropriate thresholds. Finally, corners are tracked down to the finest scale to improve localization.

The experimental results in [17, 25] showed that in geometric transformations the ARCSS detector outperformed existing CSS detector [19], which had outperformed many other detectors including the Harris interest-point detector [22].

3.2. CPDA Detector

The CPDA corner detector [18] first extracts planar curves from the edge image detected by the Canny edge detector [31]. Each curve is then smoothed with a small width Gaussian kernel in order to remove quantization noise and trivial details. In order to make strong and weak corners more distinguishable, we first use three chords of different lengths to estimate three normalized discrete curvature values on each point of the smoothed curve. Then we multiply the normalized curvatures to obtain the curvature product (a single estimated curvature) at each point. The maxima of the absolute curvature products along the smoothed curve are then obtained as candidate corners. Finally, it follows a two-step refinement process that removes weak and false corners using thresholds.

The experimental results in [18] showed that the CPDA detector outperformed many contour-based detectors including CSS [19], multi-scale CSS [20], and adaptive threshold-based CSS [21] detectors.

3.3. DoG Keypoint Detector

The DoG keypoint detector [1] first blurs the input image with a small width Gaussian filter to prevent aliasing and to increase the stability of the keypoints. It then builds the Gaussian pyramid by repeatedly smoothing the blurred image. The difference-of-gaussian (DoG) pyramid is built by subtracting the adjacent Gaussian smoothed images in the same octave of the Gaussian pyramid. Finally, they obtain the local extrema positions in the DoG pyramid as keypoints. When an extrema is found, two tests are applied before labeling it as a keypoint. First, it must have sufficient contrast and second, it should not be an edge point.

In one of our solutions, we scale down the original images before DoG keypoint detection. We name this modified detector as 'scale-DoG detector' in the rest of the paper.

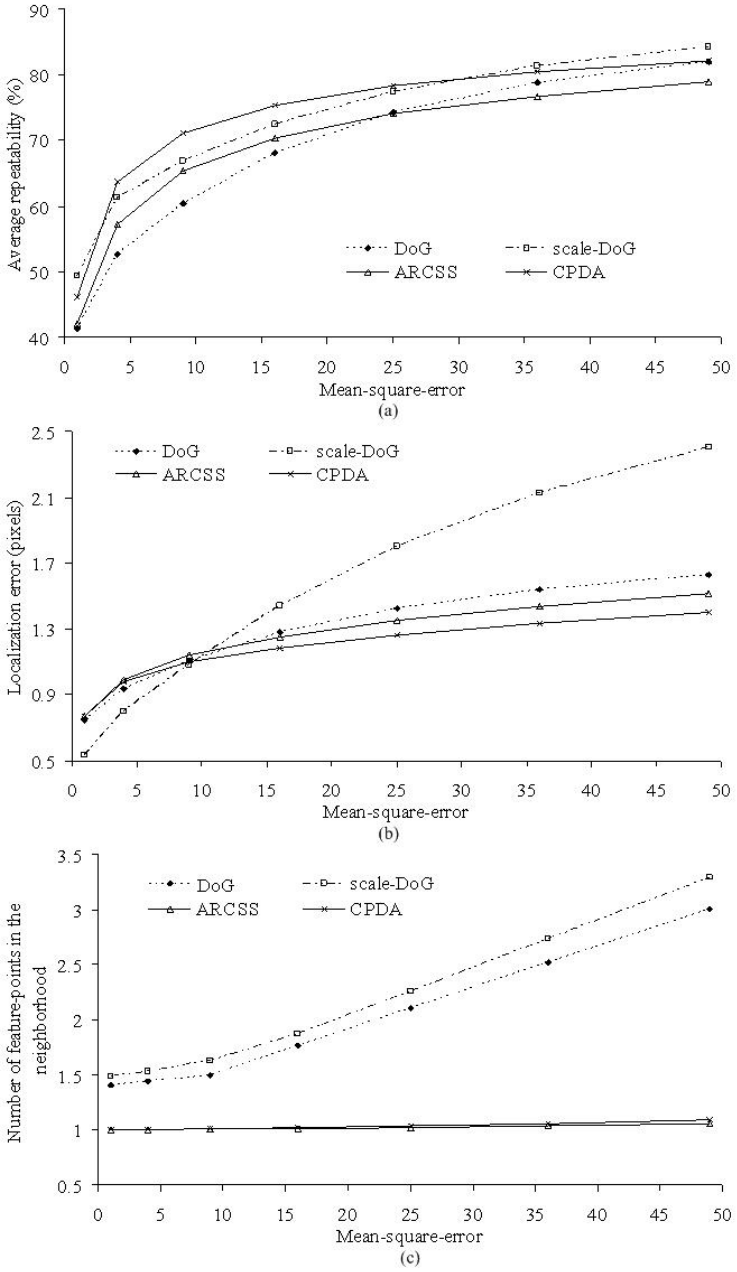

Figure 2: Comparing Detector's Performance: (a) average repeatability, (b) localization error, and (c) number of feature-points in the neighborhood.

Table 1: Performance summary of different detectors.

Properties	**DoG**	**scale-DoG**	**ARCSS**	**CPDA**
Feature-points in images	586	82	30	41
Feature detection time (seconds)	2.71	0.19	0.17	2.6
Average repeatability (%)	65.36	70.47	66.40	71.04
Localization error (pixels)	1.24	1.46	1.21	1.15
Feature-points in (transformed) neighborhood	1.93	2.12	1.06	1.06

3.4. Performance Comparisons

We used a database of 23 original images and their more than 8,500 test (transformed and signal processed, see [25, 29] for details of this database) images to evaluate the performance of the detectors. In the case of the scale-DoG detector, the original (512×512) images were down scaled to (128×128). The test images were then generated from the scaled-down images. In order to find repeated corners between the original and transformed images, we transformed the feature-points detected in the original image using the known transformation matrix and then found their repetitions with the feature-points detected in the test image. In this case, we allowed a *mean-square-error* (MSE) of e square pixels, which means a feature-point in the original image was considered as repeated if at least one feature-point was found in its (transformed) neighborhood in the test image when the maximum distance was e square pixels. The nearest feature-point in the neighborhood was considered as repeated.

We evaluated the performance in terms of *average repeatability* and *localization error* [25, 29]. *Repeatability* indicates how stable the detected feature-points are under different geometric transformations and signal processing attacks. The *average repeatability* is defined as the percentage of the total observed feature-points repeated between the original and test images:

$$R_{avg} = \frac{100 \times N_r}{2} \left(\frac{1}{N_0} + \frac{1}{N_t} \right), \tag{1}$$

where N_0 and N_t are the number of feature-points in the original and test images, respectively, and N_r is the number of repeated feature-points between them.

Localization error shows how accurately a detected feature-point is localized by the detector. A lower localization error indicates a better accuracy.

It is measured using the *root-mean-square-error* (RMSE) of the detected feature-points:

$$L_e = \sqrt{\frac{1}{N_r}\sum_{i=1}^{N_r}[(x_{oi} - x_{ti})^2 + (y_{oi} - y_{ti})^2]}, \tag{2}$$

where (x_{oi}, y_{oi}) and (x_{ti}, y_{ti}) are the positions of the i-th repeated feature-point in the original and test images respectively.

We also calculated the number of detected feature-points in the original images, the feature detection time, and the number of detected feature-points in the neighborhood (in the test image) for each repeated feature-point in the original image [29]. A low number of feature-points is essential for many applications like image copyright protection [2], fast image registration [15], and mobile robot vision [32]. A low feature detection time is always important. A larger number of detected feature-points in the neighborhood of a repeated feature-point indicates a higher chance of mismatching during image matching.

Fig. 2 shows the detail comparative performance and Table 1 presents the performance summary (average results). The scale-DoG detector offered the higher average repeatability than the original DoG detector. However, the scale-DoG detector offered the higher localization error with the increase of the neighborhood size. The ARCSS and CPDA detectors showed higher average repeatability than the DoG and scale-DoG detectors respectively when the neighborhood size was small. The opposite scenario was observed when the neighborhood size was large. The reason is, in the case of the DoG and scale-DoG detectors the detected keypoints are very close to each other. Fig. 2(c) shows that for these two detectors the number of keypoints in the transformed image is very high. Consequently, a close but different keypoint could be chosen as repeated offering high repeatability. Moreover, the DoG and scale-DoG detectors suffered from high localization error as shown in Fig. 2(b).

From the results in Table 1 we can say that the ARCSS detector was the fastest and it detected the lowest number of feature-points among the four. On the other hand, the CPDA detector offered the highest average repeatability and the lowest localization error. However, it was slower than the ARCSS and scale-DoG detectors. In the next section we will use all these four feature detectors to compare the performance of different TII approaches.

Table 2: Proposed transformed image identification approaches.

Approaches	**Detectors**	**Descriptors**	**Matching**
Approach 1	ARCSS	curvature	GPM
Approach 2	CPDA	curvature	GPM
Approach 3	scale-DoG	SIFT	GPM
Approach 4	ARCSS	SIFT	GPM
Approach 5	CPDA	SIFT	GPM

4. Proposed Transformed Image Identification Approaches

For identifying the transformed images for a given test image, all three stages – feature detection, description, and matching – should be efficient and effective. The original SIFT approach detects a several hundreds to a few thousands of keypoints from a medium size image. Its detection stage is slow because of the use of scale-space. Moreover, for many applications, e.g., image copyright protection, this huge number of keypoints also make the later stages unnecessarily slow.

To overcome this problem, we propose two solutions. The first solution is to down scale the input image before DoG keypoint detection. So, the detection stage becomes faster and the number of detected keypoints is reduced largely, which also speeds up the later stages. The highly distinctive SIFT descriptors are used to represent the keypoints and the GPM technique is used for keypoint matching.

As a second solution, we propose using corners instead of DoG keypoints. We use corners detected by our previously proposed two detectors: ARCSS [17, 25] and CPDA [18] detectors. Either the curvature descriptor or the highly distinctive SIFT descriptors at corner locations can be used to represent corners and the GPM is used for corner matching. While finding the SIFT descriptors at a corner location we considered all the SIFT descriptors (detected by Low's approach) within the 3-pixel neighborhood around the corner in question.

Consequently, in this paper we propose five approaches of TII as shown in Table 2.

4.1. Approach 1: ARCSS corners with Curvature Descriptors

Approach 1 uses the ARCSS corners as feature-points, the curvature descriptors to represent the corners and the GPM for corner matching. In the

first step (see Section 2.3) of the GPM technique, the candidate matches are obtained by matching the curvature descriptors of the query image and a database image. Once the candidate point matching set is obtained, Steps 2 and 3 of the GPM technique are applied to identify the transformed images.

4.2. Approach 2: CPDA corners with Curvature Descriptors

Approach 2 uses the CPDA corners as feature-points, the curvature descriptors to represent the corners and the GPM for corner matching.

The differences of Approach 2 with Approach 1 are – (*i*) instead of ARCSS corners Approach 2 uses the CPDA corners and (*ii*) instead of affine-length parameterized Euclidean curvature (see [18]) Approach 2 uses the CPDA discrete curvature (see [25]) to represent corners.

4.3. Approach 3: Scale-DoG Keypoints with SIFT Descriptors

Approach 3 uses the DoG keypoints as feature-points, the SIFT descriptors to represent the keypoints and the GPM for keypoint matching.

As discussed above, in Low's approach [1], the DoG detector detects several hundred to a few thousand keypoints from a medium sized image. Applying the GPM technique to such a huge number of keypoints will be computationally very expensive. Moreover, since each descriptor of a keypoint is a 128-dimensional vector of floating point numbers, storing the descriptor information of an image would take a few megabytes of memory, which in turn would take several gigabytes of memory for a moderate image database.

In order to reduce both of the matching and storage costs, the images are scaled down before feature detection to reduce the number of keypoints. In this case, only the coarse scale features are detected which are more robust than the fine scale features.

The candidate keypoint matches are obtained by matching the SIFT descriptors of the detected keypoints using the NNDR matching technique described in Section 2.3. Once the candidate point matching set is obtained, Steps 2 and 3 of the GPM technique are applied to identify the transformed images.

4.4. Approach 4: ARCSS corners with SIFT Descriptors

Approach 4 uses the ARCSS corners as feature-points, the SIFT descriptors to represent the corners and the GPM for corner matching.

It was observed that the number of false candidate matches is very high with Approach 1 and, therefore, the matching stage becomes expensive. To

overcome this problem, Approach 4 uses the highly distinctive SIFT descriptor, instead of the curvature descriptor, at each corner position while obtaining the candidate corner matching set. This strategy improves the matching performance significantly in terms of both efficiency and effectiveness, though it increases the storage requirement. Similar to the Approach 3 discussed above, the candidate corner matching set is obtained using the NNDR matching and the transformed images are identified using the GPM technique.

Once the candidate point matching set is obtained, Steps 2 and 3 of the GPM technique are applied to identify the transformed images.

4.5. Approach 5: CPDA corners with SIFT Descriptors

Approach 5 uses the CPDA corners as feature-points, the SIFT descriptors to represent the corners and the GPM for corner matching.

The only difference between Approach 5 and Approach 4 is, instead of using the ARCSS corners Approach 5 uses the CPDA corners.

5. Performance Study

We implemented the five proposed approaches using MATLAB 7 on a machine with the following configuration: Dual Core AMD Opteron(tm) Processor (265×2) with 4GB RAM. This section presents the experimental results on a large database and compares the proposed approaches with the existing most popular Low's approach [1]. It presents the results using the precision-recall graph [4].

5.1. Test Database

We randomly selected 1,050 images from David Nister's recognition database [33].[1] Each of the images was of size 480×640 and converted to gray-scale. Then the following 10 simple transformations were applied to each image:

(i) rotation-crop: $\theta = 30°$,

(ii) scale: $s_x = 1.2, s_y = 0.8$,

(iii) rotation-scale: $\theta = 20°, s_x = 1.2, s_y = 0.8$,

(iv) shear: $s_{hx} = s_{hy} = 0.012$,

(v) rotation-scale-shear: $\theta = 10°, s_x = 1.1, s_y = 0.9$, $s_{hx} = s_{hy} = 0.01$,

[1] Available at http://www.vis.uky.edu/d̃nister/. Permission is provided for research purposes on this web site.

(*vi*) jpeg: $quality factor = 20$,
(*vii*) Gaussian noise: $mean = 0$, $variance = 0.001$,
(*viii*) Gaussian blurring: $sigma = 3$, $window = 3 \times 3$,
(*ix*) rotation-scale-jpeg: $\theta = 20°, s_x = 1.2, s_y = 0.8$, $quality = 20$ and
(*x*) rotation-scale-shear-jpeg: $\theta = 20°, s_x = 1.2, s_y = 0.8, s_{hx} = s_{hy} = 0.01, quality = 20$.

As a result, there were a total of 1,050 groups of images in the database. Each group had 10 relevant images. Therefore, there were a total of 10,500 images in the database.

5.2. Evaluation Metrics

We used *precision* and *recall* [4] collectively to measure the identification performance. *Recall* measures the system's capacity to retrieve the relevant images from the database. It is defined as the ratio between the number of retrieved relevant images r and the total number of relevant images r_l (group size) in the database:

$$Recall = \frac{r}{r_l}. \tag{3}$$

Precision measures the retrieval accuracy. It is defined as the ratio between r and the number of retrieved images r_t:

$$Precision = \frac{r}{r_t}. \tag{4}$$

In practice, the performance of an information retrieval system is presented using the *precision-recall graph*, where the higher the precision at a given recall value the better the performance of the retrieval system [4].

In order to calculate the *precision* and *recall* for a given query image using (3) and (4), we first rank all the database images based on the number of feature-point matches with the query image. We then consider the minimum top r_t images as the retrieved images such that they include all $r_l = 10$ relevant images of the query image.

5.3. Approaches to be Compared

In the performance study, seven different TII approaches were considered as shown in Table 3. Approaches 1 to 5 in Table 3 are the proposed ones discussed in Section 4. Approach 3 scaled down each original image of size 480×640 to 60×80. Approaches 6 and 7 were implemented to compare with the proposed approaches. Approach 6 is Low's approach [1] which used the DoG keypoints, SIFT descriptors and NNDR matching. Approach 7 is

Table 3: Transformed image identification approaches used in the performance study.

Approaches	**Detectors**	**Descriptors**	**Matching**
Approach 1	ARCSS	curvature	GPM
Approach 2	CPDA	curvature	GPM
Approach 3	scale-DoG	SIFT	GPM
Approach 4	ARCSS	SIFT	GPM
Approach 5	CPDA	SIFT	GPM
Approach 6	DoG	SIFT	NNDR
Approach 7	scale-DoG	SIFT	NNDR

similar to Approach 3, but uses the NNDR matching instead of the GPM technique to show the superior performance of the GPM technique.

Note that the GPM technique was found computationally very expensive when it was applied to match the DoG keypoints due to the large number of keypoints, where each keypoint is represented using a 128-dimensional SIFT feature-vector (so it is not reported here). However, the accuracy was found better than using the NNDR matching (Approach 6).

5.4. Experimental Results and Discussions

Fig. 3 shows the TII performance of different approaches. Due to high computational complexity of some of the approaches, the graphs in Fig. 3 were averaged on 177 random queries only. The following sub-sections present and discuss the experimental results.

5.4.1. Scale-DoG vs Usual DoG Keypoint Detectors

Approach 7 performed worse than Low's approach (Approach 6) for the following two reasons: (*i*) in Approach 7, though the number of feature-points is decreased to speed up both of the detection and matching stages, the number of matching features using the NNDR matching is also decreased; and (*ii*) in contrast, the main reason behind the success of Low's approach is, it detects a lot of features and a large number of them are found matched between the relevant images.

5.4.2. GPM vs NNDR matching

The NNDR matching is based on the individual descriptors (see Section 2.3). In contrast, the GPM matching first estimates the transformation matrix between images using the candidate matches, transforms corners in one

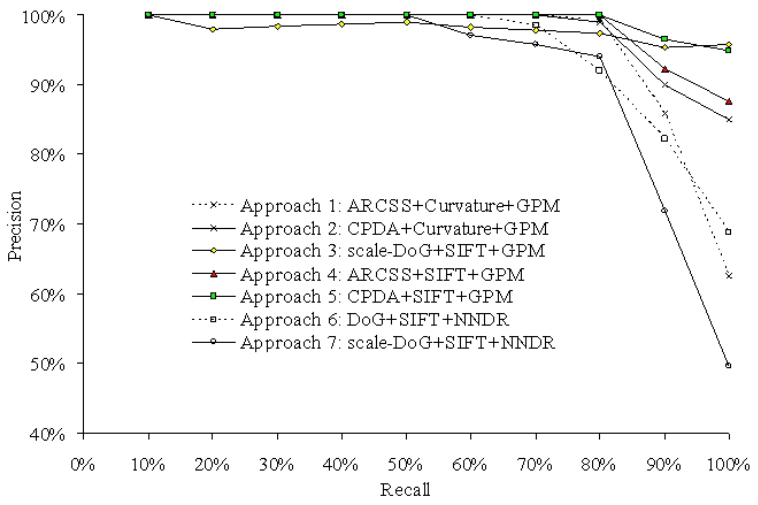

Figure 3: Transformed image identification performance by different approaches.

image and then finds matches with the other image. Consequently, the GPM matching is like a global matching technique and offers a higher number of matches than the NNDR matching.

Approach 7 uses the NNDR matching only. Approach 3 uses the GPM technique which in addition to obtaining the candidate matches using the NNDR matching, estimates the transformation matrix before finding the corner matches. As a result, Approach 3 performed better than Approach 7 as shown in Fig. 3. This shows the superior performance of the GPM technique compared to the NNDR matching.

5.4.3. Corner-Curvature vs DoG Keypoint-SIFT Combinations

Approach 1 required lower storage and total computational costs, but still showed slightly higher identification accuracy than Approach 6. While the former used the curvature descriptor with the GPM technique, the latter used the SIFT descriptors with the NNDR matching technique. Approach 2 offered higher identification accuracy than Approach 1, but required a higher computational cost. This is because the number of CPDA corners was higher than the number of ARCSS corners and, thus, there may be a larger number of false candidate matches in Approach 2 than in Approach 1.

However, Approach 6 outperformed Approach 1 at higher recall values as shown in Fig. 3. The reason is, as the average number of detected corners

was low using the ARCSS detector (Approach 1), some irrelevant images (usually with a higher number of detected corners than the relevant images) may also be retrieved. The performance was increased at higher recall values by increasing the number of corners, as evident from the performance of Approach 2 (using CPDA corners). However, this made the matching procedure expensive.

Table 4: Comparing different transformed image identification approaches of Table 3.

Approaches	**Features** [a]	**Storage**	**DR time**	**Mat. time**	**Total time**	**Accuracy**
Approach 1	46	*low*	0.17	4.330	4.50	94.8
Approach 2	58	*low*	2.6	5.36	7.96	97.4
Approach 3	**37**	*moderate*	**0.54**	**0.004**	**0.544**	**97.9**
Approach 4	**71**	*moderate*	**0.17**	**0.045**	**0.215**	**98.0**
Approach 5	**90**	*moderate*	**2.6**	**0.17**	**2.77**	**99.1**
Approach 6	585	*high*	2.70	3.380	6.08	94.2
Approach 7	37	*moderate*	0.54	0.002	0.542	90.9

[a] *Features* = number of features per image; *DR time* = feature detection and representation time per image; *Mat. time* = time to match a pair of images (a query image and a database image); all times are in seconds; *Accuracy* is the average precision over all the recall values shown in Fig. 3; total number of features and *DR time* were averaged over the whole database discussed in Section 5.1; and *Mat. time* was averaged over 177 random queries.

5.4.4. Corner-Curvature vs Corner-SIFT Combinations

When the SIFT descriptors were used to represent the corners the matching cost reduced drastically as the number of false candidate matches decreased considerably due to the highly distinctive nature of the SIFT descriptor [1] (see Approaches 4 and 5 in Fig. 3).

5.4.5. The Best TII Approaches

Table 4 summarizes the performance of different approaches. Approach 5 performed the best in terms of identification accuracy, followed by Approaches 3 and 4. However, Approach 5 was much slower than Approaches 3 and 4 due to its slow corner detector. Approach 6 required very high storage for descriptors and all of its stages were quite expensive. The identification accuracies of Approaches 1 and 2 were slightly better than the SIFT. Moreover, Approaches 1 and 2 required much lower storage than Approach

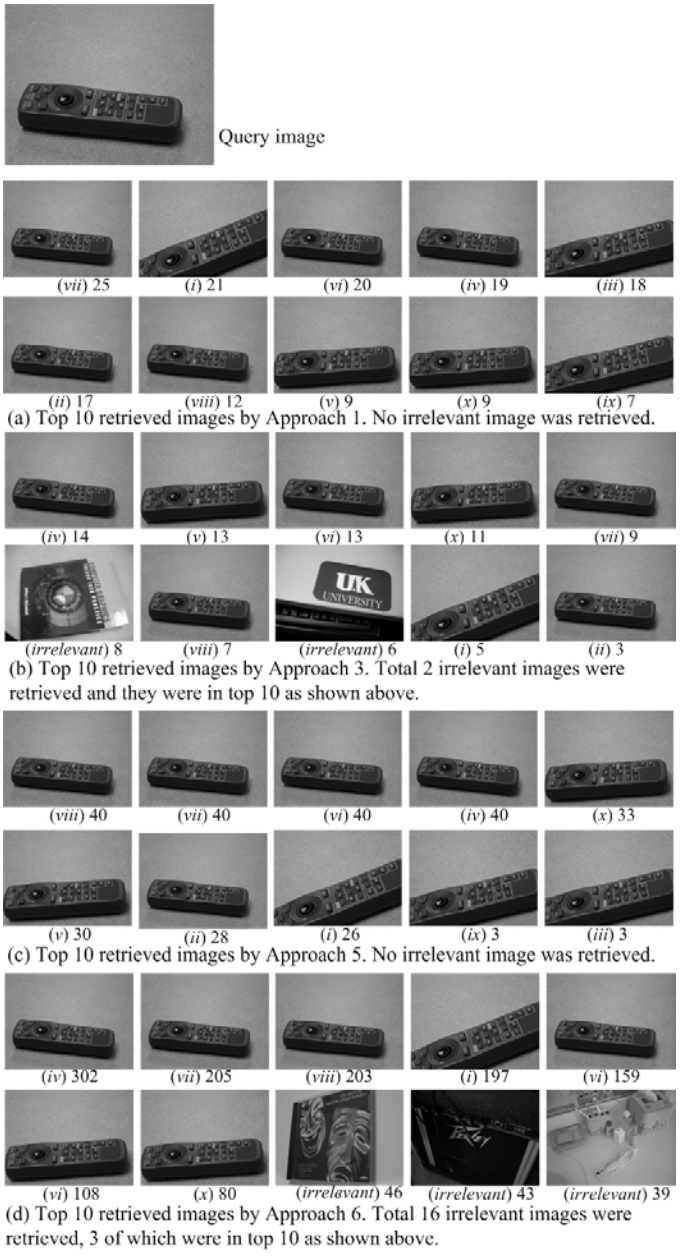

Figure 4: Transformed image retrieval examples by Approaches 1, 3, 5 and 6. Approach 4 performed the same (no irrelevant images were retrieved) as Approach 1 as shown in (a) above. And Approach 2 performed the same (no irrelevant images were retrieved) as Approach 5 as shown in (c) above. The number of detected features in the query image by these Approaches were 19, 37, 66 and 488 respectively. The roman number (inside parentheses) under each image indicates a particular transformation discussed in Section 5.1 and the number after it is the number of feature matches with the query image in (a).

6. The storage requirement, total time and the identification accuracy of Approach 6 were significantly improved by the application of the proposed first feature-point reduction solution (see Approach 3). The total time and the identification accuracy of Approaches 4 and 5 (the second solution) were also considerably improved by using the distinctive SIFT descriptors at the corner locations, though the storage requirement was moderately increased. This proves that the highly distinctive SIFT descriptor is more effective than the curvature descriptor.

Therefore, by combining the advantages of corners, SIFT descriptors and the GPM technique, Approaches 3, 4 and 5 can be considered the overall best.

Note that we used the machine code provided by D. Lowe[3] for the feature detection and representation stages of Approaches 3, 6 and 7. The DR time shown in Table 4 is not important when the feature-point detection and representation stages are done offline. However, the DR time for query processing is important.

5.5. Transformed Image Identification Examples

Fig. 4 presents an example of the transformed image identification by Approaches 1, 3, 5 and 6. In this example, Approach 6 retrieved 16 irrelevant images, Approach 3 retrieved 2 irrelevant images and Approaches 1 and 5 retrieved no irrelevant images. The top 10 retrieved images are shown in Fig. 4 for each Approach, where 3 irrelevant retrieved images by Approach 6 and 2 irrelevant images by Approach 3 are shown. Note Approaches 2 and 4 performed the same (no irrelevant images were retrieved) as Approach 5 (Fig. 4(c)) and 1 (Fig. 4(a)) respectively.

6. Conclusion and Future Work

We have shown that both of the feature detection and matching stages of TII can be significantly speeded up by reducing the number of detected feature-points. We have presented two feature-point reduction solutions.

The first solution is to down scale the image before the DoG keypoint detection. We have found that this solution significantly increases the TII performance when the scale-DoG keypoints are matched using the GPM technique, which finds more repeated features than the existing NNDR matching

[3]Available at http://www.cs.ubc.ca/~lowe/keypoints/

technique with the expense of little more computational time. However, this additional expense can be considered negligible because of high improvement in the identification accuracy (comparing accuracy of Approach 3 with that of Approaches 6 and 7 in Table 4).

The second solution is to use corners instead of DoG keypoints. Either the curvature descriptor or the highly distinctive SIFT descriptors at corner locations can be used to represent corners. The corner-curvature combinations, though require higher matching time, perform better than the keypoint-SIFT combinations (comparing performance of Approaches 1 and 2 with that of Approaches 6 and 7 in Table 4). However, when the corners are represented using the SIFT descriptors (corner-SIFT combinations), not only the matching time is significantly reduced, but also the TII accuracy (robustness) is improved (comparing performance of Approaches 4 and 5 with that of Approaches 1 and 2 in Table 4).

Therefore, the two feature-point reduction solutions combined with the SIFT descriptors and the GPM technique not only improve the computational efficiency and decrease the storage requirement, but also improve the TII accuracy.

However, there are at least two more directions for further improvement. Firstly, the reduction in the number of feature-points in the proposed solutions could limit the applicability in some applications like matching images with different lighting conditions and backgrounds where the repeatability rate of the detected feature-points is very low. Increasing the repeatability rate significantly in such cases, while keeping the number feature-points low, could be a future research interest.

For example, we could use the tensor-based feature representation (in the form of a high-order matrix [34]), which preserves the distinctive information. As a result, the repeatability rate could be improved despite the presence of a small number of feature-points. Xu et al. [35] successfully applied tensor features for human gait recognition.

Secondly, the matching time shown in Table IV is the time to match a pair of images. Consequently, the query time (to get search results after searching the whole database) is very high for a large database. There could be two solutions: 1) Representation of the database as a vocabulary tree [33], where each image is represented as a number of nodes (features) in the tree. 2) Interactive image searching techniques, which are currently being used for user feedback-based CBIR systems [5, 6], could be employed. A set of initial searched images are first shown for user feedbacks. Then based on

user feedbacks, the CBIR system iteratively refines the search results.

References

[1] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. Journal of Comp. Vision 60 (2) (2004) 91–110.

[2] M. Awrangjeb, M. Murshed, Robust signature-based geometric invariant copyright protection, in: Proc. IEEE International Conference on Image Processing, Atlanta, USA, 2006, pp. 1961–1964.

[3] C. Liu, Enhanced independent component analysis and its application to content based face image retrieval, IEEE Transactions on Systems, Man, and Cybernetics, Part B 34 (2) (2004) 1117–1127.

[4] G. Lu, Multimedia Database Management Systems, Artech House Inc., Norwood, 1999.

[5] J. Li, N. Allinson, D. Tao, X. Li, Multitraining support vector machine for image retrieval, IEEE Transactions on Image Processing 15 (11) (2006) 3597–3601.

[6] D. Tao, X. Tang, X. Li, Which components are important for interactive image searching?, IEEE Transactions on Circuits and Systems for Video Technology 18 (1) (2008) 3–11.

[7] D. Xu, T. J. Cham, S. Yan, S. F. Chang, Near duplicate image identification with spatially aligned pyramid matching, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, Alaska,, USA, 2008, pp. 1–7.

[8] W. L. Zhao, C.-W. Ngo, H.-K. Tan, X. Wu, Near duplicate keyframe identification with interest point matching and pattern learning, IEEE Transactions on Multimedia 9 (5) (2007) 1037–1048.

[9] D. Zhang, S. F. Chang, Detecting image near duplicate by stochastic attribute relational graph matching with learning, in: Proc. ACM Multimedia, New York, USA, 2004.

[10] Y. Ke, R. Sukthankar, L. Huston, Efficient near duplicate detection and sub-image retrieval, in: Proc. ACM Multimedia, New York, USA, 2004.

[11] H. Bay, T. Tuytelaars, L. V. Gool, Surf: speeded up robust features, in: Proc. European Conf. on Comp. Vision, Vol. LNCS 3951, Austria, 2006, pp. 404–417.

[12] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. on Patt. Anal. and Machine Intel. 27 (10) (2005) 1615–1630.

[13] Y. Ke, R. Sukthantar, Pca-sift: a more distinctive representation for local image descriptors, in: Proc. Int. Conf. on Comp. Vision and Pattern Recognition, Vol. 2, 2004, pp. 506–513.

[14] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. Journal of Comp. Vision 60 (1) (2004) 63–86.

[15] M. Awrangjeb, G. Lu, A robust corner matching technique, in: Proc. IEEE International Conference on Multimedia & Expo, Beijing, China, 2007, pp. 1483–1486.

[16] T. Lindeberg, Feature detection with automatic scale selection, Int. Journal of Comp. Vision 30 (2) (1998) 77–116.

[17] M. Awrangjeb, G. Lu, M. Murshed, An affine resilient curvature scale-space corner detector, in: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, Hawaii, USA, 2007, pp. 1233–1236.

[18] M. Awrangjeb, G. Lu, Robust image corner detection based on the chord-to-point distance accumulation technique 10 (6) (2008) 1059–1072.

[19] F. Mokhtarian, R. Suomela, Robust image corner detection through curvature scale space 20 (12) (1998) 1376–1381.

[20] X. Zhang, M. Lei, D. Yang, Y. Wang, L. Ma, Multi-scale curvature product for robust image corner detection in curvature scale space, Pattern Recognistion Letters 28(2007) (2007) 545–554.

[21] X. C. He, N. H. C. Yung, Curvature scale space corner detector with adaptive threshold and dynamic region of support, in: Proc. Internatinal Conference on Pattern Recognition, Vol. 2, Cambridge, UK, 2004, pp. 791–794.

[22] C. Harris, M. Stephens, A combined corner and edge detector, in: Proc. Alv. Vis. Conf., 1988, pp. 147–151.

[23] E. D. Sinzinger, A model-based approach to junction detection using radial energy, Pattern Recognition 41 (2) (2008) 494–505.

[24] C. Schmid, R. Mohr, C. Bauckhage, Comparing and evaluating interest points, in: Proc. International Conference on Computer Vision, Beijing, China, 1998, pp. 230–235.

[25] M. Awrangjeb, G. Lu, An improved curvature scale-space corner detector and a robust corner matching technique for transformed image identification 17 (12) (2008) 2425–2441.

[26] J. H. Han, T. T. Poston, Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature, Pattern Recognition Letters 22(2001) (2001) 1133–1144.

[27] D. Zhou, G. Li, Y. Liu, Effective corner matching based on delaunay triangulation, in: Int. Conf. on Robotics and Automation, Vol. 3, 2004, pp. 2730–2733.

[28] D. P. Huttenlocher, Fast affine point matching: an output-sensitive method, in: Proc. International Conference on Computer Vision and Pattern Recognition, 1991, pp. 263–268.

[29] M. Awrangjeb, Contour-based corner detection and robust geometric point matching techniques, Ph.D. thesis.

[30] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[31] J. Canny, A computational approach to edge detection 8 (6) (1986) 679–698.

[32] G. Cao, J. Chen, J. Jiang, A novel local invariant descriptor adapted to mobile robot vision, in: Proc. American Control Conference, Vol. 3, Boston, MA, USA, 2004, pp. 2196–2201.

[33] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, New York, USA, 2006, pp. 2161–2168.

[34] D. Tao, X. Li, W. Hu, S. Maybank, X. Wu, Supervised tensor learning, in: Proc. IEEE International Conference on Data Mining, Texas, USA, 2005.

[35] D. Xu, S. Yan, D. Tao, L. Zhang, X. Li, H. J. Zhang, Human gait recognition with matrix representation, IEEE Transactions on Circuits and Systems for Video Technology 16 (7) (2006) 896–903.