FAST RETRIEVAL METHODS FOR IMAGES WITH SIGNIFICANT VARIATIONS

Paul W. Fieguth and Riyin Wan

Department of Systems Design Engineering
University of Waterloo, Waterloo, Ontario, *N2L 3G1*, Canada

ABSTRACT

Fast image retrieval is the key to success for operations on large image databases, and a great many techniques have been developed for efficient retrieval. However, most of these methods are tailored to visual scenes or to images having limited variations. In this paper we investigate the searching of enormous databases (of up to 10^7 images) for the matching and identification of precious stones (principally diamonds). Because of the size of the database, we propose a hierarchy of classifiers, which successively prune candidate images such that the more complex classifiers are required to test only tiny portions of the data. The new classifier developed here applies a wavelet transform to image histograms and is capable of rejecting 99.9% of bad matches.

1. INTRODUCTION

Diamonds are small and expensive! They look essentially alike, except for variations in color, mass, and shape. Diamond thefts are relatively common and among the most infamous of crimes, so law enforcement authorities are very interested in a mechanism to determine the identity (that is, whether stolen or not) of a given diamond.

The number of diamonds is huge and increasing; in North America it is estimated that there are more than two million. Because of the small dimensions and few colors, we cannot rely on these attributes to differentiate all diamonds, however it has been found that the diamond *signature* (Figure 1) produced under red laser illumination is very nearly unique. The FBI has already used such images as the basis for diamond identification, however at present the process is intolerably slow. Furthermore the differences between diamond image signatures can be subtle, so the matching accuracy of standard image-database algorithms is either unacceptable in terms of reliability, or in terms of computational time. Existing diamond matching algorithms, performing a sequential search through the database, would require about one month of time.

In standard image-database systems, images taken at different times, by different people, and using different cameras need to be compared. These images may have different noise levels or be captured from different perspectives; a perfect matching is no longer possible, nor expected. If the variations are small, many techniques can be used. However, if the variations and random noise are significant, most known algorithms cannot be extended. Furthermore, because images can't be sorted into ascending or descending orders, brute-force sequential searches are undertaken

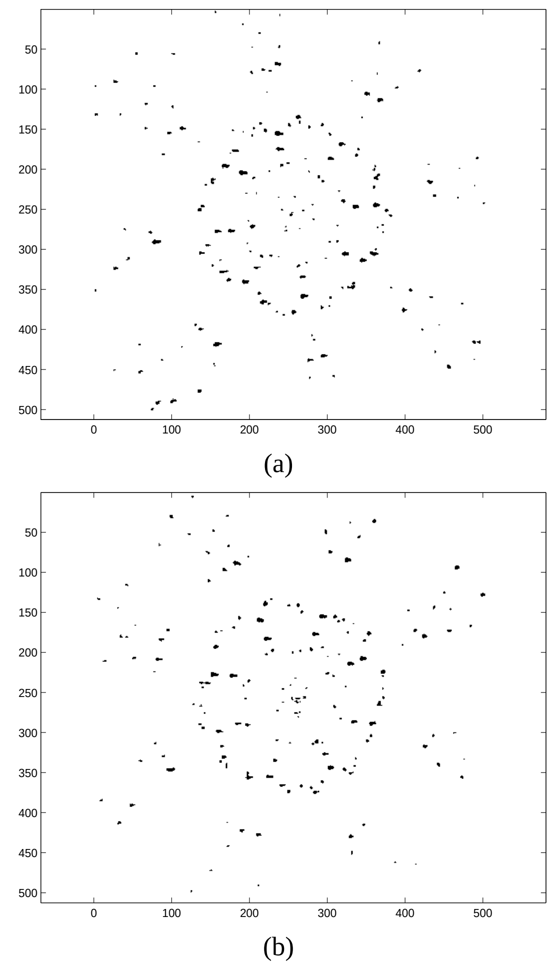
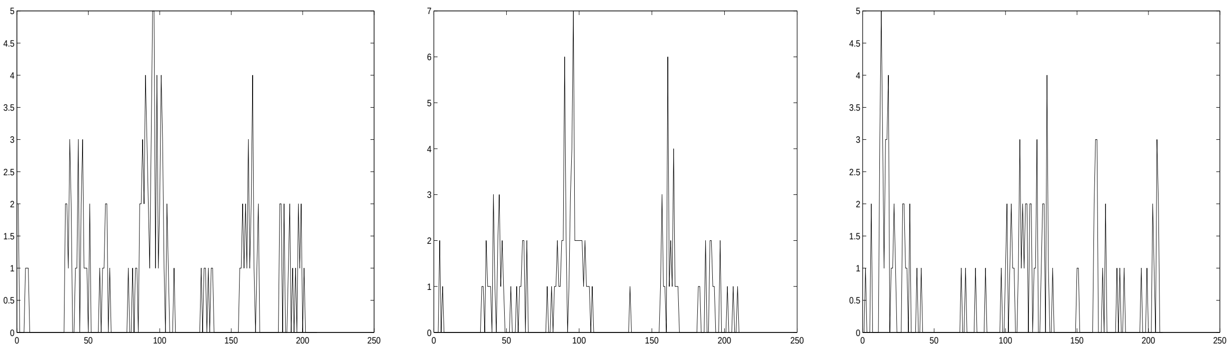


Fig. 1. Two red-laser images from the same diamond; the two images are very similar, except for a rotation of the diamond.



Radial Histogram of Figure 1 (a) Radial Histogram of Figure 1 (b) Radial Histogram of Different Gem

Fig. 2. A comparison of three radial histograms: the first two from the same diamond, the third from a different one. Although the first two histograms are not identical, they share basic similarities different from the third.

— obviously problematic for very large sets of images.

This paper proposes an effective method to match diamond images, despite significant noise and image variations. We propose a coarse-to-fine matching, to successively strip away images from the database, leading to an extremely fast result.

2. PROBLEM FORMULATION

Given a diamond image we need to find all images (if any) in the database which correspond to the same diamond. The centre of each image is known, so matching images possess only an unknown relative rotation, no translation. Each image is binary, 512×512 pixels, where the spots are found based on a predetermined thresholding.

We have no prior knowledge regarding how images from different gems should differ, however we can establish models for the variations of spots in images takes from the same gem. Generally, the spot radii vary only very slightly (a standard deviation on the order of 5 pixels), however the number of spurious, mismatching points can be as high as 40% (38% of the points mismatch between the two images in Figure 1).

Obviously directly storing and matching 512×512 pixel images is inefficient; rather we can use the spot centers to represent each image. The image matching problem is thus transformed into the matching of two point sets in two-dimensional space. The point-set matching problem has been well-studied in the graph theory literature[4, 5], so there are many algorithms which could deal with the unknown angular offset between images, however the large fraction of spot mismatches, the enormous number of images in the database, and the required classification speed lead to additional challenges.

3. EXACT MATCHING

Point matching is generally computationally expensive. The majority of time is required to identify matching pairs or to find the rotational alignment. Chang et al. [5] use a 2-D cluster approach to find the optimal matching, having a complexity of $\mathcal{O}(nm)$, where n and m are the numbers of points in the two respective images. Mount et al. [4] search the possible transformation space based on branch-and-bound approach. They all conclude their algorithms are efficient, which is true in general, although they are impractical for problems of the scale considered here because the general algorithms do not take advantage of the unique properties of diamond images.

We have developed algorithms specialized to the diamond matching problem. Because the unknown parameter is only the rotation about the known origin, polar coordinates are a convenient and efficient representation of the spot centres, since a problem involving rotation is transformed into a more efficient one of translation. This first algorithm performs an exact matching (*no* missed detections based on our database):

- Find the sets of point features P_1 and P_2 of the two images to be matched; each point feature is a spot center.
- Choose the geometric center of the image as the origin and represent the feature points using polar coordinates:

$$P_1 = \{p_{1i} = (r_{1i}, \theta_{1i}) | 1 \leq i \leq n\} \quad (1)$$
$$P_2 = \{p_{2j} = (r_{2j}, \theta_{2j}) | 1 \leq j \leq m\} \quad (2)$$

- Sort P_1 and P_2 as a function of radius r.

- Estimate the rotation $\Delta\theta$ by brute force. Let $C(\Delta\theta)$ denote the number of point pairs such that

$$(r_{1i}, \theta_{1i}) \approx (r_{2j}, \theta_{2j} - \Delta\theta) \qquad (3)$$

where a matching pair is subject to thresholds τ_r in radius and τ_θ in angle (3 pixel and 2 degrees respectively), and where $\Delta\theta$ is sampled in increments:

$$\Delta\theta = k * 2\pi/360, \;\; 1 \leq k \leq 360 \qquad (4)$$

- Find the optimum rotation angle

$$\Delta\theta_{opt} = \arg_{\Delta\theta} \max C(\Delta\theta) \qquad (5)$$

- Based on $\Delta\theta_{opt}$, find the percentage $M\%$ of matching points between the two images:

$$M = \begin{cases} > 60 & \text{Same Diamond} \\ 40 - 60 & \text{Never Observed} \\ < 40 & \text{Different Diamond} \end{cases} \qquad (6)$$

For images in which the matching percentage between 40% and 60%, it is hard reach any conclusion, however none of the images in our database ever have a matching percentage in this range; that is, our same-image and different-image clusters are very well separated, and errors are expected to be rare. Although this exact algorithm is faster than more general approaches, its computational effort is on the order of many milliseconds, which is still inadequate for a huge search through millions of images, which motivates an approximate classifier, presented next.

4. APPROXIMATE MATCHING

The obvious problem with the previous algorithm is that it attempts exact matching. Instead, one or two appropriately-chosen crude features should allow grossly different diamonds to be discriminated with only a very minimal effort, leaving only a tiny fraction to be evaluated using more sophisticated approaches. Because the images from the same diamond possess a random angle, sensible features should have a radial dependence only. In principle, the two-dimensional point-matching problem can be reduced to the comparison of one-dimensional radial histograms (shown in Figure 2). Certainly the details of the histograms will vary, but the coarse-scale radial distribution of points will be very similar for images of a given diamond.

There are, however, two crucial criteria that any successive matching algorithm has to satisfy:

1. We must never have missed detections

$$Pr\{\text{Classify Different}|\text{Same Diamond}\} = 0, \qquad (7)$$

since an image, once rejected, will never be reconsidered by a later classifier.

2. Secondly, to be useful a crude classifier should reject large fraction of mismatching images:

$$Pr\{\text{Classify Same}|\text{Different Diamond}\} = \epsilon \ll 1 \qquad (8)$$

As a fast, approximate matching tool, we propose to apply a wavelet transform to the radial histogram of the spot distribution. If we keep only the few coarsest-scale coefficients (implicitly corresponding to a "denoised" histogram, Figure 4), then we have an effective representation of the basic histogram structure, and we are left with only a tiny number of scalars to compare — an extremely fast operation, insignificant compared with the original exact spot-matching problem. The proposed approximate matching algorithm proceeds as follows:

- Construct histograms H_1 and H_2 from point features P_1 and P_2, where

$$H_i = \{n_{ij}\} \qquad (9)$$

where n_{ij} is the number of point features in image i present in radius bin j.

- Use a Daubechies wavelet to denoise the histograms and save the wavelet coefficients of the denoised histograms as the attributes in the database; that is, we compute a vector

$$W_i = \mathcal{W}_L\{H_i\} \qquad (10)$$

where $\mathcal{W}_L$ returns the wavelet approximation coefficients at level L.

The wavelet transform is a preprocessing step, applied to each image as it is added to the database.

- We then propose to define the image-image difference to be measured by

$$d_{12} = ||W_1 - W_2||_1, \qquad (11)$$

the L_1 distance between W_1 and W_2, which leads immediately to a classifier

$$\begin{array}{ll} d_{12} > \tau & \text{Images definitely different} \\ d_{12} < \tau & \text{Defer judgment to finer classifier} \end{array} \qquad (12)$$

That is, the two diamonds are either determined not to be the same, or the case is declared ambiguous, and the images are kept for further testing with a finer algorithm.

- For the images passed the previous matching, use the coefficients of histograms with smaller bin size.

The effectiveness of this algorithm - a rejection rate of over 99.9%, coupled with a rejection of about 99.8% based on physical criteria, implies that only a tiny fraction remains to be tested using an exact test.

	# *Images to Analyze*	*Complexity*	*Rejection Rate*
Physical Criterion – Carats	≈ 10000000	$\mathcal{O}(10^{-6})$s	95%
Physical Criterion – Colour	≈ 500000	$\mathcal{O}(10^{-6})$s	86.7%
Radial Histogram Matching	≈ 83333	$\mathcal{O}(10^{-4})$s	99.9%
2D Image Pattern Matching	≈ 83	$\mathcal{O}(10^{-1})$s	100%

Fig. 3. Approximate breakdown of the database matching problem: Of the 10 million images in the database, we successively reject bad matches based on successively more sensitive algorithms. The proposed radial-histogram classifier rejects about 99% of its images, allowing the entire search to run in near real-time.

5. EXPERIMENTAL RESULTS

The full database matching approach proceeds hierarchically, as illustrated in Figure 3, based on a test with several hundred images. The fastest, and also most approximate, operations are applied first, stripping the most obvious mismatches from the database. Obviously the 1D histogram matching problem itself admits numerous algorithms, with different complexity and accuracy. Figure 4 shows the effect of wavelet DB3 with a level 3 approximation: the two classes (same and different gems) are well separated, and the rejection rate is over 99%. Because the wavelet coefficients can be computed ahead of time and saved in the database, and the comparison of the wavelet coefficient vectors is very fast, therefore the total searching time is reduced to only a few minutes.

6. REFERENCES

[1] M. J. Swain and D. H. Ballard, “Color indexing”, *Int. J. Computer Vision*, 7(1):11-32, 1991.

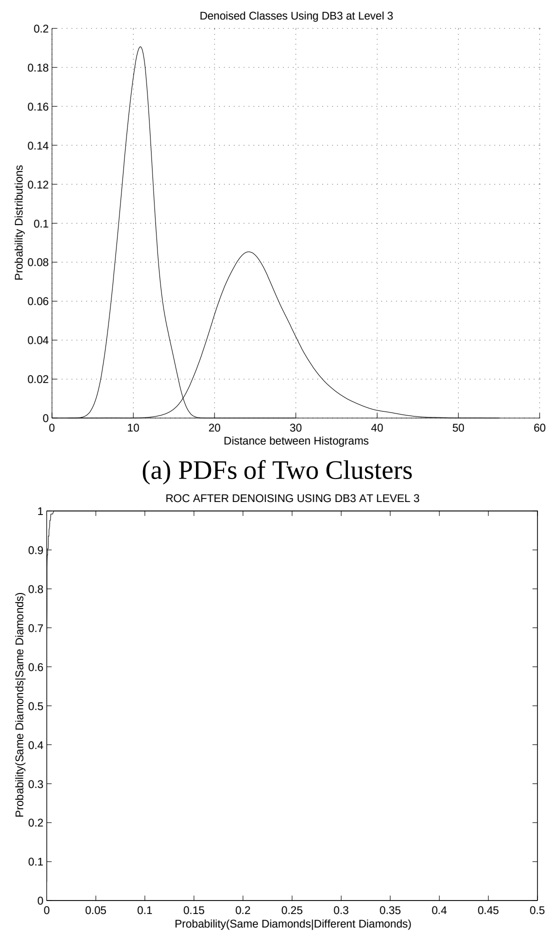
[2] J. Lee and B. W. Dickinson, “Multiresolution video indexing for subband coded video databases”, *Proc. SPIE*, V. 2185, San Jose, CA, Feb. 1994.

[3] M. K. Mandal, T. Aboulnasr, and S. Panchanathan, “Image indexing using moments and wavelets”, *IEEE Trans. Consumer Electronics*, 42(3):557-565, 1996.

[4] D. M. Mount, N. S. Netanyahu and Jacqueline L. Moigne, “Efficient algorithms for robust feature matching”, *Pattern Recognition*,32(1):17-38, Jan. 1999

[5] S. H. Chang, F. H. Cheng, W. H. Hsu and G. Z. Wu, “Fast algorihtm for point matching: Invariant to translations, rotations and scale changes”, *Pattern Recognition*, 30(2):311-320, 1997.

[6] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, “Image coding using wavelet transform”, *IEEE Trans. Image Proc.*, 1(2):205-219, 1992.



(a) PDFs of Two Clusters

(b) ROC Curves

Fig. 4. Performance of Using DB3 at Level 3: (a) distributions of the difference d_{12} for identical (left) and different (right) gems; (b) the false-alarm / detection graph, illustrating the high rejection ratio.