ROBUST PERCEPTUAL IMAGE HASHING USING FEATURE POINTS

Vishal Monga and Brian L. Evans

Embedded Signal Processing Laboratory, Center for Perceptual Systems
The University of Texas at Austin, Austin, TX 78712
{vishal,bevans}@ece.utexas.edu

ABSTRACT

Perceptual image hashing maps an image to a fixed length binary string based on the image's appearance to the human eye, and has applications in image indexing, authentication, and watermarking. In this paper, we present a general framework for perceptual image hashing using feature points. The feature points should be largely invariant under perceptually insignificant distortions. To satisfy this, we propose an iterative feature detector to extract significant geometry preserving feature points. We apply probabilistic quantization on the derived features to further enhance perceptual robustness. The proposed hash algorithm withstands standard benchmark (e.g. Stirmark) attacks including compression, geometric distortions of scaling and small angle rotation, and common signal processing operations. Content changing (malicious) manipulations of image data are also accurately detected.

1. INTRODUCTION

In cryptography, hash functions are typically used for digital signatures to authenticate the message being sent so that the recipient can verify its source. A key feature of conventional hashing algorithms such as MD5 and SHA-1 is that they are extremely sensitive to the message [1]; i.e., a one bit change in the input changes the output dramatically. Data such as digital images, however, undergo various manipulations such as compression and enhancement.

An *image hash function* takes into account changes in the visual domain. In particular, a perceptual image hash is required to be invariant under image manipulations that do not alter the image appearance significantly. Such a function could be useful for identification/search of images in large databases. Other applications of image hashing lie in the area of authentication and watermarking [2].

Significant attention has been given to generating digital signatures for image authentication under certain attacks. This includes methods based on statistics of the image (or its transformed version) [3, 4], relation-based methods that use information about the DCT/Wavelet coefficients of the image [5, 6], and methods based on extraction of low-level image features such as edges and corners [7, 8].

Research supported by a gift from the Xerox foundation

A common characteristic of the methods in [3]– [8] is that while they authenticate the image under special forms of manipulation, e.g. JPEG compression, they are still vulnerable to several incidental modifications that do not cause perceptually significant changes.

Robust image hashing methods that tolerate a wider range of perceptually insignificant distortions have been proposed recently [9–12]. The methods in in [9] and [10] are relation based. In [11] Venkatesan *et al.* form a hash based on an image statistics vector extracted from subbands in a wavelet decomposition of the image. Mihcak *et al.* [12] develop another hash by using an iterative approach to binarize the DC subband.

We present a framework for perceptual image hashing using feature-points. Current approaches based on feature points have limited utility as they have poor robustness properties. We extract significant image features by using a wavelet based feature detection algorithm based on the characteristics of the visual system [13]. Further, an iterative procedure based on observations in [12] is used to lock onto a set of image feature-points with excellent invariance properties to perceptually insignificant perturbations. Unlike, the use of public-key encryption schemes in [7], [8] probabilistic quantization is used to binarize the extracted feature vector. Previous work on image hashing has focussed extensively on the problem of capturing image characteristics but performance trade-offs such as those between perceptual robustness, fragility and randomization of the hash are not explicitly analyzed. These trade-offs are directly addressed via parameters in our hash algorithm.

2. FEATURE EXTRACTION

2.1. End-Stopped Wavelets

Psychovisual studies have identified the presence of certain cells, called hypercomplex or end-stopped cells, in the primary visual cortex [13]. For real-world scenes, these cells respond strongly to extremely robust image features such as corner like stimuli and points of high curvature in general [14], [15]. Bhattacherjee *et al.* [15] constructed "end-stopped" wavelets to capture this behavior. Morlet wavelets can be used to detect linear structures having a specific orientation. In spatial domain, the two dimensional (2-D)

Morlet wavelet is given by [16]

$$\psi_M(\mathbf{x}) = (e^{j\mathbf{k}_0.\mathbf{x}} - e^{-\frac{1}{2}|\mathbf{k}_0|^2})(e^{-\frac{1}{2}|\mathbf{x}|^2}) \quad (1)$$

where $\mathbf{x} = (x, y)$ represents 2-D spatial co-ordinates, and $\mathbf{k}_0 = (k_0, k_1)$ is the *wave-vector* of the mother wavelet, which determines scale-resolving power (SRP) and angular-resolving power (ARP) of the wavelet [16]. The frequency domain representation, $\psi_M(\mathbf{k})$, of a Morlet wavelet is

$$\hat{\psi}_M(\mathbf{k}) = (e^{-\frac{1}{2}|\mathbf{k}-k_0|^2} - e^{-\frac{1}{2}|\mathbf{k}_0|^2})(e^{-\frac{1}{2}|\mathbf{x}|^2}) \quad (2)$$

Here, $\mathbf{k}$ represents the 2-D frequency variable (u, v). In two dimensions, the end points of linear structures can be detected by applying the first-derivative of Gaussian (FDoG) filter parallel to the orientation of structures in question. The two filtering stages, the first to detect lines having a specific orientation and the second to detect the end-points of such lines, can be combined into a single filter. This results in an "end-stopped" wavelet [15]. An example of an end-stopped wavelet and its 2-D Fourier transform follow:

$$\psi_E(x, y) = \frac{1}{4} y e^{-\left(\frac{x^2+y^2}{4} + \frac{k_0}{4}(k_0 - 2jx)\right)} \quad (3)$$

$$\hat{\psi}_E(u, v) = 2\pi \left(e^{-(\frac{(u-k_0)^2+(v)^2}{2})}\right)\left(jv e^{-(\frac{u^2+v^2}{2})}\right) \quad (4)$$

Equation (4) shows that $\hat{\psi}_E$ is a product of two components. The first is a Morlet wavelet oriented along the $u-$axis. The second factor is a FDoG operator applied along the frequency-axis v, that is in the direction perpendicular to the Morlet wavelet. Hence, this wavelet detects line ends and high curvature points in the vertical direction.

2.2. Proposed feature detection method

Our approach to feature detection computes a wavelet transform based on an *end-stopped* wavelet obtained by applying the FDoG operator to the Morlet wavelet:

$$\psi_E(x, y, \theta) = (FDoG)\ o(\psi_M(x, y, \theta)) \quad (5)$$

Orientation tuning is given by $\theta = \tan^{-1}(\frac{k_1}{k_0})$. Let the orientation range $[0, \pi]$ be discretized into M intervals and the scale parameter α be sampled exponentially as α^i, $i \in Z$. This results in the wavelet family

$$\left(\psi_E(\alpha^i(x, y, \theta_k)\right), \alpha \in \mathcal{R},\ i \in Z \quad (6)$$

where $\theta_k = (k\pi)/M$, k = 0,...,M-1. The transform is

$$W_i(x, y, \theta) = \int f(x_1, y_1)\psi_E{}^*\left(\alpha^i(x - x_1, y - y_1), \theta\right) dx_1 dy_1 \quad (7)$$

The sampling parameter α is chosen to be 2.

Fig. 1 describes the proposed feature detection method. The method has two free parameters: integer scale i and real threshold T. We adapt the threshold T to select a fixed number (P) of feature points from the image. The length P feature vector is labeled as $\mathbf{f}$.

---

1. Compute the wavelet transform in (7) at a suitably chosen scale i for several different orientations. The coarsest scale ($i = 1$) is not selected as it is too sensitive to global variations. Finer the scale, the more sensitive it is to distortions such as quantization noise. We choose $i = 3$.

2. Locations (x, y) in the image that are identified as candidate feature points satisfy

$$W_i(x, y, \theta) = \max_{(x', y') \in N_{(x,y)}} |W_i(x', y', \theta)| \quad (8)$$

where $N_{(x,y)}$ represents the local neighborhood of (x, y) within which the search is conducted.

3. From the candidate points selected in step 2, qualify a location as final feature point if

$$\max_{\theta} W_i(x, y, \theta) > T \quad (9)$$

where T is a user-defined threshold.

Figure 1: Feature detection method that preserves significant image geometry feature points of an image.

Previous approaches [3, 7, 8] have used public-key encryption methods on image features to arrive at a digital signature. Such a signature would be very sensitive to small perturbations in the feature points. The feature points detected for perceptually similar images may not always be identical but "close". For example, a small angle rotation would result in the original feature points shifted slightly based on the rotation. More generally, we observe that under perceptually insignificant distortions to the image, the feature points are preserved in a "probabilistic" sense. To maintain perceptual robustness, we quantize the feature vector based on the probability distribution of feature points extracted from the image. In particular, we use the normalized histogram of $\mathbf{f}$ as an estimate of its distribution. The normalized histogram appears to be largely insensitive to attacks that do not cause significant perceptual changes.

3. HASH ALGORITHMS

The hash function for image I is represented as $\mathbf{H}(I)$ and let $D_H(\cdot, \cdot)$ denote the normalized Hamming distance between its arguments (binary strings).

3.1. Algorithm 1 – Deterministic

Mihcak *et al.* [12] observe that primary geometric features of the image are largely invariant under small perturbations to the image. They propose an iterative filtering scheme that minimizes the presence of "geometrically weak components" and enhances "geometrically strong components" by

means of *region growing*. We adapt the algorithm in [12] to lock onto a set of feature-points that are largely preserved in perceptually similar versions of the image. The *stopping criterion* for our proposed iterative algorithm is achieving a *fixed point* for the binary string obtained on quantizing the vector of feature points $\mathbf{f}$. The algorithm follows:

1. Get parameters MaxIter, ϵ and P, set count = 1
2. Use the feature detector in Fig. 1 to extract the length P vector of feature points $\mathbf{f}$
3. Quantize $\mathbf{f}$ in a probabilistic sense to obtain a binary string $\mathbf{b}_f^1$
4. Perform order-statistics filtering. Let $I_{os} = OS(I; p, q, r)$ which is the 2-D *order statistics* filtering of the input I. For a 2-D input X, $Y = OS(X; p, q, r)$ where $\forall i, j$, $Y(i, j)$ is equal to the r^th element of the sorted set of $X(i', j')$, where $i' \in \{i - p, i - p + 1, ..., i + p\}$ and $j' \in \{j - q, j - q + 1, ..., j + q\}$.
5. Perform low-pass linear shift invariant filtering on I_{os} to obtain I_{lp}.
6. Repeat steps (2) and (3) with I_{lp} to obtain $\mathbf{b}_f^2$
7. If (count = maxIter) go to step 8.
   else if $D_H(\mathbf{b}_f^1, \mathbf{b}_f^2) < \epsilon$ go to step 8.
   else set $I = I_{lp}$ and go to step 2.
8. Set $\mathbf{H}(I) = \mathbf{b}_f^2$

Step 4 eliminates isolated significant components. Step 5 preserves the "geometrically strong" components by low-pass filtering (which introduces blurred regions). The success of the deterministic algorithm relies upon the *self-correcting* nature of the iterative algorithm as well as the robustness of the feature detector. The above iterative algorithm is fairly general in that any feature detector that extracts visually robust image features may be used.

3.2. Algorithm 2 – Randomized

Randomizing the hash output is desirable not only for security against inputs designed by an adversary (malicious attacks) but also for scalability, i.e. the ability to work with large data sets while keeping the collision probability for distinct inputs in check. Randomness results from using the secret key K to generate N random locations (row-column pairs) in the image. A *partitioning* of an image into N regions $\{R_i\}_{i=1}^N$ is then obtained using $k-$means clustering [17]. The random locations serve as initial guesses for the cluster centers. Finally, feature points are extracted from each randomly selected region (sub-image) and combined to form the complete feature vector.

We employ k-means clustering to partition the image for two major reasons. First, the clustering achieved by k-means depends heavily on the initial choice of cluster centers [18]. Since the initial centers are generated randomly by the secret key, this makes the partioning also random and in general the same partioning cannot be achieved (with a high probability) unless the secret key is available. Second, the resulting clusters have few or no regions that are very small. It is important to avoid very small regions since they would not yield interesting and robust image features. As long as most of the R_i are big enough the geometric robustness properties of Algorithm 1 are retained in Algorithm 2.

Attack	*Lena*	*Bridge*	*Peppers*
JPEG, QF = 10	0.04	0.04	0.06
AWGN, $\sigma = 20$	0.04	0.03	0.02
Contrast enhancement	0.00	0.06	0.04
Gaussian smoothing	0.01	0.03	0.05
Median filter (3×3)	0.02	0.03	0.07
Scaling by 60%	0.02	0.04	0.05
Shearing by 5%	0.08	0.14	0.10
Rotation by 3°	0.13	0.15	0.15
Rotation by 5°	0.18	0.20	0.19
Cropping by 10%	0.12	0.13	0.15
Cropping by 20%	0.21	0.22	0.24

Table 1: Normalized Hamming distance between hash values of original and attacked (similar) images.

4. RESULTS

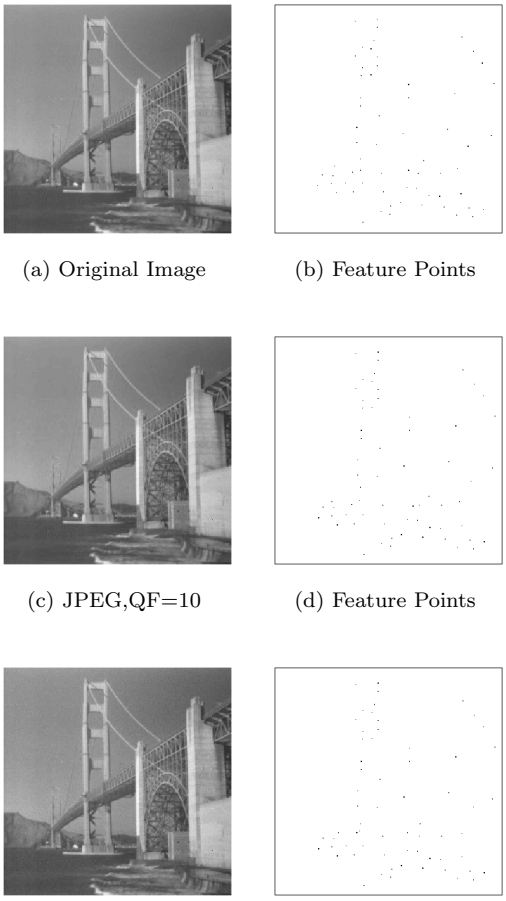
We compare the hash values obtained from two different images for closeness (but not equality) in Hamming distance. Let I_{sim} represents the class of images such that I_{sim} is perceptually similar to I. Likewise, a perceptually distinct image will be denoted by I_{diff}. Then, we require

$$D_H(\mathbf{H}(I), \mathbf{H}(I_{sim})) < 0.2 \quad (10)$$

$$D_H(\mathbf{H}(I), \mathbf{H}(I_{diff})) > 0.3 \quad (11)$$

This is a reasonable approach as perceptual similarity is not transitive. Perceptual similarity of a pair of objects A and B and of another pair B and C does not imply the similarity of A and C. However, modeling of perceptual similarity by equality of hash values would lead to a transitive relationship. A similar approach was taken in [12].

Fig. 2 shows three perceptually similar images and the extracted feature points at algorithm convergence. The original *bridge* image is shown in Fig. 2 (a). Figs. 2(c) and (e), respectively, are the image in (a) attacked by JPEG compression with quality factor (QF) of 10 and additive white Gaussian noise (AWGN) with $\sigma = 20$. The final feature points for the three images (Figs. 2(b), (d) and (f)) are largely the same (or close). Table 1 tabulates the quantitative deviation as the normalized Hamming distance between the hash values of the original and manipulated images for various attacks. The attacked images were generated using the Stirmark benchmark software [19]. The deviation is less than 0.2 except for large rotation (greater than 5°) and cropping (more than 20%). We also tested under several content changing attacks including object insertion and removal, addition of excessive noise, alteration of the position of image elements, and alteration of a significant image characteristic such as texture and structure. In all cases, the detection was accurate. That is, the normalized Hamming distance between the image and its attacked



(a) Original Image (b) Feature Points

(c) JPEG,QF=10 (d) Feature Points

(e) AWGN, $\sigma = 20$ (f) Feature Points

Figure 2: Original/attacked images with feature points at algorithm convergence. Feature points overlaid on images

version was found to be greater than 0.3. The perceptual significance of the hash is hence validated.

5. PERFORMANCE TRADE-OFFS

As the number of feature points P increases, the specification of image characteristics becomes more precise and the fragility to perceptually distinct inputs improves as well. However, the class of perceptually similar inputs becomes smaller. The parameter P facilitates a perceptual robustness vs. fragility trade-off. P is in turn determined by the size of the search neighborhood and the threshold parameter T (e.g. a small T implies more feature points).

When the number of random partitions N is one, no randomness is involved and algorithms 1 and 2 are the same. If N is very large, then the random regions shrink to an extent that they do not contain significant chunks of geometrically strong components and hence the resulting features are not robust. The parameter N facilitates a randomness vs. perceptual robustness trade-off.

The choice of algorithm parameters is governed largely by the application. For image indexing, there is no motivation to randomize ($N = 1$). Also in indexing, the hash computation should be fast, whereas randomization as in Algorithm 2 comes at the cost of partitioning the image prior to feature extraction. For security applications, it may be desirable to randomize as much as possible to minimize the vulnerability against malicious attacks.

6. REFERENCES

[1] A. Menezes, V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1998.

[2] G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Trans. Consumer Electronics*, vol. 39, pp. 905–910, Nov. 1993.

[3] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," *Proc. IEEE Conf. Image Proc.*, Sept. 1996.

[4] C. Kailasanathan and R. Safavi Naini, "Image authentication surviving acceptable modifications using statistical measures and k-mean segmentation," *IEEE-EURASIP Work. Nonlinear Sig. and Image Proc.*, June 2001.

[5] C. Y. Lin and S. F. Chang, "Generating robust digital signature for image/video authentication," *Proc. ACM Work. on Multimedia and Security*, Sept. 1998.

[6] C. Y. Lin and S. F. Chang, "A robust image authentication system distingushing JPEG compression from malicious manipulation," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 153–168, Feb. 2001.

[7] S. Bhatacherjee and M. Kutter, "Compression tolerant image authentication," *Proc. IEEE Conf. Image Proc.*, 1998.

[8] J. Dittman, A. Steinmetz, and R. Steinmetz, "Content based digital signature for motion picture authentication and content-fragile watermarking," *Proc. IEEE Int. Conf. Multimedia Comp. and Sys.*, pp. 209–213, 1999.

[9] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," *Proc. IEEE Int. Conf. Info. Tech.: Coding and Comp.*, Mar. 2000.

[10] C.-S. Lu and H.-Y. M. Liao, "Structural digital signature for image authentication," *IEEE Trans. Multimedia*, pp. 161–173, June 2003.

[11] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," *Proc. IEEE Conf. Image Proc.*, Sept. 2000.

[12] K. Mihcak and R. Venkatesan, "New iterative geometric techniques for robust image hashing," *Proc. ACM Work. Security and Privacy in Dig. Rights Man.*, Nov. 2001.

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas of the cat," *J. Neurophysiology*, pp. 229–289, 1965.

[14] A. Dobbins, S. W. Zucker, and M. S. Cynader, "End-stopping and curvature," *Vision Research*, 1989.

[15] S. Bhatacherjee and P. Vandergheynst, "End-stopped wavelets for detecting low-level features," *Proc. SPIE Wavelet Appl. in Sig. & Image Proc.*, pp. 732–741, 1999.

[16] J.-P. Antoine and R. Murenzi, "Two-dimensional directional wavelets and the scale-angle representation," *Sig. Proc.*, pp. 259–281, 1996.

[17] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. Sym. Math, Statistics and Probability*, pp. 281–297, 1967.

[18] A. Gersho and R. M. Gray, *Vector Quantization and Sig. Compression*, Kluwer Academic, 1991.

[19] "Fair evaluation procedures for watermarking systems," http://www.petitcolas.net/fabien/watermarking/stirmark, 2000.