Perceptual Image Hashing Via Feature Points: Performance Evaluation And Trade-Offs

Vishal Monga, *Student Member, IEEE*, and Brian L. Evans, *Senior Member, IEEE*

Abstract—

We propose an image hashing paradigm using visually significant 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 introduce randomness, which in turn reduces vulnerability to adversarial attacks. 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. Detailed statistical analysis in the form of receiver operating characteristic (ROC) curves is presented and reveals the success of the proposed scheme in achieving perceptual robustness while avoiding misclassification.

EDICS category— (5-DTBS, 5-SRCH, 5-AUTH)

Index terms— hashing, feature extraction, image indexing, image authentication.

I. 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 function should have the property that two images that look the same to the human eye map to the same hash value, even if the images have different digital representations, e.g. separated by a large distance in mean squared error.

An immediately obvious application for a *perceptual* image hash is identification/search of images in large databases. Several other applications have been identified recently in content authentication, watermarking [2], and anti-piracy search. Unlike traditional search, these scenarios are adversarial, and require the hash to be a randomized digest.

The underlying techniques for constructing image hashes can roughly be classified into methods based on image statistics [3], [4] [5], relations [6], [7], preservation of coarse image representation [8], [9], [10], and low-level image feature extraction [11], [12].

The approaches in [3], [4] compute statistics such as mean, variance, and higher moments of intensity values of image blocks. They conjecture that such statistics have good robustness properties under small perturbations to the image. A serious drawback with these methods is that it is easy to modify an image without altering its intensity histogram. This jeopardizes the security properties of any scheme that relies on intensity statistics. Venkatesan *et al.* [5] develop an image hash based on an image statistics vector extracted from the various sub-bands in a wavelet decomposition of the image. They observe that statistics such as averages of coarse sub-bands and variances of other (fine detail) sub-bands stay invariant under a large class of content-preserving modifications to the image. Although statistics of wavelet coefficients have been found to be far more robust than intensity statistics, they do not necessarily capture content changes[1] well, particularly those that

V. Monga is with the Xerox Innovation Group in El Segundo, CA, vishal.monga@xeroxlabs.com. B. L. Evans is with the Embedded Signal Processing Laboratory in the Dept. of Electrical and Computer Engineering at The University of Texas at Austin, Austin, TX 78712 USA, bevans@ece.utexas.edu. Research was supported by a gift from the Xerox Foundation and an equipment donation from Intel.

[1] A content change here signifies a perceptually meaningful perturbation to the image, e.g. adding/removing an object, significant change in texture, morphing a face etc. In general, a perceptual hash should be sensitive to both incidental as well as malicious content changes. A major challenge in secure image hashing is to develop al-

are maliciously generated.

A typical relation-based technique for image authentication, which tolerates JPEG compression, has been reported by Lin and Chang [6]. They extract a digital signature by using the invariant relationship between any two discrete cosine transform (DCT) coefficients, which are at the same position of two different 8 × 8 blocks. They found these invariance properties could be preserved before and after JPEG compression as long as it is perceptually lossless. This scheme, although robust to JPEG compression, remains vulnerable to several other perceptually insignificant modifications, e.g. where the statistical nature of distortion is different from the blur caused by compression. Recently, Lu *et al.* [7] have proposed a "structural digital signature" for image authentication. They observe that in a sub-band wavelet decomposition, a parent and child node are uncorrelated, but they are statistically dependent. In particular, they observe that the difference of the magnitude of wavelet coefficients at consecutive scales (i.e. a parent node and its 4 child nodes) remains largely preserved for several content-preserving manipulations. Identifying such parent-child pairs and subsequently encoding the pairs form their robust digital signature. Their scheme, however, is very sensitive to global (e.g. small rotation and bending) as well as local (Stirmark) geometric distortions, which do not cause perceptually significant changes to the image.

Fridrich and Goljan [8] propose a robust hash based on preserving selected (low-frequency) DCT coefficients. Their method is based on the observation that large changes to low frequency DCT coefficients of the image are needed to change the appearance of the image significantly. Mihcak and Venkatesan [9] develop another image hashing algorithm by using an iterative approach to binarize the DC subband (lowest resolution wavelet coefficients) in a wavelet decomposition of the image. Very recently, Kozat *et al.* [10] proposed an image hashing scheme by retaining the strongest singular vectors and values in a Singular Value Decomposition (SVD) of image blocks. Approaches based on coarse image representations [8], [9], [10] have been shown to possess excellent robustness under perceptually insignificant modifications to the image, but they remain vulnerable to local tampering or content changes.

In this paper, we present a framework for perceptual image hashing using feature points. Feature point detectors are attractive for their inherent sensitivity to content changing manipulations. Current approaches based on feature points [11], [12] have limited utility in perceptual hashing applications because they are sensitive to several perceptually insignificant modifications as well. We propose to 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 [9] 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 [11], [12] probabilistic quantization is used to binarize the extracted feature vector.

We develop both deterministic as well as randomized hash algorithms. Randomization has been identified [5], [9], [10] to be significant for lending unpredictability to the hash and hence reducing the vulnerability of the hash algorithm to malicious inputs generated by an adversary.

Our hash exhibits excellent robustness under benchmark attacks (e.g. Stirmark) including compression, geometric distortions of scaling and small angle rotation, and common signal processing operations while successfully discriminating between perceptually distinct images.

Section II-A formally defines the desired properties of a perceptual image hash. Section II-B then presents a novel two-stage framework for perceptual image hashing that consists of feature extraction followed by feature vector compression. We call the output of the first stage an intermediate hash. The rest of the paper then focuses on building such intermediate hash vectors. Section III presents a robust feature detector based on visually significant end-stopped wavelets [14]. Section IV presents a probabilistic quantization approach to binarize image feature vectors that enhances robustness, and at the same time, introduces randomness. Iterative algorithms (both deterministic and randomized) that construct intermediate hash vectors are described in Section V. Experimental results demonstrating perceptual robustness, sensitivity to content changes, and ROC analysis across 1000 different images are reported in Sections VI-A through VI-D. Concluding remarks and suggestions for future work are collected in Section VII.

II. A unifying framework for perceptual hashing

A. Perceptual Image Hash: Desired Properties

Section I describes many possible applications for an image hash. The hash algorithm developed in this paper however, is specifically targeted at two scenarios: 1.) content authentication, and 2.) anti-piracy search.

The former requires that the hash should remain invariant if the image undergoes a "content preserving" (even though lossy) transformation. Then a robust hash can serve as substitute for a robust watermark meant to be retained as long as the image is not significantly changed. The difference with watermarking is that hashing is passive and does not require any embedding into the image. Anti-piracy search is aimed at thwarting a pirate or attacker who may claim ownership of proprietary image content by suitably modifying it. As an example consider a pirate/attacker who may steal images from an official website and post them as his own by compressing them or sending them through geometric distortions such as print-scan. Since comparing with all images on the web is impractical, the true/official content owner may employ a web-crawler software that computes hashes of images on randomly accessed webpages. If the hashes match those of their own image content, pirates can be caught.

gorithms that can detect (with high probability) malicious tampering of image data.

In view of the above, we develop formally, the desired properties of a perceptual image hash. Let $\mathcal{I}$ denote a set of images (e.g., all natural images of a particular size) with finite cardinality. Also, let $\mathcal{K}$ denote the space of *secret* keys[2]. Our hash function then takes two inputs, an image $I \in \mathcal{I}$ and a *secret* key $K \in \mathcal{K}$, to produce a q-bit binary hash value $h = H(I, K)$. Let $I_{ident} \in \mathcal{I}$ denote an image such that I_{ident} looks the same as I. Likewise, an image in $\mathcal{I}$, perceptually distinct from I will be denoted by I_{diff}. Let θ_1, θ_2 satisfy $0 < \theta_1, \theta_2 < 1$: Then, three desirable properties of a *perceptual* hash are identified as:

1. **Perceptual robustness**:
$Probability(H(I, K) = H(I_{ident}, K)) \geq 1 - \theta_1$, for a given θ_1
2. **Fragility to visually distinct images**:
$Probability(H(I, K) \neq H(I_{diff}, K)) \geq 1 - \theta_2$, for a given θ_2
3. **Unpredictability of the hash**:
$Probability(H(I, K) = v) \approx \frac{1}{2^q}, \quad \forall\ v \in \{0, 1\}^q$

Let $\mathcal{Q} = \{H(I, K) \mid I \in \mathcal{I},\ K \in \mathcal{K}\}$; i.e., the set of all possible realizations of the hash algorithm on the product space $\mathcal{I} \times \mathcal{K}$. Also, for a fixed $I_0 \in \mathcal{I}$ define $\mathcal{O} = \{H(I_0, K) \mid K \in \mathcal{K}\}$; i.e., for a fixed image the set of all possible realizations of the hash algorithm over the key space $\mathcal{K}$.

Note that the probability measure in the first two properties is defined over the set $\mathcal{Q}$. For example, property 1 requires that for any pair of "perceptually identical" images in $\mathcal{I}$ and any $K \in \mathcal{K}$ the hash values must be identical with high probability. The probability measure in the third property however, is defined on $\mathcal{O}$; i.e., the third property requires that as the secret key is varied over $\mathcal{K}$ for a fixed input image; the output hash value must be approximately uniformly distributed among all possible q-bit outputs.

Further, the three desirable hash properties conflict with one another. The first property amounts to robustness under small perturbations whereas the second one requires minimization of collision probabilities for perceptually distinct inputs. There is clearly a trade-off here; e.g., if very crude features were used, then they would be hard to change (i.e., robust), but it is likely that one is going to encounter collision of perceptually different images. Likewise for perfect randomization, a uniform distribution on the output hash values (over the key space) would be needed which in general, would deter achieving the first property. From a security viewpoint, the second and third properties are very important; i.e., it must be extremely difficult for the adversary to manipulate the content of an image and yet obtain the same hash value. It is desirable for the hash algorithm to achieve these (conflicting) goals to some extent and/or facilitate trade-offs.

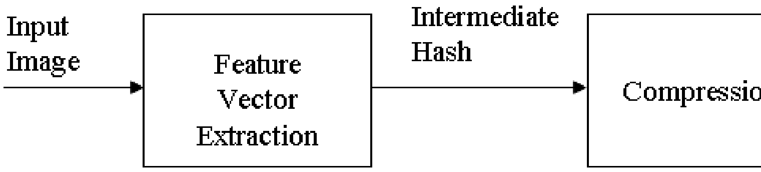


Fig. 1. Block diagram of the Hash Function

B. Hashing Framework

We partition the problem of deriving an image hash into two steps, as illustrated in Fig. 1. The first step extracts a feature vector from the image, whereas the second stage compresses this feature vector to a final hash value. In the feature extraction step, the two-dimensional image is mapped to a one-dimensional feature vector. This feature vector must capture the perceptual qualities of the image. That is, two images that appear identical to the human visual system should have feature vectors that are close in some distance metric. Likewise, two images that are clearly distinct in appearance must have feature vectors that differ by a large distance. For the rest of the paper, we will refer to this visually robust feature vector (or its quantized version) as the "intermediate hash". The second step then compresses this intermediate hash vector to a final hash value.

This paper focuses on a feature-point based solution to stage 1 of the hash algorithm. Solutions for stage 2 have been proposed in [5], [15], [16], [17].

III. Feature Extraction

For the perceptual hashing problem, two major attributes of the feature detector are identified: (1) generality, and (2) robustness. The generality criterion addresses the issue of whether or not the feature detector can be used over a variety of images and applications. Robustness is desirable for the features to be retained in perceptually identical images.

Feature detection continues to be an important vision problem and a plethora of algorithms have been reported. For an extensive coverage of feature detection, the reader is referred to [18]. In this Section, we review selected and best known robust interest point detectors that are particularly well-suited for the hashing problem. We also propose our feature detection method and experimentally evaluate it against existing approaches.

A. Harris Detector

Possibly the best known corner detector, the Harris detector [19] uses differential features of the image. The construction of the detector is based on a corner detection created by Moravec [20]. The response function $E_{x,y}$ is calculated for a shift (x, y) from the central point (u, v):

[2] The key space in general can be constructed in several ways. A necessary but not sufficient condition for secure hashing is that the key space should be large enough to preclude exhaustive search. For this paper, unless specified explicitly we will assume the key space to be the hamming space of 32-bit binary strings.

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u,y+v} - I_{x,y}|^2 \tag{1}$$

where $I_{u,v}$ represents the luminance of the image at the co-ordinate (u, v), and the function $w_{u,v}$ represents a rectangular Gaussian response window centered at (u, v). In essence, this corner detector functions by considering a local window in the image (represented by the span of $w(x, y)$), and determining the average changes of image intensity by shifting the window by small amounts in various directions. A corner can thus be detected when the change produced by any of the shifts (i.e. in all possible directions) is large.

Harris reformulated the detection function using a matricial formulation. Let

$$\mathbf{x} = \left[\begin{array}{c} x \\ y \end{array} \right] \quad I_x = \tfrac{\partial I}{\partial x} \quad I_y = \tfrac{\partial I}{\partial y} \tag{2}$$

$$A = (I_x)^2 * w \quad B = (I_y)^2 * w \quad C = I_x I_y * w \tag{3}$$

The detection function $E_{x,y}$ is then given by $\mathbf{x}^T\mathbf{M}\mathbf{x}$ and

$$\mathbf{M} = \left[\begin{array}{cc} A_{x,y} & C_{x,y} \\ C_{x,y} & B_{x,y} \end{array} \right] \tag{4}$$

$E_{x,y}$ may hence be interpreted as the auto-correlation of the image with the shape function $\mathbf{M}$. Harris and Stephens gave a new definition of the detector function using the eigenvalues α and β of the matrix $\mathbf{M}$. These values are invariant by rotation and if their magnitudes are high, the local auto-correlation function is represented by a local peak. To avoid computing the eigenvalue of $\mathbf{M}$, the new criterion is based on the trace and determinant of $\mathbf{M}$

$$\mathbf{M} = \begin{array}{rccl} Tr(\mathbf{M}) = & \alpha + \beta & = A + B \\ det(\mathbf{M}) = & \alpha.\beta & = A.B - C^2 \\ R_H = & det(\mathbf{M}) & -k.(Tr(\mathbf{M}))^2 \end{array} \tag{5}$$

where k is an arbitrary constant. Feature points extraction is achieved by applying a threshold on the response R_H and searching for local maxima.

B. Hessian Affine

A similar idea is explored in the detector based on the Hessian matrix [21]. The difference is in the matrix $\mathbf{M}$ which is now given by:

$$\mathbf{M} = \left[\begin{array}{cc} I_{x,x} & I_{x,y} \\ I_{x,y} & I_{y,y} \end{array} \right] * w \tag{6}$$

where $I_{xx} = \frac{\partial^2 I}{\partial x^2}$ and $I_{yy} = \frac{\partial^2 I}{\partial y^2}$.

The second derivatives, which are used in this matrix give strong responses on blobs and ridges. The interest points are similar to those detected by a Laplacian operator but a function based on the determinant of the Hessian matrix [21] penalizes very long structures for which the second derivative in one particular orientation is very small. A local maximum of the determinant indicates the presence of a bob structure.

C. Maximally Stable Extremal Region (MSER) Detector

A Maximally Stable Extremal Region (MSER) [22] is a connected component of an appropriately thresholded image. The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary. The maximally stable in MSER describes the property optimized in the threshold selection process.

The set of extremal regions $\mathcal{E}$, i.e., the set of all connected components obtained by thresholding, has a number of desirable properties. Firstly, a monotonic change of image intensities leaves $\mathcal{E}$ unchanged, since it depends only on the ordering of pixel intensities which is preserved under monotonic transformation. This ensures that common photometric changes modelled locally as linear or affine leave $\mathcal{E}$ unaffected. Secondly, continuous geometric transformations preserve topology pixels from a single connected component are transformed to a single connected component. Thus after a geometric change locally approximated by an affine transform, homography or even continuous non-linear warping, a matching extremal region will be in the transformed set $\mathcal{E}^{'}$. Finally, there are no more extremal regions than there are pixels in the image. So a set of regions was defined that is preserved under a broad class of geometric and photometric changes and yet has the same cardinality as e.g. the set of fixed-sized square windows commonly used in narrow-baseline matching.

For a detailed description of particular techniques to select extremal regions, the reader is referred to [22].

D. Feature Detection Based on End-stopping Behavior of the Visual System

D.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 [23], [14]. The term end-stopped comes from the strong sensitivity of these cells to end-points of linear structures. Bhattacherjee *et al.* [14] construct “end-stopped” wavelets to capture this behavior. The construction of the wavelet kernel (or basis function) combines two operations. First, linear structures having a certain orientation are selected. These linear structures are then processed to detect line-ends (corners) and/or high curvature points.

Morlet wavelets can be used to detect linear structures having a specific orientation. In the spatial domain, the two dimensional (2-D) Morlet wavelet is given by [24]

$$\psi_M(\mathbf{x}) = (e^{j\mathbf{k_0}\cdot\mathbf{x}} - e^{-\frac{1}{2}|\mathbf{k_0}|^2})(e^{-\frac{1}{2}|\mathbf{x}|^2}) \tag{7}$$

where $\mathbf{x} = (x, y)$ represents 2-D spatial coordinates, and $\mathbf{k_0} = (k_0, k_1)$ is the *wave-vector* of the mother wavelet, which determines scale-resolving power and angular-resolving power of the wavelet [24]. The frequency domain representation, $\psi_M(\mathbf{k})$, of a Morlet wavelet is

$$\hat{\psi}_M(\mathbf{k}) = (e^{-\frac{1}{2}|\mathbf{k}-\mathbf{k_0}|^2} - e^{-\frac{1}{2}|\mathbf{k_0}|^2})(e^{-\frac{1}{2}|\mathbf{k}|^2}) \qquad (8)$$

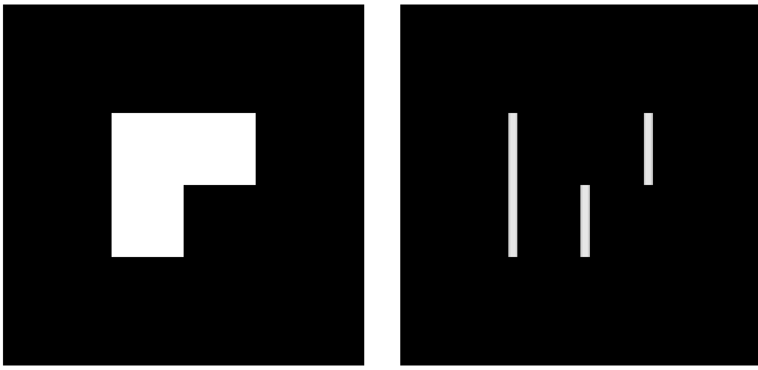
Here, $\mathbf{k}$ represents the 2-D frequency variable (u, v). The Morlet function is similar to the Gabor function, but with an extra correction term $e^{-\frac{1}{2}(|\mathbf{k_0}|^2+|\mathbf{x}|^2)}$ to make it an admissible wavelet. The orientation of the wave-vector determines the orientation tuning of the filter. A Morlet wavelet detects linear structures oriented perpendicular to the orientation of the wavelet.

In two dimensions, the end points of linear structures can be detected by applying the first-derivative of Gaussian (FDoG) filter in the direction parallel to the orientation of structures in question. The first filtering stage detects lines having a specific orientation and the second filtering stage detects end-points of such lines. These two stages can be combined into a single filter to form an "end-stopped" wavelet [14]. An example of an end-stopped wavelet and its 2-D Fourier transform follow:

$$\psi_E(x,y) = \frac{1}{4} y e^{-\left(\frac{x^2+y^2}{4} + \frac{k_0}{4}(k_0 - 2jx)\right)} \qquad (9)$$

$$\hat{\psi}_E(u,v) = 2\pi \left(e^{-\frac{(u-k_0)^2+(v)^2}{2}}\right)\left(jve^{-\frac{u^2+v^2}{2}}\right) \qquad (10)$$

Eqn. (10) shows $\hat{\psi}_E$ as a product of two factors. The first factor is a Morlet wavelet oriented along the $u-$axis. The second factor is a FDoG operator applied along the frequency-axis v, i.e. in the direction perpendicular to the Morlet wavelet. Hence, this wavelet detects line ends and high curvature points in the vertical direction. Fig. 2 illustrates the behavior of the end-stopped wavelet as in (9)-(10). Fig. 2 (a) shows a synthetic image with L-shaped region surrounded by a black background. Fig. 2 (b) shows the raw response of the vertically oriented Morlet wavelet at scale $i = 2$. Note that this wavelet responds only to the vertical edges in the input. The response of the end-stopped wavelet is shown in Fig. 2 (c) also at scale $i = 2$. The responses are strongest at end-points of vertical structures and negligibly small elsewhere. The local maxima of these responses in general correspond to corner-like stimuli and high curvature points in images.



(a) Synthetic L-shaped image

(b) Response of a Morlet wavelet, orientation = 0^o

Fig. 2. Behavior of the end-stopped wavelet on a synthetic image: note the strong response to line-ends and corners

1. Compute the wavelet transform in (13) 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. The 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

$$W_i(x,y,\theta) = \max_{(x',y')\in N_{(x,y)}} |W_i(x',y',\theta)| \qquad (14)$$

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 a final feature point if

$$\max_{\theta} W_i(x,y,\theta) > T \qquad (15)$$

where T is a user-defined threshold.

Fig. 3. Feature detection method that preserves significant image geometry feature points of an image.

D.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:

$$\psi_E(x,y,\theta) = (FDoG)\ o(\psi_M(x,y,\theta)) \qquad (11)$$

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

$$\left(\psi_E(\alpha^i(x,y,\theta_k))\right), \alpha \in \mathcal{R},\ i \in \mathcal{Z} \qquad (12)$$

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

$$W_i(x,y,\theta) = \int\int f(x_1,y_1)\psi_E{}^*\left(\alpha^i(x-x_1,y-y_1),\theta\right) dx_1 dy_1 \qquad (13)$$

The sampling parameter α is chosen to be 2.

Fig. 3 describes the proposed feature detection method. Step 1 computes the wavelet transform in (13) for each image location. Step 2 identifies significant features by looking for local maxima of the magnitude of the wavelet coefficients in a preselected neighborhood. We chose a circular neighborhood to avoid increasing detector anisotropy. Step 3 applies thresholding to eliminates spurious local maxima in featureless regions of the image.

The method in Fig. 3 has two free parameters: integer scale i and real threshold T. The threshold T is adapted to select a fixed number (user defined parameter P) of feature points from the image. An image feature vector is formed by collecting the magnitudes of the wavelet coefficients at the selected feature points.

E. Detector Evaluation

To evaluate a detector for the robust hashing application, we *employ* a robustness function $\mathcal{R}$ defined as

$$\mathcal{R} = \frac{N_{ret} - (N_{new} + N_{rem})}{N_{ret} + N_{rem}} \tag{16}$$

where $N_{tot} = N_{ret} + N_{rem}$ represents the total number of feature points detected in the original image. N_{ret} denotes the number of feature points after the image goes a content preserving (but lossy) transformation. Similarly, N_{rem} represents the number of feature points that are removed as a result of the transformation, and N_{new} is the number of new feature points that may be introduced. Clearly, the maximum value of this function is 1 which happens when $N_{new} = N_{rem} = 0$.

We obtained this function for the four feature point detectors reviewed above under a large class of perceptually insignificant and typically allowable transformations including JPEG image compression, rotation, scaling, additive white Gaussian noise addition, print-scan and linear/non-linear image filtering operations. Then an averaged evaluation measure $\mathcal{R}_{ave}$ was obtained by averaging the values of $\mathcal{R}$ from each transformation.

A plot of $\mathcal{R}_{ave}$ for tests on four different images is plotted for the four detectors in Fig. 4. Clearly, the best values of $\mathcal{R}_{ave}$ are obtained for the feature detector based on end-stopped wavelets followed by MSER. In practice, we observed that while most detectors fared comparably under geometric attacks of rotation, scaling and translation, the real gains of the end-stopped wavelet based detector were under lossy transformations such as compression, noise addition and non-linear filtering.

IV. Probabilistic Quantization

Let $\mathbf{f}$ denote the length P feature vector. The next step then is to obtain a binary string from the feature vector that would form the intermediate hash. Previous approaches have [3], [11] used public-key encryption methods on image features to arrive at a digital (binary) signature. Such a signature would be very sensitive to small perturbations in the extracted features (here, the magnitude of the wavelet coefficients). We observe that under perceptually insignificant distortions to the image, although the actual magnitudes of the wavelet coefficients associated with the feature points may change, the "distribution" of the magnitudes of the wavelet coefficients is still preserved.

In order to maintain robustness, we propose a quantization scheme based on the probability distribution of the features extracted from the image. In particular, we use the normalized histogram of the feature vector $\mathbf{f}$ as an estimate of its distribution. The normalized histogram appears to be largely insensitive to attacks that do not cause significant perceptual changes. In addition, a randomization rule [25] is also specified which adds unpredictability to the quantizer output.

Let L be the number of quantization levels, $\mathbf{f_q}$ denote the quantized version of $\mathbf{f}$, $\mathbf{f}(k)$ and $\mathbf{f_q}(k)$ denote the k^{th} elements of $\mathbf{f}$ and $\mathbf{f_q}$, respectively. The binary string obtained from the quantized feature vector $\mathbf{f_q}$ is hence of length $P\lceil\log_2(L)\rceil$ bits. If quantization were deterministic, the quantization rule would be given by

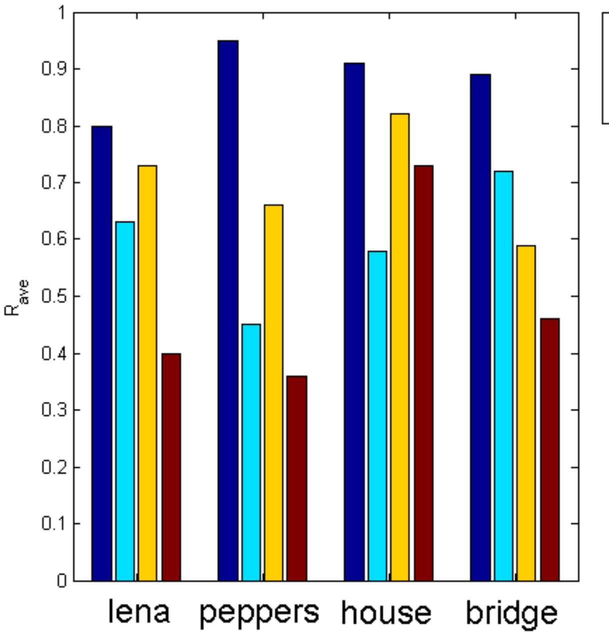


Fig. 4. Detector benchmark $\mathcal{R}_{ave}$ for the four detectors across four different images. Note, the closer $\mathcal{R}_{ave}$ is to 1, better the performance.

$$l_{i-1} \leq \mathbf{f}(k) < l_i, \quad \mathbf{f_q}(k) = i \tag{17}$$

where $[l_{i-1}, l_i)$ is the $i^t h$ quantization bin. Note, the quantized values are chosen to be i, $1 \leq i \leq L$. This is because unlike traditional quantization for compression, there is no constraint on the quantization levels for the hashing problem. These may hence be designed to convenience as long as the notion of "closeness" is preserved. Here, we design quantization bins $[l_{i-1}, l_i)$ such that

$$\int_{l_{i-1}}^{l_i} p_f(x)dx = \frac{1}{L}, \quad 1 \leq i \leq L \tag{18}$$

where $p_f(x)$ is the estimated distribution of $\mathbf{f}$. This ensures that the quantization levels are selected according to the distribution of image features. In each interval $[l_{i-1}, l_i)$, we obtain center points C_i with respect to the distribution, given by

$$\int_{C_i}^{l_i} p_f(x)dx = \int_{l_{i-1}}^{C_i} p_f(x)dx = \frac{1}{2L} \tag{19}$$

Then, we find *deviations* P_i, Q_i about C_i where $l_{i-1} \leq P_i \leq C_i$ and $C_i \leq Q_i \leq l_i$, such that

$$\frac{\int_{C_i}^{Q_i} p_f(x)dx}{\int_{C_i}^{l_i} p_f(x)dx} = \frac{\int_{P_i}^{C_i} p_f(x)dx}{\int_{l_{i-1}}^{C_i} p_f(x)dx}, \quad 1 \leq i \leq L \tag{20}$$

P_i, Q_i are hence symmetric around C_i with respect to the distribution $p_f(x)$. By virtue of the design of C_i's in (19), the denominators in (20) are both equal to $\frac{1}{2L}$ and hence only the numerators need to be computed. The probabilistic quantization rule is then completely given by

$$P_i < \mathbf{f}(k) < Q_i, \quad \mathbf{f_q}(k) = \begin{cases} i & \textit{with probability } \frac{\int_{P_i}^{\mathbf{f}(k)} p_f(x)dx}{\int_{P_i}^{Q_i} p_f(x)dx} \\ i-1 & \textit{with probability } \frac{\int_{\mathbf{f}(k)}^{Q_i} p_f(x)dx}{\int_{P_i}^{Q_i} p_f(x)dx} \end{cases}$$

$$l_{i-1} \leq \mathbf{f}(k) \leq P_i, \quad \mathbf{f_q}(k) = i-1 \quad \textit{with probability} \quad 1 \tag{21}$$

$$Q_i \leq \mathbf{f}(k) \leq l_i, \quad \mathbf{f_q}(k) = i \quad \textit{with probability} \quad 1 \tag{22}$$

The output of the quantizer is deterministic except in the interval (P_i, Q_i) Note, if $\mathbf{f}(k) = C_i$ for some i, k then the assignment to levels i or $i-1$ takes place with equal probability, i.e. 0.5, the quantizer output in other words is completely randomized. On the other hand, as $\mathbf{f}(k)$ approaches P_i or Q_i the quantization decision becomes almost deterministic.

In the next section, we present iterative algorithms that employ the feature detector in Section III-D.2, and the quantization scheme described in this section to construct binary intermediate hash vectors.

V. Intermediate Hash Algorithms

A. Algorithm 1 - Deterministic

The intermediate 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).

Mihcak *et al.* [9] 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 [9] 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 $\mathbf{f}$.

Fig. 4 describes the proposed intermediate hash algorithm. 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.

B. 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. The algorithm as presented in Fig. 4 does not make use of a secret key and hence there is no randomness involved.

```
1. Get parameters MaxIter, $\rho$ and $P$ (number of features), and set count = 1
2. Use the feature detector in Fig. 3 to extract the length $P$ feature vector $\mathbf{f}$.
3. Quantize $\mathbf{f}$ according to the rule given by (17) and (18) (i.e. deterministic quantization) 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 $j' \in \{j-q, j-q+1, ..., j+q\}$. Note, for $r = (2p+1)(2q+1)/2$ this is same as median filtering.
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 $D_H(\mathbf{b_f^1}, \mathbf{b_f^2}) < \rho$ go to step 8.
else set $I = I_{lp}$ and go to step 2.
8. Set $\mathbf{h}(I) = \mathbf{b_f^2}$
```

Fig. 5. Deterministic intermediate hash algorithm

In this section, we will construct randomized hash algorithms using a secret key K, which is used as the seed to the pseudo-random number generator for the randomization steps in the algorithm. For this reason, we now denote the intermediate hash vector as $\mathbf{h}(I, K)$, i.e. function of both the image, and the secret key. We present a scheme that employs a *random partitioning* of the image to introduce unpredictability in the hash values. A stepwise description is given in Fig. 5.

Qualitatively, Algorithm 2 enhances the security of the hash by employing Algorithm[3] 1 on randomly chosen regions or sub-images. As long as these sub-images are sufficiently unpredictable (i.e. they differ significantly as the secret key is varied), then the resulting intermediate hashes are also different with high probability[4]. Examples of random partitioning of the *lena* image using Algorithm 2, are shown in Fig. 7. In each case, i.e. Figs. 7 (a), (b), and (c), a different secret key was used.

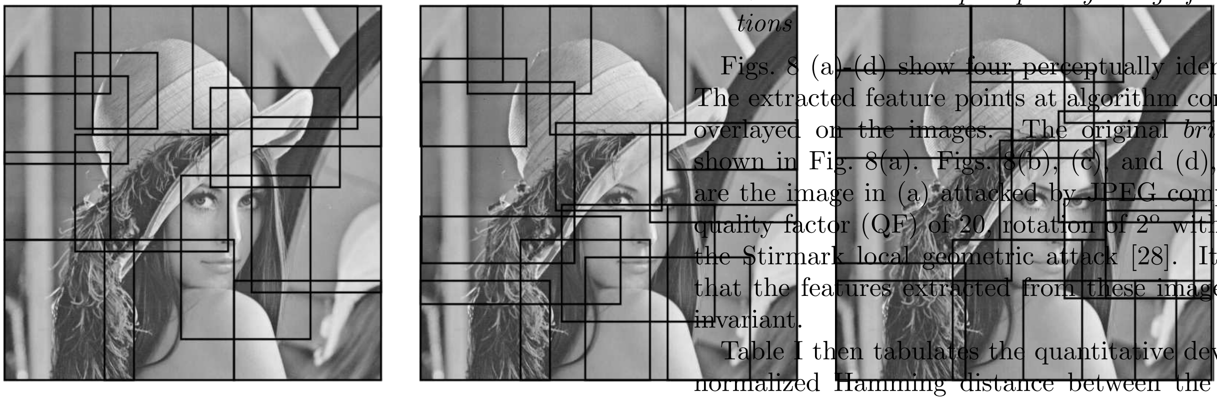
The approach of dividing the image into random rectangles for constructing hashes was first proposed by Venkatesan *et al.* in [5]. However, their algorithm is based on image statistics. In our framework, by applying the fea-

[3] This would now use a probabilistic quantizer.

[4] Although we do not implement it, as suggested by an anonymous reviewer another avenue for randomization is to employ the feature detector in Fig. 3 on randomly chosen wavelet scales. In that case however, the choice of wavelet coefficients (from different scales) should be done very carefully to retain necessary hash robustness.

1. (Random Partitioning) Divide the image into N (overlapping) random regions. In general, this can be done in several ways. The main criterion is that a different partitioning should be obtained (with high probability) as the secret key is varied. In our implementation, we divide the image into overlapping circular/elliptical regions with randomly selected radii. Label, these N regions as C_i, $i = 1, 2, ..., N$.
2. (Rectangularization) Approximate each C_i by a rectangle using a *waterfilling* like approach. Label the resulting random rectangles (consistent with the labels in step 1.) as R_i, $i = 1, 2, ..., N$.
3. (Feature Extraction) Apply Algorithm 1 on all R_i, denote the binary string extracted from each R_i as $\mathbf{b}_i$. Concatenate all $\mathbf{b}_i$'s into a single binary vector $\mathbf{b}$ of length B bits.
4. (Randomized Subspace Projection) Let $A < B$ be the desired length of $\mathbf{h}(I, K)$. Randomly choose distinct indices $i_1, i_2, ..., i_A$ such that each $i_m \in [1, B], m = 1, 2, ..., A$.
5. The intermediate hash $\mathbf{h}(I, K) = \{\mathbf{b}(i_1), \mathbf{b}(i_2), ..., \mathbf{b}(i_A)\}$

Fig. 6. Randomized intermediate hash algorithm



(a) Secret key K_1

(b) Secret key K_2

(c) Secret key K_3

Fig. 7. Examples of random partitioning of the *lena* image into $N = 13$ rectangles. Note that the random regions vary significantly based on the secret key.

ture point detector to these semi-global rectangles, we gain an additional advantage in capturing any local tampering of image data (results presented later in Section VI-B). These rectangles in Fig. 7 are deliberately chosen to be overlapping to further reduce the vulnerability of the algorithm to malicious tampering. Finally, the randomized sub-space projection step adds even more unpredictability to the intermediate hash. Trade-offs among randomization, fragility and perceptual robustness are analyzed later in Section VI-C.

VI. Results

We compare the binary intermediate hash vectors obtained from two different images for closeness in (normalized) Hamming distance. Recall from Section II-A, that $(I, I_{ident}) \in \mathcal{I}$ denote a pair of perceptually identical images, and likewise $(I, I_{diff}) \in \mathcal{I}$ represent perceptually distinct images. Then, we require

$$D_H(\mathbf{h}(I), \mathbf{h}(I_{ident})) < \epsilon \tag{23}$$

$$D_H(\mathbf{h}(I), \mathbf{h}(I_{diff})) > \delta \tag{24}$$

where the natural constraints $0 < \epsilon < \delta$ apply. For results presented in Sections VI-A and VI-B, the following parameters were chosen for Algorithm 1: a circular (search) neighborhood of 3 pixels was used in the feature detector, $P = 64$ features were extracted, the order statistics filtering was $OS(3, 3, 4)$ and a zero-phase 2-D FIR low-pass filter of size 5×5 designed using McClellan transformations [26] was employed. For Algorithm 2, the same parameters were used except that the image was partitioned into $N = 32$ random regions. For this choice of parameters, we experimentally determine $\epsilon = 0.2$ and $\delta = 0.3$. A more elaborate discussion of how to choose the best ϵ and δ will be given in Section VI-D. All input images were resized to 512×512 using bicubic interpolation [27]. For color images, both Algorithm 1 and 2 were applied to the luminance plane since it contains most of the geometric information.

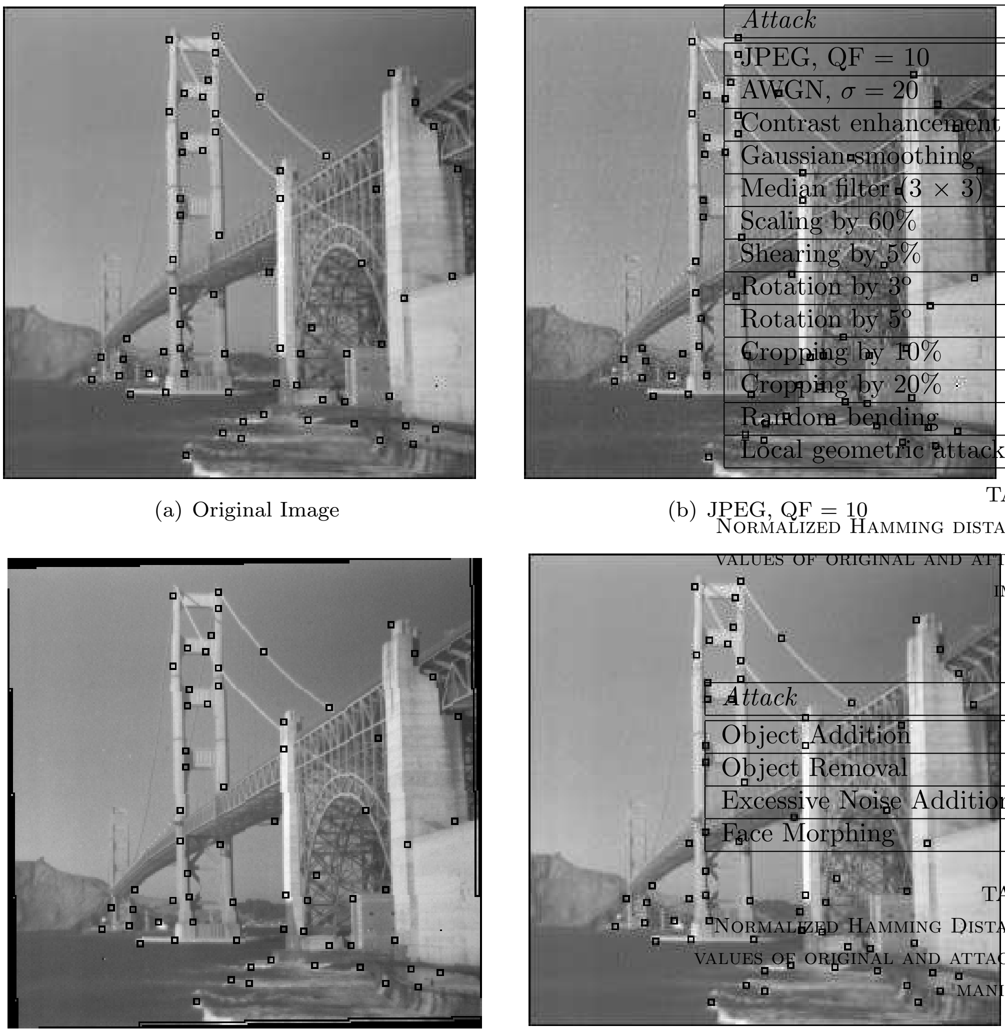
A. Robustness under perceptually insignificant modifications

Figs. 8 (a)-(d) show four perceptually identical images. The extracted feature points at algorithm convergence are overlayed on the images. The original *bridge* image is shown in Fig. 8(a). Figs. 8(b), (c), and (d), respectively, are the image in (a) attacked by JPEG compression with quality factor (QF) of 20, rotation of 2° with scaling, and the Stirmark local geometric attack [28]. It can be seen that the features extracted from these images are largely invariant.

Table I then tabulates the quantitative deviation as the normalized Hamming distance between the intermediate hash values of the original and manipulated images for various perceptually insignificant distortions. The distorted images were generated using the Stirmark benchmark software [28]. The results in Table I reveal that the deviation is less than 0.2 except for large rotation (greater than 5°), and cropping (more than 20%). More severe geometric attacks like large rotation, affine transformations, and cropping can be handled by a search based scheme. Details may be found in [29].

B. Fragility to Content Changes

The essence of our feature point based hashing scheme lies in projecting the image onto a visually meaningful wavelet basis, and then retaining the strongest coefficients to form the content descriptor (or hash). Our particular choice of the basis functions, i.e. end-stopped type exponential kernels, yield strong responses in parts of the image where the significant image geometry lies. It is this very characteristic that makes our scheme attractive for detecting content changing image manipulations. In particular, we observe that a visually meaningful content change is



(a) Original Image

(b) JPEG, QF = 10

(c) 2° rotation and scaling

(d) Stirmark local geometric attack

Fig. 8. Original/attacked images with feature points at algorithm convergence. Feature points overlaid on images.

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
Random bending	0.15	0.17	0.14
Local geometric attack	0.12	0.02	0.13

TABLE I

Normalized Hamming distance between intermediate hash values of original and attacked (perceptually identical) images.

Attack	*Lena*	*Clinton*	*Barbara*
Object Addition	0.43	0.42	0.46
Object Removal	0.47	0.44	0.52
Excessive Noise Addition	0.53	0.45	0.38
Face Morphing	0.50	0.44	0.34

TABLE II

Normalized Hamming Distance between intermediate hash values of original and attacked images via content changing manipulations

effected by making a significant change to the image geometry.

Fig. 8 shows two examples of malicious content changing manipulation of image data and the response of our feature extractor to those manipulations. Fig. 8 (a) shows the original *toys* image. Fig. 8 (b) then shows a tampered version of the image in Fig. 8(a), where the tampering is being brought about by addition of a "toy bus". In Fig. 8 (d), an example of malicious tampering is shown where the face of the lady in Fig. 8 (c) has been replaced by a different face from an altogether different image.

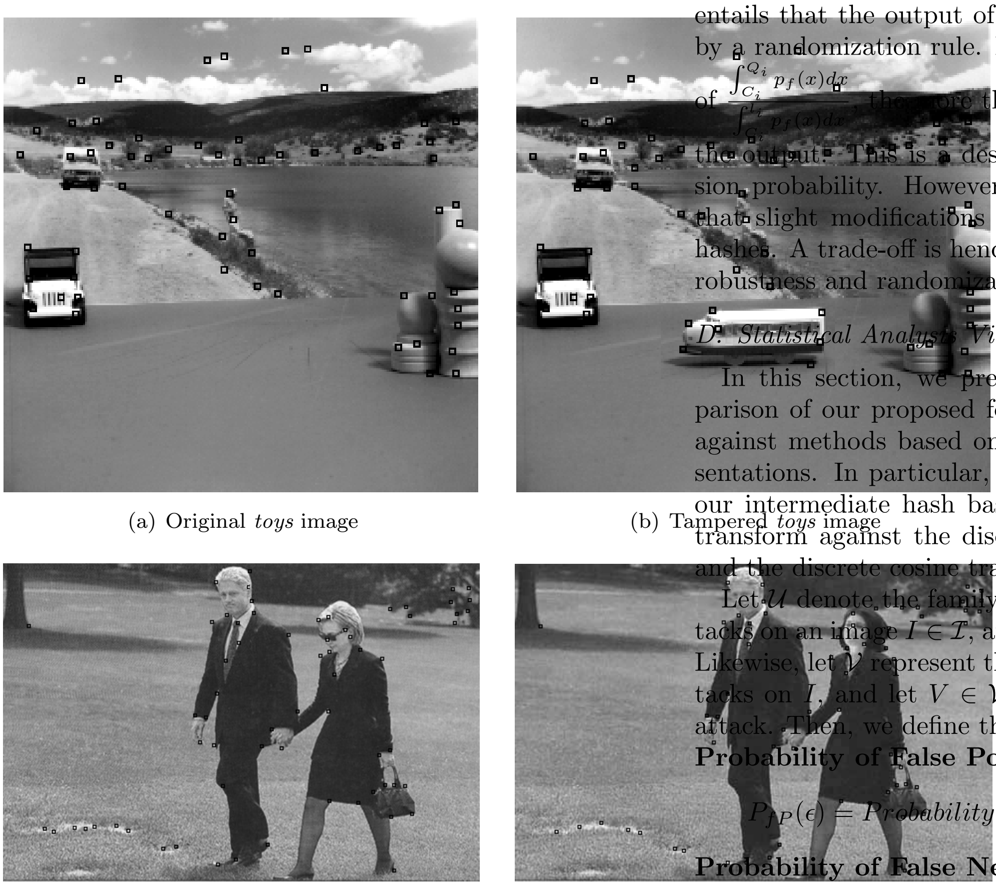
Comparing Figs. 8 (a) and (b), and Figs. 8 (c) and (d) it may be seen that several extracted features do not match. This observation is natural since our algorithm is based on extracting the *P strongest* geometric features from the image. In particular, in Fig. 8 (d), tampering of the lady's face is easily detected since most differences from Fig. 8 (c) are seen in that region. Quantitatively, this translates into a large distance between the intermediate hash vectors.

With complete knowledge of the iterative feature extraction algorithm, it may still be possible for a malicious adversary to generate inputs (pairs of images) that defeat our hash algorithm, e.g. tamper content in a manner such that the resulting features/intermediate hashes are still close. This, however, is much harder to achieve, when the randomized hash algorithm (Algorithm 2) were used.

We also tested under several other content changing attacks including object insertion and removal, addition of excessive noise, alteration of the position of image elements, tampering with facial features, 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 version was found to be greater than 0.3. Table II shows the normalized Hamming distance between intermediate hash values of original and maliciously tampered images for many different content changing attacks. Algorithm 2 with $N = 32$ was used for these results.

C. Performance Trade-Offs

A large search neighborhood implies the maxima of wavelet responses are taken over a larger set and hence the feature points are more robust. Likewise, consider selecting the feature points so that $T_1 < \max_\theta W_i(x, y, \theta) < T_2$.



(a) Original *toys* image

(b) Tampered *toys* image

(c) Original *clinton* image

(d) Tampered *clinton* image

Fig. 9. Content changing attacks and feature extractor response. Feature points overlayed on the images.

Note, the feature detection scheme as described in Fig. 3 implicitly assumes T_2 to be infinity. If T_1 and T_2 are chosen to be large enough, then the resulting feature points are very robust, i.e. retained in several attacked versions of the image. Similarly, if the two thresholds are chosen to be very low, then the resulting features tend to be easily removed by several perceptually insignificant modifications. The thresholds and the size of the search neighborhood facilitate a perceptual robustness vs. fragility trade-off.

When the number of random partitions N is one, and a deterministic quantization rule is employed in Section IV, 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.

Recall from Section IV that the output of the quantization scheme for binarizing the feature vector is completely deterministic except for the interval (P_i, Q_i). In general, more than one choice of the pair $(P_i,\, Q_i)$ may satisfy (20). Trivial solutions to (20) are (a) $P_i = Q_i = C_i$ and (b) $P_i = l_{i-1}$, $Q_i = l_i$. While (a) corresponds to the case when there is no randomness involved, the choice in (b) entails that the output of the quantizer is always decided by a randomization rule. In general, the greater the value of $\frac{\int_{C_i}^{Q_i} p_f(x)dx}{\int_{C_i}^{l_i} p_f(x)dx}$, the more the amount of unpredictability in the output. This is a desired property to minimize collision probability. However, this also increases the chance that slight modifications to the image result in different hashes. A trade-off is hence facilitated between perceptual robustness and randomization.

D. Statistical Analysis Via ROC Curves

In this section, we present a detailed statistical comparison of our proposed feature-point scheme for hashing against methods based on preserving coarse image representations. In particular, we compare the performance of our intermediate hash based on the end-stopped wavelet transform against the discrete wavelet transform (DWT) and the discrete cosine transform (DCT).

Let $\mathcal{U}$ denote the family of perceptually insignificant attacks on an image $I \in \mathcal{I}$, and let $U \in \mathcal{U}$ be a specific attack. Likewise, let $\mathcal{V}$ represent the family of content changing attacks on I, and let $V \in \mathcal{V}$ be a specific content changing attack. Then, we define the following terms:

Probability of False Positive:

$$P_{fP}(\epsilon) = Probability(D_H(\mathbf{h}(I), \mathbf{h}(V(I))) < \epsilon \qquad (25)$$

Probability of False Negative:

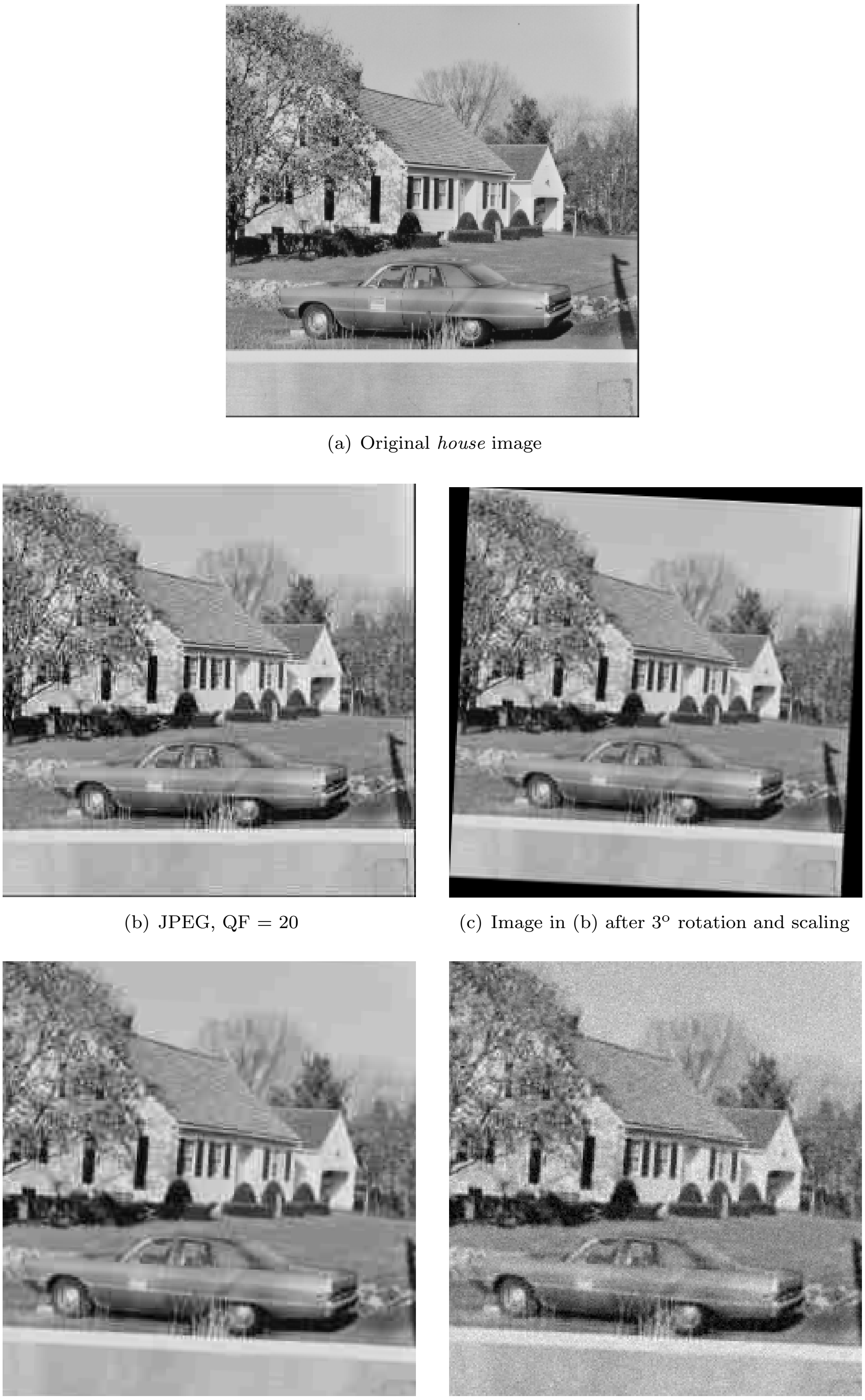
$$P_{fN}(\delta) = Probability(D_H(\mathbf{h}(I), \mathbf{h}(U(I))) > \delta \qquad (26)$$

To simplify the presentation, we construct two representative attacks:

- **A strong perceptually insignificant attack** in $\mathcal{U}$: A composite attack was constructed for this purpose. The complete attack (in order) is described as: (1) JPEG compression with QF = 20, (2) 3° rotation and rescaling to the original size, (3) 10% cropping from the edges, and (4) Additive White Gaussian Noise (AWGN) with $\sigma = 10$ (image pixel values range in 0-255). Fig. 9 (a) through (e) show the original and modified *house* images at each stage of this attack.
- **A content changing attack** in $\mathcal{V}$: The content changing attack consisted of maliciously replacing (a randomly selected) region of the image by an alternate unrelated image. An example of this attack for the *lena* image is shown in Fig. 10.

For fixed ϵ and δ, the probabilities in (25) and (26) are computed by applying the aforementioned attacks to a natural image database of 1000 images and recording the failure cases. As ϵ and δ are varied, $P_{fP}(\epsilon)$ and $P_{fN}(\delta)$ describe an ROC (receiver operating characteristic) curve.

All images were resized to 512×512 prior to applying the hash algorithms. For the results to follow, our intermediate hash was formed as described in Section V-A by retaining the P strongest features. The intermediate hash/feature vector in the DWT based scheme was formed by retaining the lowest resolution sub-band in an M-level DWT decomposition. In the DCT scheme, correspondingly, a



(a) Original *house* image

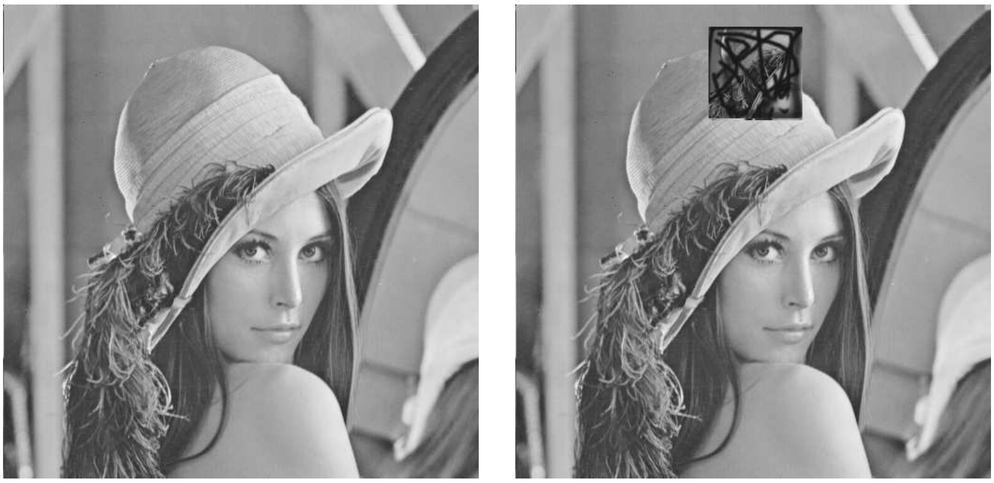
(b) JPEG, QF = 20

(c) Image in (b) after 3° rotation and scaling

(d) Image in (c) cropped 10% on the sides and rescaled to original size

(e) Final attacked image: AWGN attack on the image in (d)

Fig. 10. Representative perceptually insignificant attack on the *house* image: images after each stage of the attack.



(a) Original Lena Image

(b) Tampered lena image

Fig. 11. Example of the representative content changing attack on the *lena* image: 15% of the image area is being corrupted.

certain percentage of the total DCT coefficients were retained. These coefficients would in general belong to a low frequency band (but not including DC, since it is too sensitive to scaling and/or contrast changes).

Fig. 12 shows ROC curves for the three schemes for extracting intermediate features of images: 1.) preserving low-frequency DCT coefficients, 2.) low resolution wavelet coefficients, and 3.) the proposed scheme based on end-stopped kernels. Each point on these curves represents a (P_{fP}, P_{fN}) pair computed as in (25) and (26) for a fixed ϵ and δ. We used $\delta = \frac{3}{2}\epsilon$ in all cases. For the ROC curves in Fig. 12, we varied ϵ in the range $[0.1, 0.3]$. A typical application, e.g. image authentication or indexing, will choose to operate at a point on this curve.

To ensure a fair comparison among the three schemes, we consider two cases for each hashing method. For the DWT, we show ROC curves when a 6-level DWT and 5-level DWT transform was applied. A 6-level DWT on a 512×512 image implies that 64 transform coefficients are retained. In a 5-level DWT similarly, 256 coefficients are retained. Similarly, for the DCT based scheme we show two different curves in Fig. 12, respectively, corresponding to 64 and 256 low-frequency DCT coefficients. For our intermediate hash, we show curves corresponding to $P = 64$ and $P = 100$.

In Fig. 12, both the false positive as well as the false negative probabilities are much lower for our proposed hash algorithm. Predictably, as the number of coefficients in the intermediate hash are increased for either scheme, a lower false positive probability (i.e. less collisions of perceptually distinct images) is obtained at the expense of increasing the false negative probability. Recall however, from Section VI-C that this trade-off can be facilitated in our intermediate hash even with a fixed number of coefficients - an option that the DWT/DCT does not have.

In Fig. 12, with $P = 64$ features, our hash algorithm based on end-stopped kernels vastly outperforms the DCT as well as DWT based hashes[5] in achieving lower false positive probabilities, even as a much larger number of coefficients is used for them. We also tested our hash algorithms on all possible pairings (499500) of the 1000 distinct images in our experiments. Only 2 collision cases (same/close intermediate hash vectors for visually distinct images) were observed with $P = 100$, and 5 with $P = 64$.

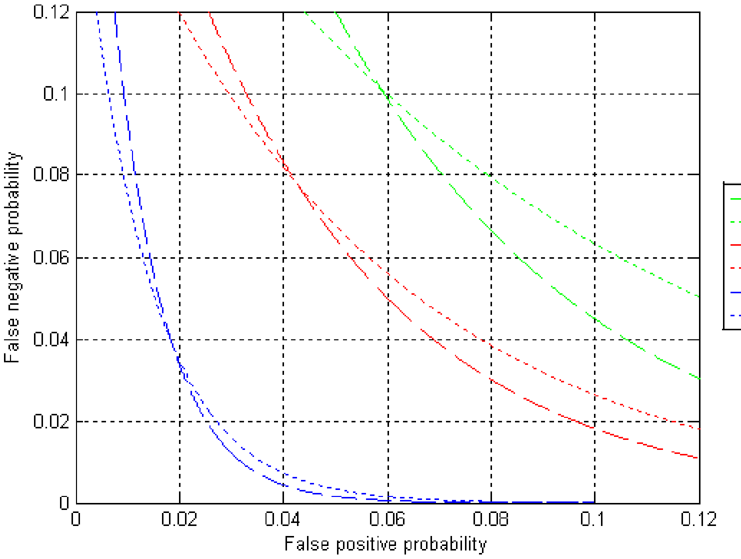


Fig. 12. ROC curves for hash algorithms based on three approaches: DCT transform, DWT transform, proposed intermediate hash based on end-stopped kernels. Note that error probabilities are significantly lower for the proposed scheme.

[5] All the wavelet transforms in the MATLAB wavelet toolbox version 7.0 were tested. The results shown here are for the discrete Meyer wavelet "dmey" which gave the best results amongst all DWT families.

VII. Discussions & Conclusion

This paper proposes a two-stage framework for perceptually-based image hashing consisting of visually robust feature extraction (intermediate hash) followed by feature vector compression (final hash). A general framework for constructing intermediate hash vectors from images via visually significant feature points is presented.

An iterative feature extraction algorithm based on preserving significant image geometry is proposed. Several robust feature detectors may be used within the iterative algorithm. Parameters in our feature detector enable tradeoffs between robustness and fragility of the hash, which are otherwise hard to achieve with traditional DCT/DWT based approaches.

We develop both deterministic and randomized algorithms. Randomization is particularly desirable is adversarial scenarios to lend unpredictability to the hash and reduce its vulnerability to attacks by a malicious adversary. ROC analysis is performed to demonstrate the statistical advantages of our hash algorithm over existing schemes based on preserving coarse image representations.

It is useful to think of features extracted via our randomized hash algorithm as a pseudo-random signal representation scheme for images, i.e. a different representation, each sufficient to characterize the image content, is obtained (with high probability) as the secret key is varied. Future work could explore alternate pseudo-random signal representations for image identification and hashing. In particular, the goal of secure hashing can be understood as developing the pseudo-random image representation that leaks the minimum amount of information about the image.

References

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

[2] K. Mihcak, R. Venkatesan, and T. Liu, "Watermarking via optimization algorithms for quantizing randomized semi-global image statistics," *ACM Multimedia Systems Journal*, Apr. 2005.

[3] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," *Proc. IEEE Conf. on Image Processing*, vol. 3, pp. 227–230, Sept. 1996.

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

[5] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," *Proc. IEEE Conf. on Image Processing*, pp. 664–666, Sept. 2000.

[6] C. Y. Lin and S. F. Chang, "A robust image authentication system distingushing JPEG compression from malicious manipulation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 153–168, Feb. 2001.

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

[8] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," *Proc. IEEE International Conf. on Information Technology: Coding and Computing*, pp. 178–183, Mar. 2000.

[9] K. Mihcak and R. Venkatesan, "New iterative geometric techniques for robust image hashing," *Proc. ACM Workshop on Security and Privacy in Digital Rights Management Workshop*, pp. 13–21, Nov. 2001.

[10] S. S. Kozat, K. Mihcak, and R. Venkatesan, "Robust perceptual image hashing via matrix invariances," *Proc. IEEE Conf. on Image Processing*, pp. 3443–3446, Oct. 2004.

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

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

[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] S. Bhatacherjee and P. Vandergheynst, "End-stopped wavelets for detection low-level features," *Proc. SPIE, Wavelet Applications in Signal and Image Processing VII*, pp. 732–741, 1999.

[15] M. Johnson and K. Ramachandran, "Dither-based secure image hashing using distributed coding," *Proc. IEEE Conf. on Image Processing*, 2003.

[16] V. Monga, A. Banerjee, and B. L. Evans, "Clustering algorithms for perceptual image hashing," *Proc. IEEE Digital Sig. Proc. Work.*, pp. 283–287, Aug. 2004.

[17] V. Monga, A. Banerjee, and B. L. Evans, "A clustering based approach to perceptual image hashing," *IEEE Trans. on Signal Processing*, 2005, accepted for publication.

[18] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A compasion of affine region detectors," *accepted to International Journal on Computer Vision*, 2006.

[19] C. Harris and M. Stephen, "A combined corner and edge detector," *Proc. Alvey Visual Conference*, pp. 147–151, Sept. 1988.

[20] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seen robot rover," *Technical Report CMU-RI-TR-3, Robotics Institute, Carnegie-Mellon University,*, Sept. 1980.

[21] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal on Computer Vision*, pp. 63–68, 2004.

[22] J. Matas, O. Schum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Proc. British Machine Vision Conference*, pp. 384–393, 2002.

[23] A. Dobbins, S. W. Zucker, and M. S. Cynader, "End-stopping and curvature," *Vision Research*, pp. 1371–1387, 1989.

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

[25] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1996.

[26] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, 1984.

[27] G. Sharma, *Digital Color Imaging Handbook*, CRC Press, 2002.

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

[29] V. Monga, D. Vats, and B. L. Evans, "Image authentication under geometric attacks via structure matching," *IEEE Int. Conf. on Multimedia and Expo*, July 2005.