

Geometric Distortion-Resilient Image Hashing Scheme and Its Applications on Copy Detection and Authentication *

Chun-Shien Lu and Chao-Yong Hsu
Institute of Information Science, Academia Sinica
Taipei, Taiwan 115, Republic of China
E-mail: {lcs,cyhsu}@iis.sinica.edu.tw

Abstract

Media hashing is an alternative approach to many applications previously accomplished with watermarking. The major disadvantage of the existing media hashing technologies is their poor resistance to geometric attacks. In this paper, a novel geometric distortion-invariant image hashing scheme which can be employed to perform copy detection and content authentication of digital images, is proposed. Our major contributions are threefold: (i) mesh-based robust hashing function is proposed; (ii) sophisticated hash database for error-resilient and fast matching is constructed; and (iii) the application scalability of our scheme for content copy tracing and authentication is studied. In addition, we further investigate several media hashing issues, including robustness and discrimination, error analysis, and complexity, for the proposed image hashing system. Exhaustive experimental results obtained from benchmark attacks confirm the excellent performance of the proposed method.

keywords: Authentication, Copy detection, Geometric distortion, Hash, Robustness, Searching

*This paper was, in part, published in the *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2004 [18] and *Proc. ACM Multimedia and Security Workshop*, 2004 [7].

1 Introduction

With the advances in multimedia and networking technologies, it has become easy to copy original material completely and distribute illegal copies rapidly over the Internet. In order to trace the unauthorized use of digital contents, media hashing technologies have been applied to digital content management. In contrast to data hiding, the main characteristic of media hashing is its non-invasive property, which means that no information has to be embedded in the digital content. On the other hand, a hash sequence for specific media data needs to be extracted to represent its condensed essence. Analogous to media hashing, some synonymous terminologies have been reported in the literature, including fingerprinting, digital signature, and passive/non-invasive watermarking. The major feature that distinguishes media hashing from watermarking is that the former measures “similarity” and needs to work together with a feature database, while the latter measures “originality” and can operate as a standalone system. On the other hand, media hashing is also similar to media retrieval in that both need to transform media data into a short string for the sake of compact representation. The technical difference between them is that media hashing must resist (either malicious or incidental) attacks. Therefore, several applications that call for robust identification of media contents require the use of robust media hashing methods [8, 22]. The technical requirements for media hashing cannot be satisfied by means of traditional cryptographic hashing functions because even a one bit error in a hash sequence can lead to entirely different contents. However, limited distortions are not harmful to the visual quality and commercial value of multimedia data.

Many existing media hashing methods were developed for audio identification [22] and video authentication [4, 14]. In this paper, we focus on image hashing. First of all, a review of the literature on image hashing will be presented in the following. In 1998, Chang *et al.* proposed a wavelet-based Replicated IMage dEtector (RIME) [2] to search for unauthorized image copying on the Internet. They used color features only to represent an image and then used the vector quantization (VQ) technique to index images. Their system cannot resist extensive geometric distortions. To speed up the detection of near-replicas of images in their RIME system, Chang *et al.* proposed a new clustering approach [3] that can improve I/O efficiency by clustering and retrieving relevant information sequentially on and from the disk. Recently, Meng and Chang [20] used multi-scale color and texture features to characterize images and employed the dynamic partial function (DPF) to measure the perceptual similarity of images. Basically, the idea behind DPF [13] is to dynamically activate partial features (thereby discarding the other larger feature distance) in order to reveal the similarity between a pair of images. Although DPF outperformed traditional distance metrics, the adopted image feature was global in

that resistance to geometric distortions was inherently limited. In [15, 16], the digital signature was proposed for image authentication. Lin and Chang [15] created the mutual relationship of pairwise block-DCT coefficients to distinguish JPEG compressions from malicious modifications. Lu and Liao [16] built the so-called “structural digital signature” based on the multiscale structure of the wavelet transform to tolerate incidental manipulations and reflect intentional manipulations. However, the ability to resist geometric manipulations was lacking [15, 16]. In [5, 6], Fridrich and Goljan proposed a robust/visual hash function for digital watermarking. Their hash digests of digital images were created by projections of DCT coefficients to key-dependent random patterns. In [26], Venkatesan *et al.* proposed an image hashing technique, which includes (i) random tiling of an image; and (ii) hash generation based on statistical feature extraction of tiles. However, the two methods [5, 26] only achieve limited resistance to geometric distortions. In [11, 12, 25], image hashing methods were proposed based on the Radon transform and exploited its affine invariance. However, resistance to geometric distortions was greatly limited if the incoming attacks went beyond affine distortions. In [21], Mihcak and Venkatesan proposed an iterative geometric image hashing method, which contained two major steps. First, an image was converted into a binary image by thresholding the lowest-frequency subband in the wavelet domain in order to identify the geometric shapes. Second, iterative filtering was employed on the resultant binary image to obtain the hash by enhancing geometrically strong components and erasing geometrically weak components. Their method could only withstand slight geometric distortions. In [10], Kim proposed an image copy detection scheme that employed ordinal measures of AC coefficients in the 8×8 DCT domain; i.e., the magnitudes of the AC coefficients in a block were ranked in descending order to represent an image. Extensive signal processing attacks were conducted to test the method’s robustness and discrimination in the case of a large database. However, this system basically could not resist geometric distortions.

It is evident from the above survey that the common disadvantage of the existing image hashing techniques is their limited robustness against geometric distortions. (For instance, resistance to rotations is restricted to very small angles.) The main reason for this weakness is that the previous methods did not really deal with the problem of combating geometric attacks. In view of this fact, the purpose of this paper is to deal with this challenging problem seriously. We shall propose a robust mesh-based image hashing scheme for content copy detection, identification, and tracing in a large database. Our major contribution is the capability of achieving robustness against extensive geometric distortions (e.g., standard benchmarks like Stirmark3.1 and Stirmark4.0 [23, 24]). Although the concept of image meshing has been applied to watermarking before [1], we have considered the

stability of mesh generation, which is closely related to mesh-based applications. Consequently, we present a robust mesh extraction technique so that it cannot easily harm the performance of mesh-based hashing. We also present a robust mesh-based hash extraction technique that considers content position-dependent features. Extensive results obtained from benchmark attacks further confirm the robustness of the proposed scheme.

In addition to robustness, the practical use of an image hashing system requires the ability to quickly search in a large database. In this paper, we will also show how an efficient hash database can be built to facilitate fast hash matching. Moreover, we shall present error analyses and investigate the complexity, granularity, and scalability of the proposed image hashing system. In particular, we will demonstrate how our image hashing system can be applied to content authentication.

The remainder of this paper is organized as follows. Sec. 2 discusses the difference between cryptographic hashing and media hashing, and states the problems of media hashing that need to be dealt with. In Sec. 3, the proposed image hashing system, which includes mesh generation, mesh-based hash generation, coarse-to-fine hash database construction, and fast hash matching, is described. Media hashing issues, including robustness, error analysis, complexity, and scalability, are discussed in Sec. 4. Extensive experimental results are given in Sec. 5 to verify the performance of our scheme. Finally, concluding remarks are given in Sec. 6.

2 Problem Statement

Media hashing is recognized as an alternative approach to several applications that were previously performed using digital watermarking. Here, a scenario of copy detection and tracing is given to outline how an image hashing approach could be employed to manage digital image contents. Given an image owned by its creator, an image copy detection system needs to find out whether illegal copies of the image exist on the Internet, and if they exist, return a list of suspect URLs. This content searching strategy is accomplished by means of image hashing, and the output of the hashing system can offer owners information about unauthorized use of their precious media data.

Referring to the image space shown in Fig. 1, let \mathbf{I} denote an image, and let \mathcal{X} denote the set of images that are modified from \mathbf{I} by means of content-preserving operations (e.g., filtering, compression, geometric distortions) and are defined to be perceptually similar to \mathbf{I} . Although perceptual similarity is still an ill-posed concept [13, 27], we will propose a hash-based matching metric in the next section for image searching. We further use \mathcal{Y} to denote those images that are actually modified from \mathbf{I} but

can hardly be recognized as originating in \mathbf{I} . For example, severe noise adding and severe cropping are two representative attacks that can generate the elements of \mathcal{Y} . In addition, let us denote as \mathcal{Z} a set which contains all the other images that are irrelevant to \mathbf{I} and its modified versions. Consequently, $\{\mathbf{I}\} \cup \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ is a case that forms an entire image space.

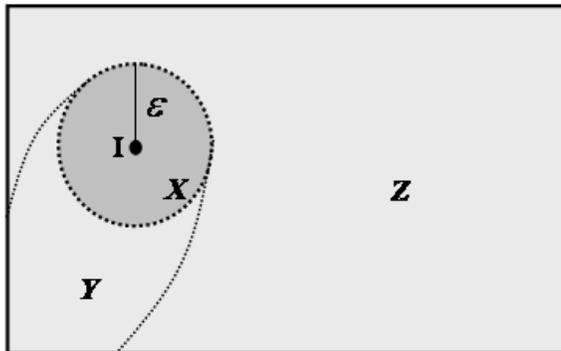


Figure 1: **The Image Space.** \mathbf{I} is an element in the image space. \mathcal{X} denotes the set of images modified from \mathbf{I} that are still perceptually similar to \mathbf{I} . \mathcal{Y} denotes the set of images modified from \mathbf{I} that are allowed to be unrecognizable. \mathcal{Z} is the set of images that are irrelevant to \mathbf{I} .

In order to represent the condensed essence of an image for perceptual similarity measurement, a hash function is usually employed. Conventionally, a cryptographic hash function, H^c , is used to map an image \mathbf{I} as a short binary string, $H^c(\mathbf{I})$. One of the most important properties of cryptographic hashing is that it is collision-free, which means that it is hard to find two different images that can be transformed to produce the same hashes. Let $z \in \mathcal{Z}$, and let z and \mathbf{I} be distinct. The collision-free property of cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(z)$. Furthermore, let $x \in \mathcal{X}$; then, cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(x)$. This implies that cryptographic hashing inherently produces totally different hash sequences if the media content has been modified.

However, this characteristic is too restricted to be suitable for multimedia applications since multimedia content permits acceptable distortions. As a result, it is necessary to develop a media hashing function, H^m , that can provide error-resilience. The error-resilience property of media hashing is defined as follows. It is said that $x (\in \mathcal{X})$ is successfully identified as having been modified from \mathbf{I} if $d(H^m(\mathbf{I}), H^m(x)) \leq \epsilon$ holds, where $d(\cdot, \cdot)$ indicates a Hamming distance function. In other words, if two images are perceptually similar, their corresponding hashes need to be highly correlated. In addition, the desired media hash function still needs to possess the collision-free property, like cryptographic hashing, except that the distance measure is changed to $d(H^m(\mathbf{I}), H^m(x)) > \epsilon$. On the

other hand, it is insignificant whether y ($\in \mathcal{Y}$) can be identified as having been modified from \mathbf{I} or not. It should be noted that the traditional cryptographic hash function is a special case of the media hash function in that its ϵ value is set to be 0. Overall, the main idea behind media hashing is to develop a robust hash function that can identify perceptually similar media contents and achieve the collision-free property. Issues related to media hashing will be discussed in Sec. 4.

3 Proposed Image Hashing Approach

The block diagrams of the proposed mesh-based image hashing system and image query system are depicted in Fig. 2 and Fig. 3, respectively. Our image hashing method is operated on normalized meshes such that geometric-resilience is not restricted to affine transformations [11, 12, 25]. Our previous approaches proposed via this paradigm were reported in [7, 18], and the capabilities of robustness and discrimination were obtained. In this paper, we generalize our previous methods to provide more analyses of hashing issues and a more extensive set of verification results. The major components of our proposed image hashing system will be sequentially described in this section.

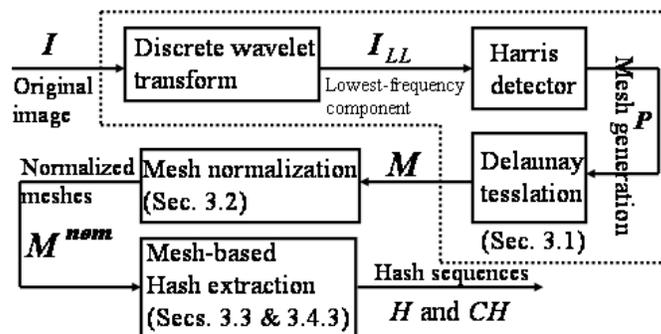


Figure 2: Block diagram of the proposed mesh-based image hashing system.

3.1 Robust Image Mesh Generation

The extraction of robust meshes plays an important role in our method since it is a prerequisite for withstanding geometric distortions. To generate meshes, the first step is to detect the salient points of an image. Among the ubiquitous feature point extraction methods, the Harris detector has been widely used. However, the original Harris detector is still not robust enough to be used for our purposes. Thus, we propose to improve its robustness by applying it in the lowest-frequency subband of the downsampled-discrete wavelet transform (DS-DWT) domain. Our intention is twofold: (i) to

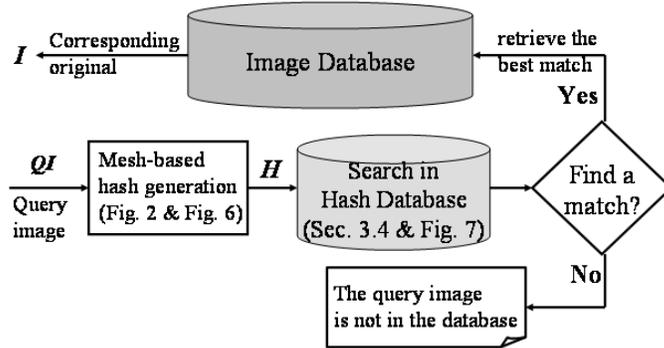


Figure 3: **Block diagram of the proposed image query system: a query image (QI) enters the hash database for possible retrieval of its original image from the image database.**

filter out noisy points for robust detection of salient points; (ii) to reduce the space for mesh and hash generations.

Once the feature point extraction process is finished, the Delaunay tessellation is exploited to decompose the image into a set of disjointed triangles. Each triangle (called a mesh in this paper) is regarded as the minimum unit for robust hash extraction. The overall mesh generation process is summarized as follows: (i) the original image \mathbf{I} is transformed via downsampled discrete wavelet decomposition and the lowest-frequency subband signal, \mathbf{I}_{LL} , is selected; (ii) the set of feature points \mathcal{P} is generated by applying the Harris detector to \mathbf{I}_{LL} ; and (iii) Delaunay tessellation is performed using \mathcal{P} to obtain a set, \mathcal{M} , of meshes ($|\mathcal{M}|$ is used to denote the number of meshes in \mathcal{M}).

An example of mesh extraction is presented in Fig. 4, which shows meshes generated from the original Lenna and its Stirmark attacked Lenna images. From visual inspection, we can find that several meshes are consistently extracted. Although we can see a few differences between the generated meshes (e.g., the top-right part of Fig. 4(d) and the top-left part of Fig. 4(e)), these differences are not certain to affect the hash-based similarity measurement, as will be explained in Sec. 3.4. These results illustrate the effectiveness of mesh extraction from the lowest-frequency subband of an image.

3.2 Mesh Normalization

Once the set of meshes in an image has been produced, each original mesh $\mathbf{M}_{\mathbf{k}} (\in \mathcal{M})$ is normalized as $\mathbf{M}_{\mathbf{k}}^{\text{nom}}$ to generate a mesh-based hash $\mathbf{H}_{\mathbf{k}}$, where $\mathbf{M}_{\mathbf{k}}^{\text{nom}}$ is a right-angled isosceles triangle. The aim of normalization is to keep all normalized meshes the same size and the extracted mesh-based hashes the same length to make mesh-based hash comparisons possible. Fig. 5 illustrates the relationship between $\mathbf{M}_{\mathbf{k}}$ and $\mathbf{M}_{\mathbf{k}}^{\text{nom}}$, where $\langle A, B, C \rangle$ and $\langle \acute{A}, \acute{B}, \acute{C} \rangle$ denote the corners of $\mathbf{M}_{\mathbf{k}}$ and

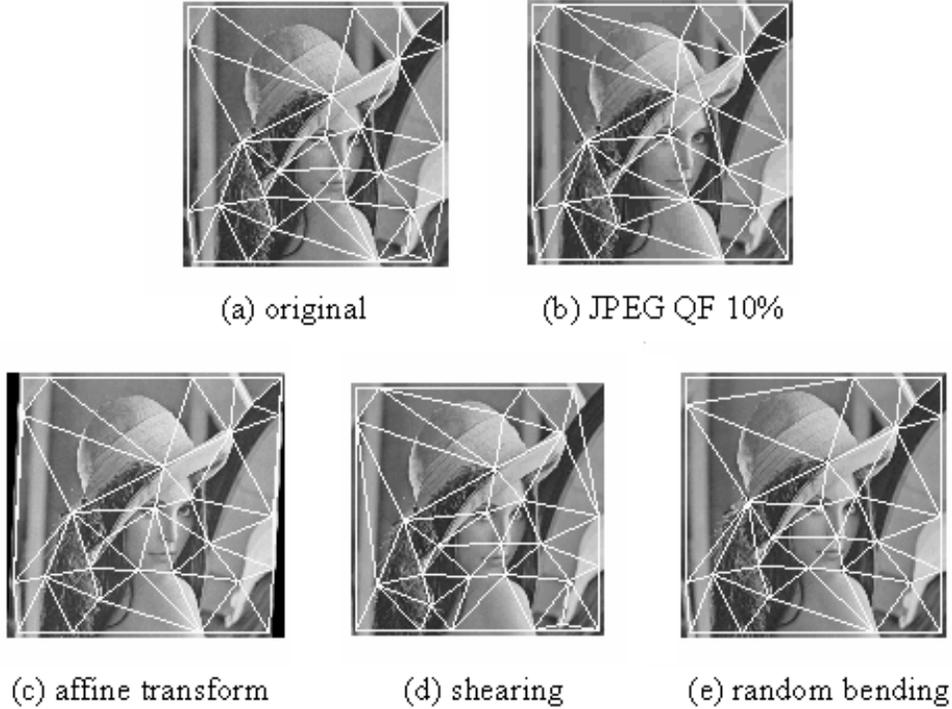


Figure 4: **Illustration of meshes extracted from original and attacked Lenna images (initially, they were color images).** Note that the meshes were detected at the lowest-frequency subband of an image after two-level wavelet decomposition was performed.

$\mathbf{M}_k^{\text{nom}}$, respectively. In addition, let $\langle A, B, C \rangle$ be arranged to satisfy $\angle BAC \geq \angle ABC \geq \angle ACB$, where $\angle BAC$ denotes an angle centered at corner A . When the normalization process is performed, $\langle A, B, C \rangle$ is first mapped to $\langle \acute{A}, \acute{B}, \acute{C} \rangle$ sequentially. That is, this “angle order” must be maintained to keep uniform warping in order to not affect the generation of normalized meshes and their corresponding hashes. Here, non-uniform warping implies that an original mesh and its attacked mesh are normalized in a different angle order; thus, the resultant normalized meshes will produce different hashes. After angle order-based corner mapping is performed, \mathbf{M}_k is transformed into $\mathbf{M}_k^{\text{nom}}$ through the procedures of affine transformation and interpolation.

There are two major factors that will affect the angle order. One is caused by severe geometric distortions that change the order of three angles in a mesh. However, this factor will lead to apparent destruction of the visual quality of images, which will thus lose their commercial value. Therefore, we can ignore this problem. The other one results from the situation where two or three angles are nearly the same in magnitude such that even a slight distortion can change their order. This problem must be dealt with since the visual quality of an image is only imperceptibly modified. Our solution is to

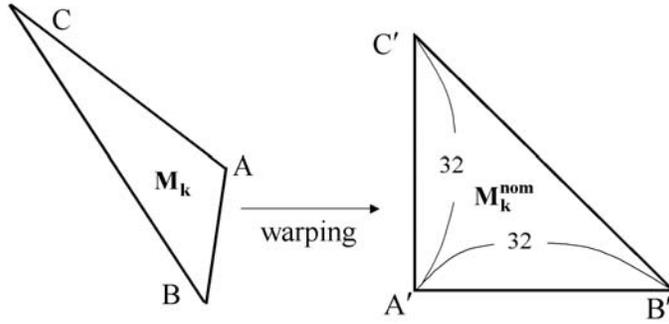


Figure 5: **The angle ordering between an original mesh and its normalized mesh. The angle ordering is determined by sorting the three angles in a descending order.**

generate two (if two angles are nearly the same) or six (if three angles are nearly the same) different hashes for such a mesh by changing the angle order sequentially with respect to a query image. In addition, we still keep one mesh in order to have one corresponding hash in the hash database to save space.

In our implementation, a normalized mesh is a right-angle isosceles triangle with a length of both sides that was experimentally verified to be 32, based on the following observations: (i) if the side's length is small enough, partial information of an original mesh will be lost, and not enough discriminable features will be provided among different meshes; (ii) if the side's length is large enough, execution time will be mostly spent on mesh warping that becomes the bottleneck of the whole scheme. As a result, the number 32 is empirically selected to obtain a trade-off between discrimination of the hash and the complexity of normalization.

The change of a mesh by attacks can affect the mesh normalization result. However, since the proposed hash generation technique is conducted in the block-DCT domain (described in the next section), somewhat different normalization results won't be certain to lead to totally different hashes.

3.3 Robust Mesh-based Hashing

Image hashing attempts to transform an image content into a feature sequence to represent its condensed essence. This feature sequence must be short enough for fast matching and, meanwhile, preserve distinguishable features for feasible similarity measurement. In addition, a robust hashing algorithm is necessary to accommodate possible changes of normalization results. With the above crucial factors taken into consideration, in this paper, the robust hash of each normalized mesh M_k^{nom} is extracted in the block-DCT domain. In fact, the extracted hash bits are position-dependent and belong to a

certain type of local feature.

First, each triangle $\mathbf{M}_k^{\text{nom}}$ is flipped and padded with its flipped version to form a 32×32 block. Each 32×32 block is divided into 64 4×4 blocks, as illustrated in Fig. 6(a). Second, the 4×4 DCT transform is performed, and the first AC coefficient (located at the lowest frequency subband except for the DC term) of each 4×4 block is selected. All the selected AC coefficients form an AC sequence with a length of 64. It should be noted that due to the effect of padding, the upper triangular and the lower triangular capture different features. For example, as shown in Fig. 6(a), if the dark-gray block in the upper triangular captures the horizontal feature, its padding counterpart will capture the vertical feature. The DC coefficients will not be selected because they are not helpful for capturing identifiable features. Finally, this AC sequence is sorted according to the magnitudes of its 64 elements, and the hash bits, $H_k(s)$'s, are assigned as follows:

$$H_k(s) = \begin{cases} 1, & \text{if } |AC_k^s(1)| \text{ belongs to the first 32} \\ & \text{largest AC coefficients} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $H_k(\cdot)$ is a hash bit in a binary hash sequence \mathbf{H}_k and $AC_k^s(1)$ ($0 \leq s \leq 63$) denotes the first AC coefficient in a 4×4 block s of a normalized mesh $\mathbf{M}_k^{\text{nom}}$.

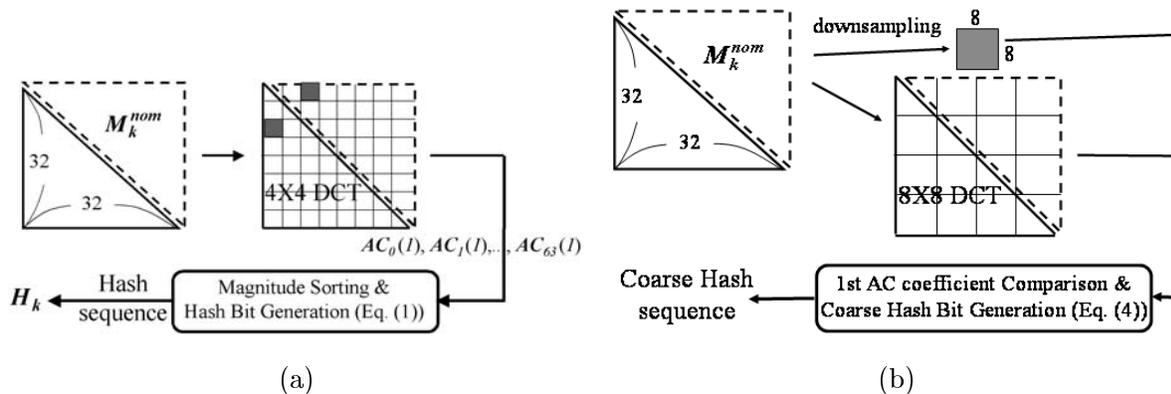


Figure 6: Mesh padding for hash extraction in the block-DCT domain: (a) full hash generation and (b) coarse hash generation.

It is worth mentioning that the hash bits determined by Eq. (1) are image position-dependent (i.e., s). Unlike other hashing methods that have adopted global or statistical features, this additional security measure should be used to avoid the collision problem, where two dissimilar images have similar hashes. We shall further discuss this problem in Sec. 4.1.

In Eq. (1), one hash bit is generated from a 4×4 block. In addition, the mesh-based hash designed

according to Eq. (1) guarantees that the number of 1's and 0's is the same, i.e., uniform distribution is achieved, to avoid any bias that will affect hash matching. This uniform distribution of hash bits is necessary to ensure that both collision-free and false positive can be efficiently controlled (this will be discussed in Secs. 4.1 and 4.2, respectively). We say that this feature value $H_k(\cdot)$ is robust because this magnitude relationship obtained after sorting can be approximately preserved. Please refer to Sec. 4.1 for detailed robustness analyses.

Note that the number of meshes is equal to the number of hashes in an image. $|\mathbf{H}_k|$ is used to denote the length of a binary hash sequence \mathbf{H}_k . In this paper, the hash dimensionality, $|\mathbf{H}_k|$, is fixed at 64, as explained previously. After mesh generation and mesh-based hash extraction are performed, the feature vector of an image \mathbf{I} could be expressed as

$$\{\mathbf{H}_1^{\mathbf{I}}, \mathbf{H}_2^{\mathbf{I}}, \dots, \mathbf{H}_{|\mathcal{M}^{\mathbf{I}}|}^{\mathbf{I}}\}, \quad (2)$$

where $|\mathcal{M}^{\mathbf{I}}|$ and $\mathbf{H}_k^{\mathbf{I}}$ denote the number of meshes and the k -th hash sequence in image \mathbf{I} , respectively.

3.4 Hash Database Creation and Mesh-based Fast Matching

In this section, we discuss how to create an image hash database with which the mesh-based matching process can be performed when an incoming query is given. Our hash database is designed to be suitable for two-stage fast search, where the first stage produces potential candidates through a coarse matching process, and the second stage is a full matching process used to identify the final winner (if there is one) from among the candidates. The overall image query system is depicted in Fig. 3.

3.4.1 Similarity Measurement

Since the objective of this paper is to provide a high degree of robustness against various attacks (including geometric distortions), partial matching is considered when measuring similarity. In content copy detection and tracing applications, two images (\mathbf{I}_m and \mathbf{I}_n) are considered to be similar if at least N mesh-pairs are matched. Moreover, it is said that a pair of meshes is matched if the bit error rate (BER) between their corresponding hashes is smaller than a threshold T ($0 \leq T \leq 1$), i.e.,

$$BER(\mathbf{H}_i^{\mathbf{I}_m}, \mathbf{H}_j^{\mathbf{I}_n}) = \frac{\#\{t | H_i^{\mathbf{I}_m}(t) \neq H_j^{\mathbf{I}_n}(t)\}}{|\mathbf{H}_i^{\mathbf{I}_m}|} \leq T, \quad (3)$$

where $H_i^{\mathbf{I}_m}(t)$ denotes the t -th element of the i -th hash in \mathbf{I}_m and $\#\{\}$ denotes the number of bit errors. The two thresholds, T and N , will be derived based on a sufficiently small false positive probability in Sec. 4.2. In addition, so-called valid or invalid retrieval in a large database based on the image similarity measurement (Eq. (3)) will be described in Sec. 3.4.5.

3.4.2 Full Matching

Let $|\mathcal{M}^{\mathbf{I}_m}|$ denote the number of meshes in an image \mathbf{I}_m . Conventionally, the image hash database collects and stores the hashes of all images. Thus, $|\mathcal{M}^{\mathbf{I}_m}| \times |\mathcal{M}^{\mathbf{I}_n}|$ mesh pairs have to be compared in order to determine whether two images, \mathbf{I}_m and \mathbf{I}_n , are similar or not according to the similarity measurement described in Sec. 3.4.1. However, we cannot rely on the exhaustive matching process only. This is because when the database is huge, the time consumed by exhaustive hash matching becomes tremendous. Thus, this kind of search is not suitable for many applications that require real-time processing. We call this kind of matching process “full matching.” Therefore, full matching is only performed on those candidates that have been retrieved through a rapid coarse matching process (this will be described in Sec. 3.4.4). A coarse-to-fine image hash database with the error-robust capability for fast search will be described in the next section.

3.4.3 Creation of Error-Resilient Tree-Structured Image Hash Database for Fast Search

In order to speed-up the matching process, we propose a fast matching technique that comprises two stages: (i) “coarse matching” for rapid selection of candidates; (ii) “full matching” for determining the final target from among the selected candidates. In fact, this technique looks like a coarse-to-fine searching paradigm. The so-called coarse matching stage is mainly used to coarsely find a set of candidates (whose size is usually significantly smaller than the entire search space) that may contain the desired target. Next, full matching is conducted on the set of candidates to exhaustively find the final result. Therefore, our fast matching paradigm needs to be used with a hash database that is designed in a sophisticated manner. This sophisticated hash database is built as shown in Fig. 7. The hash database comprises “entries,” each of which links to a chain that contains the indices of images. It is said that an image can be linked to a specific entry if one hash of that image and the entry are similar. In practice, each entry is the seed of a group. It is also observed that a group associated with an entry can proliferate rapidly if this entry is a common feature among images. As a result, our approach to building the hash database for fast searching is a kind of clustering method. However, unlike other clustering methods that have been proposed for content-based image retrieval, our clustering paradigm adopts partial clustering instead of global clustering. That is to say, an image can be linked to different entries as long as its hashes are similar to more than one entry.

There are two issues that should be considered when constructing a hash database; i.e., entries should (i) be short enough for practical implementation and (ii) possess the error-robust capability in order to accommodate modifications of meshes due to attacks. By taking the above two issues

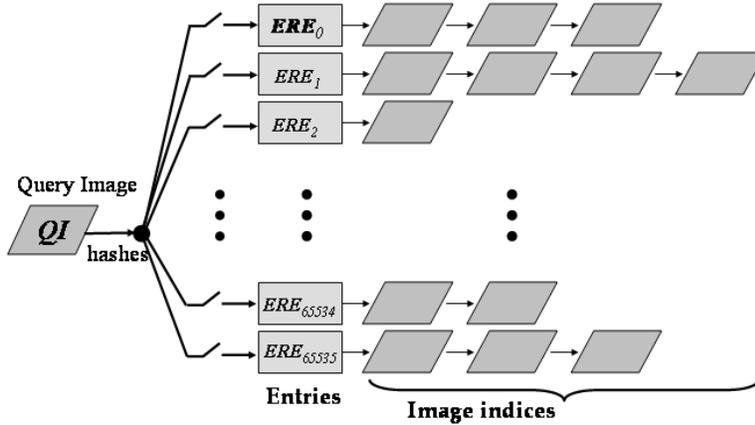


Figure 7: **Creation of an image hash database for fast search in a coarse-to-fine manner. Our image hash database includes (i) error-resilient entries; (ii) image indices; and (iii) image hashes.**

into account simultaneously, in this paper, each entry is designed as a 16-bit long hash sequence that also represents a coarse representation of a mesh. This “coarse hash,” which is generated in a way similar to the one described in Sec. 3.3, is described as follows. As shown in Fig. 6(b), each normalized mesh of size 32×32 is downsampled to obtain an 8×8 coarse block from which 8×8 DCT is performed and the first AC coefficient, denoted as B_{ds} , is selected. Meanwhile, each normalized mesh is also partitioned into 16 8×8 blocks from which 8×8 DCT is performed and 16 1^{st} AC coefficients, denoted as $B_p(b)$ ($0 \leq b \leq 15$), are obtained. With the above setting, the coarse hash bits are designed as the results obtained by comparing each $B_p(b)$ with B_{ds} . More specifically, the coarse hash bit of a mesh is defined as

$$CH(b) = \begin{cases} 1, & \text{if } B_p(b) \geq B_{ds} \\ 0, & \text{otherwise;} \end{cases} \quad (4)$$

where $CH(b)$ is a hash bit in a coarse hash sequence \mathbf{CH} . As indicated in Eq. (4), each coarse hash is a 16-bit vector, which implies that each entry is also composed of 16 bits, and that there are a total of 65536 entries. The entries are expressed as $ERE_0, ERE_1, \dots, ERE_{65535}$, where ERE_i is a binary representation of i . In our implementation, the size of the ERE_i 's needs to be controllable so that they can be stored in an array for rapid indexing. This corresponds to the first issue. As for error resilience, this means that even if an image has been modified, its coarse hashes are largely unaffected. Since the proposed coarse block has the low-frequency characteristic and the coarse hash bits are designed based on the magnitude relationship between two blocks, both are stable and not easy to change. Readers can refer to [16] for robustness analyses. Consequently, coarse matching is

able to reliably select candidates that contain the desired target.

The clustering operation associated with each entry is performed as follows. It is said that an image’s index id is linked to an entry ERE_i if at least one coarse hash \mathbf{CH} of the image \mathbf{I}_{id} and ERE_i are the same, i.e.,

$$BER(\mathbf{CH}, ERE_i) = 0. \tag{5}$$

Through the above process, the image hash database can be built in an off-line manner. Basically, the built image hash database is error-resilient and tree-structured, and it permits newcomers to join at any time.

The relationship between the image database and hash database will be discussed in Sec. 4 with respect to the complexity issue. Based on the proposed error-resilient coarse-to-fine image hash database, the coarse matching part of the proposed fast matching process will be described in the next section.

3.4.4 Coarse Matching

For an incoming query image, \mathbf{QI} , each of its mesh-based hashes tries to enter the hash database through the entries. It is said that the j -th coarse hash of \mathbf{QI} , $\mathbf{CH}_j^{\mathbf{QI}}$, is allowed to enter an entry E_i if $\mathbf{CH}_j^{\mathbf{QI}}$ and E_i satisfy Eq. (5). Since the idea behind our coarse matching method is to rapidly select candidates for advanced full matching, we first exploit the entries of the hash database to filter out those targets in the image database that are identified as being dissimilar to the incoming query. The goal is to reduce the number of images that are needed for full matching and, thus, to save time.

In our coarse matching process, if a coarse hash of an incoming query \mathbf{QI} is allowed to enter an entry E_i , then the hit indicators of all the image indices that are linked to E_i will be increased by 1 to indicate the gradual increase of the possibility that the images are similar to the query. Let us denote by $\delta(id)$ the hit indicator of an image \mathbf{I}_{id} . When all the coarse hashes of \mathbf{QI} have gone through the above process, we retain those images (in the database) that have hit indicators larger enough as candidates for full matching in order to determine the final winner, i.e., the target with the best match. In fact, our empirical observations indicate that the desired target could be found from only a few candidates (e.g., smaller than 10). Compared with the millions of images in a database, this number of candidates greatly reduces the time required for searching. This also implies that most of the target images have been early eliminated through coarse matching.

3.4.5 Valid or Invalid Retrieval

In the proposed two-stage matching paradigm, “valid retrieval” is defined as follows. Given a query image (**QI**), a hash database, and an image database, it is said that a target image is effectively retrieved to match **QI** if (i) candidates are retrieved that satisfy Eq. (5) during the coarse matching process (Sec. 3.4.4); (ii) the target image is the candidate, together with **QI**, that has the number, N^v , of mesh pairs satisfying Eq. (3) (see Sec. 3.4.2) and $N^v \geq N$. Furthermore, the importance of valid retrievals is determined according to their N^v values. For example, the top n valid retrieval are the ones that have N^v values ranked top n among all the valid retrievals. In content searching and retrieval, the top 1 valid retrieval is regarded as the best match.

On the other hand, if all N^v 's are smaller than N , then this search is considered invalid. As a result, it is concluded that the query image does not exist in the image database.

4 Analyses of Media Hashing Issues

In this section, several challenging issues [9], including error analyses, robustness, granularity, complexity, and scalability of image hashing, will be discussed along with the parameters used in our implementation.

4.1 Robustness and Discrimination

Robustness refers to the ability of image hashes to resist digital operations (including filtering, compression, geometric distortions, and so on) such that the hashes generated before and after attacks are similar. Usually, these digital operations are “incidental” in that they might inevitably be applied for different purposes. Therefore, it is necessary for an image hashing scheme to resist incidental modifications. As indicated in Eq. (1), an image hash is generated as a binary sequence, in which the hash bits are determined according to the magnitudes of the AC coefficients in a normalized mesh. In this situation, keeping the magnitude relationship nearly unchanged (the hash bits nearly unchanged) is necessary to achieve robustness. In the following, we will discuss how the magnitude relationship between AC coefficients can be approximately maintained when incidental operations are employed. It is said that a false negative or miss detection occurs if a query that is a modified version of an image in the image database cannot be correctly identified.

As shown in Eq. (1), an AC coefficient with a larger (or smaller) magnitude is assigned hash bit 1 (or 0). It is said that a bit error occurs if a larger (or smaller) AC coefficient becomes smaller

(or larger). Under this circumstance, it is not hard to realize based on Eq. (1) that bit errors will appear in pairs, i.e., 2 bit errors, 4 bit errors, ..., and so on. In addition, it has been noted in [19] that the intrinsic content of an image can be sufficiently reconstructed from larger transformed coefficients only because they carry significant information. In our previous work [16], we investigated the magnitude relationships between wavelet coefficients as the digital signature of an image for content authentication. These magnitude relationships were found to be mostly preserved under incidental manipulations and to be easily destroyed by malicious distortions. This implies that if a larger (or smaller) coefficient becomes smaller (or larger), then the content of an image will be changed as well. In particular, if more wavelet coefficients in the first image \mathbf{I}_1 are adjusted significantly, based on the same positions of the significant wavelet coefficients in the second image \mathbf{I}_2 , then the content of the \mathbf{I}_1 will gradually look like that of \mathbf{I}_2 . (The reader may refer to [17] for more detailed analyses.) Our previous works [16, 17] revealed that it is necessary to keep larger (smaller) wavelet coefficients larger (smaller) in order to keep the perceptual content of an image relatively unchanged, and we can apply the same principle to the proposed normalized mesh-based hashing that is operated in the block-DCT domain. The major difference is that the inter-scale relationship is considered in the wavelet domain, while the intra-scale relationship is considered in the block-DCT domain to construct an image hash.

On the other hand, discrimination means that a pair of image hashes that are randomly obtained from two different sources should ideally be uncorrelated; i.e., the BER obtained using Eq. (3) should be approximately 0.5. Otherwise, any query is likely to be falsely recognized as existing in the image database. This false detection probability is also known as the false positive probability. Fig. 8 shows the numerical distribution of BERs calculated from 1 million hash pairs that were randomly generated. The length of each hash is 64 to fit the proposed method. The mean and standard deviation of this distribution are 0.5 and 0.062, respectively. It can be found from this result that random hash pairs can statistically achieve an uncorrelated distribution.

In addition, our observations of real images are exactly consistent with the theoretical result shown in Fig. 8. One practical result is shown in Fig. 9 for comparison. The result shown in Fig. 9 was obtained by comparing a hash pair that was composed of a hash of Lenna and a hash of an image within a database (which contained all the hashes of 20,000 images, excluding Lenna). The mean and standard deviation of this distribution are surprisingly found to be 0.5 and 0.062, respectively. This result indicates that our hash extraction method achieves sufficient randomness.

When the best match criterion, as shown in Eq. (3), is taken into consideration, the mean of the BERs obtained from real hash pairs will be shifted. Fig. 10 shows the result obtained by using each

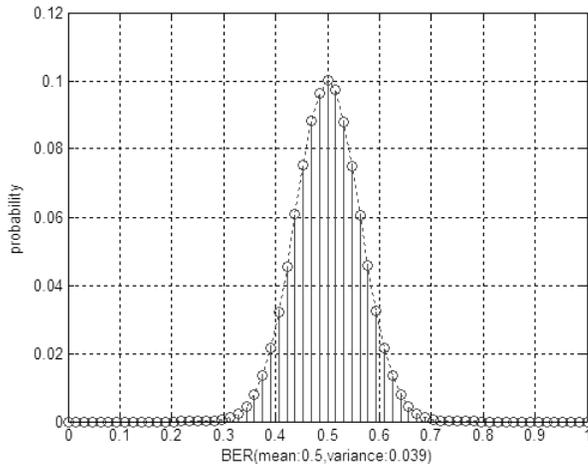


Figure 8: **Distribution of BERs calculated from random hash pairs (each hash has a length of 64). The horizontal axis indicates the bit error rate, while the vertical axis indicates the probability of occurrence of a bit error rate. In this numerical distribution, the mean and standard deviation are 0.5 and 0.062, respectively.**

hash of Lenna as a query to find the best match (with the lowest BER) in a large hash database (which contained all the hashes of 20,000 images, excluding Lenna). Comparing Fig. 10, and Figs. 8 and 9, we can see that the mean of the BERs in Fig. 10 has shifted to a value lower than 0.5. The main reason for this result is that the best match that is produced from among those with the lowest BER is considered.

As previously mentioned in Sec. 3.4, two thresholds, T and N , are relevant to both the false positive and false negative probabilities. Based on the practical results we have observed, it can be experimentally verified that T can be reasonably set to 0.25. The method used to determine the remaining threshold N will be discussed in the next section. Of course, T can also be varied to find different N 's. This tedious task will be ignored in this paper.

4.2 Error Analyses

Error analyses that will be discussed here for threshold selection include false positive and false negative probabilities. Under a sufficiently small false positive and with $T = 0.25$ (note that T can also be used as a variable for analyses), the parameter N that defines the number of matched mesh pairs (as previously mentioned in Sec. 3.4) can be derived. Recall that the hash size of a normalized mesh is 64 bits. It is said that two random meshes (one from the query and the other from the database) are

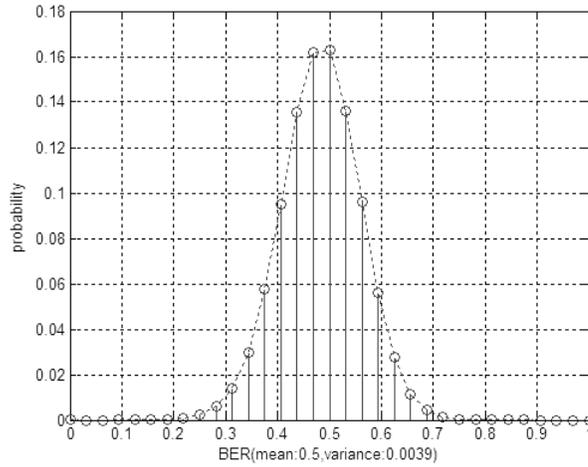


Figure 9: **Distribution of BERs calculated from hash pairs (each hash has a length of 64), where one hash is Lenna’s hash, and the other hash is from the hash database. In this distribution, the mean and standard deviation are 0.5 and 0.062, respectively.**

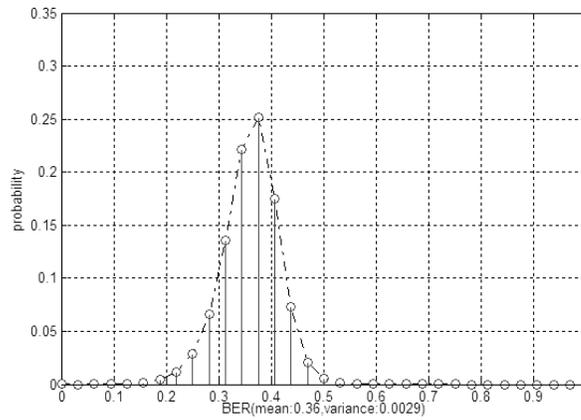


Figure 10: **Distribution of BERs calculated based on the best matched hash pairs (each hash has a length of 64), where one hash is Lenna’s hash, and the other hash is from the image database. In this distribution, the mean and standard deviation are 0.36 and 0.054, respectively.**

similar if their hash comparison satisfies Eq. (3); i.e., at most 16 bits are different. More specifically, the probability, p_m , of finding a pair of hashes that satisfy a matching score (i.e., BER) that is equal to T can be expressed as

$$p_m = \frac{(C_0^{32})^2 + (C_1^{32})^2 + \dots + (C_8^{32})^2}{(C_0^{32})^2 + (C_1^{32})^2 + \dots + (C_8^{32})^2 + \dots + (C_{32}^{32})^2} \approx 6.70 \times 10^{-5}, \quad (6)$$

where C_b^{32} denotes the number of possible cases where $2b$ bits are found to be different between two compared hashes. Since p_m is far larger than those probabilities of finding a match with BER smaller than T , we approximately use the worse case, p_m , to denote the probability of a match. Similarly, $(1 - p_m)$ denotes the probability of mismatch for a pair of hashes.

Based on Eq. (6) and a given value of N , the false positive probability, p_{fp} , is defined as

$$p_{fp} = \sum_{n=N}^{|\mathcal{H}|} C_n^{|\mathcal{H}|} (1 - p_m)^{|\mathcal{H}|-n} p_m^n \geq C_N^{|\mathcal{H}|} (1 - p_m)^{|\mathcal{H}|-N} p_m^N \approx C_N^{|\mathcal{H}|} p_m^N, \quad (7)$$

where $|\mathcal{H}|$ denotes the number meshes in an image, $C_n^{|\mathcal{H}|} (1 - p_m)^{|\mathcal{H}|-n} p_m^n$ with $n > N$ is sufficiently smaller than $C_N^{|\mathcal{H}|} (1 - p_m)^{|\mathcal{H}|-N} p_m^N$, and $(1 - p_m)^{|\mathcal{H}|-N}$ is approximately 1.

On the other hand, the probability of failing to retrieve the desired target should be smaller than the false negative probability, p_{fn} . However, it is hard to analyze p_{fn} since various discrepancies between the characteristics of digital image operations exist. Based on our observations, the false negative probability can increase in the following two cases: (i) the query image is modified to a certain extent such that most of the detected meshes are different from the originals; (ii) the mesh is kept unchanged, but its hash is mostly changed. From our experimental results, we find that most false negatives occur in the first cause.

Substituting some cases of $|\mathcal{H}|$ and N into Eq. (7) and solving p_{fp} , we find several relationships between $|\mathcal{H}|$, N , and sufficiently small p_{fp} , which are depicted in Table 1. From Table 1, we can observe that if the number of meshes, $|\mathcal{H}|$, is on the order of $20 \sim 40$, then $N = 3$ or 4 is feasible for obtaining a sufficiently small p_{fp} . Of course, we can use a larger N value to get an even smaller false positive probability. However, a larger N value also implies that false negatives can easily occur. In this paper, it has been analyzed that $N = 3$ is feasible under the constraints of sufficiently small false positive and false negative probabilities.

Table 1: **Relationship between parameters, N , $|\mathcal{H}|$, and p_{fp} under $T = 0.25$.**

N	$ \mathcal{H} $	p_{fp}
3	20	3.46×10^{-10}
3	30	1.23×10^{-9}
3	40	2.99×10^{-9}
4	20	9.90×10^{-14}
4	30	5.59×10^{-13}
4	40	1.87×10^{-12}

4.3 Complexity

The complexity of an image hashing system that will be discussed here includes the time spent on mesh-based hash extraction, hash comparison, and hash database construction. As for mesh-based hash extraction, it is actually the most time-consuming step in our method because most of the time is spent on warping during the mesh normalization process. As for hash comparison, this step has been previously described in Sec. 3.4. Since a sophisticated hash database is created for fast coarse matching followed by full matching, the time required for hash comparison, when compared with that required for full matching [18], is significantly reduced. This is because (i) the time-cost of full matching depends on the number of images in the image database; (ii) the time-cost of fast matching depends on the number of meshes in the query image and on the small but fixed number of candidates that should be used for full matching. As for the size of the proposed hash database, it is the sum of the total size of the entries, the total size of the image indices, and the total size of the image hashes, as illustrated in Fig. 7. Because each entry is a binary vector of 16 bits, the entry size is, in total, $2^{16} \times 16$ bits, i.e., 131,072 bytes. In addition, the total size of the image indices can be calculated as $\frac{|ID| \times \lceil \log_2^{|ID|} \rceil \times |\mathcal{M}|}{8}$ bytes, where $|ID|$ denotes the number of images in the image database, $\lceil \log_2^{|ID|} \rceil$ denotes the number of bits used to represent an image’s index ($\lceil \cdot \rceil$ denotes the ceiling operation), and $|\bar{\mathcal{M}}|$ is the average number of meshes in an image. Furthermore, the total size of the image hashes can be calculated as $\frac{|ID| \times |\bar{\mathcal{M}}| \times 64}{8}$ bytes. Therefore, the ratio σ between the size of a hash database and the size of its corresponding image database can be calculated as follows:

$$\sigma = \frac{131072 + \frac{|ID| \times \lceil \log_2^{|ID|} \rceil \times |\bar{\mathcal{M}}|}{8} + \frac{|ID| \times |\bar{\mathcal{M}}| \times 64}{8}}{|ID| \times \bar{L} \times \bar{W}}$$

$$\approx \frac{(\lceil \log_2^{|ID|} \rceil + 64) \times |\bar{\mathcal{M}}|}{8 \times \bar{L} \times \bar{W}}, \quad (8)$$

where $\bar{L} \times \bar{W}$ is used to denote the average size of various images.

To clarify Eq. (8), let us take an image database of size 131,072; i.e., $|ID| = 2^{17}$, as an example. In addition, suppose $\bar{L} \times \bar{W} = 2^{16}$ and $|\bar{\mathcal{M}}| = 2^6$ approximately hold. Then, we find that $\sigma \approx 0.01$, which means that the size of the hash database when compared with the size of the image database can be limited within the order of 10^{-2} .

4.4 Granularity

Granularity here means the minimum size of a query that can be identified in a feasible way in an image hashing system. As explained in Sec. 3.4.5, a valid retrieval is defined one in which at least N hash pairs are matched; thus, the granularity is basically the size of a query that can accommodate at least N meshes. However, it is rather difficult to settle on a fixed value for this size since the mesh's size is image-dependent and may vary under different attacks.

4.5 Scalability

The scalability of an image hashing system as discussed here is application scalability; i.e., the same hash database can be used for different purposes. In this paper, copy detection/tracing and content authentication applications will be investigated. Since we have described the proposed image hashing method for copy detection, in the following we will explain how the proposed hash database can be adopted for image content authentication.

Imagine the scenario shown in Fig. 11, in which Alice would like to send an image to Bob through the network. During the transmission stage, in addition to the transmitted image, \mathbf{I} , Alice will also send the hashes of \mathbf{I} to Bob, who can verify the authenticity of the received image, \mathbf{I}^a , according to the hashes, where \mathbf{I}^a is a modified version of \mathbf{I} . Image content authentication [4, 15, 16] demands that hashes be robust against incidental modifications and sensitive to malicious distortions.

In this paper, the proposed mesh-based image authentication scheme, as indicated by the authentication block shown in Fig. 11, operates as follows. First, the mesh-based hashes are extracted from the received image, \mathbf{I}^a , and compared with the received hashes of \mathbf{I} using Eq. (3). Then, those meshes whose corresponding hashes do not satisfy Eq. (3) will be marked as “incredible.”

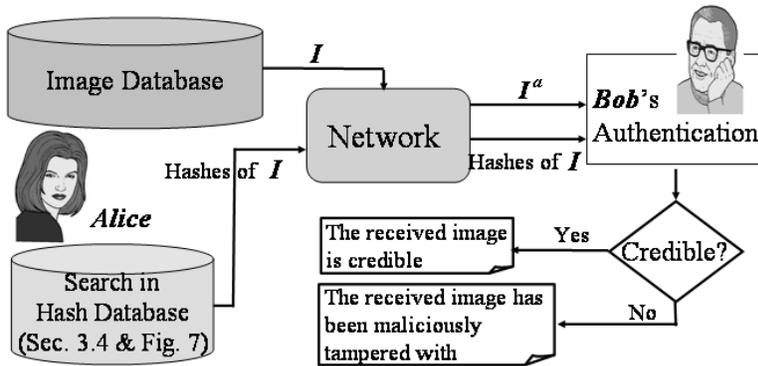


Figure 11: **The same hash database can be used for image authentication. In an authentication scenario, the sender (Alice) sends an image and its hashes to the receiver (Bob). Then, Bob verifies the authenticity of the received image by comparing the retrieved hashes with the received hashes.**

5 Experimental Results

In this study, several experiments were conducted to evaluate the performance of the proposed mesh-based image hashing and query system. In Secs. 5.1 and 5.2, the performance of our copy detection scheme will be demonstrated, while in Sec. 5.3, the performance of our image content authentication approach will be demonstrated.

5.1 Robustness: Resistance to Miscellaneous Attacks

First, ten color images with different contents (I_1 : Pepper; I_2 : Lenna; I_3 : Bridge; I_4 : Sailboat; I_5 : Goldhill; I_6 : F16; I_7 : Baboon; I_8 : Clock; I_9 : Tank; and I_{10} : Splash) were used to verify the robustness of our scheme against miscellaneous attacks. The standard benchmark, Stirmark versions 3.1 and 4.0, was quite suitable for simulating various manipulations of the digital images. The reader may refer to [23, 24] for more detailed parameters of Stirmark. In this test, the original image was used as a query to find out how many modified versions could be successfully detected. The results for robustness verification are summarized in Tables 2 and 3, respectively. In the two tables, each attack's name is followed by a digit, which indicates the number of times that the attack was performed with different parameters. In addition, each field indicates the number of modified images that were successfully identified. Here, $N = 3$ and $T = 0.25$, as explained in Secs. 3.4 and 4.2, were adopted. According to Tables 2 and 3, among 1,910 modified images, 1761 of them could be correctly identified, which indicates that the correct recognition rate was 92.2%.

Table 2: **Robustness of our scheme vs. Stirmark 3.1:** attacks are denoted as SPA: the Signal Processing Attack, including median filtering, Gaussian filtering, sharpening, and Frequency Mode Laplacian Removal (FMLR); JPEG: compression with quality factors ranging from 90% to 10%; GLGT: General Linear Geometric Transform; CAR: Change of the Aspect Ratio; LR: Line Removal; RC: Rotation+Cropping; Scaling: scaled with factors ranging from 0.5 to 2.0; RRS: Rotation+ReScaling; RB: Random Bending.

Stirmark 3.1	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈	I ₉	I ₁₀
SPA (6)	6	6	6	6	6	6	6	6	6	6
JPEG (12)	12	12	12	12	12	12	12	12	12	12
GLGT (3)	3	3	3	3	3	3	3	3	3	3
CAR (8)	8	8	8	8	8	8	7	8	8	8
LR (5)	5	5	5	5	5	5	5	5	5	5
Flipping (1)	1	1	1	1	1	1	1	1	1	1
Cropping (9)	8	7	7	8	8	8	4	8	3	6
RC (16)	15	15	15	15	14	15	12	15	14	15
Scaling (6)	6	6	4	6	6	6	2	6	4	4
RRS (16)	15	15	15	16	15	16	10	16	12	14
Shearing (6)	6	6	6	6	6	6	6	6	6	6
RB (1)	1	1	1	1	1	1	1	1	1	1

Table 3: **Robustness of our scheme vs. Stirmark 4.0:** attacks are denoted as **AffineT:** Affine Transformation; **ConvF:** Convolution Filtering; **Cropping:** cropped to $\frac{3}{4}$, $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{5}$ the original size; **JPEG:** compression with quality factors ranging from 90% to 10%; **MF:** Median Filtering; **Noise:** noise addition; **SS:** Self-Similarities; **Scaling:** scaled with factors ranging from 0.5 to 2.0; **RML:** Removing Lines; **PSNR:** all pixel values increased by the same quantity; **Rotation:** pure rotation; **RRS:** Rotation+ReScaling; and **RC:** Rotation+Cropping.

Stirmark 4.0	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈	I ₉	I ₁₀
AffineT (8)	8	8	8	8	8	8	8	8	8	8
ConvF (2)	2	2	2	2	2	2	1	2	1	1
Cropping (4)	2	1	1	1	2	2	0	1	1	2
JPEG (12)	12	12	12	12	12	12	12	12	12	12
MF (4)	4	4	4	4	4	4	4	4	4	4
Noise (6)	1	1	1	2	1	2	1	1	1	1
SS (3)	3	3	3	3	3	3	3	3	3	3
Scaling (6)	5	6	4	6	5	6	4	6	4	6
RML (10)	10	10	10	10	10	10	10	10	10	10
PSNR (11)	11	11	11	11	11	11	11	11	11	11
Rotation (16)	16	16	16	16	16	16	16	16	16	16
RRS (10)	10	10	10	10	10	10	10	10	10	10
RC (10)	10	10	10	10	10	10	10	10	10	10

Moreover, it can be observed that most of the modified images could be successfully detected except for few exceptions. A few attacked images that failed to be identified are shown in Fig. 12 for visual inspection. We can observe from Fig. 12 that it was not possible to correctly extract meshes from attacked images involving markedly degraded fidelity and content elimination. In particular, severe cropping and heavy noise addition could break the connections among the meshes and thereby affect the hashes, thus defeating our system even though the attacked images might have lost their commercial value. However, compared with the existing methods [2, 3, 5, 10, 11, 12, 21, 25, 26], it is evident that our scheme indeed achieves promising resistance to extensive geometric distortions.

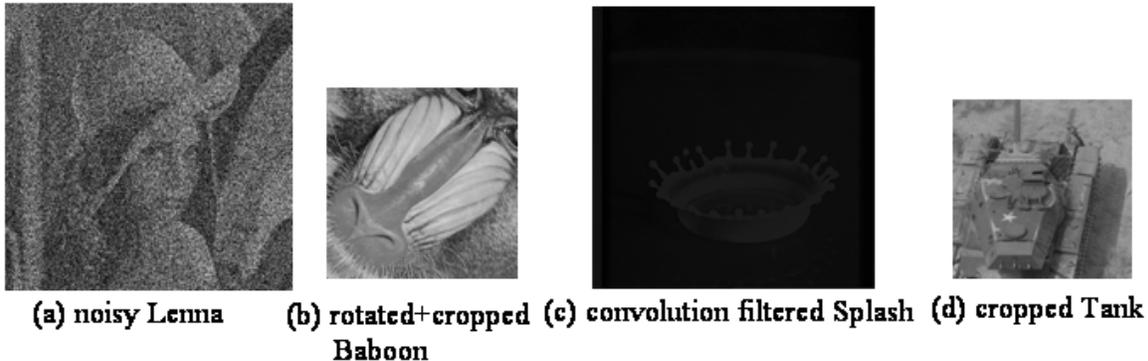


Figure 12: **Examples of failures in the robustness test. (a) and (c) were caused by Stirmark 4.0, and (b) and (d) were caused by Stirmark 3.1.**

5.2 Identification: Searching in a Large Database

The second group of experiments focused on the problem of searching in a large image database. In this searching system, the database was composed of so-called original color images (which consisted of the Corel image database containing 20,000 images and ten traditional images, such as Lenna, Baboon, and so on), while the query image was suspect in the sense that it may have been a modified version generated from our database or may have been totally irrelevant to the database. We used the attacked images, obtained from Stirmark 3.1 and 4.0, as queries of the database.

Two measures, the recall rate and precision rate, were used to evaluate the searching performance. They are dependent on the parameters T , N , and n , and are, respectively, defined as follows:

$$Recall(T, N, n) = \frac{\text{No. of queries satisfying Eq.(3)}}{\text{No. of total queries}}, \quad (9)$$

$$Precision(T, N, n) = \frac{\text{No. of queries satisfying Eq.(3)}}{\text{No. of detections satisfying Eq.(3)}}, \quad (10)$$

where n means the number of valid retrievals that may include the desired target. Both the full matching and fast matching procedures were applied to searching, and their results were compared.

According to the full matching process (described in Sec. 3.4.2), a so-called “successful search” needs to be defined to evaluate the searching performance. Here, a successful search means that at least one of the valid retrievals (described in Sec. 3.4.5) contains the desired target. In this study, the top n valid retrievals were adopted to check whether at least one of them contained the desired target, where different values of n included 1, 2, 5, 10, and 100, respectively. In addition, we provide here information about the overall performance in terms of both the recall and precision rates, as depicted in Table 4 [†]. As can be seen from Table 4, the overall searching results could only be significantly improved (up to $\approx 90\%$) if the number of valid retrievals was less than 10. This outcome demonstrates that the proposed searching strategy is very efficient in finding the desired target without the need to check many valid retrievals. Note that the number of miss detections was slightly larger than that obtained in the robustness test (as explained in Sec. 5.1) because the search space had been broadened. Basically, these results reveal that the desired originals were hard to identify when the query images (e.g., Fig. 12) had been severely modified. Moreover, the miss detected queries were mostly consistent with those that could not be identified in the robustness test.

Table 4: **Recall rate vs. precision rate for full searching with Stirmark.**

Searching style	Full Searching									
	<i>Stirmark 3.1 (890 queries)</i>					<i>Stirmark 4.0 (1,020 queries)</i>				
Query sources										
Top n matches	1	2	5	10	100	1	2	5	10	100
Recall rate (%)	82.1	86.5	90.7	93.3	94.5	84.4	87.1	89.4	90.4	91.2
Precision rate (%)	82.1	43.3	18.1	9.3	0.9	84.4	43.5	17.9	9.0	0.9

In order to speed up the search process in the case of a large database, the proposed fast matching process (as described in Sec. 3.4) was also employed. The time-cost can be greatly saved due to entry entrance checking offers early elimination of those images that are dissimilar to the query. The overall performance of fast searching in terms of both the recall and precision rates is depicted in Table 5. Comparing Table 4 and Table 5, we find that the proposed fast searching strategy is efficient and

[†]It should be noted that the traditional precision rate vs. recall rate measurement used in content retrieval may not be suitable for media hashing. This is because we are interested in identifying the correct target instead of just similar ones. As a result, increasing the number of detections will dramatically decrease the precision rate.

achieves comparable performance as full searching.

Table 5: **Recall rate vs. precision rate for fast searching with Stirmark.**

Searching style	Fast Searching									
	<i>Stirmark 3.1 (890 queries)</i>					<i>Stirmark 4.0 (1,020 queries)</i>				
Query sources										
Top n matches	1	2	5	10	100	1	2	5	10	100
Recall rate (%)	80.5	84.5	87.2	87.2	87.2	83.6	85.1	86.0	86.2	86.2
Precision rate (%)	80.5	42.5	17.5	8.7	0.9	83.6	42.5	17.2	8.6	0.9

5.3 Content Authentication

In this section, image content authentication performed using the proposed hash database will be analyzed. As shown in Fig. 11, the original image that is sent from Alice to Bob may be modified by an attacker who can access the transmitted data. If the attacker only manipulates the image by means of incidental modifications, the authenticated results will be the same as the results in the robustness test, as explained in Sec. 5.1. In addition, we will further consider two scenarios of actions that the attacker may perform: (i) malicious modification; and (ii) malicious modification+incidental modification. Some results with respect to the above two scenarios are shown in Figs. 13 ~ 17, respectively.

Fig. 13 shows two original images that were transmitted from the sender Alice to the receiver Bob. During transmission, the original images were maliciously tampered with by pasting in an additional object to form modified images, as shown in Figs. 14(a) and 15(a). Since the newly added objects destroyed the original mesh structures and their corresponding hashes, the areas (containing a number of meshes) that contained the pasted object were located, as shown in Figs. 14(b) and 15(b). On the other hand, the images shown in Figs. 14(a) and 15(a) were further manipulated using some incidental operations. The resultant attacked images are shown in Figs. 16(a) and 17(a), respectively. After our authentication scheme was performed, the areas that were detected as having been maliciously tampered with were shown in Figs. 16(b) and 17(b), respectively.

It can be observed from the above authentication results that the areas that were detected as having been maliciously tampered were composed of two kinds of meshes. The first kind was meshes newly generated from pasted objects, while the second kind is meshes generated due to some salient points that changed due to incidental manipulations. However, our method in its current form could not

distinguish between the two situations. A more robust salient point extraction technique is required.

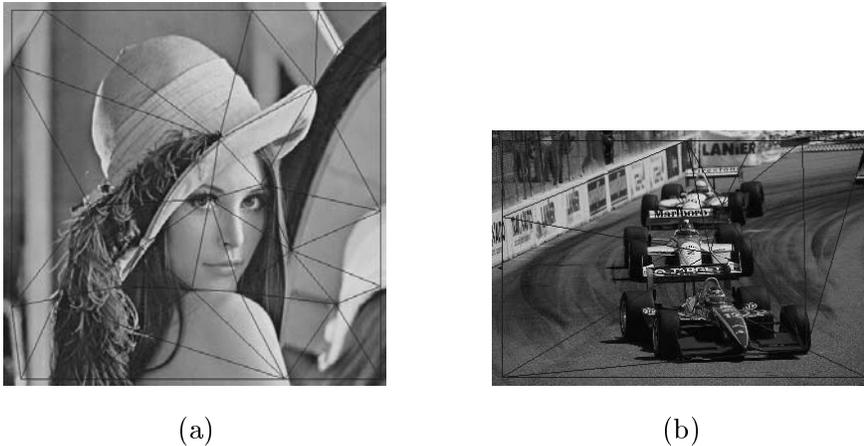


Figure 13: **Original images for content authentication. The detected meshes are imposed on the images.**

6 Conclusions

A robust mesh-based image hashing scheme for content management of digital images has been proposed in this paper. Our scheme is mainly composed of two components, including (i) mesh-based robust hash generation and (ii) hash database construction for error-resilient and fast searching. In comparison with the existing methods, the main contribution of our approach is that it significantly improves the resistance of image hashing to geometric distortions. Furthermore, we have investigated several media hashing issues, including robustness and discrimination, error analysis, complexity, granularity, and scalability. We have also demonstrated the application of the robust mesh-based image hashing system to both copy detection and content authentication.

However, our scheme is somewhat complex in that most of the time is consumed in mesh normalization. Fortunately, the hash database used for querying and searching can be built in an off-line manner. As a result, time is mainly spent on the incoming query image. However, our scheme compensates for this cost with promising robustness against geometric distortions. A fast matching process has also been proposed to speed up searching in a large image database. To understand the impact of different parameters on the false alarm rate, error analyses were conducted to derive guidelines for determining the necessary parameters.

Some directions for further research have been identified as follows. First, we will study the challenging problem of more robust feature point extraction for mesh generation. This problem is



(a)

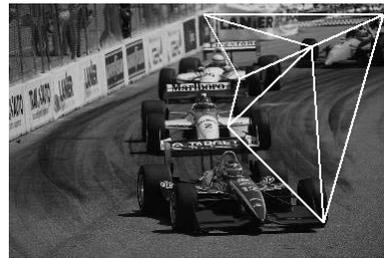


(b)

Figure 14: Content authentication under malicious tampering: (a) the image modified through face substitution on the middle part of Fig. 13(a); (b) the areas that were detected as having been maliciously tampered with are indicated by meshes with white lines.



(a)



(b)

Figure 15: Content authentication under malicious tampering: (a) the image modified through object substitution applied to the top-right part of Fig. 13(b); (b) the areas that were detected as having been maliciously tampered with are indicated by meshes with white lines.



(a)



(b)

Figure 16: **Content authentication under malicious tampering+incidental modification (affine transformation): (a) the modified image and its detected meshes; (b) the areas that were detected as having been maliciously tampered with are indicated by meshes with white lines.**



(a)



(b)

Figure 17: **Content authentication under malicious tampering+incidental modification (JPEG with quality factor 30%): (a) the modified image and its detected meshes; (b) the areas that were detected as having been maliciously tampered with are indicated by meshes with white lines.**

particularly crucial for both identification of small images and mesh-based image authentication. Second, we will extend the scope of our method to searching and identifying images in URLs. Finally, the security issue of media hashing that may be application-dependent is worth studying.

Acknowledgment: This research was supported, in part, by the National Science Council under NSC grant 92-2422-H-001-004. The authors thank Mr. Sun for providing the mesh generation algorithm.

References

- [1] P. Bas, J. M. Chassery, and B. Macq, “Geometrically invariant watermarking using feature points,” *IEEE Trans. Image Processing*, Vol. 11, pp. 1014-1028, 2002.
- [2] E. Y. Chang, J. Z. Wang, C. Li, and G. Wiederhold, “RIME: A Replicated Image Detector for the World-Wide-Web,” *Proc. SPIE: Multimedia Storage and Archiving Systems*, Vol. III, 1998.
- [3] E. Y. Chang, C. Li, J. Z. Wang, P. Mork, and G. Wiederhold, “Searching Near-Replicas of Images via Clustering,” *Proc. SPIE Symposium of Voice, Video, and Data Communications*, pp. 281-92, Boston, 1999.
- [4] J. Dittmann, A. Steinmetz, and R. Steinmetz, “Content-based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarking,” *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, Italy, Vol. II, pp. 209-213, 1999.
- [5] J. Fridrich, “Visual Hash for Oblivious Watermarking,” *Proc. SPIE: Security and Watermarking of Multimedia Contents II*, 2000.
- [6] J. Fridrich and M. Goljan, “Robust Hash Functions for Digital Watermarking,” *Proc. IEEE Int. Conf. on Information Technology: Coding and Computing*, pp. 178-183, 2000.
- [7] C. Y. Hsu and C. S. Lu, “Geometric Distortion-Resilient Image Hashing System and Its Application Scalability,” *Proc. ACM Multimedia and Security Workshop*, Magdeburg, Germany, 2004.
- [8] *IEEE Int. Conf. on Multimedia and Expo: special session on Media Identification*, J. Oostveen, C. S. Lu, and Q. Sun (co-organizers), June 2004.
- [9] T. Kalker, “Applications and Challenges for Audio Fingerprinting,” *Proc. 111th AES Convention, in the “Watermarking versus Fingerprinting” Workshop*, December 3, 2001.

- [10] C. Kim, "Content-based Image Copy Detection," *Signal Processing: Image Communication*, Vol. 18, pp. 169-184, 2003.
- [11] F. Lefebvre and B. Macq, "RASH : RADon Soft Hash algorithm," *Proc. European Signal Processing Conference*, Toulouse, France, 2002.
- [12] F. Lefebvre, J. Czyz, and B. Macq, "A Robust Soft Hash Algorithm for Digital Image Signature," *Proc. IEEE Int. Conf. on Image Proc.*, 2003.
- [13] B. Li, E. Chang, and C. T. Wu, "DPF – A Perceptual Distance Function for Image Retrieval," *Proc. IEEE Int. Conf. on Image Processing*, 2002.
- [14] C. Y. Lin and S. F. Chang, "Generating Robust Digital Signature for Image/Video Authentication," *Proc. ACM Multimedia and Security Workshop*, UK, 1998.
- [15] C. Y. Lin and S. F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 11, No. 2, pp. 153-168, 2001.
- [16] C. S. Lu and H. Y. Mark Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Trans. on Multimedia*, Vol. 5, No. 2, pp. 161-173, 2003.
- [17] C. S. Lu, "On the Security of Structural Information Extraction/Embedding for Images," *Proc. IEEE Int. Symposium on Circuits and Systems*, Vancouver, Canada, 2004.
- [18] C. S. Lu, C. Y. Hsu, S. W. Sun, and P. C. Chang, "Robust Mesh-based Hashing for Copy Detection and Tracing of Images," *Proc. IEEE Int. Conf. on Multimedia and Expo: special session on Media Identification*, Taipei, Taiwan, 2004.
- [19] S. Mallat and S. Zhong, "Characterization of Signals from Multiscale Edges," *IEEE Trans. on Pattern Anal. and Machine Intell.*, Vol. 14, No. 7, pp. 710-732, 1992.
- [20] Y. Meng and E. Chang, "Image Copy Detection Using Dynamic Partial Function," *Proc. SPIE Storage and Retrieval for Media Database*, Vol. 5021, pp. 176-186, 2003.
- [21] M. K. Mihcak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," *Proc. ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, PA, 2001.

- [22] *IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, special session on Media Recognition, Virgin Islands, USA, 2002.
- [23] F. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on Copyright Marking Systems," *Proc. Int. Workshop on Information Hiding*, LNCS 1575, pp. 219-239, 1998.
- [24] F. Petitcolas, "Watermarking Schemes Evaluation," *IEEE Signal Processing Magazine*, Vol. 17, No. 5, pp. 58-64, 2000.
- [25] J. S. Seo, J. Haitisma, T. Kalker, and C. D. Yoo, "A Robust Image fingerprinting system Using the Radon Transform," *Signal Processing: Image Communication*, Vol. 19, pp. 325-339, 2004.
- [26] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," *Proc. IEEE Int. Conf. Image Processing*, 2000.
- [27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. on Image Processing*, Vol. 13, No. 4, pp. 600-612, 2004.