Robust Image Hashing for Searching Illegal Copies

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

ABSTRACT

Media hashing is an alternative to achieve 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, our aim is to propose a geometry-invariant image hashing scheme, which can be employed for copy detection and content authentication of digital images. Our scheme is mainly composed of two components: (i) mesh-based robust hash generation and (ii) hash database construction for error-resilient and fast matching. In addition, we further investigate the issues of robustness, error analyses, complexity, granularity, and scalability for the proposed image hashing system. Exhaustive experimental results obtained from benchmark attacks have confirmed the performance of the proposed method.

Keywords: Copy detection, Media hashing, Mesh, Robustness, Searching, Similarity

1. INTRODUCTION

With the advancement of multimedia and networking technologies, it becomes easy to copy the original completely and distribute the illegal copies rapidly over the Internet. In order to trace unauthorized uses of digital contents, media hashing technologies have attracted much attention recently in digital content management. In contrast with data hiding, the main characteristic of media hashing is its non-invasive property, which means that no information is required to be embedded into digital content. On the contrary, a hash sequence for a specific media data needs to be extracted to represent its condensed essence. Analogous to media hashing, there exists some synonymous terminologies in the literature including fingerprinting, digital signature, and passive/non-invasive watermarking. The major difference 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 needs to transfer a media data into a short string for compact representation. The technical diversity between them is that media hashing is required to resist (either malicious or incidental) attacks. Therefore, several applications that call for robust identification of media contents are seeking robust media hashing methods [6, 16]. The technical need for media hashing cannot be accomplished by traditional cryptographic hashing functions because even one bit error in a hash sequence can lead to represent entirely different contents. However, reasonable distortions are not harmful to the visual quality and commercial value of multimedia data.

Most of the existing media hashing methods are developed for audio identification [16], in this paper, we are devoted to the study of image hashing. First of all, literature review about image hashing will be discussed as follows. In 1998, Chang *et al.* proposed a wavelet-based Replicated IMage dEtector (RIME) [2] to search 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 the images. Their system's capability is remarkably prohibited from resisting 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 the I/O efficiency by clustering and retrieving relevant information sequentially on and from the disk. Recently, Meng and Chang [14] used multi-scale color and texture features to characterize images and employed dynamic partial function (DPF) to measure image perceptual similarity. Although DPF outperformed traditional distance metrics, the adopted image feature was global in that resistance to geometric distortions is inherently limited. In [9, 10], digital signature has been proposed for image authentication. Lin and Chang [9] created the mutual relationship of pairwise block-DCT coefficients to distinguish JPEG compressions from malicious modifications. Lu and Liao [10] built the so-called "structural digital signature" from the multiscale structure of wavelet transform to tolerate incidental manipulations and reflect intentional manipulations. However, the ability of resisting geometric manipulations was a lack of [9, 10]. In [4], Fridrich and Goljan proposed a robust/visual hashing method. Their hash digests of digital images were created by projections of DCT coefficients to key-dependent random patterns. In [20], Venkatesan *et al.* proposed an image hashing technique, which contains (i) random tiling of an image's coarse

*Corresponding author: C. S. Lu

subband (using a three-level Haar wavelet decomposition); and (ii) hash generation from statistical feature extraction of tiles. However, the two methods [4, 20] only allows limited resistance to geometric distortions. In [8, 19], image hashing methods were proposed based on the Radon transform by exploiting its affine invariance. However, resistance to geometric distortions is greatly limited if the incoming attacks go beyond affine distortions. In [15], Mihcak and Venkatesan proposed an iterative geometric image hashing method. Their method can only withstand slight geometric distortions. In [7], Kim proposed an image copy detection scheme by means of ordinal measures of AC coefficients in the 8×8 DCT domain, i.e., the magnitudes of AC coefficients in a block were ranked in descending order to represent an image. However, this system basically cannot 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 cause lies in the fact that the previous methods did not really deal with the problem of combating geometric attacks. In view of this, the purpose of this paper aims to treat 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 to achieve robustness against extensive geometric distortions (e.g., standard benchmark like Stirmark3.1 and Stirmark4.0 [17, 18]). Although the concept of image meshing has been applied to watermarking [1], we have taken notice of the stability of mesh generation that is closely related to the success of intended purposes. Consequently, we present a robust mesh extraction method such that it won't be easily harmful to mesh-based hashing. We also present a robust mesh-based hash extraction method by considering content position-dependent features. Extensive results obtained from benchmark attacks have further confirmed the robustness of the proposed scheme.

In addition to the robustness issue, the practical use of an image hashing system requires fast search in a large database. In this paper, we will also describe how an efficient hash database could be built to facilitate fast hash matching. Moreover, we shall investigate the error analyses, complexity, granularity, and scalability issues of the proposed image hashing system.

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 needs to cope with. In Sec. 3, the proposed image hashing system that is composed of mesh generation, mesh-based hash generation, coarse-to-fine hash database construction, and fast hash matching, will be described. Many issues of media hashing including robustness, error analyses, complexity, and scalability can be found in [5, 12]. Extensive experimental results will be given in Sec. 4 to verify the performance of our scheme. Finally, concluding remarks will be drawn in Sec. 5.

2. PROBLEM STATEMENT

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

Referring to the image space as shown in Fig. 1, let $\mathbf{I}$ denote an image, and $\mathcal{X}$ denote the set of images that are modified from $\mathbf{I}$ by means of content-preserving operations (e.g., filtering, compression, geometric distortions and etc) and are defined to be perceptually similar to $\mathbf{I}$. We further use $\mathcal{Y}$ to denote those images that are actually modified from $\mathbf{I}$ but are hardly recognized to be originated from $\mathbf{I}$. 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 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.

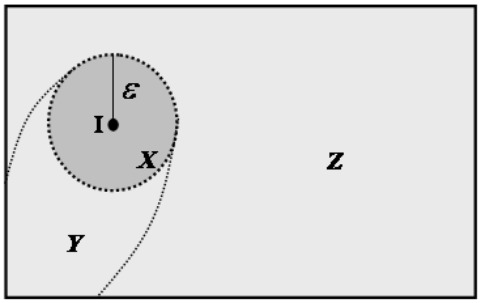

Figure 1: The Image Space. I is an element in the image space. $\mathcal{X}$ denotes the set of images modified from I that are still perceptually similar to I. $\mathcal{Y}$ denotes the set of images modified from I that permits to be unrecognizable. $\mathcal{Z}$ is the set of images that are irrelevant to 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 collision-free, which means that it is hard to find two different images that can be transferred to produce the same hashes. Let $z \in \mathcal{Z}$, and z and $\mathbf{I}$ are distinct. The collision-free property of cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(z)$. Furthermore, let $x \in \mathcal{X}$ and cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(x)$, too. This implies that cryptographic hashing inherently produces totally different hash sequences if 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 of media hashing is defined as follows. It is said that x ($\in \mathcal{X}$) is successfully identified to be 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 collision-free property like cryptographic hashing except that the distance measure is changed as $d(H^m(\mathbf{I}), H^m(x)) > \epsilon$. On the other hand, it is insignificant about whether y ($\in \mathcal{Y}$) can be identified to be modified from $\mathbf{I}$ or not. It should be noted that traditional cryptographic hash function is a special case of media hash function in that its ϵ is set to be 0. Overall, the main theme of media hashing is to develop a robust hash function that can identify perceptually similar media contents and achieve collision-free. Relevant issues of media hashing have been analyzed in [5].

3. PROPOSED 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 the geometric-resilience is not restricted to affine transformations [8, 19]. Previous approaches proposed via this paradigm were reported in [5, 12] and promising capabilities of robustness and discrimination were obtained. In this paper, we generalize our previous methods to provide more analyses of hashing issues and more extensive set of verification results. The major components of our proposed image hashing system will be sequentially described in this section.

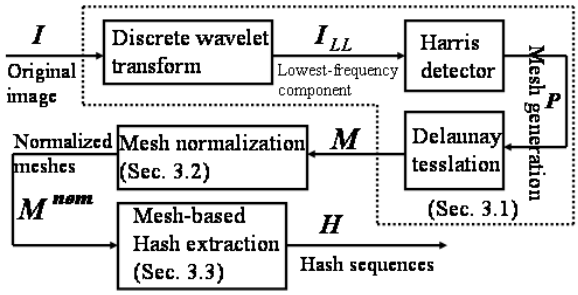

Figure 2: Block diagram of the proposed mesh-based image hashing system.

3.1 Robust Image Mesh Generation

Extraction of robust meshes plays an important role in our method since it is a prerequisite in withstanding geometric distortions. To generate meshes, the first step is to detect salient points of an image. Among the ubiquitous feature point extraction methods, Harris detector has been popularly 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 carrying out it in the lowest-frequency subband of the discrete wavelet transform (DWT) domain [12]. Our intention is to filter out noisy points before salient point detection.

Once the feature point extraction process is finished, the Delaunay tessellation is exploited to decompose the image into

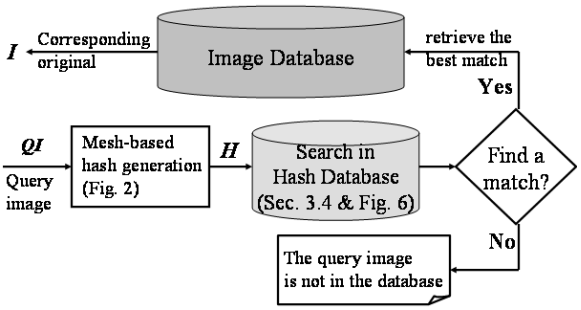

Figure 3: Block diagram of the proposed image query system: a query image (QI) enters into the hash database for possible retrieval of its original from the image database.

a set of disjointed triangles. Each triangle (called a mesh hereafter) 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 discrete wavelet transformed to obtain the lowest-frequency subband signal, $\mathbf{I}_{LL}$; (ii) the set of feature points $\mathcal{P}$ are generated by means of applying the Harris detector on $\mathbf{I}_{LL}$; 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 shown in Fig. 4, which contains the generated meshes from the original Lenna and its Stirmark attacked Lenna images. By visual inspection, we can find that several meshes are consistently extracted. Although we can see a few differences between the generated meshes (e.g., top-right of Fig. 4(d) and top-left of Fig. 4(e)), these differences are not certain to affect the hash-based similarity measurement, as will be described in Sec. 3.4. These results illustrate the effectiveness of mesh extraction from the lowest-frequency subband of an image.

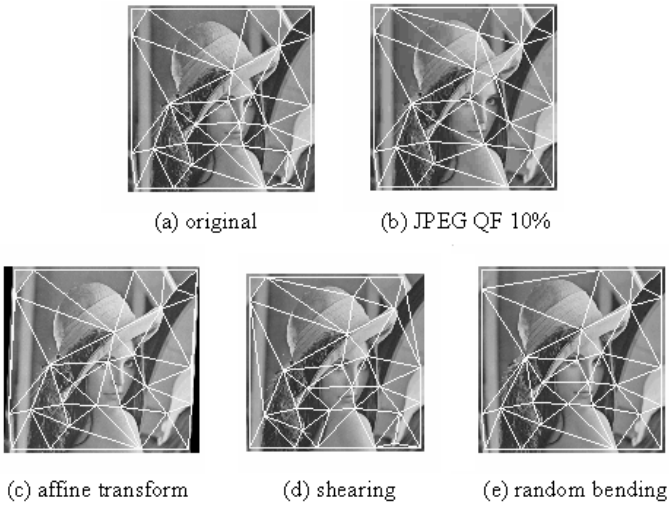

Figure 4: Illustration of extracted meshes from original and attacked Lenna images (Initially, they are color.). Note that the meshes are detected at the lowest-frequency subband of an image after two-level wavelet decomposition.

3.2 Mesh Normalization

Once the set of meshes in an image has been produced, each original mesh $\mathbf{M_k}(\in \mathcal{M})$ will be normalized as $\mathbf{M_k^{nom}}$ to generate a mesh-based hash $\mathbf{H_k}$, where $\mathbf{M_k^{nom}}$ is a right-angled isosceles triangle. The aim of normalization is to maintain that all normalized hashes are of the same size and the extracted mesh-based hashes are of the same length to enable mesh-based hash comparisons. Fig. 5 illustrates the relationship between $\mathbf{M_k}$ and $\mathbf{M_k^{nom}}$, where $< A, B, C >$ and $< \acute{A}, \acute{B}, \acute{C} >$ denote the corners of $\mathbf{M_k}$ and $\mathbf{M_k^{nom}}$, respectively. In addition, let $< A, B, C >$ 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, $< A, B, C >$ is first mapped to $< \acute{A}, \acute{B}, \acute{C} >$ sequentially. That is, this "angle order" must be maintained to keep uniform warping in order not to 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 with different angle order and thereby the resultant normalized meshes will produce different hashes. After angle order-based corner mapping, $\mathbf{M_k}$ is transformed into $\mathbf{M_k^{nom}}$ through the procedures of affine transformation and interpolation.

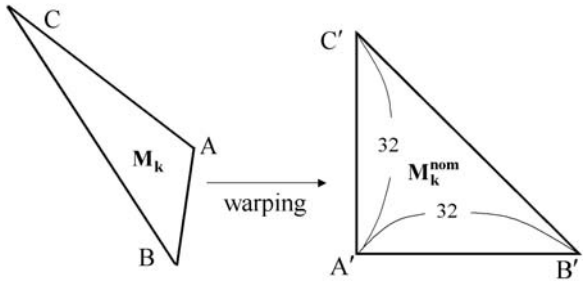

Figure 5: The angle ordering between an original mesh and its normalized mesh. The angle ordering is determined by sorting the three angles in a descending order.

There are two major factors that will affect the angle order. One is raise by severe geometric distortions that change the order of three angles in a mesh. However, this factor will lead to apparent destruction of visual quality in images, which lose their commercial value. Therefore, we can ignore this problem. The other one is resulted from the situation that two or three angles are nearly the same in magnitude such that even a slight distortion can change their order. This problem is required to be dealt with since the visual quality of an image is only imperceptibly modified. Our solution is to generate two (if two angles are nearly the same) or six (if three angles are nearly the same) different hashes for such mesh by changing angle order subsequently with respect to a query image. In addition, we still keep one mesh to have one corresponding hash in the hash database to save space.

In our implementation, a normalized mesh is a right-angled isosceles triangle with the length of both sides 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 would be lost to not provide enough discriminable features among different meshes; (ii) if the side's length is large enough, certain information of an original mesh would be redundant to pose the time-consuming problem in hash generation and comparison. As a whole, the number, 32, is empirically selected to meet the trade-off between the hash's discrimination and the normalization's complexity.

3.3 Robust Mesh-based Hashing

Image hashing attempts to transfer an image content to a feature sequence to represent its condensed essence. This feature sequence is required to be short enough for fast matching and meanwhile to preserve distinguishable features for feasible similarity measurement. By taking the above two conflicting factors into consideration, in this paper, the robust hash of each normalized mesh $\mathbf{M_k^{nom}}$ is extracted in the block-DCT domain. In fact, the extracted hash bits are position-dependent and belong to a kind of local features.

First, each triangle $\mathbf{M_k^{nom}}$ is flipped and padded with its flipped version to form a 32×32 block. For each 32×32 block, it is divided into 64 4×4 blocks. Second, 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 of length 64. It should be noted that due to the effect of padding the upper triangular and the lower triangular exactly capture different features. The DC coefficients will not be selected because they are not helpful in 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:

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

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 $\mathbf{M_k^{nom}}$.

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

In Eq. (1), there is one hash bit generated from a 4×4 block. Besides, the mesh-based hash designed according to Eq. (1) is to guarantee the number of 1's and 0's the same, i.e., uniform distribution, to avoid any bias that will affect hash matching. This uniform distribution of hash bits is extremely indispensable to the requirement of collision-free or false positive. We call this feature value $H_k(\cdot)$ robust because this magnitude relationship obtained after sorting can be approximately preserved.

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 to be 64, as explained previously. After mesh generation and mesh-based hash extraction, the

feature vector of an image $\mathbf{I}$ could be expressed as

$$\{\mathbf{H_1^I}, \mathbf{H_2^I}, ..., \mathbf{H}^{\mathbf{I}}_{|\mathcal{M}^{\mathbf{I}}|}\}, \quad (2)$$

where $|\mathcal{M}^{\mathbf{I}}|$ and $\mathbf{H_k^I}$ denote the number of meshes and 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 from which the mesh-based matching process is performed when an incoming query is given. Our hash database will be designed to be suitable for two-stage fast search where the first stage is to produce potential candidates through a coarse matching process and the second stage is a full matching process used to identify the final winner (if any) from 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 high robustness of an image hashing scheme against various attacks (including geometric distortions), partial matching is considered in measuring similarity. In the applications of content copy detection and tracing, two images ($\mathbf{I_m}$ and $\mathbf{I_n}$) are determined to be similar if there are at least N mesh-pairs 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.,

$$BER(\mathbf{H_i^{I_m}}, \mathbf{H_j^{I_n}}) = \frac{\sharp\{t | H_i^{I_m}(t) \neq H_j^{I_n}(t)\}}{|\mathbf{H_i^{I_m}}|} \leq T, \quad (3)$$

where $H_i^{\mathbf{I_m}}(t)$ denotes the t-th element of the i-th hash in $\mathbf{I_m}$ and $\sharp\{\}$ denotes the number of bit errors. The two thresholds, T and N, are determined meaningfully to be related to the false positive probability [5]. In addition, the 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. Under this circumstance, there will be $|\mathcal{M}^{\mathbf{I_m}}| \times |\mathcal{M}^{\mathbf{I_n}}|$ mesh pairs required 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 for the exhaustive hash matching becomes tremendous such that this kind of search is not suitable for many applications that need real-time processing. We call this kind of matching process "full matching." Therefore, full matching will be constrained to be performed on those candidates retrieved through a rapid coarse matching process (to be described in Sec. 3.4.4). A coarse-to-fine image hash database with 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 the selected candidates. In fact, this technique looks like a coarse-to-fine searching paradigm. The so-called coarse matching is mainly used to coarsely find a set of candidates (usually, its size is significantly smaller than the entire search space) that may contain the desired target. Next, a full matching is conducted on the set of candidates to find the final result exhaustively. Therefore, our fast matching paradigm needs to be cooperative with a hash database that is designed in a sophisticated manner. This sophisticated hash database is built, as shown in Fig. 6. The hash database comprises "entries" and each entry links to a chain that contains the indices of images. It is said that an image could 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 method of building the hash database for fast searching is a kind of clustering. However, unlike other clustering methods that are proposed for content-based image retrieval, our clustering paradigm belongs to partial clustering instead of global clustering. That is to say, an image can be linked into different entries as long as its hashes are similar to more than one entries.

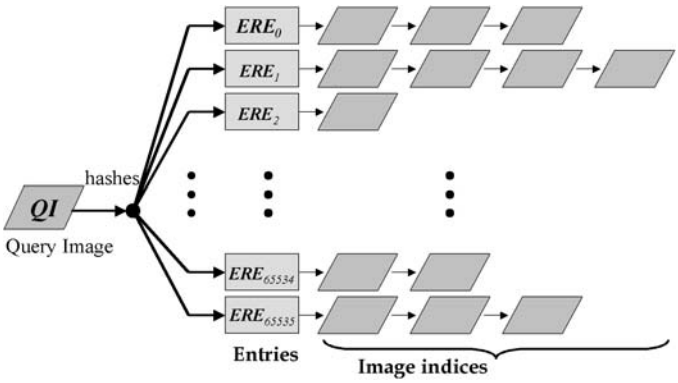

Figure 6: Creation of an image hash database for fast search in a coarse-to-fine manner. Our image hash database includes (i) error-resilient entries; (ii) image indices; and (iii) image hashes.

There are two issues that should be considered in constructing the desired hash database, i.e., entries should (i) be short enough for realizing practical implementation and (ii) possess error-robust capability to accommodate modifications of meshes due to attacks. In order to take the above two issues into account simultaneously, in this paper, each entry is designed as a 16-bit long hash sequence and a coarse representation of a mesh. This "coarse hash" that is similarly generated from the one described in Sec. 3.3 is described as follows. Now, each normalized mesh of size 32×32 is downsampled into an 8×8 coarse block from which a coarse hash, B_{ds}, is extracted to indicate the approximate characteristic of a mesh. Meanwhile, each mormalized mesh is also partitioned into 16 8×8 blocks from which block-based hashes, denoted as $B_p(b)$ ($0 \leq b \leq 15$), are extracted. With

the above setting, the coarse hash bits are designed as the results obtained from 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 **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 there are in total 65536 entries. The entries are expressed as ERE_0, ERE_1, ,... ERE_{65535}, where ERE_i is a binary representation of i. In our implementation, the size of ERE_i's needs to be controllable so that they can be stored into an array for rapid indexing. This corresponds to the first issue. As for the issue of error resilience, it means that even an image has been modified, its coarse hashes could be largely unaffected. Since the proposed coarse block carries low-frequency characteristic and the coarse hash bits are designed as the magnitude relationship between two blocks, both are stable and not easy to be changed. Readers can refer to [10] for the robustness analyses. Consequently, coarse matching is able to reliably select candidates that contain the desired target.

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

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

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

3.4.4 Coarse Matching

For an incoming query image, **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 **QI**, $\mathbf{CH_j^{QI}}$, is permitted to enter the entry E_i if $\mathbf{CH_j^{QI}}$ and E_i satisfy Eq. (5). Since the rationale behind our coarse matching method is to rapidly select the candidates for advanced full matching, we first design to exploit entries of the hash database to filter out those targets in the image database that are identified to be dissimilar to the incoming query. This goal is to reduce the number of images that are required for full matching and thereby the time-cost is saved.

In our coarse matching process, if a coarse hash of an incoming query **QI** is permitted to enter into an entry E_i, then the hit indicators of all the image indices that are linked to E_i will be added by 1 to indicate the gradual increase of the possibility that the images are potentially similar to the query. Let us denote by $\delta(id)$ as the hit indicator of an image $\mathbf{I_{id}}$. When all coarse hashes of **QI** have gone through the above process, we retain those images (in the database) that have the magnitude of their 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 million number of images in a database, this choice of candidates has greatly reduced the time required for searching. This also implies that most of the target images have been early obviated through coarse matching.

3.4.5 Valid or Invalid Retrieval

In the proposed two-stage matching paradigm, a so-called "valid retrieval" is defined as follows. Given a query image (**QI**), a hash database, and an image database, it is said that a best target image is effectively retrieved to match **QI** if (i) there are candidates retrieved to satisfy Eq. (5) during the coarse matching process (Sec. 3.4.4); (ii) the target image is the candidate, together with **QI**, that have the maximum number, N^{max}, of mesh pairs satisfying Eq. (3) (Sec. 3.4.2) and $N^{max} \geq N$. Furthermore, the importance of valid retrievals is determined according to their N^{max}'s. For example, top 1 valid retrieval means the one that have the maximum N^{max} among all valid retrievals.

On the other hand, if N^{max} is smaller than N, this search is treated to be invalid. As a result, it is concluded that the query image does not exist in the image database.

4. EXPERIMENTAL RESULTS

In this paper, several experiments were conducted to evaluate the proposed mesh-based image hashing and query system. In Secs. 4.1 and 4.2, the performance of copy detection is demonstrated.

4.1 Robustness: Resistance to Miscellaneous Attacks

First, ten color images with different contents ($\mathbf{I_1}$: Pepper, $\mathbf{I_2}$: Lenna, $\mathbf{I_3}$: Bridge, $\mathbf{I_4}$: Sailboat, $\mathbf{I_5}$: Goldhill, $\mathbf{I_6}$: F16, $\mathbf{I_7}$: Baboon, $\mathbf{I_8}$: Clock, $\mathbf{I_9}$: Tank, and $\mathbf{I_{10}}$: Splash) were used to verify the robustness of our scheme against miscellaneous attacks. The standard benchmark, Stirmark, with versions 3.1 and 4.0 quite fits our goal in simulating various manipulations of digital images. Please refer to [17, 18] 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 of robustness verification are summarized in Tables 1 and 2, 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. Besides, each field indicates the number of modified images that have been successfully identified. Here, $N = 3$ and $T = 0.25$, as explained in Sec. 3.4, were adopted. According to Tables 1 and 2, among 1910 modified images 1761 of them could be correctly identified, which indicates that the correct recognition rate is 92.2%.

Moreover, it can be observed that most modified images could be successfully detected except for some exceptions. Several attacked images that were failed to be identified are shown in Fig. 7 for visual inspections. We can observe from Fig. 7 that it is still not easy to correctly extract the meshes from the attacked images involving remarkably degraded fidelities and content eliminations. In particular, severe cropping and heavy noise adding are efficient in breaking the connectivity of meshes and thereby affect the hashes to defeat our system even the attacked images have lost their commercial value. However, compared with the existing meth-

ods [2, 3, 4, 7, 8, 15, 19, 20], it is evident that our scheme indeed achieves promising resistance to extensive geometric distortions.

Figure 7: Failed examples in the robustness test. (a) and (c) are from Stirmark 4.0, and (b) and (d) are from Stirmark 3.1.

4.2 Identification: Searching in a Large Database

The second part of our experiments was related to a retrieval problem in a large image database. In this searching system, the database is composed of the so-called original color images (which is composed of the Corel image database that contains 20000 images and ten traditional images such as Lenna, Baboon, ..., etc.) while the query image is either suspect in the sense that it may be a modified version generated from our database or totally irrelevant to the database. We have used the attacked images, obtained from Stirmark 3.1 and 4.0, as queries of the database.

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

$$Recall(T, N, n) = \frac{\text{No. of queries satisfying } Eq.(3)}{\text{No. of total queries}}, \quad (6)$$

$$Precision(T, N, n) = \frac{\text{No. of queries satisfying } Eq.(3)}{\text{No. of detections satisfying } Eq.(3)}, \quad (7)$$

where n means the number of valide retrievals that may include the desired target. Both the full matching and fast matching procedures were exploited for searching and their results were compared.

According to the process of full matching (described in Sec. 3.4.2), the co-called "successful search" needs to be defined to evaluate the performance of searching. Here, successful research means that at least one of the valid retrievals (described in Sec. 3.4.5) contains the desired target. In this paper, top n valid retrievals is adopted to check whether at least one of them contains the desired target, where different n's include 1, 2, 5, 10, and 100, respectively. In addition, we provide the information about the overall performance in terms of both recall and precision rates, as depicted in Table 3 [2]. As can be seen in Table 3, the overall searching results can only be significantly improved (up to $\approx 90\%$) if the number of valid retrievals is constrained to be within 5. This outcome demonstrates that the proposed searching strategy is very efficient in finding the desired target without relying on checking too many valid retrievals. Note that the number of miss detections is slightly larger than that obtained in the robustness test (as described in Sec. 4.1) because the search space has been broadened. Basically, these results reveal that the desired originals are hard to be identified for those query images (e.g., Fig. 7) that have been severely modified. Moreover, the miss detected queries are mostly consistent with those of failed identification in the robustness test.

In order to speed up search in 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 offers early elimination of those images that are dissimilar to the query. The overall performance of fast searching in terms of both recall and precision rates is depicted in Table 4. By comparing Table 4 and Table 3, we find that the proposed fast searching strategy is efficient while retaining comparable performance as full searching.

5. CONCLUSIONS

A robust mesh-based image hashing scheme has been proposed in this paper for content management of digital images. 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, our major contribution is to significantly improve the resistance of image hashing to geometric distortions. Furthermore, we have investigated several media hashing issues including robustness and discrimination, error analyses, complexity, granularity, and scalability. We have also demonstrated the use of the robust mesh-based image hashing system for both copy detection and content authentication.

On the other hand, our scheme is somewhat complex in that most time is consumed for mesh normalization. Fortunately, the hash database used for query and searching could be built in an off-line manner. As a result, the time is mainly spent for the incoming query image. However, it should be noted that this cost is compensated for the promising robustness against geometric distortions. Besides, a fast matching process has been proposed to speeded up search in a large image database. To understand the impact of different parameters on the false alarm rate, error analyses are conducted to derive a guideline of determining the necessary parameters.

Still, some directions that are worth of further researching are identified as follows. First, robust identification of small images is still a challenging problem because it is not robust enough to extract mesh-based hashes from small regions. Fortunately, precious images are usually with large sizes and only attacked images can be of small sizes (may lose their commercial value). Second, we will be devoted to study the challenging problem of robust feature point extraction for mesh generation. This problem is particularly crucial for mesh-based image authentication. Finally, we will extend the scope of our method to search and identify images in

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

the URLs.

Acknowledgment: This paper was supported, in part, by the National Science Council under NSC grants 91-2213-E-001-037 and 92-2422-H-001-004.

6. 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. Fridrich, "Visual Hash for Oblivious Watermarking," *Proc. SPIE: Security and Watermarking of Multimedia Contents II*, 2000.

[5] 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.

[6] *IEEE Int. Conf. on Multimedia and Expo*: special session on Media Identification, J. Oostveen, C. S. Lu, and Q. Sun (co-organizers), June 2004.

[7] C. Kim, "Content-based Image Copy Detection," *Signal Processing: Image Communication*, Vol. 18, pp. 169-184, 2003.

[8] F. Lefebvre, J. Czyz, and B. Macq, "A Robust Soft Hash Algorithm for Digital Image Signature," *Proc. IEEE Int. Conf. on Image Proc.*, 2003.

[9] 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.

[10] 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.

[11] C. S. Lu, "On the Security of Structural Information Extraction/Embedding for Images," *Proc. IEEE Int. Symposium on Circuits and Systems*, Vancouver, Canada, 2004.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] *IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, special session on Media Recognition, Virgin Islands, USA, 2002.

[17] 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.

[18] F. Petitcolas, "Watermarking Schemes Evaluation," *IEEE Signal Processing Magazine*, Vol. 17, No. 5, pp. 58-64, 2000.

[19] J. S. Seo, J. Haitsma, 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.

[20] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," *Proc. IEEE Int. Conf. Image Processing*, 2000.

Table 1: Robustness of our scheme vs. Stirmark 3.1: attacks are denoted as SPA: Signal Processing Attack including median filtering, Gaussian filtering, sharpening, and Frequency Mode Laplacian Removal (FMLR); JPEG: compressed 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 2: Robustness of our scheme vs. Stirmark 4.0: attacks are denoted as AffineT: Affine Transformation; ConvF: Convolution Filtering; Cropping: cropped into $\frac{3}{4}$, $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{5}$ sizes; JPEG: compressions with quality factors ranging from 90% to 10%; MF: Median Filtering; Noise: noise adding; SS: Self-Similarities; Scaling: scaled with factors ranging from 0.5 to 2.0; RML: Removing Lines; PSNR: all pixel values added with 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

Table 3: Recall rate vs. Precision rate for full searching on Stirmark

Searching style	**Full Searching**									
Query sources	*Stirmark 3.1 (890 queries)*					*Stirmark 4.0 (1020 queries)*				
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

Table 4: Recall rate vs. Precision rate for fast searching on Stirmark

Searching style	**Fast Searching**									
Query sources	*Stirmark 3.1 (890 queries)*					*Stirmark 4.0 (1020 queries)*				
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