A Visual Model-Based Perceptual Image Hash for Content Authentication

Xiaofeng Wang, Kemu Pang, Xiaorui Zhou, Yang Zhou, Lu Li, and Jianru Xue, *Member, IEEE*

***Abstract*— Perceptual image hash has been widely investigated in an attempt to solve the problems of image content authentication and content-based image retrieval. In this paper, we combine statistical analysis methods and visual perception theory to develop a real perceptual image hash method for content authentication. To achieve real perceptual robustness and perceptual sensitivity, the proposed method uses Watson's visual model to extract visually sensitive features that play an important role in the process of humans perceiving image content. We then generate robust perceptual hash code by combining image-block-based features and key-point-based features. The proposed method achieves a tradeoff between perceptual robustness to tolerate content-preserving manipulations and a wide range of geometric distortions and perceptual sensitivity to detect malicious tampering. Furthermore, it has the functionality to detect compromised image regions. Compared with state-of-the-art schemes, the proposed method obtains a better comprehensive performance in content-based image tampering detection and localization.**

***Index Terms*— Perceptual image hash, content authentication, Watson's visual model, tampering detection, tampering localization.**

I. Introduction

PERCEPTUAL image hash, also known as perceptual image signature, has been proposed as a primitive method to solve problems of image content authentication. A perceptual image hash is a short summary of an image's perceptual content. It has many important applications, for example, image content authentication, tampering detection, image retrieval, digital watermarking, and image registration. Recently, perceptual image hash has been developed as a frontier research topic in the field of digital media content security and multimedia applications.

The generation of a perceptual image hash is based on well designed image features that are in accordance with the perceptual characteristic of the human visual system. Such features are extracted from the image perceptual content. Image authentication is performed via comparing the hash value of an original image with the hash value of a doubted image. A perceptual image hash is expected to be able to survive unintentional distortion and reject malicious tampering within an acceptable extend.

The process of human cognizing multimedia data is a complex psychological activity. According to cognitive science theory, this process can be divided into three stages: sensation inputting, perceptual content abstracting and extracting, and cognitive identification [1]. The generation of a perceptual image hash is also following this process. Therefore, perceptual image hash is based on the subjective means by which human cognize image content, rather than a simply objective description. Moreover, unlike cryptographic hash functions, which are highly sensitive to bit changes, perceptual image hash is scalable and tolerates the fuzziness associated with how computers understand image content, although they are similar in form. Perceptual image hash, which maps the perceptual content of an image to a short binary string, can be considered a digital digest of the image content. It is essentially a mapping that meets some constraints.

Given an image I, a perceptually content-similar copy I', a perceptually content-changed version I_T (coming from I), and a perceptually content-different image I_d, a perceptual image hash algorithm $H_K(\cdot)$ depends on a secret key K and the perceptual content of I. According to the literature [2], the desirable properties of $H_K(\cdot)$ can be summarized as follows.

(1) *One-Way:* The hash value is easily calculated from the image and secret key, and this process should be noninvertible. That is, it is computationally infeasible to regain image content from the corresponding perceptual hash value, even when the secret key is usable. It can be formalized as $\Pr\{H_K(I) \rightarrow I\} < \varepsilon$.

(2) *Perceptual Sensibility/Visual Fragility:* The changing of image perceptual content should cause a distinct difference in hash value. That is, $\Pr\{H_K(I) \neq H_K(I_T)\} > 1 - \sigma$.

(3) *Collision Resistance/Discrimination:* Different images should have different perceptual hash values. That is, $\Pr\{H_K(I) = H_K(I_d)\} < \delta$.

(4) *Perceptual Robustness:* The algorithm produces the same or very similar hash values for perceptual content-similar images. That is, $\Pr\{H_K(I) = H_K(I')\} > 1 - \tau$.

Manuscript received April 25, 2014; revised September 18, 2014, January 19, 2015, January 25, 2015, and February 20, 2015; accepted February 23, 2015. Date of publication February 26, 2015; date of current version May 15, 2015. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2012CB316400 and in part by the National Natural Science Foundation of China under Grant 61075007 and Grant 61273252. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Barni.

X. Wang is with the Shaanxi Key Laboratory for Network Computing and Security Technology, and School of Science, Xi'an University of Technology, Xi'an 710048, China (e-mail: xfwang66@sina.com.cn).

K. Pang, X. Zhou, and Y. Zhou are with the School of Science, Xi'an University of Technology, Xi'an 710048, China (e-mail: kemupang@xaut.edu.cn; xiaoruizhou@xaut.edu.cn; yangzhou@xaut.edu.cn).

L. Li was with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China. He is now with the State University of New York at Buffalo, Buffalo, NY 14228 USA (e-mail: lilu.acrobat@gmail.com).

J. Xue is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jrxue@mail.xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2407698

(5) *Randomness:* Ideally, the hash code should be random. That is, $\Pr\{Entropy(H_K(I)) = Size(H_K(I))\} > 1 - \gamma$.

(6) *Compactness:* The size of the hash value should be much less than the size of the image. That is, $Size(H_K(I)) \ll Size(I)$.

(7) *Security:* Calculating the perceptual hash value without the secret key is intractable. That is, $\Pr\{H_K(I) = H_{K'}(I)\} < \theta$.

Here, $\Pr\{\cdot\}$ denotes the probability, $0 < \varepsilon, \sigma, \delta, \tau, \gamma, \theta < 1$, and $\varepsilon, \sigma, \delta, \tau, \gamma, \theta$ should be close to zero.

A. Connections With Other Technologies

The technologies related to the perceptual image hash are robust image hash and image digital fingerprinting.

(1) The difference between robust image hash and perceptual image hash. Robust image hash [3], [4] is a type of technology very close to the perceptual image hash. Both require that the core mapping be robust. However, the robust image hash tends to select invariant features to establish the core mapping, whereas the perceptual image hash is generated by using the perceptual features that are in accordance with human's visual characteristics. Therefore, the latter provide a more efficient approach to analyzing changes of image perceptual content.

(2) The difference between image digital fingerprinting and perceptual image hash. Image digital fingerprinting includes two categories. The first is the robust watermark that is primarily applied to copyright protection [5], and the second is the image hashing technology that is primarily applied to image content retrieval and object recognition [6], [7]. The perceptual image hash is similar to the latter, but not identical. It requires a stronger emphasis on both perceptual robustness to tolerate content-preserving manipulations and perceptual sensitivity to detect malicious tampering attacks.

B. Prior Works

Although many existing works refer to "perception hash", they are not really related to human's visual perceptual characteristics. Based on a general survey of the research on image hash technologies [3]–[9], the earliest hash methods answer whether the image content is authentic, but are unable to detect changed image regions. Recent works emphasize robustness to tolerate content-preserving manipulations and geometric distortion such as rotation/scaling and to detect changed image regions. Thanks to the theory and technology of image processing and pattern recognition, many robust image hash methods were developed in the past dozen years. Based on differences in feature extraction, existing methods in the literatures can be classified into four main categories [8]–[10].

Statistics-based methods [3], [4], [11]: these methods extract image features by calculating statistics such as mean, variance, moments of image blocks, histogram.

Relationship-based methods [12]–[16]: these methods use the invariant relationships of the image transform coefficients, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT), to generate image features and hash codes.

Coarse representation or sketch-based approaches [17]–[19]: the hash codes are calculated by using the global coarse features of an image, such as the spatial distribution of the wavelet coefficients or the low-frequency coefficients of the Fourier Transform.

Lower-level features based methods [20]–[23]: the hash codes are extracted by detecting salient image feature points. These hash values are very sensitive to local distortions that do not cause perceptually significant changes.

In addition, there are some other methods such as a clustering-based method [24] and a mesh-based method [25].

With more image hash methods being presented, the emphasis of recent works is primarily focused on developing the functionality to detect compromised image regions. The main methods in the literatures can be divided into two categories: image-block-based methods and feature-point-based methods. The former uses the statistical features of image blocks to generate short binary representations [14], [15], [26]–[28]. These methods can detect image content changes and changed image regions. However, their inherent defect is that the detection accuracy is not reliable [26]. Furthermore, many of them are incapable of handling geometric distortions such as rotation and scaling. The latter uses image feature points to generate hash code [22]–[24], [29]. Because feature points are sparse in an image, these methods still have the difficulty of providing ideal detection accuracy. Moreover, some of these methods cannot process some images that contain textureless regions. In addition to the above methods, Lu [30], [31] proposed a concept of forensic hash, which can detect both content authenticity and processing history.

The essence of image content authenticity should be able to identify the authenticity of the perceptual content of what the image expresses. Therefore, how to extract image features that are able to express image perceptual content has become a key issue of image content authentication. Recently, image hash related to "perception" has received increasing attention. The pioneer work was reported by Bhattacharjee et al. [20]. In their work, "salient" image feature points were extracted via a scale interaction model and Mexican-hat wavelets, and the positions of these points were used to generate hash code. The advantage of this method is the compact hash length. However, the method [32] might not be adequate for detecting crop-and-replacement manipulations inside image objects because the relevance of selected points is not clear.

Monga et al. [22] reported that psychovisual studies had identified the presence of certain cells, called hypercomplex or end-stopped cells, in the human's visual cortex. These cells respond strongly to extremely robust image features such as corners and points of high curvature in general. The work in [21] constructed an "end-stopped" wavelet to capture this behavior, in which the feature points were invariant under perceptually insignificant distortions [23]. They applied probabilistic quantization to the position coordinates of derived feature points to generate a perceptual hash code. This method can detect content changing caused by malicious attacks. However, because the hash code was generated only by the "end-stopped" points of the image objects, which have linear structures with a specific orientation, it has little

or no robustness to geometric distortions. Subsequently, Monga et al. introduced Nonnegative Matrix Factorization (NMF) into their new hash algorithm [24]. The major benefit of NMF hashing is the structure of the basis resulting from its nonnegative constraints, which leads to a parts-based representation. This method is robust under a large class of perceptually insignificant attacks, and it significantly reduces misclassification of perceptually distinct images.

Zhao [33] reported a perceptual image hash method based on texture and shape features. In [33], an image was divided into non-overlapped blocks and each image block was mapped into a circle. Then, the Zernike moments and texture features of each circle were connected to form a hash code. In their other work [34], the global and local features of an image were used to generate hash code. The global features were the Zernike moments of the luminance and chrominance parameters, and the local features were composed of position and texture information of salient regions in the image.

Lv et al. [2] proposed a shape-contexts-based perceptual image hash approach using robust local feature points. In their scheme, the SIFT-Harris detector was used to extract keypoints. These keypoints were used to generate local features, and the image hash was then generated by embedding local features into shape-contexts-based descriptors.

Khelifi and Jiang [35] presented a perceptual image hash method from virtual watermark detection. The idea came from the fact that a non-embedded watermark detector would yield similar responses to perceptually close images. In [35], a linear high-pass filtered image was divided into overlapping blocks, and the means of the absolute coefficients in the image blocks were computed to form a set of features. Then, the Weibull model was used to extract the statistical values of feature coefficients to generate a perceptual image hash code.

Hou et al. [36] formulated the figure-ground separation problem of an image in the framework of sparse signal analysis. They first used DCT coefficients to define an image signature, and then used Gaussian smoothing to make this signature approximate the spatial location of a sparse foreground hidden in a spectrally sparse background. The experimental data show that the approximate foreground location highlighted by the image signature is remarkably consistent with the locations of human eye movement fixations.

Recently, compressive sensing (CS) was used to construct image hash schemes. Tagliasacchi et al. [37] reported an image hash scheme based on CS and distributed source coding (DSC), in which, the hash was derived from the DSC-encoded quantized random projection coefficients of an image. To perform authentication, the hash code was served as side information, the DSC decoder decoded the received hash bits, and the authenticity determination depended on the success/failure of DSC decoding. This method produces a very long hash code. Kang et al. [38] exploited the property of dimensionality reduction inherent in CS for image hash designing. Their CS-based hash is computationally secure and with a small size. Sun and Zeng [39] proposed a method in which the Fourier-Mellin transform was used to improve the performance of the algorithm under rotation, scale, and transition attacks, and CS was used to reduce the vector dimension to generate a hash value.

Additionally, most existing image hash methods only focus on feature extraction and ignore security analysis. In a perceptual image hash system for content authentication, security is also an important performance. Hadmi et al. [40] proposed a perceptual image hash scheme that focuses particularly on a quantization step analysis. Its purpose is to enhance perceptual robustness and to minimize collision probabilities to improve security.

Although great progress has been made in perceptual image hash technology, there still are very few methods that focus on visual perceptual feature. To tally with human's visual system, Qin et al. [41] presented an HVS (Human Visual System)-based image hash method. In [41], hash codes were generated using DCT coefficients that were weighted via Watson's visual sensitivity matrix. This method has poor robustness because of a lack of comprehensive image features.

C. Our Contributions

In this paper, a real perceptual image hash method for content authentication is proposed. Using this hash, an image tampering detection and tampering localization method is developed. Our contributions are as follows. (1) Watson's visual model is used to extract the visually sensitive features that play an important role in the process of humans perceiving image content. (2) Image-block-based features and key-point-based features are combined to generate intermediate hash code. Gaussian random matrices are used to reduce the vector dimension, and encryption and randomization are used to generate the final hash code. (3) The proposed method is robust to a wide range of geometric distortions and content-preserving manipulations such as JPEG compression, adding noise, and filtering. Furthermore, it is sensitive to content changes caused by malicious attacks. (4) The proposed method has the functionality of tampering localization. Additionally, it achieves a trade-off between robustness to tolerate geometric distortion and tampering localization. Compared with state-of-the-art schemes, the proposed scheme yields better performance.

The rest of this paper is organized as follows. In section 2, we introduce the preliminaries used in this paper. The proposed algorithms are described in section 3, and supported by experimental results in section 4. Section 5 concludes the paper with some thoughts on future works.

II. Preliminaries

A. Framework of Perceptual Image Hash

In general, a perceptual image hash system consists of four stages: image preprocessing and transformation, perceptual feature extraction and description, compression and coding, and encryption and randomization. The general framework is shown in Fig. 1.

The purpose of image preprocessing and transformation is to eliminate irrelevant information, recover useful information and enhance image features that are important in subsequent processing. To ensure perceptual robustness and perceptual

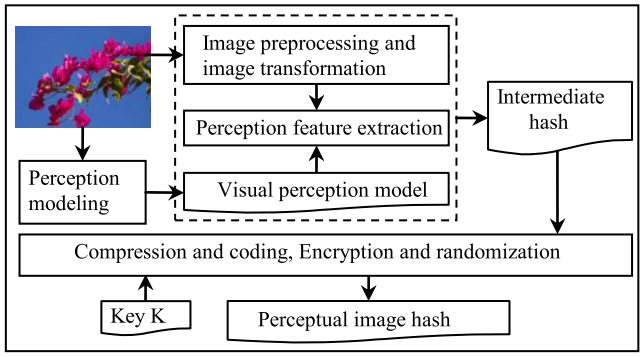

Fig. 1. The framework of perceptual image hash.

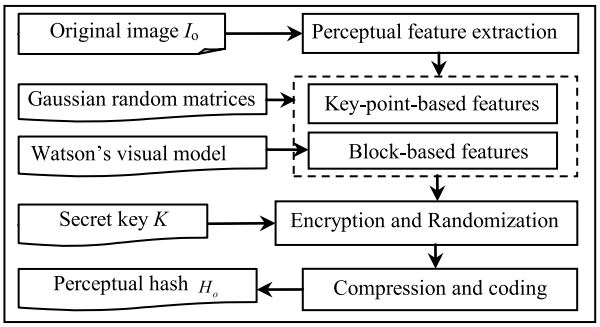

Fig. 2. Flowchart of the proposed perceptual image hash.

sensibility, the selection and extraction of perceptual features are very important. Perceptual feature extraction is based on the human visual perception model that is established by the cognitive science theory. It is accomplished via signal processing methods that remove redundant data but retain perceptually significant features. Moreover, to reduce hash length and improve convenience for storage and hardware implementation, post-processing such as compression and coding is necessary. Encryption and randomization are used to reduce hash collisions to improve the security of the algorithm.

B. Watson's Visual Model

Watson's DCT-based visual model [42] attempts to account for frequency sensitivity, luminance masking, and contrast masking in the DCT domain. It estimates the perceptibility of changes in individual terms of image's block DCT and then pools those estimates into a single estimate of perceptual distance.

1) Frequency Sensitivity: Watson's visual model defines the frequency sensitivity in a matrix t, in which, each entry $t[i, j]$ $(i, j = 1,\ldots,8)$ is the smallest visible value of the corresponding DCT coefficients of an image block without any masking noise (i.e., the amount of change in the coefficient that produces one Just Noticeable Difference (JND)). Smaller values of $t[i, j]$ indicate the higher sensitivity of the human's eye to the corresponding frequency [42], [43]. Lower frequencies (in the top left corner of matrix t) have smaller values. The high sensitivity of the human visual system to these frequencies results in detecting perceptible phenomena in the image despite the triviality of the introduced changes.

$$t = \begin{bmatrix} 1.40 & 1.01 & 1.16 & 2.40 & 3.43 & 4.79 & 4.79 & 6.56 \\ 1.01 & 1.45 & 1.32 & 1.52 & 2.00 & 2.71 & 3.67 & 4.93 \\ 1.16 & 1.32 & 2.24 & 2.59 & 2.98 & 3.64 & 4.80 & 5.88 \\ 1.66 & 1.52 & 2.56 & 3.77 & 4.56 & 5.30 & 6.28 & 7.60 \\ 2.40 & 2.00 & 2.98 & 4.56 & 6.15 & 7.46 & 8.71 & 10.17 \\ 3.43 & 2.71 & 3.64 & 5.30 & 7.46 & 9.62 & 11.56 & 13.51 \\ 4.79 & 3.67 & 4.60 & 6.28 & 8.71 & 11.58 & 14.50 & 17.29 \\ 6.56 & 4.93 & 5.88 & 7.60 & 10.17 & 13.51 & 17.29 & 21.15 \end{bmatrix}$$

2) Luminance Masking: The sensitivity of human visual system also depends on the average intensity of the image block. This means that in image blocks with higher luminance, larger changes in DCT coefficients maybe imperceptible. Watson's model modifies matrix $t[i, j]$ in each block k according to the block average luminance or the block's DC term. The luminance-masking threshold is defined as follows:

$$t_L[i, j, k] = t[i, j](C_0[0, 0, k]/C_{0,0})^{a_T}, \quad 0 \le i, j \le 7, \quad 0 \le k \le N-1$$

where a_T is a constant with a suggested value of 0.649 by [42] and [43], $C_0[0,0,k]$ is the DC coefficient of the k-th block in an image, $C_{0,0}$ is the average of the DC coefficients in the image, and $\boldsymbol{N}$ is the block numbers of the image.

3) Contrast Masking: The reduction of visibility of a change in a frequency due to its energy is called contrast masking. Watson's DCT-based visual model also defines the contrast masked threshold $s[i, j, k]$ for each DCT frequency in an image block k. It is defined as:

$$s[i, j, k] = max\{t_L[i, j, k], |C_0[i, j, k]|^{w(i,j)} t_L[i, j, k]^{1-w(i,j)}\}$$

where $w(i, j)$ is a constant between 0 and 1. (The value 0.7 has been used by Watson for all i and j [42], [43].)

This formula clearly decides that the contrast threshold value depends on not only the energy present in that frequency but also the luminance masked threshold for that frequency.

III. Proposed Method

The proposed perceptual image hash scheme includes a hash generation algorithm, a tampering detection algorithm, and a tampering localization algorithm. The hash generation algorithm consists of three stages: feature extraction, encryption and randomization, and compression and coding. The process is shown in Fig. 2.

A. Hash Generation Algorithm

Considering that the SIFT (Scale Invariant Feature Transform) [44] features are invariant to translation, rotation and scaling transformations in image domain and robust to moderate perspective transformations and illumination variations, we begin our work with the SIFT feature extraction.

1) Perceptual Feature Extraction:

a) Key-point-based features: For an $M \times N$ image $I_o(x, y)$, the SIFT feature point set is denoted as $S_o = \{S_{o1}(x_{o1}, y_{o1}), \ldots, S_{oi}(x_{oi}, y_{oi}), \ldots, S_{om}(x_{om}, y_{om})\}$. Here, (x_{oi}, y_{oi}) is the location coordinate of S_{oi}. Let T_{oi} represent the 128-dimension vector of S_{oi}, $i = 1, \ldots, m$. To obtain a sparse representation of the image, we take a 1-level db1 wavelet transformation to T_{oi}, where the coefficients are denoted as $DT_{oi} = (dt_{oi}^1, dt_{oi}^2, \ldots, dt_{oi}^{128})^T$ $(i = 1, \ldots, m)$.

b) Block-based features: The $M \times N$ image $I_o(x, y)$ is divided into non-overlapping $P \times P$ $(P = 8)$ blocks, where each block is denoted as B_{ok}. The DCT coefficients of B_{ok} $(k = 1, \ldots, (M \times N)/P^2)$ are denoted as:

$$CB_{ok} = \begin{bmatrix} b_{ok}(1,1) & b_{ok}(1,2) & \ldots & b_{ok}(1,8) \\ \ldots & \ldots & \ldots & \ldots \\ b_{ok}(8,1) & b_{ok}(8,2) & \ldots & b_{ok}(8,8) \end{bmatrix} \quad (1)$$

To extract the image features that can represent image perceptual content, we use Watson's DCT-based visual model to adjust the DCT coefficients. Considering that each matrix entry $t[i, j]$ $(i, j = 1,\ldots,8)$ of matrix t is the smallest visible value of the corresponding DCT coefficients of an image block, we use the inverse of each $t[i, j]$ to weight its corresponding DCT coefficient, i.e.

$$db_{ok}(i, j) = b_{ok}(i, j)/t[i, j] \quad (2)$$

Then, we obtain the weighted matrix:

$$MB_{ok} = \begin{bmatrix} db_{ok}(1,1) & db_{ok}(1,2) & \ldots & db_{ok}(1,8) \\ \ldots & \ldots & \ldots & \ldots \\ db_{ok}(8,1) & db_{ok}(8,2) & \ldots & db_{ok}(8,8) \end{bmatrix} \quad (3)$$

Finally, we zigzag MB_{ok} to obtain the following vector:

$$DB_{ok} = (db_{ok}^1, db_{ok}^2, \ldots, db_{ok}^{64})^T \quad (4)$$

c) Compression and projection: We use two Gaussian random matrices derived from the compressive sensing model to achieve compression and projection. That is, two Gaussian random matrices G_{s_1} and G_{s_2} are generated; we then use equations (5) and (6), respectively, to obtain compressed matrices GT_{oi} and GB_{ok}. Specifically

$$GT_{oi} = G_{s_1} \cdot DT_{oi}, \quad i = 1, \ldots, m \quad (5)$$

$$GB_{ok} = G_{s_2} \cdot DB_{ok}, \quad k = 1, \ldots, (M \times N)/P^2 \quad (6)$$

where projection rates s_1 and s_2 are selected via the experiments.

Let

$$\begin{aligned} F_o = (&GT_{o1}, GT_{o2}, \ldots, GT_{om}, GB_{o1}, GB_{o2}, \ldots, \\ &GB_{o(M\times N/P^2)}, (x_{o1}, y_{o1}), (x_{o2}, y_{o2}), \ldots, \\ &(x_{om}, y_{om}), M, N) \end{aligned} \quad (7)$$

F_o can be considered an intermediate hash of I_0. The length of F_o is $L_{F_0} = m \cdot s_1 + [(M \times N)/P^2] \cdot s_2 + 2m + 2$.

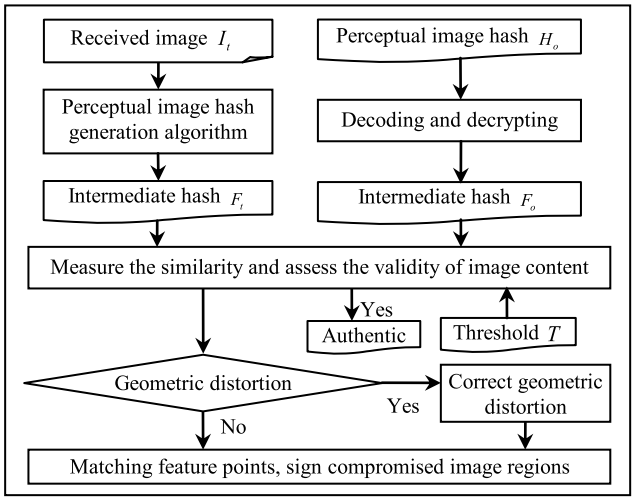

Fig. 3. The schematic diagram of the tampering detection and localization.

2) Encryption and Randomization:

① *Key generation:* We use chaotic encryption because it is sensitive to initial value change, and calculated rapidly. To do so, we generate a chaotic sequence via a logistic mapping. Let $K \in (0, 1)$ be a secret key shared by a sender and a receiver, and let $L(\cdot)$ represent the logistic mapping $x_{n+1} = x_n - x_n^2$.

Let $k_1 = K$, $k_2 = k_1 - k_1^2$, ..., $k_{n+1} = k_n - k_n^2$,...,

Let $k = (k_1, k_2, \ldots, k_L)$, where $L = m + (M \times N)/P^2 + 2m + 2$.

②*Encryption and randomization:* Let

$$\begin{aligned} &\tilde{F}_o \\ &= (\tilde{G}T_{o1}, \tilde{G}T_{o2}, \ldots, \tilde{G}T_{om}, \tilde{G}B_{o1}, \tilde{G}B_{o2}, \ldots, \tilde{G}B_{o(M\times N)/P^2}, \\ &\quad (\tilde{x}_{o1}, \tilde{y}_{o1}), (\tilde{x}_{o2}, \tilde{y}_{o2}), ..., (\tilde{x}_{om}, \tilde{y}_{om}), \tilde{M}, \tilde{N}) \\ &= (GT_{o1} \times k_1, GT_{o2} \times k_2, \ldots, GT_{om} \times k_m, \\ &\quad GB_{o1} \times k_{m+1}, GB_{o2} \times k_{m+2}, \ldots, GB_{o(M\times N)/P^2} \\ &\quad \times k_{m+(M\times N)/P^2}, (x_{o1} \times k_{m+(M\times N)/P^2+1}, \\ &\quad \times y_{o1} k_{m+(M\times N)/P^2+2}), \ldots, (x_{om} \times k_{L-3}, y_{om} \times k_{L-2}), \\ &\quad M \times k_{L-1}, N \times k_L) \end{aligned} \quad (8)$$

$\tilde{F}_o$ is the encrypted hash code of image I_0.

3) Compression and Coding: To obtain a compact hash code, $\tilde{F}_o$ can be compressed by using Huffman coding. The final hash code H_o is generated by adjoining the Huffman codes of leaves in Huffman tree HT_o which correspond to each element of $\tilde{F}_o$.

B. Tampering Detection and Tampering Localization

Regarding image content changing, it is difficult to define a clear boundary between perceptually insignificant distortion and malicious tampering because some content-preserving manipulations such as JPEG compression are lossy. This results in an intriguing question, that is, the trade-off between robustness to tolerate content-preserving manipulations and sensitivity to malicious tampering. In our work, tampering detection and tampering localization are realized by comparing a distance metric to measure the similarity between hash values. The algorithm process is as follows, and the schematic diagram is shown in Fig. 3.

Step 1: For received perceptual hash H_o and Huffman tree HT_o, Huffman decoding and decrypting are applied to obtain intermediate hash F_o of the original image I_o.

Step 2: For received image I_t, an interpolation operation $I'_t = I(I_t)$ is applied to obtain the same size with I_o. Then SIFT features are extracted and the feature points set is denoted as $S_t = \{S_{t1}(x_{t1}, y_{t1}), \ldots, S_{ti}(x_{ti}, y_{ti}), \ldots, S_{tm'}(x_{tm'}, y_{tm'})\}$, where (x_{ti}, y_{ti}) is the location coordinate of feature point S_{ti}. We denote T_{ti} as the 128-dimension feature vector of $S_{ti} (i = 1, \ldots, m')$.

After applying 1-level db1 wavelet transformation to T_{ti}, the resulting coefficients are denoted as $DT_{ti} = (dt_{ti}^1, dt_{ti}^2, \ldots, dt_{ti}^{128})^T$. Then, the Gaussian random matrix G_{s_1} is used to generate the vector:

$$GT_{ti} = G_{s_1} \cdot DT_{ti}, \quad i = 1, \ldots, m' \tag{9}$$

Step 3: To tolerate geometric distortions, it is necessary to match feature point sets GT_{oi} and GT_{ti}. By applying the SIFT matching algorithm [44], n pairs of most-similar feature points can be obtained, where $4 \le n \le \min(m, m')$. Then, we update the feature point sets as matched feature points:

$$S_o = \{S_{o1}(x_{o1}, y_{o1}), S_{o2}(x_{o2}, y_{o2}), \ldots, S_{oi}(x_{oi}, y_{oi}), \ldots, S_{on}(x_{on}, y_{on})\} \tag{10}$$

$$S_t = \{S_{t1}(x_{t1}, y_{t1}), S_{t2}(x_{t2}, y_{t2}), \ldots, S_{ti}(x_{ti}, y_{ti}), \ldots, S_{tn}(x_{tn}, y_{tn})\} \tag{11}$$

We estimate an affine transformation matrix Π by using S_o and S_t. Let

$$MS_o = \begin{bmatrix} y_{o1} & y_{o2} & \ldots & y_{on} \\ x_{o1} & x_{o2} & \ldots & x_{on} \\ 1 & 1 & \ldots & 1 \end{bmatrix},$$
$$MS_t = \begin{bmatrix} y_{t1} & y_{t2} & \ldots & y_{tn} \\ x_{t1} & x_{t2} & \ldots & x_{tn} \\ 1 & 1 & \ldots & 1 \end{bmatrix} \tag{12}$$

We then solve the equation system $\Pi \cdot MS_o = MS_t$ to obtain the affine transformation matrix Π. Applying affine transformation to I'_t, we obtained an $M \times N$ image I''_t.

Step 4: I''_t is divided into non-overlapping $P \times P$ $(P = 8)$ blocks B_{tk}, $k = 1, \ldots, (M \times N)/P^2$. Let $B_{tk}(x, y)$ represent the gray value of (x,y) in B_{tk}, where $1 \le x, y \le P$. The Watson's frequency sensitivity matrix t is used to weight the DCT coefficients of B_{tk}, and the weighted coefficients are denoted as vector: $DB_{tk} = (db_{tk}^1, db_{tk}^2, \ldots, db_{tk}^{64})^T$. After applying the Gaussian random matrix G_{s_2} to DB_{tk}, we obtain vector

$$GB_{tk} = G_{s_2} \cdot DB_{tk}, \quad k = 1, \ldots, (M \times N)/P^2 \tag{13}$$

The intermediate hash of I''_t is denoted as:

$$F_t = (GT_{t1}, GT_{t2}, \ldots, GT_{tn}, GB_{t1}, GB_{t2}, \ldots, GB_{t(M\times N)/P^2}, (x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \ldots, (x_{tn}, y_{tn}), M, N) \tag{14}$$

Using the same encryption and randomization, compression and coding described in section III.A, the final hash code H_t of the received image I_t can be generated.

Step 5: To measure the similarity between image blocks in I_o and image blocks in I_t, we estimate the Euclidean distance D_k between GB_{ok} and GB_{tk}:

$$D_k = \sqrt{(GB_{ok} - GB_{tk}) \cdot (GB_{ok} - GB_{tk})^T}, \quad k = 1, \ldots, (M \times N)/P^2 \tag{15}$$

Let $D = \max\ (D_1, D_2, \ldots, D_{(M\times N)/P^2})$.

Step 6: If $D \ge T$, then the tested image should be considered inauthentic; go to Step 7. Else, the tested image should be considered authentic, where T is a threshold.

Step 7: We estimate the Euclidean distance D_{a_i} between each pair of matched feature points in sets S_o and S_t.

$$D_{a_i} = \sqrt{(x_{oi} - x_{ti})^2 + (y_{oi} - y_{ti})^2}, \quad i = 1, \ldots, n \tag{16}$$

Let

$$D_a = \min\ (D_{a_1}, \ldots, D_{a_n}) \tag{17}$$

If $D_a > T_a$, the tested image I_t should have undergone geometric transformation. We investigate the values of $D_k (k = 1, \ldots, (M \times N)/P^2)$ and sign all of the blocks B_{tk} of $D_k > T_{p2}$. These blocks should be compromised regions.

If $D_a \le T_a$, tested image I_t should not be considered undergoing geometric transformation. We investigate the values of $D_k (k = 1, \ldots, (M \times N)/P^2)$, and sign all of the blocks B_{tk} of $D_k > T_{p1}$. These blocks should be compromised regions. Here, T_{p1} and T_{p2} are thresholds estimated via experiments, and $T_{p2} > T_{p1}$.

Here, the settings of T_{p1} and T_{p2} are based on the following analysis. Image tampering will cause the change of feature points, that is, if an image has been tampered with, then the feature points detected from the compromised regions will be different from the feature points defined in its original version. As a result, the distance between feature points defined in the original version and those detected will be greater. We can use the distance change to detect changed image regions. That is, an image block can be considered as a tampered region if it contains changed feature points, and the change of feature points can be measured via distances between the original feature points and detected feature points in the corresponding image region. Because distance values are clearly distinguishable, we can estimate the cut-off values of the distance change by using the statistical values that are obtained from experimental results; they are thresholds T_{p1} and T_{p2}.

IV. Experimental Results and Performance Analysis

In this section, we evaluate the proposed method. We implemented and tested our method using MATLAB2010a and famous Stirmark benchmark. The test is running on a computer with a Dual-Core CPU, i5-2400 @ 3.10GHz, 4.00GB RAM.

A. Parameter Setting

An ideal perceptual image hash scheme should have some desirable properties: perceptual robustness to content-preserving manipulations and geometric distortions, perceptual

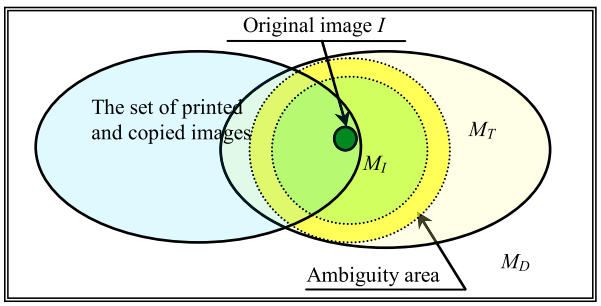

Fig. 4. The association relationship of image sets.

sensibility to malicious tampering, tampering detection and localization functionality, compact hash length, and security-resistance to forgery attacks. Generally, perceptual robustness and perceptual sensibility are mutually contradictory because of the lack of a clear boundary to distinguish content-preserving manipulations from malicious tampering. Moreover, compact hash codes include less image information, which will contribute to the algorithm's stronger perceptual robustness. However, the perceptual sensibility will be weaker. In contrast, hash values of longer length will include abundant image information and hence will contribute to the algorithm's ideal tampering localization functionality. Thus, perceptual sensibility will be stronger, and perceptual robustness will be weaker. The design goal of the perceptual image hash should emphasize the trade-off among above-mentioned factors that will determine the final performance of the algorithm.

To achieve satisfactory performance, we estimate the parameters used in our method via experiments. To this end, we first define notations. Let M denote the image space, and $I \in M$ denote an image in M. According to the relationship between $I$and other images in M, M can be divided into three subsets. 1) $M_D \subset M$, includes the different images with I. 2) $M_I \subset M$, consists of the images that come from I and have undergone content-preserving manipulations such as moderate RST, filtering, adding noise, and JPEG compression. 3) $M_T \subset M$, includes the images that come from image I and have undergone tampering attacks. The relationship of these image sets is shown in Fig. 4.

According to [9] and [10], typical image attack types include adding image objects, image objects being replaced, copy-move attack (moving image elements or changing their positions), and image characteristics (e.g., color, textures) being changed.

Let $\Delta(\cdot)$ denote a hash-based image-tampering detection algorithm. For image I, $\Delta(I)$ returns a decision "1" (True) or "0" (False). $\Delta(I)$ =1 indicates the image content is authentic, that is, $I \in \{I\} \cup M_I$. $\Delta(I) = 0$ indicates the image content is inauthentic, $I \in M_T \cup M_D$; that is, the image has been tampered with, or it is an irrelevant image. For $I \in \{I\} \cup M_I$, $\Delta(I) = 1$ indicates a correct detection, and $\Delta(I) = 0$ indicates an incorrect detection. For $I \in M_T \cup M_D$, $\Delta(I) = 1$ indicates an incorrect detection, and $\Delta(I) = 0$ indicates a correct detection.

To discuss the algorithm performance in detail, we define a term of "Detection passing rate". It refers to the probability of correct detection, and it is defined at the bottom of this page [see (18)], where |•| means the cardinality of a set.

The experiments are performed on the UCID image database and the Columbia university image database. In experiments, 3000 images with various sizes are tested. Examples of content-preserving manipulations include JPEG compression, adding noise, and filtering. The geometric transformations include rotation, scaling, cropping, and shear mapping. Equation (15) is used to measure the hash distances between an original image and its manipulated versions. Considering that the proposed method presents a satisfactory detection passing rate for all tested content-preserving manipulations when $T \geq 5.0$, we set $T = 5.0$; the choice of threshold T references the method of the robustness experiment (see Fig. 5).

1) Estimating the Number of Feature Points: In feature matching, considering that too many feature points will reduce computing efficiency, whereas too few feature points will reduce the detection accuracy, we estimate the number of feature points via experimental results. Let $s_1 = 9.4\%$, and $s_2 = 9.4\%$. First, we calculate the hash value of each tested original image. We then manipulate images via content-preserving manipulations with StirMark software and calculate the hash value of each manipulated image. Finally, we compare these hash values pair by pair, using equation (15) and estimate the Detection passing rate under various numbers of feature points. The experimental results are shown in Tables I and II. Here, i represents the number of feature points. Integrating the numbers of feature points and the detection passing rates, we select $36 \leq i \leq 40$ as the number of feature points.

2) Setting Projection Rate s_2: To choose an appropriate value for s_2, we tested the detection passing rate with different s_2 under both conditions $36 \leq i \leq 40$, and $s_1 = 9.4\%$. The experimental results are shown in Tables III and IV. As shown, when $s_2 = 3.0\%$, the detection passing rates are higher than others for each content-preserving manipulation.

3) Setting Projection Rate s_1: To choose an appropriate value for s_1, we tested the detection passing rate with different s_1 under both conditions $36 \leq i \leq 40$ and $s_2 = 3.0\%$.

$$\text{Detection passing rate} = \frac{|\{I'|(I' \in \bigcup_I \{\{I\} \cup M_I\}) \wedge (\Delta(I') = 1)\} \cup \{I''|(I'' \in M_T \cup M_D) \wedge (\Delta(I'') = 0)\}|}{|\{\bigcup_I \{\{I\} \cup M_I\}\} \cup M_T \cup M_D|} \times 100\%$$
$$= \frac{\text{The number of correct detection}}{\text{The total number of tested images}} \times 100\% \quad (18)$$

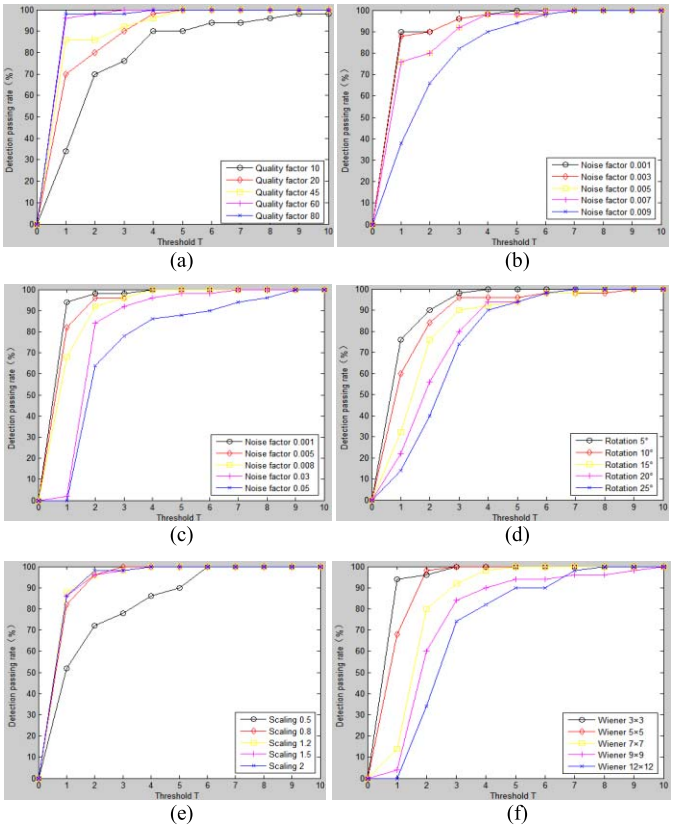

Fig. 5. Robustness testing for content-preserving manipulations. (a) Robustness for JPEG compression. (b) Robustness for Gaussian noise. (c) Robustness for salt and pepper noise. (d) Robustness testing for rotation. (e) Robustness testing for scaling. (f) Robustness testing for wiener filtering.

TABLE I
DETECTION PASSING RATE (%) UNDER VARIOUS NUMBERS OF FEATURE POINTS

i	Salt & pepper noise NF=0.03	Gaussian noise NF=0.005	Wiener filtering 9×9	JPEG compression QF=20	Scaling SF=0.5	Rotation Ang=25°
$5 \le i \le 10$	77.48	80.79	84.11	80.13	34.44	35.10
$11 \le i \le 15$	85.43	87.42	88.47	92.05	57.62	50.99
$16 \le i \le 20$	96.03	98.01	94.70	98.68	70.86	62.25
$21 \le i \le 25$	97.35	99.34	96.03	98.01	72.85	65.56
$26 \le i \le 30$	97.35	99.34	96.69	98.01	80.13	66.89
$31 \le i \le 35$	97.35	99.34	96.69	98.01	84.77	70.20
$36 \le i \le 40$	**98.68**	**99.34**	**98.01**	**100.00**	**88.08**	**71.52**
$41 \le i \le 45$	98.68	99.34	98.01	100.00	90.07	71.52
$46 \le i \le 50$	98.68	99.34	98.68	100.00	90.73	73.51

TABLE II
DETECTION PASSING RATE (%) UNDER VARIOUS NUMBERS OF FEATURE POINTS

i	Salt & pepper noise NF=0.008	Gaussian noise NF=0.001	Wiener filtering 3×3	JPEG compres-sion QF=80	Scaling SF=1.5	Rotation Ang=5°
$5 \le i \le 10$	97.35	96.69	93.38	98.68	80.13	58.28
$11 \le i \le 15$	98.01	98.68	99.34	99.34	90.07	80.70
$16 \le i \le 20$	98.68	98.68	99.34	99.34	92.72	92.72
$21 \le i \le 25$	98.68	99.34	99.34	99.34	96.03	94.70
$26 \le i \le 30$	98.68	99.34	99.34	99.34	96.03	94.70
$31 \le i \le 35$	98.68	99.34	99.34	99.34	97.30	94.70
$36 \le i \le 40$	**99.34**	**100.00**	**100.00**	**100.00**	**97.35**	**94.70**
$41 \le i \le 45$	99.34	100.00	100.00	100.00	97.35	97.35
$46 \le i \le 50$	99.34	100.00	100.00	100.00	98.68	96.69

TABLE III
DETECTION PASSING RATE (%) UNDER VARIOUS S_2

s_2	Salt & pepper noise NF=0.03	Gaussian noise NF=0.005	Wiener filtering 9×9	JPEG compression QF=20	Scaling SF=0.5	Rotation Ang=25°
3.00%	**99.34**	**99.34**	**98.01**	**98.68**	**86.75**	**86.75**
4.70%	95.36	98.68	96.03	98.68	74.17	78.15
6.30%	95.36	98.68	95.36	98.01	70.86	70.86
7.80%	96.69	99.34	96.03	98.01	76.82	84.77
9.40%	96.69	98.68	95.36	98.01	72.85	65.56

TABLE IV
DETECTION PASSING RATE (%) UNDER VARIOUS S_2

s_2	Salt & pepper noise NF=0.008	Gaussian noise NF=0.001	Wiener filtering 3×3	JPEG Compression QF=80	Scaling SF=1.5	Rotation Ang=5°
3.00%	**99.34**	**99.34**	**99.34**	**100.00**	**98.01**	**98.01**
4.70%	97.35	99.34	98.68	99.34	96.69	96.03
6.30%	97.35	99.34	98.68	99.34	96.69	94.04
7.80%	98.68	99.34	99.34	99.34	96.69	96.69
9.40%	98.68	99.34	98.68	99.34	96.03	94.70

TABLE V
DETECTION PASSING RATE (%) UNDER VARIOUS S_1

s_1	Salt & pepper Noise NF=0.03	Gaussian noise NF=0.005	Wiener filtering 9×9	JPEG Compression QF=20	Scaling SF=0.5	Rotation Ang=25°
3.10%	91.39	94.70	82.78	92.71	66.23	66.89
3.90%	98.68	98.01	91.39	96.69	74.17	80.79
4.70%	99.34	98.68	94.70	97.35	78.81	80.79
5.50%	98.01	98.68	94.04	98.01	80.79	84.77
6.30%	**98.68**	**100.00**	**94.70**	**99.34**	**84.77**	**86.09**
7.00%	98.68	99.34	97.35	99.34	82.78	82.12
7.80%	98.68	99.34	96.03	97.35	88.08	88.08
8.60%	99.34	99.34	96.03	98.68	89.40	87.42
9.40%	99.34	99.34	98.01	98.68	86.75	86.75
10.20%	99.34	99.34	97.35	98.68	87.42	86.75
10.90%	99.34	99.34	96.69	98.68	87.42	86.75

The experimental results are shown in Tables V and VI. As shown, when $s_1 = 6.3\%$, the detection passing rates are higher than others for most content-preserving manipulations.

As shown by the experimental results, the comprehensive performance of the proposed method changes with various parameters setting. In fact, for image hash technology, there is no absolute standard for performance evaluation due to various applications including image content authentication, image retrieval, watermarking, and so on. Different applications require different performance. For applications such as image retrieval and watermarking, shorter and fixed-length hash codes are necessary. However, for image content authentication, the abilities to detect malicious tampering and tampering localization are of crucial importance.

B. Robustness Analysis and Comparison

This experiment is designed to test whether the proposed method is robust to incidental changes caused by content-preserving manipulations. In our experiment, 3000 tested images come from the UCID image database and the Columbia university image database. We calculate the hash value of each tested original image; manipulate the images via JPEG compression, add noise, filtering, rotation, and scaling using StirMark and calculate the hash values of

TABLE VI
DETECTION PASSING RATE (%) UNDER VARIOUS S_1

S_1	Salt & pepper noise NF=0.008	Gaussian noise NF=0.001	Wiener filtering 3×3	JPEG Compression QF=80	Scaling SF=1.5	Rotation Ang=5°
3.10%	98.68	100.00	98.68	99.34	91.39	80.79
3.90%	100 .00	100.00	98.68	100 .00	98.01	89.40
4.70%	100 .00	100.00	98.68	100.00	97.35	96.02
5.50%	99.34	100.00	98.68	99.34	98.01	96.69
6.30%	**100 .00**	100.00	**100 .00**	**100 .00**	**97.35**	**96.03**
7.00%	100 .00	100.00	100.00	100 .00	96.69	96.03
7. 80%	99.34	100.00	99.34	99.34	98.01	94.04
8.60%	100 .00	100.00	100.00	100 .00	98.01	96.69
9.40%	99.34	100.00	99.34	100.00	98.01	98.01
10.20%	99.34	100.00	100.00	99.34	97.35	96.69
10.90%	99.34	100.00	98.68	99.34	97.35	98.68

Note: in tables 1-6, 'NF' represents noise factor, 'QF' represents quality factor. 'SF' represents the scaling rate, and 'Ang' represents the angle of rotation.

TABLE VII
COMPARISON RESULTS OF THE PERCEPTUAL ROBUSTNESS

Works		Approaches	JPEG quality factor				Median filtering	
			90	80	60	20	2*2	4*4
Monga[23]		Feature point	100.0	100.0	98.2	82.37	98.4	93.7
Wang [28]		Feature point	100.0	99.8	97.9	81.10	93.7	78.3
Wang [29]		Image block	86.1	91.4	92.7	76.4	84.3	77.2
Proposed method	T=4	Feature point & Image block	100.0	100.0	100.0	98.5	100.0	100.0
	T=10		100.0	100.0	100.0	100.0	100.0	100.0
Works		Approaches	Salt and pepper noise				Median filtering	
			0.002	0.004	0.03	0.05	5*5	12*12
Monga[23]		Feature point	86.7	81.7	/	/	91.6	/
Wang [28]		Feature point	95.9	93.1	/	/	76.3	/
Wang [29]		Image block	88.7	87.9	/	/	72.8	/
Proposed method	T=4	Feature point & Image block	100.0	100.0	98.5	88.7	98.4	82.8
	T=10		100.0	100.0	100.0	100.0	100.0	100.0

manipulated versions. Finally, we compare these hash values pair by pair and measure the hash distance between the original image and the manipulated versions. Fig. 5 plots the tested results of the detection passing rate.

As shown in Fig. 5, as threshold T increases, the detection passing rate increases gradually. This means that the intensity of the robustness increases when threshold T is higher. In case $T > 2.5$, the detection passing rates are satisfactory. When noise is adding, the detection passing rates are satisfactory when the noise factor is less than 0.05. For the rotation transform, the detection passing rate is satisfactory when the rotation angle is less than 25°. For large rotation angles, we can correct the rotated image by using the affine transformation described in section III.C. For the translation and cropping transformations, as extracted feature points correspond to the original image feature points, the detection passing rates are maintained.

Compared with the feature point-based methods [23], [28], and the image block-based method [29], the proposed method presents satisfactory perceptual robustness. This robustness is achieved because the proposed method synthetically utilizes both feature points and image blocks to generate forensic features, in which the SIFT features are invariant to translation, rotation and scaling transformation, and the DCT-based block features are robust to content-preserving manipulations such as JPEG compression. These features contribute to our approach having stronger robustness than unilateral features do. Additionally, Watson's visual model is used to extract visually sensitive features, which contributes to our solution's real perceptual robustness and perceptual sensitivity. Table VII lists the comparison results of the detection passing rate in terms of content-preserving manipulations. Here, "/" represents an item that we have not investigated.

C. Sensitivity Analysis and Performance Comparison

In general, for perceptual image hash, perceptual robustness and perceptual sensitivity are in opposition. The former requires good stability under slight perturbation, whereas the latter requires that the algorithm is sensitive to small malicious modifications. Therefore, the trade-off between perceptual robustness and perceptual sensitivity must be considered in practical applications. To analyze these characteristics quantitatively, we present formalized definition of false negative rate (R_{FN}) and false positive rate (R_{FP}) as follows:

$$\begin{aligned} R_{FN} &= \text{False negative rate} \\ &= \frac{|\{I'|(I' \in M_T) \wedge (\Delta(I') = 1)\}|}{|\bigcup_I M_T|} \times 100\% \\ &= \frac{\text{The number of tampered images detected as authentic}}{\text{Total number of tampered images}} \times 100\% \end{aligned} \quad (19)$$

$$\begin{aligned} R_{FP} &= \text{False positive rate} \\ &= \frac{|\{I'|(I' \in \bigcup_I \{\{I\} \cup M_I\}) \wedge (\Delta(I') = 0)\}|}{|\bigcup_I \{\{I\} \cup M_I\}|} \times 100\% \\ &= \frac{\text{The number of authentic images detected as tampered}}{\text{Total number of tested images}} \times 100\% \end{aligned} \quad (20)$$

where $|\bullet|$ means the cardinality of a set.

In the experiment, 1000 tested images with different sizes come from the Ground truth Database [45]. The tampering manipulations include adding image objects, copy-move attack, object replacement attack, hiding image object attack, among others. The content-preserving manipulations and relative parameters are listed in Table VIII. We examine the perceptual robustness and perceptual sensitivity of the proposed method in terms of the receiver operating characteristics (ROC). For each original image I_o and corresponding manipulated image I_t, we compare the hash values and compute R_{FN} and R_{FP}. We repeat this process with different T thresholds and finally arrive at the ROC (Fig. 6). As shown in Fig. 6, the sensitivity becomes weaker when threshold T is higher. This means that a smaller threshold will lead to a higher sensitivity. Conversely, to obtain higher sensitivity, threshold T should be as small as possible. However, a smaller threshold will lead to a weaker robustness. In practice, an applicable threshold T should be selected according to specific application requirements.

We compare the proposed method with Zhao et al. [34] and Monga et al. [24] (NMF hashing) in terms of the ROC. As shown in the experimental results Fig. 7, the detection

TABLE VIII
CONTENT-PRESERVING MANIPULATIONS AND RELATIVE PARAMETERS

Manipulations	Parameters	Testing times
Gaussian noise	Sigma: 0-0.1	10
Salt and pepper noise	Sigma: 0-0.1	10
Speckle noise	Sigma: 0-0.1	10
Gaussian Blur	Size: 3-21 Sigma: 5	10
Circular Blur	Radius: 1-10	10
Motion blur	Len: 5-15	9
Rotation	Degree: 5-45	9
Scaling	25%-200%	5
JPEG compression	Quality factor: 5-50	10
Gamma correction	Gamma: 0.75-1.25	10

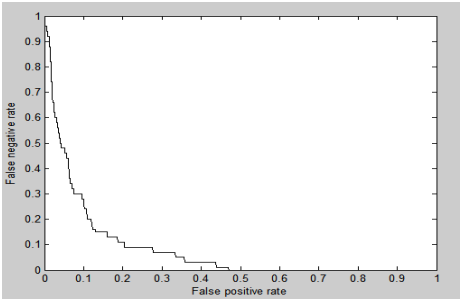

Fig. 6. ROC curve of the proposed method.

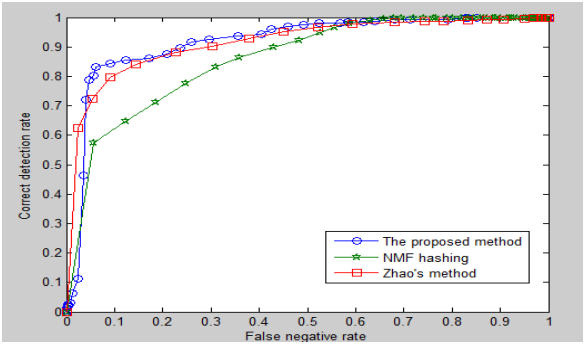

Fig. 7. Compared result of ROC.

passing rate of the proposed method is higher than that of Zhao's [34] and NMF hashing [24] under the same false negative rate.

To estimate the overall performance of the proposed method, we compare it with state-of-the-art schemes that have been developed. The comparison results are listed in Tables IX and X. Here, "/" represents an item that we have not investigated. "Yes (No)" denotes that the method has (has not) the corresponding functionality. As shown, the overall performance of the proposed method is satisfactory. In general, the global-feature-based methods such as [4], [18], and [24] have difficulty providing tampering localization because of the lack of a local description. The feature-point-based methods such as [2], [28], and [30] and the image-block-based methods such as [15] and [29] are established by the local features of the image; thus, they have the functionality of tampering localization. However, image-block-based methods such as [15], [18], [29], [34], and [35] lack good robustness to geometric transformations such as rotation/scaling, cropping,

TABLE IX
COMPREHENSIVE COMPARISON RESULTS I

Citation Index	Approaches	Tampering localization
Lv (2012) [2]	Feature point & shape context	Yes
Swaminathan(2006) [4]	Global Fourier transform coefficients	No
Fawad (2010) [15]	Image blocks	Yes
Mihcak (2001) [18]	Random rectangles	No
Monga (2006) [23]	Feature points	No
Monga 2007 [24]	Non-Negative Matrix Factorizations	No
Wang (2012) [28]	Feature points	Yes
Wang (2012) [29]	Image blocks	Yes
Lu (2010) [30]	Feature points & Image blocks	Yes
Zhao (2013) [34]	Zernike moments & salient regions	Yes
F.Khelifi (2010) [35]	Image blocks	No
Sun (2012) [39]	Image blocks	No
Proposed method	Feature points & Image blocks	Yes

TABLE X
COMPREHENSIVE COMPARISON RESULTS II

Citation Index	JPEG, (Quality factor)	Adding noise, (Noise factor)	Filtering (Filter order)	Rotation (Degress)	Scaling (%)	Cropping (%)	Shear mapping (%)
[2]	10	0.01	/	30°	0.5-1.5	10%	10%
[4]	10	0.5	11×11	20°	0.5-1.5	30%	10%
[15]	11	/	Yes	No	No	No	No
[18]	10	Yes	4×4	5°	0.5	10%	5%
[23]	10	Yes	3×3	5°	0.6	20%	5%
[24]	5	/	3×3	15°	Yes	Yes	5 pixels
[28]	10	0.004	5×5	Any	0.8-1.1	10%	No
[29]	10	0.004	5×5	No	No	No	No
[30]	10	/	No	45°	0.3-1.5	36%	No
[34]	30	0.01	/	5°	0.5-1.5	2%	No
[35]	10	Yes	11×11	5°	No	10%	No
[39]	Yes	0.1	/	45°	0.25-2	35%	No
Proposed method	10	0.05	12×12	Any	0.25-2	50%	30%

and shear mapping because these transformations would distort the image blocks, which then would lead to a larger change of the hash code. In the proposed method, the image-block-based features and the key-point-based features are combined to generate robust hash code. Additionally, Watson's visual model is used to extract visually sensitive features. These features contribute many advantages to our solution in both global and local perspectives.

D. Visual Effect of Tampering Localization

For a perceptual image hash scheme, the tampering localization functionality is of crucial importance. This functionality refers to a capability to identify compromised image regions. This functionality can be visually demonstrated via visual effect. We also investigated this functionality of the proposed method via quantitative assessment. Fig. 8-18 show the visual detection results of the proposed method for different attack types. The detected results are indicated by white color regions.

As shown by the experimental results, the proposed method can detect the locations of compromised image regions. It is valid for detecting compromised images that have undergone geometric distortions. To assess the detection performance quantitatively, we estimate the tampering rate and detection rate at the pixel level. Here, the tampering rate and the

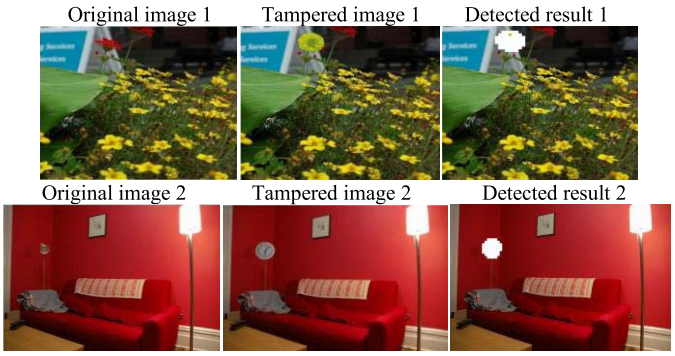

Fig. 8. Detection results for object replacement attack.

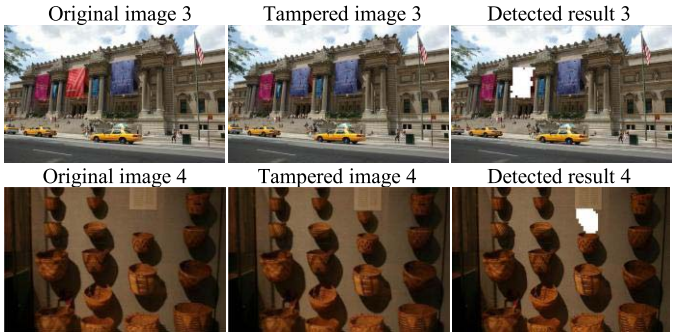

Fig. 9. Detection results for Copy-move attack.

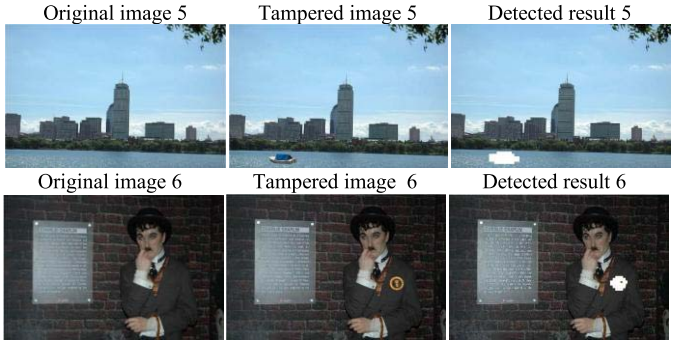

Fig. 10. Detection result for adding image object attack.

Fig. 11. Detection result for hiding image object attack.

detection rate are defined as follows:

$$Tr = \text{Tampering rate} = \frac{\text{The size of tampered regions}}{\text{The size of tested image}} \times 100\% \quad (21)$$

$$Dr = \text{Detection rate} = \frac{\text{The size of detectioned regions}}{\text{The size of tested image}} \times 100\% \quad (22)$$

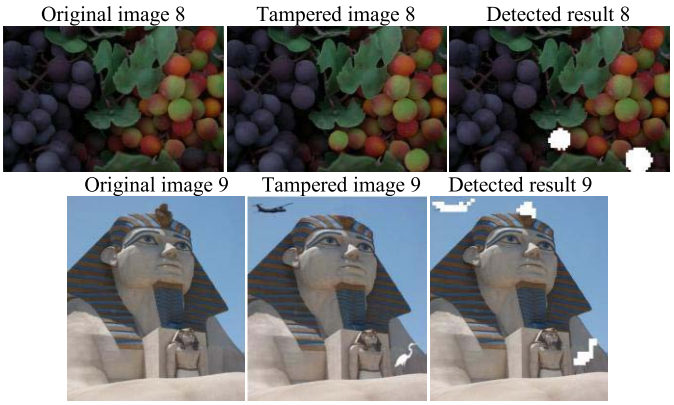

Fig. 12. Detection results for multiple compromised regions.

Fig. 13. Detection result for tampering and rotating (10°) + scaling (80%).

Fig. 14. Detection result for tampering and rotating (30°) + scaling (200%).

Fig. 15. Detection result for tampering + translating (Horizontal 40, vertical 40).

We have estimated the tampering rate and the detection rate for tested images in Fig. 8-18. The results are shown in Table XI. As shown in Table XI, the detection rates are close to the tampering rates.

E. Computational Complexity Analysis

Computational complexity includes the calculating time spent on hash generation, tampering detection and tampering localization. In our experiments, the 1000 tested images are of varying sizes. Table XII shows the statistical average values of the time cost, demonstrating that the proposed method is efficient in computing time.

F. Security Analysis

Image tampering essentially means that the content of an image has been changed. Therefore, the perceptual image hash refers to a content-based hash; that is, the hash code is closely related to the image content. Therefore, the security

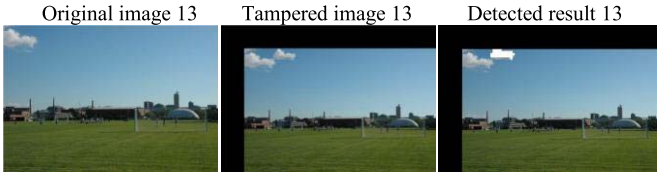

Fig. 16. Detection result for tampering + translating (Horizontal 90, vertical 90).

Fig. 17. Detection results for tampering and shear mapping (factor=0.1).

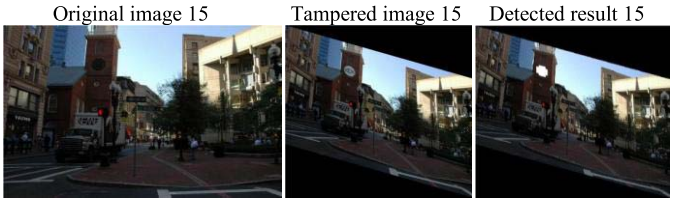

Fig. 18. Detection results for tampering and shear mapping (factor=0.3).

TABLE XI
TAMPERING RATE (Tr) AND DETECTION RATE (Dr)

Image index	Tr	Dr	Image index	Tr	Dr	Image index	Tr	Dr
1	1.36%	1.51%	6	0.46%	0.55%	11	0.71%	0.69%
2	0.91%	1.01%	7	2.85%	3.03%	12	2.37%	2.03%
3	1.95%	2.14%	8	2.33%	2.37%	13	0.67%	0.69%
4	1.06%	1.23%	9	1.60%	2.08%	14	0.75%	0.84%
5	0.79%	0.85%	10	0.45%	0.49%	15	0.42%	0.47%

TABLE XII
TIME SPENT ON HASH GENERATION T_g(s), TAMPERING DETECTION AND TAMPERING LOCALIZATION T_d(s)

Image size	128× 128	200× 200	256× 256	384× 384	512× 512	656× 656	832× 832	976× 976	1200× 1200
T_g(s)	0.26	0.55	1.21	2.51	4.32	7.29	10.56	15.23	21.36
T_d(s)	0.04	0.09	0.11	0.21	0.44	1.13	2.18	5.26	8.31

of the hash code represents content-based security, which refers to the ability to resist forgery attacks. Potential forgeries primarily include (1) The unauthorized users forged hash code from known image content, (2) Deriving image content or forging image content from a known hash code, and (3) Replacing realistic image content with other content using a hash collision (seeing Fig. 19).

The occurrence of (1) or (2) is closely related to the correlation of the image content with the corresponding hash value, and (3) is due to a hash collision. The purpose of encryption is to hide the corresponding relationship between the hash code and corresponding image content. The purpose of randomization is to scramble the hash code, increasing randomness to improve collision resistance.

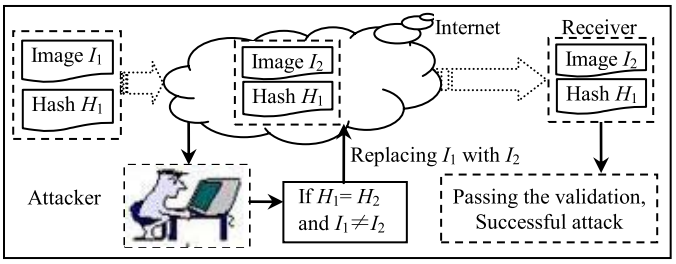

Fig. 19. Replaces realistic image by others via using hash collision.

To avoid possible forgeries, de-correlating the hash value from the corresponding image content and randomizing the hash value are necessary. In the process of generating the hash, we reduce the correlation and increase the randomness by using encryption and randomization. We analyze the security of the proposed method in two aspects:

1) Security Evaluation for Encrypting and Randomization: An ideal cryptography system requires that the cipher text be distributed fully and evenly in the cipher text space, allowing the plaintext to be covered completely. If a key sequence is random, then the cipher text will be uniformly distributed. In our work, the randomness of the final hash value is introduced by the variable k_i, a pseudorandom number produced by the chaotic key generation algorithm.

For the chaotic sequence $k_1, k_2, \ldots, k_N, \ldots$, the input and output are evenly distributed in (0,1). H. G. Schuster [46] proved that the probability density function $\rho(x)$ of $k_1, k_2, \ldots, k_N, \ldots$ is:

$$\rho(x) = \begin{cases} 1/(\pi\sqrt{x(1-x)}), & 0 < x < 1 \\ 0, & \text{else} \end{cases} \tag{23}$$

Therefore, the mean is:

$$m = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} k_i = \int_{-\infty}^{+\infty} x\rho(x)dx$$
$$= \int_0^1 \frac{x}{\pi\sqrt{x(1-x)}} dx = 0.5 \tag{24}$$

After encrypting and randomization, the final hash can be considered obeying the same probability distribution as the key sequence. Next, we analyze the statistical distribution of the final hash. In our encryption algorithm, the probability of each plaintext bit is not identical. Assuming that the probability of every plaintext bit is 0 or 1 is an independent event; the statistical distribution of the encrypted data is as follows:

Let m_i denote the i^{th} plaintext bit and q denote the probability that m_i is equal to 1, i.e., $P(m_i = 1) = q$. A chaotic key sequence is similar to white noise. For a single bit, the probability of equaling 0 or 1 is identical, that is, $P(k_i = 1) = P(k_i = 0) = 0.5$. Therefore, the probability that the i^{th} cipher text bit is equal to 1 is as follows:

$$P(c_i = 1) = P((m_i = 1) \wedge (k_i = 0)) + P((m_i = 0) \wedge (k_i = 1))$$
$$= q \times 0.5 + (1 - q) \times 0.5 = 0.5$$

Here, $k_i(m_i$ or $c_i) = 1$(or 0) denotes that the i^{th} bit of the key sequence (plaintext or cipher text) is equal to 1 (or 0), and $P(c_i = 1)$ denotes the probability of $c_i = 1$.

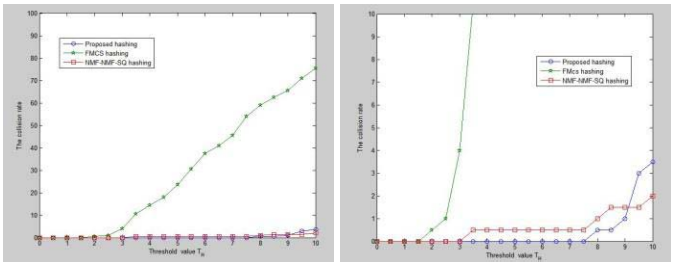

Fig. 20. Comparison results of the collision rate. The right image is a zoomed-in view of the left.

As shown by the above analysis, our encrypting and randomization algorithm achieves satisfactory security.

2) Probability of Hash Collision: A hash collision means that there is the same hash value $H(M) = H(M')$ for two different images M and M'. To investigate the collision resistance of the proposed method, we built the testing data set using the standard USC-SIPI image database, and the tested 1000 images are different in texture. The hash values are compared pair-by-pair via computing their Euclidean distance (15). In our experiment, we have not yet found a hash collision in the testing data set. With the relaxed limitation, we define a concept as follows:

Less Than a Given Value Hash Collision: Given a value T_H, for two different images M and M', a hash collision occurs if the distance D between $H(M)$ and H(M') is not greater than T_H. We define the probability of hash collision as follows: The collision rate

$$= \frac{\text{Image pairs numbers of distance } D \leq T_H}{\text{Total pairs numbers of tested images}} \times 100\% \quad (25)$$

We test the collision rate under a series of values T_H and compare the collision rates with existing methods. As shown by the experimental results in Fig. 20, the proposed method reaches the optimal collision rate when $T_H \leq 7.5$, and it is superior to the compared methods in terms of collision resistance.

V. Conclusion

In this paper, a real perceptual image hash method is proposed. Based on this hash, an image tampering detection and tampering localization method is presented. As a tool for image content authentication, the proposed method is robust to geometric deformations and content-preserving manipulations such as JPEG compression, adding noise, filtering, and others. It is sensitive to changes caused by malicious attacks, and it achieves a trade-off between robustness against geometric distortion and tampering localization. The experimental results show the effectiveness and the availability of the proposed algorithm for different tampering attacks at three performance levels: image-tampering detection (detection accuracy), compromised region localization (visual effect), and localization accuracy (detection rate at the pixel level). The proposed method can be used for content-based image authentication and for image retrieval and matching in large-scale image databases.

Acknowledgments

We would like to acknowledge the helpful comments and kindly suggestions provided by anonymous referees.

References

[1] N. A. Stillings, S. E. Weisler, C. H. Chase, M. H. Feinstein, J. L. Garfield, and E. L. Rissland, *Cognitive Science: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 1995.

[2] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1081–1093, Jun. 2012.

[3] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. Int. Conf. Image Process.*, 2000, pp. 664–666.

[4] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[5] H. G. Schaathun, "On watermarking/fingerprinting for copyright protection," in *Proc. 1st Int. Conf. Innov. Comput., Inf., Control (ICICIC)*, Aug./Sep. 2006, pp. 50–53.

[6] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[7] Y.-H. Kuo, K.-T. Chen, C.-H. Chiang, and W. H. Hsu, "Query expansion for hash-based image object retrieval," in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 65–74.

[8] G. Zhu, J. Huang, S. Kwong, and J. Yang, "Fragility analysis of adaptive quantization-based image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 133–147, Mar. 2010.

[9] S.-H. Han and C.-H. Chu, "Content-based image authentication: Current status, issues, and challenges," *Int. J. Inf. Secur.*, vol. 9, no. 1, pp. 19–32, 2010.

[10] A. Hadmi, W. Puech, B. A. E. Said, and A. A. Ouahman, "Perceptual image hashing," in *Computer and Information Science: Watermarking*, vol. 2, M. D. Gupta, Ed. Rijeka, Croatia: InTech, May 2012.

[11] M. Schneider and S.-F. Chang, "A robust content based digital signature for image authentication," in *Proc. Int. Conf. Image Process.*, Sep. 1996, pp. 227–230.

[12] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.

[13] C.-S. Lu and H.-Y. M. Liao, "Structural digital signature for image authentication: An incidental distortion resistant scheme," *IEEE Trans. Multimedia*, vol. 5, no. 2, pp. 161–173, Jun. 2003.

[14] S. Tang, J.-T. Li, and Y.-D. Zhang, "Compact and robust image hashing," in *Proc. ICCSA*, 2005, pp. 547–556.

[15] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Process.*, vol. 90, no. 5, pp. 1456–1470, 2010.

[16] X. C. Guo and D. Hatzinakos, "Content based image hashing via wavelet and radon transform," in *Advances in Multimedia Information Processing* (Lecture Notes in Computer Science), vol. 4810. Berlin, Germany: Springer-Verlag, pp. 755–764.

[17] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," in *Proc. Int. Conf. Inf. Technol., Coding Comput.*, 2000, pp. 178–183.

[18] M. K. Mihçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Proc. ACM CCS-8 Workshop Secur. Privacy Digit. Rights Manage.*, 2001, pp. 13–21.

[19] S. S. Kozat, R. Venkatesan, and M. K. Mihçak, "Robust perceptual image hashing via matrix invariants," in *Proc. Int. Conf. Image Process.*, 2004, pp. 3443–3446.

[20] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," in *Proc. Int. Conf. Image Process.*, Oct. 1998, pp. 435–439.

[21] S. K. Bhatacherjee and P. Vandergheynst, "End-stopped wavelets for detecting low-level features," *Proc. SPIE*, vol. 3813, pp. 732–741, Oct. 1999.

[22] V. Monga and B. L. Evans, "Robust perceptual image hashing using feature points," in *Proc. Int. Conf. Image Process.*, Oct. 2004, pp. 677–680.

[23] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006.

[24] V. Monga and K. M. Mihçak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 376–390, Sep. 2007.
[25] C.-S. Lu and C.-Y. Hsu, "Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication," *Multimedia Syst.*, vol. 11, no. 2, pp. 159–173, Dec. 2005.
[26] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in *Proc. IEEE Int. Conf. Image Process.*, Sep./Oct. 2007, pp. VI-117–VI-120.
[27] L. Sumalatha, V. Venkata Krishna, and V. Vijaya Kumar, "Local content based image authentication for tamper localization," *Int. J. Image, Graph., Signal Process.*, vol. 4, no. 9, pp. 30–36, 2012.
[28] X. Wang, J. Xue, Z. Zheng, Z. Liu, and N. Li, "Image forensic signature for content authenticity analysis," *J. Vis. Commun. Image Represent.*, vol. 23, no. 5, pp. 782–797, 2012.
[29] X. Wang, N. Zheng, J. Xue, and Z. Liu, "A novel image signature method for content authentication," *Comput. J.*, vol. 55, no. 6, pp. 686–701, 2012.
[30] W. Lu and M. Wu, "Multimedia forensic hash based on visual words," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 989–992.
[31] W. Lu, A. L. Varna, and M. Wu, "Forensic hash for multimedia information," *Proc. SPIE*, vol. 7541, p. 75410Y, Jan. 2010.
[32] C.-Y. Lin, "Watermarking and digital signature techniques for multimedia authentication and copyright protection," Ph.D. dissertation, Graduate School Arts Sci., Columbia Univ., New York, NY, USA, 2000.
[33] Y. Zhao, "Perceptual image hash using texture and shape feature," *J. Comput. Inf. Syst.*, vol. 8, no. 8, pp. 3519–3526, 2012.
[34] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 55–63, Jan. 2013.
[35] F. Khelifi and J. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 981–994, Apr. 2010.
[36] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 194–201, Jan. 2012.
[37] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2491–2504, Nov. 2009.
[38] L.-W. Kang, C.-S. Lu, and C.-Y. Hsu, "Compressive sensing-based image hashing," in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 1285–1288.
[39] R. Sun and W. Zeng, "Secure and robust image hashing via compressive sensing," *Multimedia Tools Appl.*, vol. 70, no. 3, pp. 1651–1665, 2014.
[40] A. Hadmi, W. Puech, B. A. E. Said, and A. A. Ouahman, "A robust and secure perceptual hashing system based on a quantization step analysis," *Signal Process., Image Commun.*, vol. 28, no. 8, pp. 929–948, 2013.
[41] C. Qin, S.-Z. Wang, and X.-P. Zhang, "Image hashing based on human visual system," *J. Image Graph.*, vol. 11, no. 11, pp. 1678–1681, 2006.
[42] A. B. Watson, "DCT quantization matrices visually optimized for individual images," *Proc. SPIE*, vol. 1913, pp. 202–216, Sep. 1993.
[43] M. Fakhredanesh, R. Safabakhsh, and M. Rahmati, "A model-based image steganography method using Watson's visual model," *ETRI J.*, vol. 36, no. 3, p. 479, 2014.
[44] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[45] Dept. Comput. Sci. Eng., Univ. Washington. (2013). *Ground Truth Database*. [Online]. Available: http://www.cs.washington.edu/research/imagedatabase/
[46] H. G. Schuster and W. Just, *Deterministic Chaos: An Introduction*, 4th ed. Weinheim, Germany: Wiley, 2005.

Xiaofeng Wang received the B.S. degree in applied mathematics from Tianjin University, China, and the M.S. degree in mathematics and the Ph.D. degree in mechanical and electronic engineering from the Xi'an University of Technology, China. In 2007, she joined the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, where she was a Post-Doctoral Researcher until 2010. In 2012, she joined the Grasp Laboratory, University of Pennsylvania, where she was a Visiting Scholar until 2013. She is currently a Professor with the Department of Mathematics, Xi'an University of Technology. Her current research interests include multimedia forensics and security, image processing, steganography, and steganalysis.

Kemu Pang received the B.S. degree in mathematics from Yanan University, Shaanxi, China, in 2011, and the M.S. degree in mathematics from the Xi'an University of Technology, Shaanxi, in 2014. His research interests include image forensics and security, and image processing.

Xiaorui Zhou received the B.S. degree in mathematics from Changzhi University, Shanxi, China, in 2012. He is currently pursuing the M.S. degree in mathematics with the Xi'an University of Technology. His research interests include multimedia forensics and security, and image processing.

Yang Zhou received the B.S. degree in mathematics from Yanan University, Shaanxi, China, in 2011, and the M.S. degree in mathematics from the Xi'an University of Technology, Shaanxi, in 2014. Her research interests include image forensics and security, and image processing.

Lu Li received the B.S. degree from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2010, and the M.S. degree in pattern recognition and intelligence systems from the Institute of Artificial Intelligence and Robotics, XJTU, in 2014. He is currently pursuing the Ph.D. degree with the State University of New York at Buffalo. His research interests include image forensics and security.

Jianru Xue (M'06) received the M.S. and Ph.D. degrees in pattern recognition and intelligence systems from the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University (XJTU), in 1999 and 2003, respectively. He was with Fuji Xerox, Tokyo, Japan, from 2002 to 2003, and visited the University of California at Los Angeles from 2008 to 2009. He has been a Professor with the Institute of Artificial Intelligence and Robotics, XJTU, since 2007. His research interests include computer vision and pattern recognition, machine learning, statistical approaches for video analysis, and image/video coding. He served as the Co-Organization Chair of the Asian Conference on Computer Vision in 2009 and the Conference on Virtual System and Multimedia in 2006. He also served as a PC Member of Pattern recognition in 2012, and the Asian Conference on Computer Vision in 2010 and 2012.