International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **2**, February **2013**

 ISSN 1349-4198
pp. 599–610

DICTIONARY-BASED DATA HIDING USING IMAGE HASHING STRATEGY

Chuan Qin[1,2], Chin-Chen Chang[2,3,*] and Pei-Ling Tsou[2]

[1]School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
No. 516, Jungong Road, Shanghai 200093, P. R. China
qin@usst.edu.cn

[2]Department of Information Engineering and Computer Science
Feng Chia University
No. 100, Wenhwa Road, Taichung 40724, Taiwan
∗Corresponding author: alan3c@gmail.com; pltsou02@gmail.com

[3]Department of Computer Science and Information Engineering
Asia University
No. 500, Lioufeng Road, Taichung 41354, Taiwan

Received November 2011; revised March 2012

Abstract. *This paper proposes a novel data hiding framework based on image hashing strategy. First, the best-matching blocks for each cover image block are searched in the existing database, which can be generated by the codebook of vector quantization technique. Then, a dictionary consisting of different versions for each cover block is constructed by a set of content-preserving manipulations, and each version is assigned with a unique binary representation. According to the secret data hidden for each block, corresponding versions of the block in the dictionary are chosen to substitute the cover block and form the stego image. Due to the properties of robustness and anti-collision of image hashing, the secret data can be extracted correctly by comparing the stego block with the blocks in the reconstructed dictionary. The experimental results show the effectiveness and flexibility of the proposed framework.*
Keywords: Data hiding, Dictionary, Image hashing, Vector quantization

1. **Introduction.** With the great improvement in digital signal processing and networking technologies, information exchanges via the Internet bring convenience to people, but some security problems have been incurred because the Internet is a public environment. Therefore, malicious attackers can easily intercept the information that is transmitted in this way. Currently, researchers' efforts to solve such security problems can be classified mainly into two categories, i.e., cryptography [1-3] and data hiding techniques [4].

To achieve secure communication, researchers in cryptography aim to convert the original information, plaintext, to unintelligible strings, i.e., ciphertext, using cryptography algorithms with secret keys. Unless the receiver has the proper keys, the encrypted information cannot be decrypted from the meaningless strings to recover the original information. A cryptography system seems to do a great job of protecting transmitted information, but, due to the pointless characteristic of the encrypted information, it attracts the attention of attackers. To solve this problem, the data hiding technique, which embeds to-be-transmitted information into carriers, such as images, has been studied extensively in recent years [5-13]. The most typical data hiding technique is the least significant bit (LSB) substitution algorithm [10], which directly replaces the least significant bits of the original pixels with secret bits. This approach is effective in achieving the goal of making

images nearly undistinguishable before and after embedding the secret information into them. In addition to the data hiding schemes in the spatial domain, such as LSB-like methods, there are some other schemes that are based on the frequency domain [11-13], in which the cover image first is transformed by a frequency transformation, such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT). The secret messages are embedded into related coefficients of the cover image. Then, the inverse transformation is performed to obtain the stego-image. Most of the data hiding methods based on spatial domain and frequency domain adopt the same strategy that embeds the secret data or its encoded form into the unimportant ingredients of the cover image, such as the LSBs of cover pixels or frequency coefficients. This strategy for data embedding restricts the hiding capacity of secret data that totally depends on the volume of the unimportant ingredients of the cover image. In addition, due to the widely use of this strategy, a large number of steganalysis schemes have been proposed to detect this type of hiding operation according to the change of statistical characteristics in cover image, which might make this kind of data hiding methods unsecure to a certain extent [14,15].

In this work, we propose a novel data hiding framework for digital images using an image hashing strategy, which, to the best of our knowledge, has not been reported by earlier researchers. Different with the traditional methods mentioned above, the hiding capacity of secret data in the proposed framework is related to the size of the constructed dictionary, which is more flexible. First, representative image blocks are chosen from the image database by comparing them with the cover image, which is like the process of vector quantization. Then, an image block dictionary is constructed by transforming each chosen block into a series of perceptually similar blocks. The transformed image blocks in the same series have the same or similar image hash values. The secret data embedding procedure can be achieved by substituting the cover image block with different blocks in the constructed dictionary. In the data extraction procedure, the hidden secret data can be extracted efficiently by comparing the image hashes of the stego-image blocks and the blocks in the reconstructed dictionary. The proposed dictionary-based data hiding framework can be easily implemented, and the hiding capacity can be flexibly increased by integrating more content-preserving manipulations. Additionally, because the process of constructing dictionary is controlled by the secret key, the proposed framework satisfies the security requirement in the sense of cryptography.

The rest of the paper is organized as follows. Section 2 introduces the conception of image hashing and the image hashing algorithm used in this paper. Section 3 describes our dictionary-based data hiding framework using the image hashing strategy. Experimental results and analysis are presented in Section 4, and Section 5 concludes this paper.

2. **Image Hashing.**

Image hash is a compact representation of the image content, which can also be called image digest or image authentication code [16]. Recently, the image hashing technique has been used extensively in various applications, including digital watermarking, tamper detection, image authentication, and content-based image retrieval (CBIR). Traditional cryptographic hash functions, such as SHA-1 and MD5, can inconvertibly map a variable length input message to a short, fixed-length string. Since the cryptographic hash function is sensitive, any slight change of the input message will influence the result of the hash value significantly. But for the scenario of image hashing, the traditional cryptographic hash functions are not suitable. This is due to the fact that the images usually must undergo various processes, many of which are content-preserving, e.g., JPEG compression, filtering, and resizing, even though the digital representations of the images have been changed. Therefore, an image hashing scheme should satisfy that perceptually similar images will produce the same or similar hash sequences, while

perceptually distinct images will generate significantly different hash sequences to avoid collision.

We denote the image for the hashing calculation as I_o, the procedure of image hashing as $\mathrm{F_H}(\cdot)$, and the normalized distance function for two hash sequences as $\mathrm{D}(\cdot)$. The two desirable properties of an ideal image hashing are listed below:

(1) *Robustness*. Suppose I_s is the processed version of I_o by some content-preserving operations, and the processed version is visually similar to I_o. The generated hashes of I_o and I_s should satisfy:

$$\mathrm{D}\left[\mathrm{F_H}(I_o), \mathrm{F_H}(I_s)\right] \leq \varepsilon_1. \tag{1}$$

(2) *Anti-collision*. Suppose I_t is an image that is perceptually distinct from I_o or that is a maliciously altered version of I_o. The generated hashes of I_o and I_t should satisfy:

$$\mathrm{D}\left[\mathrm{F_H}(I_o), \mathrm{F_H}(I_t)\right] \geq 1 - \varepsilon_2 \tag{2}$$

where ε_1 and ε_2 are small positive numbers belonging to [0, 1].

There are two main stages in most reported image hashing schemes, i.e., feature extraction and hash generation, as shown in Figure 1. In the feature extraction stage, the salient features are extracted to represent the main content of the image compactly. Then the extracted features are quantized into a binary or real number sequence to form the final hash value. Recently, several image hashing schemes have been proposed, which can be classified into two categories based on their methods of feature extraction, i.e., frequency domain based schemes [16-19] and spatial domain based schemes [20,21].

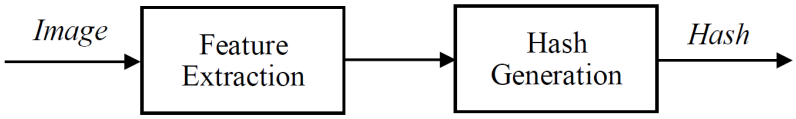

FIGURE 1. Two main stages of the general image hashing scheme

In this work, we used a classic, DCT-based image hashing algorithm [16], which is in the light of that the image cannot maintain the invariance of its low frequency coefficients when significant changes occur in the image. The DCT coefficient matrices of the input image are projected on R low-pass filtered random masks. By judging the signs of the inner product, the projected results can be quantized into the final binary hash with a fixed length of R bits. In order to ensure the security, the R low-pass filtered random masks for projection are controlled by the secret key, and different keys generate distinct hash sequences. The DC component is not involved in the calculation of the inner product. The generated hash represents the salient structural feature of the image, because the projected results of the DCT coefficients on smooth masks reflect the low-frequency characteristics. This image hashing algorithm is robust to the common content-preserving manipulations and discriminates perceptually-distinct images effectively.

We take advantage of image hashing in our proposed data hiding method. The robustness and discrimination of image hashing favors the embedding and extracting of secret data, which is described in detail in the next section.

3. **Proposed Data Hiding Method.** In the proposed method, a dictionary of image blocks should be constructed before data embedding. The dictionary construction consists of two main steps, i.e., choosing representative blocks and generating perceptual similar block sets. According to the embedding secret data, the blocks in the cover image are replaced with different image blocks in the dictionary to produce the stego-image. Due to the perceptually-similarity characteristic of the blocks in the same set, data extraction

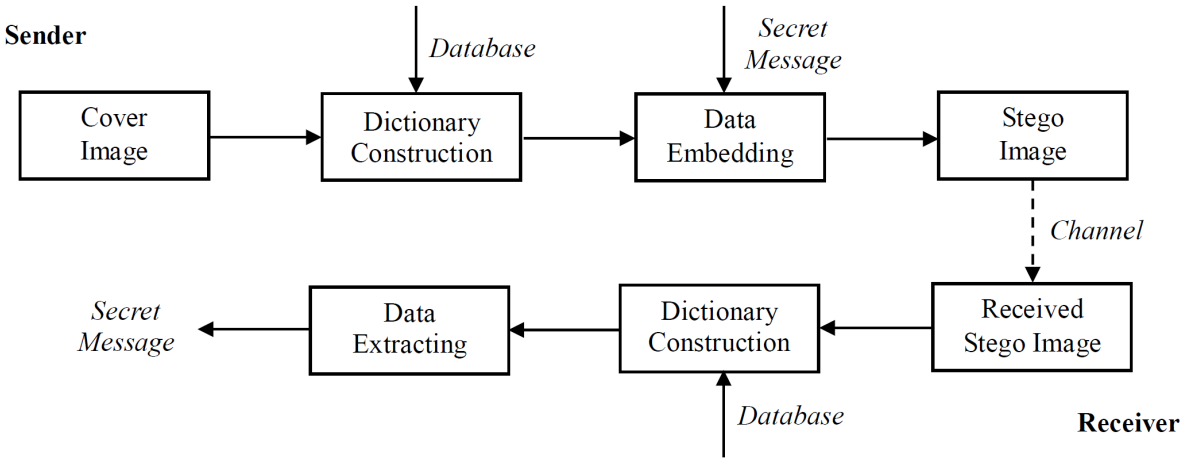

FIGURE 2. Framework of the proposed method

can be achieved by computing and comparing the hash values of the image blocks. The framework of the proposed data hiding method is shown in Figure 2. Detailed descriptions of each stage are presented below.

3.1. **Dictionary construction.** First, the cover image I is divided into non-overlapped, $n \times n$ blocks $B_1, B_2, \ldots, B_k$, where k is the number of total divided blocks. Suppose that a database exists that contains vast, distinct image blocks. For each cover image block B_i, the image block $B^{'}$ in the database with the smallest Euclidean distance d is searched as the best-matching block.

$$d = \sqrt{\sum_{x=1}^{n}\sum_{y=1}^{n}\left[B'(x,y) - B_i(x,y)\right]^2} \tag{3}$$

For color image blocks B_i and $B^{'}$, the Euclidean distances are calculated on R, G, and B channels, respectively, and added to obtain the final distance. This procedure can be conducted using the vector quantization (VQ) technique, i.e., the database can be understood as the codebook of VQ, while each image block in the database that can be transformed into a one-dimensional vector is analogized as the codeword in the codebook. Both the sender and the receiver share the image database or the VQ codebook.

After the best-matching blocks for each cover image block B_i are chosen from the database or VQ codebook, a dictionary is built using a set of perceptual, content-preserving manipulations $\mathbf{P}$, such as JPEG compression with different quality factors and Gaussian filtering with different standard deviations. Denote the chosen, best-matching block using Equation (3) as $B^{'}_{(i)}$ corresponding to each cover image block $B_i, i = 1, 2, \ldots, k$, and a set of different perceptual, content-preserving manipulations as $\mathbf{P}= \{\mathrm{P}_i, i = 1, 2, \ldots, t\}$. Therefore, a two-dimensional dictionary Φ of image blocks can be constructed:

$$\Phi_{i,j} = \mathrm{P}_i\left[B^{'}_{(j)}\right] \tag{4}$$

where $\Phi_{i,j}$ is the element of the dictionary Φ with the coordinate (i, j), and $i = 1, 2, \ldots, t$ and $j = 1, 2, \ldots, k$. Equation (4) indicates that each element $\Phi_{i,j}$ in the dictionary Φ is the processed version of $B^{'}_{(j)}$ under the content-preserving manipulation P_i. So the elements of the i^{th} row in Φ are the processed versions of different representative blocks $B^{'}_{(j)}$, $j = 1, 2, \ldots, k$ under the same manipulation P_i, while the elements of the j^{th} column

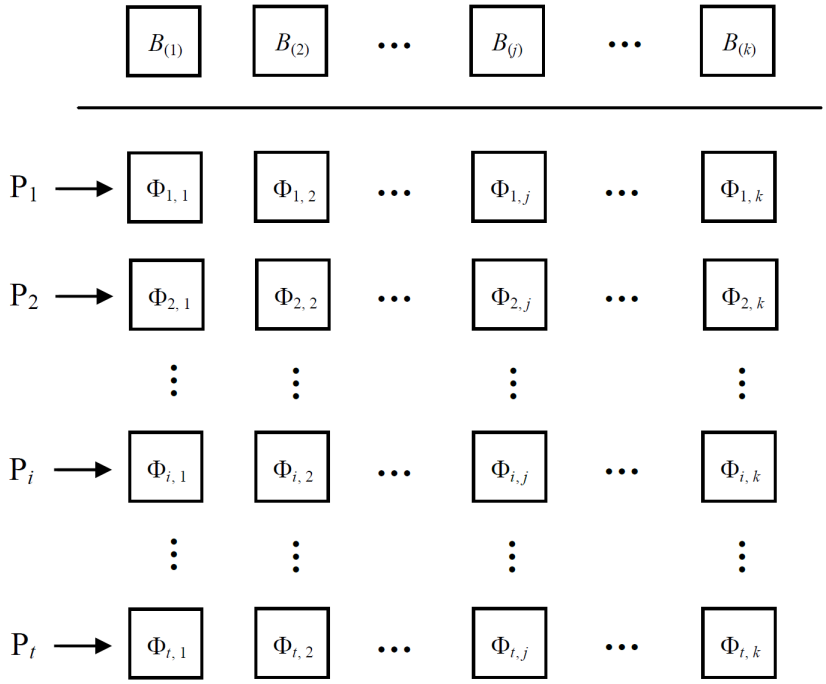

FIGURE 3. Dictionary construction

in Φ are the processed versions of the same representative block $B'_{(j)}$ under the different manipulations P_i, $i = 1, 2, \ldots, t$. According to the characteristics of image hashing introduced in Section 2, we can easily find that the image blocks in each column have the same or similar hash values, which are significantly distinct with the blocks in other columns. Figure 3 illustrates dictionary construction. Next, we present how to embed and extract the secret data using this constructed dictionary.

3.2. **Data embedding.** In the present method, we embed the secret message into the cover image block by block. Denote the secret message for hiding as $W = (w_1, w_2, \ldots, w_s)$, and each element in W is either 0 or 1. The following steps are implemented to embed the secret message to the cover image:

Step 1: W is first encrypted by a secret key to enhance security. For simplicity, suppose that s can be divided by k with no remainder. Therefore, each block of cover image hides s/k bits of secret data.

$$\frac{s}{k} = \log_2 t \tag{5}$$

Step 2: In order to hide s/k bits in each image block, the dictionary Φ mentioned in the last subsection should be constructed using $2^{s/k}$ kinds of content-preserving manipulations.

Step 3: Assign each possible binary representation of s/k bits with one kind of content-preserving manipulation P_i, $i = 1, 2, \ldots, 2^{s/k}$. Denote the secret binary bits that should be embedded into the λ^{th} block B_λ of cover image as W_λ.

$$W_\lambda = \left(w_{\lambda_1}, w_{\lambda_2}, \ldots, w_{\lambda_{\frac{s}{k}}}\right) \tag{6}$$

Step 4: The cover block B_λ is then replaced by the element $\Phi_{\kappa,\lambda}$ in dictionary Φ, which is the manipulated version of the corresponding best-matching block $B'_{(\lambda)}$ using the κ^{th}

content-preserving operation.

$$\kappa = 1 + \sum_{j=1}^{\frac{s}{k}} [w_{\lambda_j} \times 2^{j-1}] \tag{7}$$

Step 5: After all the blocks of the cover image complete the above process, the data embedding procedure finishes, and we can obtain the stego-image denoted as I_e.

Because the stego-image is composed of the content-preserving, manipulated versions of the corresponding cover-image blocks, it has similar visual quality as the cover image.

3.3. **Data extracting.** After receiving the stego-image, the receiver should conduct the following steps to extract the secret data:

Step 1: Divide the received stego-image $I_e^{'}$ into non-overlapped blocks in the same way with the embedding stage. Denote the divided blocks of stego-image as $E_1, E_2, \ldots, E_k$.

Step 2: Calculate the hash value of each stego-block E_i using the image hashing scheme. Denote the hash value of each E_i as H_i, $i = 1, 2, \ldots, k$.

Step 3: Search the database or the VQ codebook to find the best-matching image block $B_{(i)}^{''}$ for each E_i, whose hash value $H_{(i)}$ has the smallest Hamming distance with H_i.

Step 4: Construct the dictionary $\Phi^{'}$ using the searched $B_{(i)}^{''}$. The same set of content-preserving manipulations $\mathbf{P}$ is utilized in dictionary construction. The size of constructed dictionary $\Phi^{'}$ is also $t \times k$.

Step 5: Compute the Euclidean distances between the stego-block E_i with the elements of $\Phi^{'}$ in the i^{th} column. Suppose the $\kappa^{'\text{th}}$ element in the i^{th} column of $\Phi^{'}$ that has the smallest Euclidean distance with E_i is $\Phi^{'}_{\kappa^{'},i}$, which corresponds the $\kappa^{'\text{th}}$ kind of content-preserving manipulation.

Step 6: The hidden s/k secret bits in E_i can be obtained by converting the decimal $\kappa^{'} - 1$ into s/k binary bits. After all stego-blocks E_i, $i = 1, 2, \ldots, k$ finish the above steps, s bits of hidden data can be extracted.

Step 7: The original s bits of secret data can be obtained successfully after the decryption using the same key in the embedding stage.

3.4. **DCT-based image hashing.** The image hashing scheme used in *Step 2* of the extracting procedure is based on the discrete cosine transform [16]. Denote the DCT coefficients matrix of each stego-block E_i as C_i, $i = 1, 2, \ldots, k$. The size of C_i is also $n \times n$. To calculate the hash value of E_i, R low-pass, filtered, random masks with the size of $n \times n$ are generated using a secret key: $M_1, M_2, \ldots, M_R$. Therefore, the R bits hash sequence $H_i = (h_i^1, h_i^2, \ldots, h_i^R)$ for each stego-block E_i can be produced by Equation (8). Note the DC coefficient of C_i should be excluded in the summation for the performance of the image hashing scheme.

$$h_i^j = \begin{cases} 1, & \text{if } \sum_{x=1}^{n} \sum_{y=1}^{n} C_i(x,y) \cdot M_j(x,y) \geq 0 \\ 0, & \text{if } \sum_{x=1}^{n} \sum_{y=1}^{n} C_i(x,y) \cdot M_j(x,y) < 0 \end{cases} \qquad j = 1, 2, \ldots, R \tag{8}$$

Due to the robustness of the image hashing scheme, the hash values of the image blocks will remain unchanged after the content-preserving manipulations. Therefore, the best-matching image blocks $B_{(i)}^{'}$ and $B_{(i)}^{''}$, $i = 1, 2, \ldots, k$ that are found in the embedding and extracting stages must be the same. The dictionaries Φ and $\Phi^{'}$ constructed at the sender side and the receiver side, respectively, must also be identical. Because the image blocks in the database are perceptually distinct, their hashes are significantly distinct from each

other due to the anti-collision property. Based on these features, the hidden secret data can be extracted effectively using the image hashing strategy.

4. **Experimental Results and Analysis.** In the experiment, the size of the divided cover image blocks is set to 4×4, i.e., $n = 4$. The codebook of vector quantization is utilized as the image database, and each codeword corresponds to one representative image block. For efficiency and simplicity, we chose the VQ codebook, which consists of 1024 codewords that have 16 elements each, whose values all belong to [0, 255]. All the codewords are converted into two-dimensional matrices with the size of 4×4 to form the image block database. The normalized Hamming distance is used to measure the similarity between two image hash values:

$$\mathrm{D}\left(H_\alpha,\ H_\beta\right) = \frac{1}{R}\sum_{l=1}^{R}\left|\, h_\alpha^l - h_\beta^l \,\right| \tag{9}$$

where H_α and H_β denote the two vectors of hash values for two image blocks; h_α^l and h_β^l are the binary hash bits in the corresponding hash vectors, respectively; and R is the hash length that is set to 64 in the experiments. If the normalized Hamming distance between two hashes is smaller than a predetermined threshold T, we can conclude that the corresponding image blocks of the two hashes are similar perceptually. Otherwise, the two image blocks are visually distinct. By executing the Matlab 6.5 codes on a computer with Intel Core2 2.4 GHz processor and 4 GB memory under the operating system of Windows 7, the hash calculating process described in Subsection 3.4 averagely took less than 0.0129 seconds for each image sized 256×256.

The set $\mathbf{P}$ for constructing the dictionary used in experiments consists of three classes of content-preserving manipulations, i.e., JPEG compression with different quality factors, Gaussian low-pass filtering with different standard deviations, and non-linear filtering based on total variation (TV) with different iteration steps. The equation of TV-based filtering [22] is:

$$\frac{\partial u}{\partial t^*} = \mathrm{div}\left[\frac{\nabla u}{|\nabla u|}\right] + \eta(u_0 - u) \tag{10}$$

where u_0 and u are the original input and the filtered result, respectively; $\mathrm{div}(\cdot)$ is the divergence operator; η is the Lagrange multiplier; and t^* is the iteration step in numerical implementation. TV-based filtering can be utilized to remove noise and preserve salient edges simultaneously. The details of set $\mathbf{P}$ are listed in Table 1. Therefore, there is a total of 32 kinds of content-preserving manipulations used in our experiments.

Two issues should be proved to ensure the effectiveness of the proposed data hiding scheme: 1) Different image blocks in the database should have significantly distinct hashes; 2) All the versions should have similar hashes after the image block undergoes the content-preserving manipulations. The following subsections 4.1 and 4.2 demonstrate these two issues.

Table 1. Set $\mathbf{P}$ of content-preserving manipulations

Names	Descriptions	Parameters	Indices
JPEG compression	Quality factor Q_f	96, 97, ..., 100	I: 1∼5
Gaussian low-pass filtering	Standard deviation S_d	0.22, 0.23, ..., 0.32	II: 6∼16
TV-based filtering	Iteration step T_s	1, 2, ..., 16	III: 17∼32

4.1. **Discrimination of hashes.** Discrimination, also called anti-collision, means that two image blocks that are visually distinct have a very low probability of producing similar hashes. The database generated by the VQ codebook in our experiments contains 1,024 image blocks with the size of 4×4. First, we generate 1,024 hashes for these image blocks and then calculate the 523,776 normalized Hamming distances between the hash pairs of different image blocks. The histogram of the normalized Hamming distances is shown in Figure 4. Using the parameter estimation method, we find that the distribution of the normalized Hamming distance approximates a normal distribution, with its mean and standard deviation at $\mu = 0.50$ and $\sigma = 0.15$, respectively. Once the threshold T is determined, the collision probability G_c for two distinct image blocks is the probability that the normalized Hamming distance is smaller than T:

$$G_c(T) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{T} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx = \frac{1}{2}\mathrm{erfc}\left(-\frac{T-\mu}{\sqrt{2}\sigma}\right) \tag{11}$$

where $\mathrm{erfc}(\cdot)$ is the complementary error function.

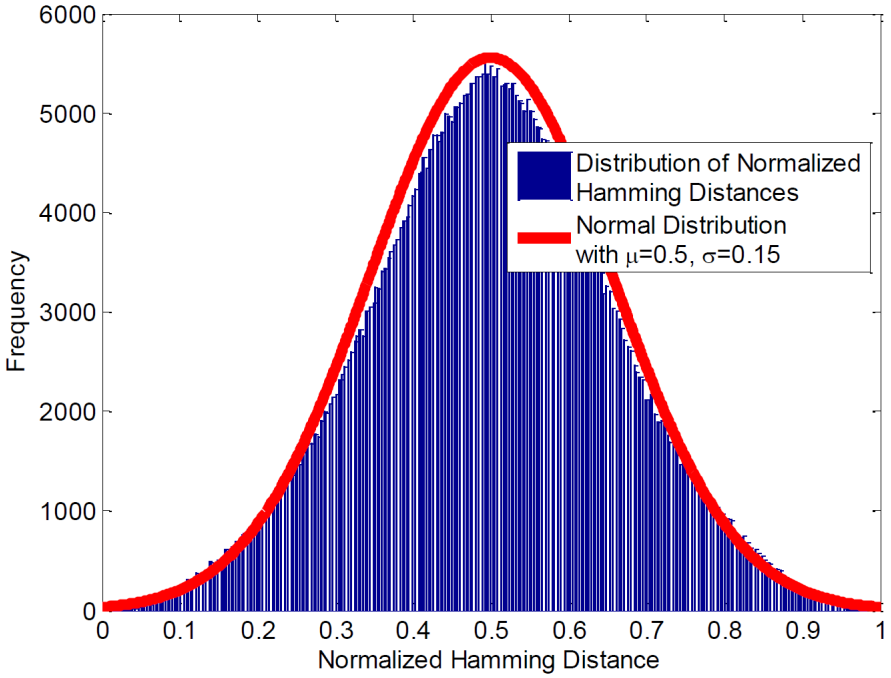

FIGURE 4. Distribution of normalized Hamming distances between the hashes of different blocks in the database

4.2. **Robustness of hashes.** We utilized the 32 kinds of content-preserving manipulations listed in Table 1 to test the robustness of the performance. In Figure 5, the abscissa is the indices of content-preserving manipulations, and the ordinate is the average value of the normalized Hamming distances between the hash pairs of the 1,024 original image blocks in our database and their manipulated versions. We find that the average Hamming distances of the present method against JPEG compression, Gaussian low-pass filtering, and TV-based filtering are all very small, which reflects satisfactory performance concerning robustness.

Figure 5 shows that the average normalized Hamming distances of all 1,024 image blocks in the database for 32 kinds of content-preserving manipulations are below 0.045. Therefore, we can set the normalized threshold $T = 0.045$, and the collision probability

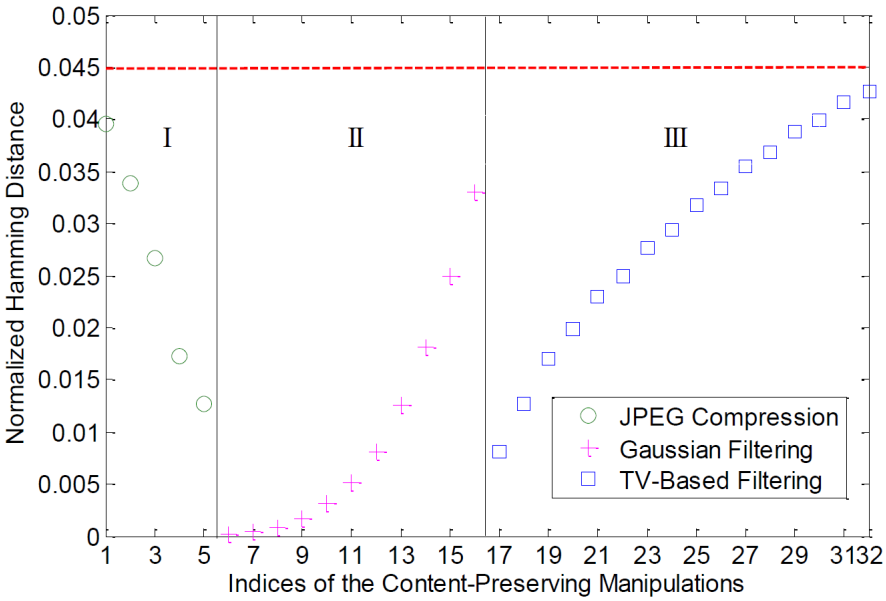

FIGURE 5. Results of robustness performance for hashes

is equal to 0.0012, as calculated by Equation (11). This demonstrates that satisfactory discrimination and robustness of hashes can be achieved simultaneously.

4.3. **Results of data hiding.** After the two issues mentioned above were demonstrated, we conducted the procedure of data hiding using our dictionary-based scheme. Experiments were carried out on a group of gray and color images of different sizes. For color images, data hiding is done on the R, G, and B channels, respectively. The obtained components are then combined to give the final stego-images.

Figure 6 shows some results of the proposed method. Figures 6(a)-(c) are the standard images with size of 256×256. We generated 20,480 binary bits randomly for data hiding by the proposed scheme. Figures 6(d)-(f) are the corresponding images for (a)-(c), which are substituted with the best-matching image blocks from the database. The stego-images with hidden secret bits are shown in Figures 6(g)-(i). Since the performance of vector quantization is irrelevant to the discussion in this paper, we directly calculated the peak signal-to-noise ratio (PSNR) of the stego-images in Figures 6(g)-(i) with respect to (d)-(f). The PSNR values of the stego-images in Figures 6(g)-(i) are 51.61 dB, 51.33 dB, and 51.90 dB, respectively. It should be noted that more secret bits can be embedded when more kinds of content-preserving manipulations are involved.

The hidden secret data can be extracted correctly by the image hashing strategy. There are two advantages of the hashing strategy: 1) Because image blocks with perceptually similar content have the similar hash, the hash value can be used as the feature for comparison and retrieval, which is more objective and accurate than using Euclidean distance directly and 2) Because the hash value is the compact representation of the original block that is just a short binary string, it is more efficient to compute the Hamming distances of the hashes rather than computing the difference of pixel values for all the pixels in two blocks.

Compared with the traditional methods, such as LSB embedding method [5], the proposed framework is more flexible, because the hiding capacity of our method is related to the size of the constructed dictionary. According to the description in Section 3, the hiding capacity of secret data for each $n \times n$ block is equal to $\log_2 t$, where t is the number

FIGURE 6. Results of data hiding. (a)-(c) show the standard images Lena, Airplane, and Pepper sized 256×256, (d)-(f) are the corresponding images of (a)-(c) that are substituted with the best-matching blocks, and (g)-(i) are the stego-images with 20,480 hidden secret bits.

of the content-preserving manipulations used for dictionary construction. For the traditional LSB embedding method, the hiding capacity for each $n \times n$ block is fixed to n^2. Therefore, the hiding capacity of the proposed method is extensible, and it is greater than that of LSB embedding method when t is set to meet $\log_2 t > n^2$.

5. **Conclusions.** In this paper, we have proposed a novel, flexible data hiding framework based on an image hashing strategy. Taking the advantage of the property that images or blocks have the same hashes after undergoing the content-preserving manipulations, a dictionary can be constructed for all the blocks of the cover image. According to the secret data hidden for each block, the corresponding version of the block in the dictionary is chosen to substitute for the cover image block, since each manipulated version has been assigned with one possible binary bits representation. On the receiver side, the same dictionary can be reconstructed using the image hashing strategy and the same set of content-preserving manipulations. The hidden binary bits can be correctly extracted by comparing the stego-image block with the blocks in the reconstructed dictionary. The

proposed dictionary-based data hiding framework can be easily implemented, and satisfies the security requirement in the sense of cryptography, because the process of constructing dictionary is controlled by the secret key.

In our further studies, we will integrate more kinds of content-preserving manipulations to further increase the data hiding capacity. The present framework also can be extended and generalized by including other multimedia hashing techniques to conduct data hiding in other carriers, e.g., carriers of audio and video. Additionally, we will further consider the property of resisting steganalysis for our scheme in order to make it more secure and practical.

Acknowledgment. This work is supported by the Shanghai Specialized Research Foundation for Excellent Young Teacher in University (slg09005).

REFERENCES

[1] National Institute of Standards & Technology, Data encryption standard (DES), *Federal Information Processing Standards Publication*, vol.46, 1977.

[2] National Institute of Standards & Technology, Announcing the advanced encryption standard (AES), *Federal Information Processing Standards Publication*, vol.197, no.l, 2001.

[3] R. L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, vol.21, no.2, pp.120-126, 1978.

[4] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding-a survey, *Proc. of the IEEE*, vol.87, no.7, pp.1062-1078, 1999.

[5] C. C. Lin, Y. H. Chen and C. C. Chang, LSB-based high-capacity data embedding scheme for digital images, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4283-4289, 2009.

[6] W. L. Tai, C. M. Yeh and C. C. Chang, Reversible data hiding based on histogram modification of pixel differences, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.19, no.6, pp.906-910, 2009.

[7] C. C. Chang and T. D. Kieu, A reversible data hiding scheme using complementary embedding strategy, *Information Sciences*, vol.180, no.16, pp.3045-3058, 2010.

[8] J. D. Lee, Y. H. Chiou and J. M. Guo, Reversible data hiding based on histogram modification of SMVQ indices, *IEEE Transactions on Information Forensics and Security*, vol.5, no.4, pp.638-648, 2010.

[9] X. P. Zhang, Efficient data hiding with plus-minus one or two, *IEEE Signal Processing Letters*, vol.17, no.7, pp.635-638, 2010.

[10] X. P. Zhang and S. Z. Wang, Efficient steganographic embedding by exploiting modification direction, *IEEE Communications Letters*, vol.10, no.113, pp.781-783, 2006.

[11] C. C. Chang, T. S. Chen and L. Z. Chung, A steganographic method based upon JPEG and quantization table modification, *Information Science*, vol.141, no.1-2, pp.123-138, 2002.

[12] Y. K. Chan, W. T. Chen, S. S. Yu, Y. A. Ho, C. S. Tsai and Y. P. Chu, A HDWT-based reversible data hiding method, *Journal of Systems and Software*, vol.82, no.3, pp.411-421, 2009.

[13] C. C. Chang, C. C. Lin and Y. S. Hu, An SVD oriented watermark embedding scheme with high qualities for the restored images, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.1-12, 2007.

[14] T. Zhang and X. J. Ping, A new approach to reliable detection of LSB steganography in natural images, *Signal Processing*, vol.83, no.10, pp.2085-2093, 2003.

[15] X. Y. Luo, D. S. Wang, P. Wang and F. L. Liu, A review on blind detection for image steganography, *Signal Processing*, vol.88, no.9, pp.2138-2157, 2008.

[16] J. Fridrich and M. Goljan, Robust hash function for digital watermarking, *Proc. of International Conf. on Information Technology: Coding and Computing*, Las Vegas, Nevada, USA, pp.173-178, 2000.

[17] A. Swaminathan, Y. N. Mao and M. Wu, Robust and secure image hashing, *IEEE Transactions on Information Forensics and Security*, vol.1, no.2, pp.215-230, 2006.

[18] C. D. Roover, C. D. Vleeschouwer, F. Lefèbvre and B. Macq, Robust video hashing based on radial projections of key frames, *IEEE Transactions on Signal Processing*, vol.53, no.10, pp.4020-4037, 2005.

[19] D. Wu, X. B. Zhou and X. M. Niu, A novel image hash algorithm resistant to print-scan, *Signal Processing*, vol.89, no.12, pp.2415-2424, 2009.

[20] S. S. Kozat, R. Venkatesan and M. K. Mihcak, Robust perceptual image hashing via matrix invariants, *Proc. of International Conf. on Image Processing*, Singapore, pp.3443-3446, 2004.

[21] V. Monga and M. K. Mhcak, Robust and secure image hashing via non-negative matrix factorizations, *IEEE Transactions on Information Forensics and Security*, vol.2, no.3, pp.376-390, 2007.

[22] T. F. Chan and J. Shen, Nontexture inpainting by curvature-driven diffusions, *Journal of Visual Communication and Image Representation*, vol.12, no.4, pp.436-449, 2001.