Multimed Tools Appl (2016) 75:3259–3280
DOI 10.1007/s11042-014-2433-0

A robust watermarking method for stereo-pair images based on unmatched block bitmap

Zhan-He Ou · Ling-Hwei Chen

Received: 5 May 2014 / Revised: 8 November 2014 / Accepted: 21 December 2014 /
Published online: 9 January 2015

Abstract A stereo-pair image contains two views of a scene called the left image and right image. This paper proposes a novel watermarking method for stereo-pair images. The proposed method is divided into three parts: watermark creation, watermark embedding, and watermark verification. Because the left and right images of a stereo pair appear to be highly similar, a robust watermark is first created based on a feature map that records the positions of the unmatched blocks between these two images. The created watermark is then embedded into the left image by swapping the AC coefficients. A feature map is first extracted from the watermarked stereo-pair image during the verification process. Subsequently, the embedded watermark is extracted from the watermarked left image and converted into an estimated feature map. Ownership is proved when the feature map and the estimated feature map are similar. Experimental results indicate that the proposed method exhibits greater robustness against malicious attacks and produces less distortion than existing methods do.

Keywords Stereo-pair images · Watermarking · Ownership proof · AC coefficient swap · Unmatched block

1 Introduction

Marketing of and research on stereo-pair images have grown rapidly because of the immersive experiences provided by the 3D content. A stereo-pair image is created by capturing a scene from slightly shifting viewpoints, and these different viewpoints create left and right images.

Two methods are used for obtaining stereo-pair images. The first method, termed depth-image-based rendering (DIBR), entails constructing a central image and a depth map. Based on the depth map, the central image is used to generate the left and right images to display the 3D content. The second method, termed stereo image recording (SIR), involves recording two captured images that can be directly displayed on a 3D monitor.

Z.-H. Ou · L.-H. Chen (✉)
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China
e-mail: lhchen@cc.nctu.edu.tw

L.-H. Chen
e-mail: id4922.cs96g@nctu.edu.tw

Copyright protection has become more crucial as applications for and research on stereo-pair images continue to grow. Kim et al. [5] proposed a watermarking method for DIBR 3D images. The central image is decomposed into nine subband pairs by using the three-level dual-tree complex wavelet transform [6]. Only four subband pairs are used to embed the watermark. Every row in each pair of embeddable subbands is used to represent one watermark bit. All coefficients in the row of one subband are quantized to represent 0, and all coefficients in the row of the other subband are quantized to represent 1. Although this method is robust against malicious attacks such as JPEG compression, noise addition, median filtering, scaling, and rotation, quantizing the wavelet coefficients reduces the image quality.

Campisi [1] proposed an object-oriented watermarking method for SIR stereo-pair images. Both the left and right images are decomposed into seven subbands by using the two-level discrete wavelet transform, and the two 2LL subbands obtained in each image are used to derive a disparity map. The right image is segmented into several objects based on the disparity map. The watermark is embedded into each 2LH, 2HL, and 2HH subband of the right image based on the position of each object. Embedding is performed by quantizing the coefficients. Although this method is robust against compression attacks such as JPEG and JPEG2000, quantizing the coefficient results in substantial distortion.

Wu et al. [12] proposed a relationship-modulation-based blind watermarking method for stereo-pair images. The left and right images are first divided into several nonoverlapping blocks, and the discrete cosine transform (DCT) is applied to each block. To embed one bit for each pair of blocks in the right and left images, an AC coefficient (*ac*) is selected, and the average of some of the previous AC coefficients (*pa*) in the same block is calculated. The sign of (*ac* – *pa*) is then obtained, and 1 is embedded if the pair of blocks exhibit different signs; otherwise, 0 is embedded. Thus, if the embedded watermark bit is 0 (1) and the pair of blocks exhibit the same (different) sign, the sign of (*ac* – *pa*) is altered by increasing or reducing the *ac* value. This method enables extracting the watermark blindly. This method is robust against JPEG compression and produces less distortion than does the method proposed by Campisi [1] because the watermark is embedded in the quantized AC coefficients. However, a malicious attack on one or both images changes the sign relationship between the left and right images, causing the watermark to be extracted incorrectly.

To increase the robustness and retain the blind verification property, this paper presents a novel method for watermarking stereo-pair images. The proposed method includes watermark creation, embedding, and verification. First, a feature map is extracted from the stereo-pair image. The feature map is then encrypted with the owner's secret key to create a watermark. Subsequently, the created watermark is embedded into the left image. Before the watermark is embedded, the left image is divided into several blocks, and one bit is embedded in each block by using a pair of quantized AC coefficients exhibiting the smallest swapping error.

Owners can prove their ownership through a verification process. The proposed verification process involves first extracting the embedded watermark from the left image and then decrypting the extracted watermark to obtain an estimated feature map (*EFM*). In addition, a feature map (*FM'*) can be extracted from the concerned stereo-pair image. The ownership of the stereo-pair image can be verified by comparing *EFM* with *FM'*. Experimental results indicate that the proposed method is robust against malicious attacks such as JPEG compression, convolution, median filtering, affine transform, rescaling, and rotation. Furthermore, this method produces less distortion than do the methods proposed by Campisi [1] and Wu et al. [12] because a watermark bit is embedded by using a pair of quantized AC coefficients that exhibit the smallest swapping error.

The rest of this paper is organized as follows. Section 2 details the proposed method. Section 3 presents experiments performed to evaluate the robustness of the proposed method. The final section presents the conclusion.

2 The proposed watermarking method

The proposed method includes watermark creation, embedding, and verification. A robust watermark is created based on a stereo-pair image, and the following subsection details the proposed method.

2.1 Watermark creation

Figure 1 shows a block diagram of the watermark creation process. First, a stereo image feature extractor is used to extract a feature map, which records the positions of the unmatched blocks between the left and right images. A robust watermark is then created based on the feature map. The stereo image feature extractor is divided into two parts: preprocessing and the feature extractor. The details are described in Sections 2.1.1 and 2.1.2.

2.1.1 Preprocessing

The stereo-pair image is preprocessed to increase the robustness of the created watermark, and Fig. 2 shows a block diagram of the preprocessing operation.

First, the left image is divided into 8×8 blocks. The DCT is then applied to each block, and the coefficients of each block are quantized. Based on the zig-zag scanning order, the first 16 quantized AC coefficients (ac_0, ac_1,…, ac_{15}) are randomly grouped into eight pairs by using a secret key. Assume the randomly grouped AC pairs are denoted as ($ac_{p(1,0)}$, $ac_{p(1,1)}$), ($ac_{p(2,0)}$,

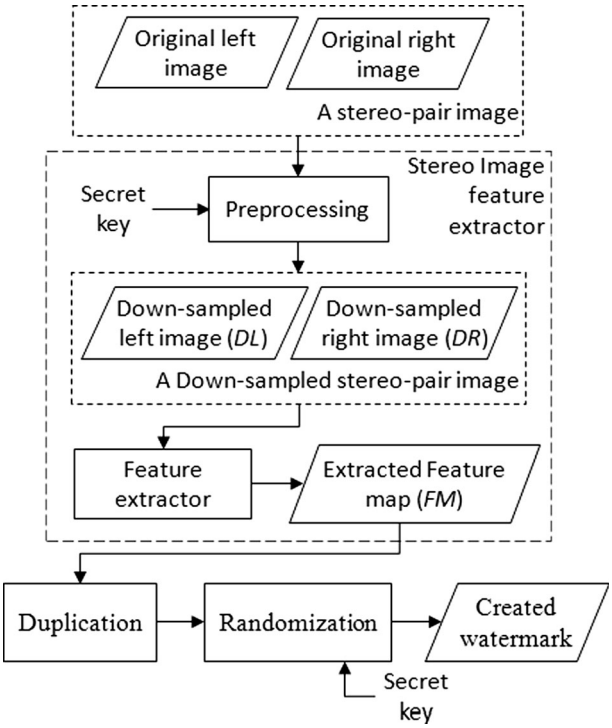

Fig. 1 Block diagram of the watermark creation process

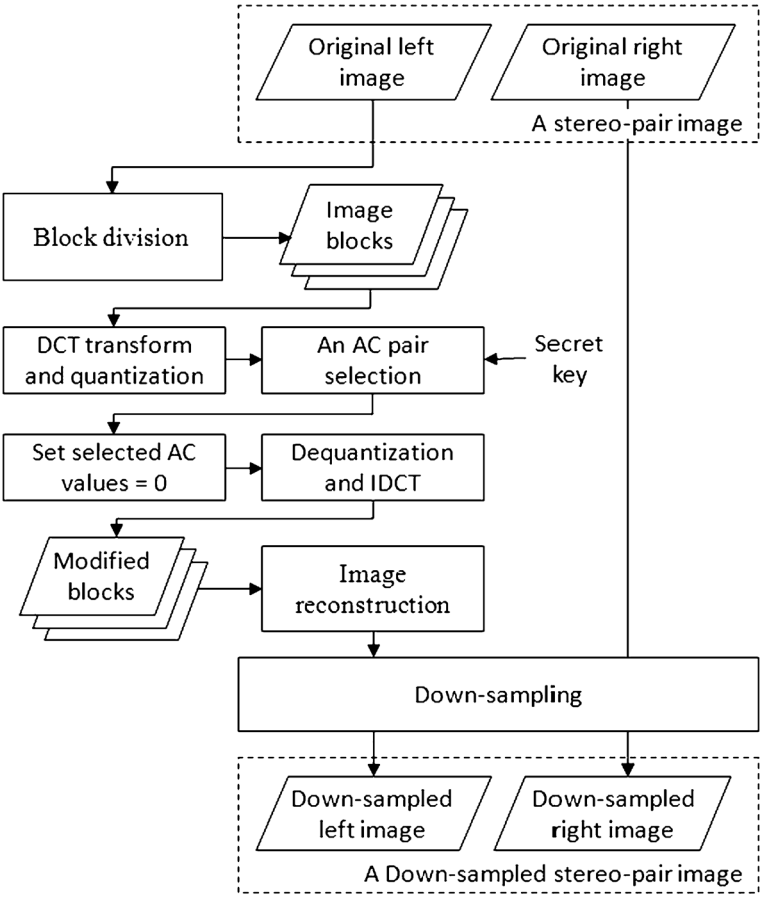

Fig. 2 Block diagram of the preprocessing operation

$ac_{p(2,1)}$),…, ($ac_{p(8,0)}$, $ac_{p(8,1)}$), where ($ac_{p(m,0)}$, $ac_{p(m,1)}$) represents the m_{th} pair of the AC coefficients, and the corresponding quantization steps are denoted as ($q_{p(1,0)}$, $q_{p(1,1)}$), ($q_{p(2,0)}$, $q_{p(2,1)}$),…, ($q_{p(8,0)}$, $q_{p(8,1)}$). One of the eight AC pairs is selected based on their swapping distortions. The swapping distortion of the m_{th} pair is defined as

$$dist(m) = \left|ac_{p(m,1)} \times q_{p(m,0)} - ac_{p(m,0)} \times q_{p(m,0)}\right| + \left|ac_{p(m,0)} \times q_{p(m,1)} - ac_{p(m,1)} \times q_{p(m,1)}\right|, \quad (1)$$

where m=1,2,…, 8. Let

$$m^* = \underset{m}{\text{argmin}}\ dist(m), \quad (2)$$

then the m^*_{th} pair with minimal swapping distortion is used to embed a watermark bit.

Figure 3 illustrates an example of the AC pair selection process. Figure 3a shows the AC coefficients in the zig-zag scanning order, Fig. 3b shows the quantization steps and Fig. 3c and d show the quantized AC coefficients and randomly grouped AC pairs, respectively. In this example, ($ac_{p(1,0)}$, $ac_{p(1,1)}$)=(ac_5, ac_6) is the first pair, and the swapping distortion $dist$(1) is 3;

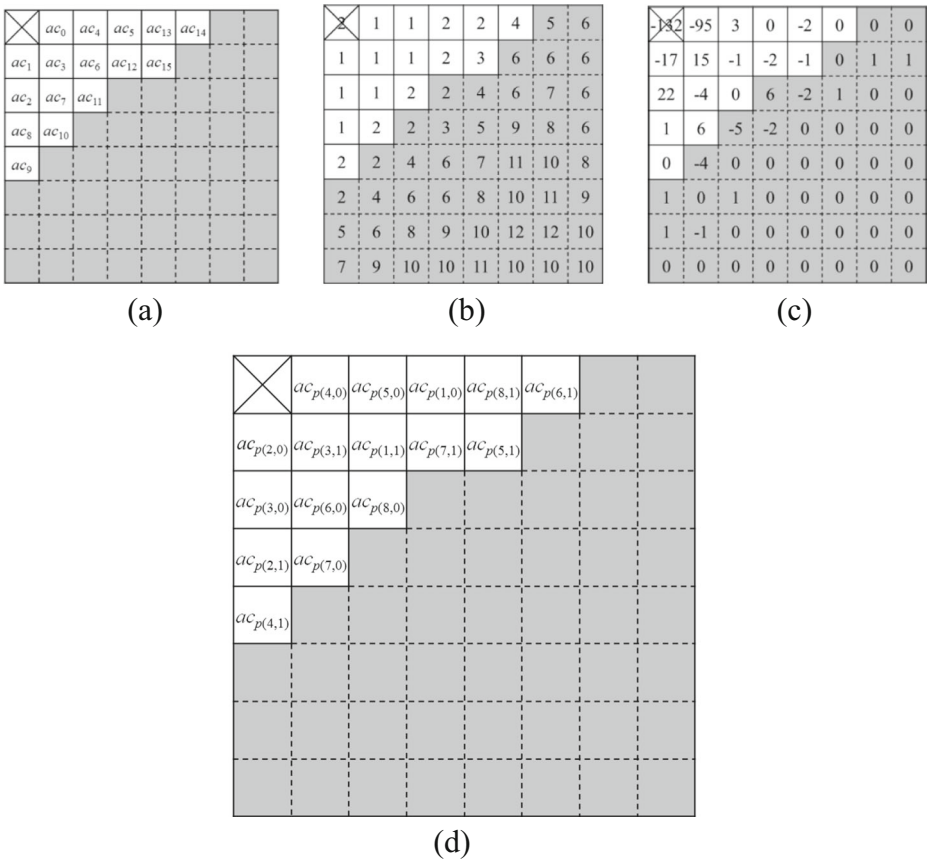

Fig. 3 Example of AC pair selection. (**a**) AC coefficients in the zig-zag scanning order. (**b**) Quantization steps. (**c**) Quantized DCT coefficients. (**d**) Eight randomly grouped AC pairs

($ac_{p(2,0)}$, $ac_{p(2,1)}$)=(ac_1, ac_8) is the second pair, and the swapping distortion *dist*(2) is 36. The swapping distortions of the remaining six pairs are 14, 285, 16, 20, 40, and 8, respectively. The first pair is used to embed the watermark because it exhibits the minimal swapping distortion. Selecting the corresponding AC coefficients to embed a watermark bit does not alter *dist*(*m*); this is explained further in Section 2.2.

Since the coefficients of the selected pair may be changed during the watermark embedding process. To ensure that both encoder and verifier can obtain the same feature, the coefficients of the selected pair will be ignored by setting to 0 before downsampling left image. That is, the AC coefficients of the selected pair are set to 0. All blocks are then dequantized and transformed to the spatial domain by using the inverse DCT. The resulting left image and original right image are downsampled to 1/4 of their original sizes (1/2 height × 1/2 width). The downsampling process can reduce spatial noise, thus facilitating the reduction of errors in the extracted feature map caused by malicious attacks.

2.1.2 Feature extractor

The feature extractor extracts a feature map from the downsampled stereo-pair image. Figure 4 shows a block diagram of the feature extraction process. The downsampled left image (*DL*)

obtained from the preprocessing operation is divided into 8×8 blocks. Let $B_{k,l}$ denote block (k, l); the first pixel (usually the top-left pixel) in $B_{k,l}$ is at position ($k\times 8$, $l\times 8$).

For each $B_{k,l}$ in the DL, a similar area matching method is applied to the downsampled right image (DR) to determine the most similar 8×8 area based on the minimal block matching error. The search area in the DR is centered at position ($k\times 8$, $l\times 8$) with size ($2r_h$, $2r_v$). The minimal block matching error ($S_{k,l}$) is calculated as follows:

$$S_{k,l}=\min_{\substack{i\in[-r_h,\,r_h)\\ j\in[-r_v,\,r_v)}}\left(\frac{1}{8}\sqrt{\sum_{x=0}^{7}\sum_{y=0}^{7}\left(D_L(8k+x,\ \ 8l+y)-D_R(8k+x+i,\ \ 8l+y+j)\right)^2}\right) \quad (3)$$

After all $S_{k,l}$ values are obtained, the distribution function (DSF) of $S_{k,l}$ can be calculated as follows:

$$DSF(t)=\frac{Number\ \ of\ \ blocks\ \ with\ \ S_{k,l}\le t}{Total\ \ number\ \ of\ \ blocks}, \quad (4)$$

where t is a threshold value and $DSF(t)$ represents the percentage of blocks with $S_{k,l}\le t$. Let t_p be the p-percentile of the DSF; that is, $DSF(t_p)=p$ / 100. A p-percentile is used as a threshold, and each $B_{k,l}$ with $S_{k,l}\ge t_p$ is regarded as an unmatched block. An unmatched block in the DL is recorded in a bitmap called the feature map (FM). In FM, 0 represents an unmatched block and 1 represents a matched block.

To increase the robustness, FM is duplicated four times (Fig. 5), yielding a duplicated feature map (DFM). The size of DFM is equivalent to the block number of the left image. A random bitmap (RBM) with the same size as DFM is generated by using a secret key, and the exclusive-or operation is applied to DFM and RBM to obtain a randomized DFM called the watermark (w).

2.1.3 Robustness of the created watermark

The robustness of the created watermark depends on the determination of the unmatched blocks. The proposed method considers $B_{k,l}$ with $S_{k,l}\ge t_p$ as an unmatched block, and the reason will be illustrated in the following paragraphs.

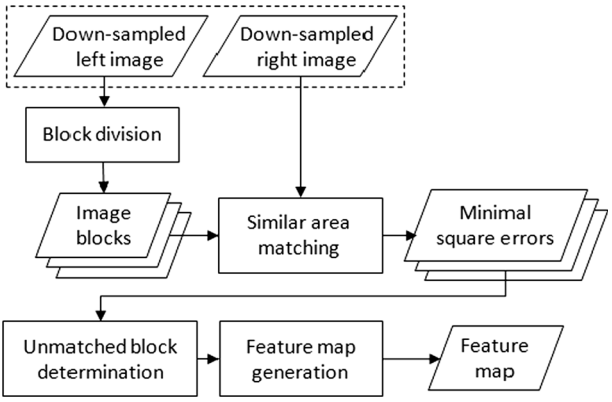

Fig. 4 Block diagram of the feature extraction process

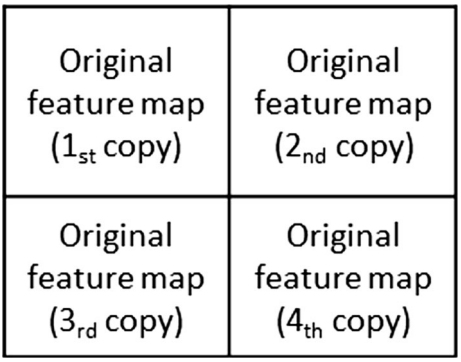

Fig. 5 Duplicated feature map

Figure 6a and b depict the left and right images, respectively, of an original stereo-pair image. All $S_{k,l}$ values for the blocks in the left image were calculated and sorted using the

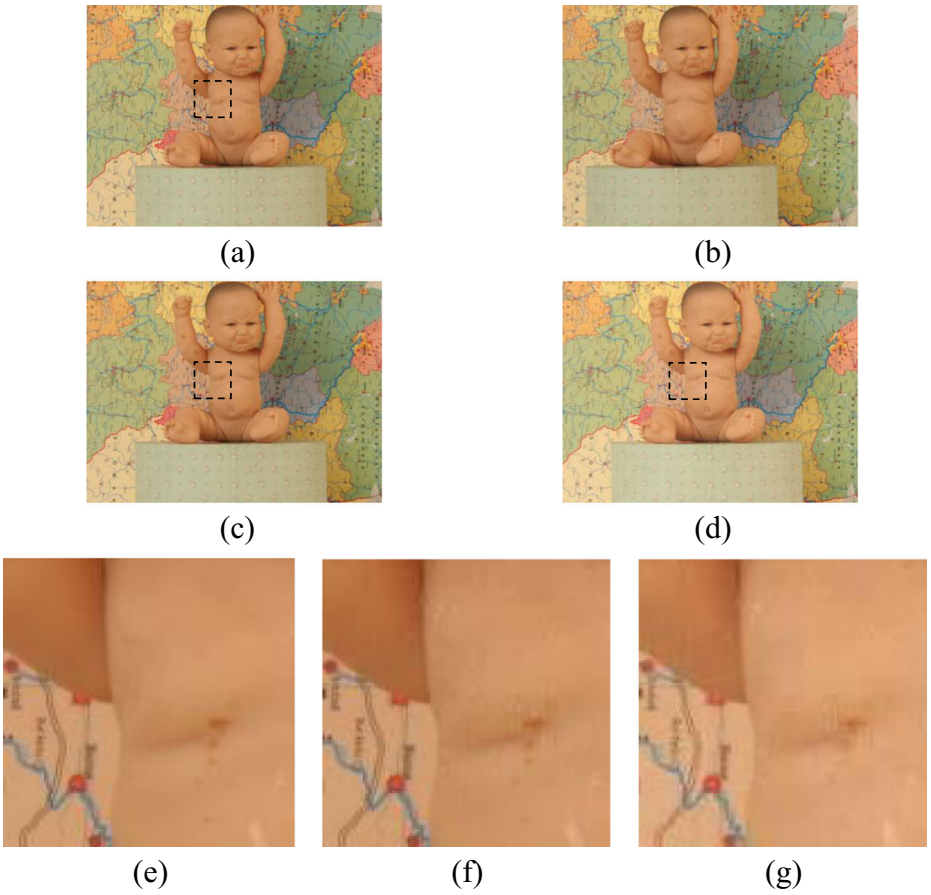

Fig. 6 Stereo-pair image with and without watermarking and attacking. (**a**) Left image. (**b**) Right image. (**c**) Watermarked left image with the 90-percentile as the threshold. (**d**) Left watermarked image attacked using JPEG compression with Q=50. (**e**) Magnified part of (**a**). (**f**) Magnified part of (**c**). (**g**) Magnified part of (**d**)

proposed method. Beside the values greater than the 90-percentile, the list of the remaining sorted $S_{k,l}$ values was divided into nine sublists, each of which comprised 10 % of the values. The list of the top 10 % of the values was divided into two sublists, each of which contained 5 % of the values. Table 1 lists part of the average and standard deviation of sublists.

Table 1 indicates that the averages and standard deviations of the top 20 % sublists (i.e., 80–100 %) were greater than 12 and 1.1, respectively. Thus, for each $B_{k,l}$ in the *DL* with $S_{k,l} \geq t_{80}$, the average difference of each pair of pixels between $B_{k,l}$ and the matched block in the *DR* was greater than 10, implying that each block in the top 20 % sublists was different from its matching block.

Before a suitable threshold value t_p is determined, the correct bit rate of the feature map extracted from an attacked watermarked stereo-pair image is defined. Let *FM* be the feature map extracted from the original stereo-pair image and *FM'* be the feature map extracted from the attacked watermarked stereo-pair image. The correct bit rate (*CBR*) is defined as

$$CBR = \frac{|FM \cap FM'|}{|FM|} \tag{5}$$

where $|FM|$ represents the bit number of *FM*, and $|FM \cap FM'|$ represents the number of bits that have the same values in *FM* and *FM'*.

In determinining a suitable threshold t_p value, various threshold values ranging from the 80-percentile to the 99-percentile were applied to 30 randomly selected stereo-pair images to generate watermarked stereo-pair images. All watermarked images (with different threshold values) were attacked using JPEG compression with a quality factor of 50, and Fig. 6 illustrates one example. Figure 6c shows the watermarked left image with a threshold of t_{90}. Figure 6d depicts the attacked result of Fig. 6c, and e–g depict the magnified parts of Fig. 6a, c, and d, respectivily.

Figure 7 shows the average *CBRs* of the 30 randomly selected images, and the *CBRs* from five selected stereo-pair test images at various threshold values. For further description, Fig. 7 indicates that the average *CBRs* of the 30 stereo-pair images are greater than 0.96. Thus, the suitable threshold is between the 80-percentile to the 99-percentile. In this study, the 90-percentile and the 95-percentile were used as the threshold in the experiments.

2.2 Watermark embedding

Figure 8 shows a block diagram of the watermark embedding process. First, the left image is divided into several 8×8 blocks. The DCT is applied to each block, and all coefficients are quantized. Based on the secret key, an AC pair is selected using the method applied in the preprocessing operation described in Subsection 2.1.1. Let the indices of the chosen pair be $(p(m^*,0), p(m^*,1))$. If the sign of $(p(m^*,0) - p(m^*,1))$ is identical to the sign of $(ac_{p(m^*,0)} - ac_{p(m^*,1)})$, 0 is embedded in the pair; otherwise, 1 is embedded in the pair. Therefore, if the representation of the embedded watermark bit is different from the representation of the pair, then $(ac_{p(m^*,0)}, ac_{p(m^*,1)})$ is swapped to fit the watermark bit.

Table 1 Part of average and standard deviation of sublists of the sorted minimal block matching errors

Sublist (%)	Average	Standard deviation
95~100	27.9	7.3
90~95	17.2	1.7
80~90	12.2	1.1
70~80	9.7	0.4
60~70	8.2	0.4
50~60	6.8	0.4

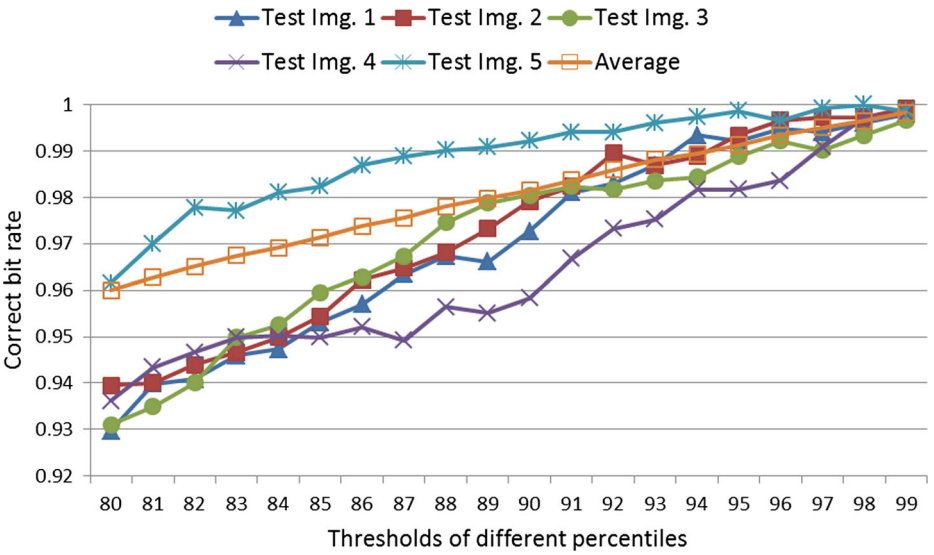

Fig. 7 Average correct bit rates of 30 stereo-pair test images and correct bit rates from five selected stereo-pair test images for the extracted feature maps (*FM'*) at various threshold values

Figure 3 illustrates an example of the AC pair selection process. The first pair (5, 6) is used to embed a watermark bit. The quantized AC coefficients (0, -1) in Fig. 3c is not swapped if the watermark bit is 1. However, if the watermark bit is 0, then the quantized AC coefficients (0, -1) are swapped, and the resulting (ac_5, ac_6) is (-1, 0).

2.3 Ownership verification

Figure 9 shows a block diagram of the proposed ownership verification procedure, which comprises three parts: the stereo image feature extractor, embedded watermark extraction, and comparison. The stereo image feature extractor, which is the same as that discussed in Section 2.1, is used to extract the feature map (*FM'*).

The embedded watermark is extracted from the watermarked left image, and converted to an estimated feature map (*EFM*), which is compared with the extracted feature map (*FM'*) in the verification process.

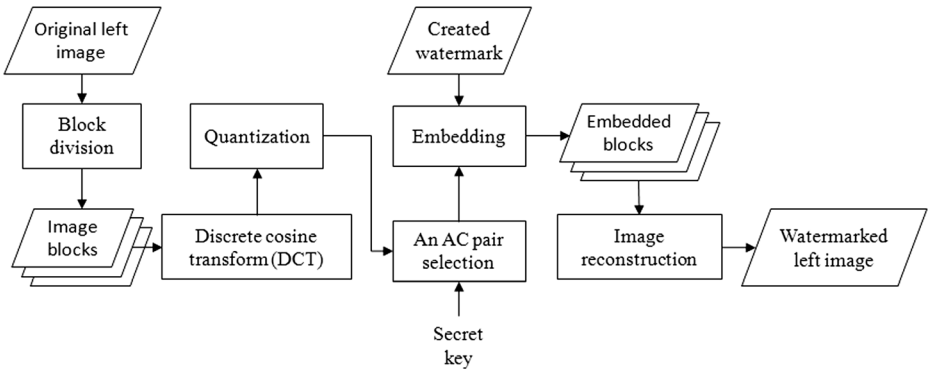

Fig. 8 Block diagram of the watermark embedding process

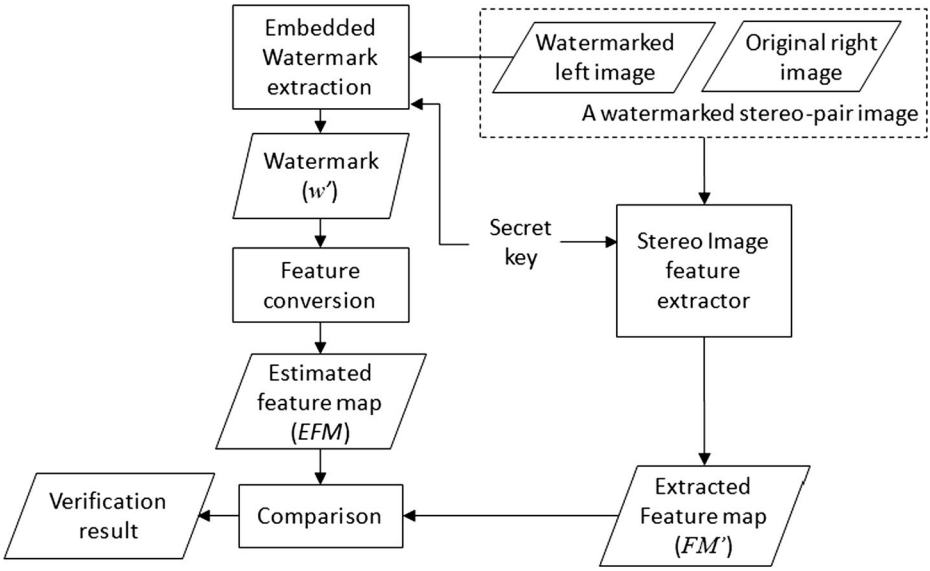

Fig. 9 Block diagram of the proposed ownership verification process

Figure 10 shows a block diagram of the embedded watermark extraction process. First, the watermarked left image is divided into several 8×8 blocks, and the DCT is applied to each block. The DCT coefficients of each block are then quantized. Subsequently, for each block, the quantized AC pair ($ac_{p(m^*,0)}$, $ac_{p(m^*,1)}$) with minimal swapping distortion is selected using the AC pair selection method described in Subsection 2.1.1. The embedded watermark bit is extracted by comparing the signs of ($p(m^*,0)$ - $p(m^*,1)$) and ($ac_{p(m^*,0)}$ - $ac_{p(m^*,1)}$). The embedded watermark (w') is formed by grouping all extracted watermark bits.

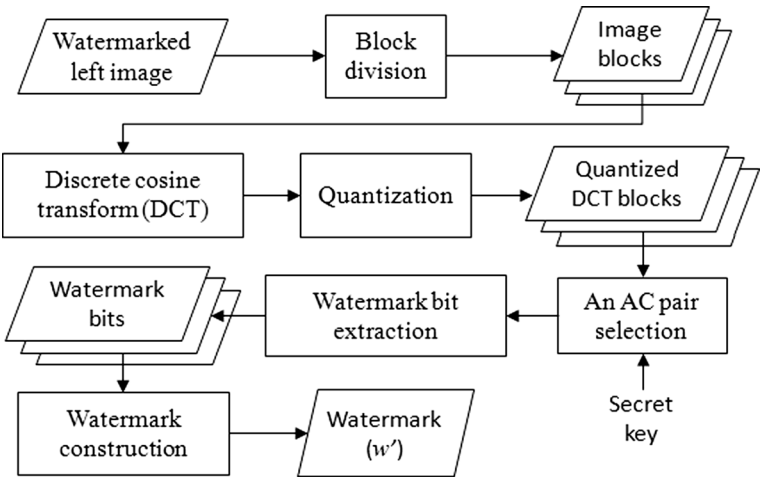

Fig. 10 Block diagram of the proposed watermark extraction process

Feature conversion is conducted after *w'* is obtained. A random bitmap (*RBM*) is generated by using the secret key, and an embedded duplicated feature map (*EDFM*) is obtained by applying the exclusive-or operation to *w'* and *RBM*. Because the *EDFM* is 4 times the size of the embedded feature map, the embedded feature map can be estimated by applying a voting scheme to the four duplicated feature bitmaps. Let $b_i^c \in \{0,1\}$ represent the i_{th} bit in the c_{th} copy, where $c \in \{1,2,3,4\}$. According to the voting scheme, the i_{th} element in the estimated embedded feature map $EFM(i)$ can be set as

$$EFM(i) = \begin{cases} 0 & if \sum_{c=1}^{4} b_i^c < 2, \\ 1 & if \sum_{c=1}^{4} b_i^c > 2, \\ -1 & if \sum_{c=1}^{4} b_i^c = 2. \end{cases} \tag{6}$$

The error rate between *FM'* and *EFM* is calculated to prove ownership. The error rate is defined as

$$ErrorRate = \frac{DifferentBits}{TotalBits - TieBits} \tag{7}$$

where *DifferentBits* is the number of bits with $EFM(i) \neq -1$ and $EFM(i) \neq FM'(i)$, *TotalBits* is the size of the *FM'*, and *TieBits* is the number of bits with $EFM(i) = -1$. Ownership is proved when the error rate is less than a predefined threshold value.

2.4 Example

This subsection presents an example illustrating the entire watermarking method. First, the pair of AC coefficients with the smallest swapping error on each block in the left image is selected and set to 0 based on the owner's secret key. Both the left and right images are then downsampled to 1/4 of their original sizes, and they are used to calculate the *FM* (Fig. 11c).

FM is then duplicated four times and encrypted by using the owner's secret key to create the watermark (Fig. 11d and e). The watermark is then embedded into the left image (Fig. 11f).

Two feature maps are first extracted when ownership must be verified. The first map (*FM'*, Fig. 12a) representing the original feature map is extracted from the downsampled stereo pair illustrated in Fig. 11f and b by using the stereo image feature extractor. The second map (*EFM*, Fig. 12b) representing the estimated map is directly extracted from the watermarked left image by using the watermark extractor. Subsequently, *FM'* and *EFM* are compared to calculate the error rate and verify ownership.

3 Experimental results

A total of 150 stereo image pairs were obtained from four sets. The first set contained 38 image pairs from the database established by Scharstein et al. [4, 9–11]. The remaining three sets were produced in this study. The second set contained 27 nature image pairs captured using a Fujifilm FinePix Real 3D W1 camera. The third set contained 70 nature image pairs captured

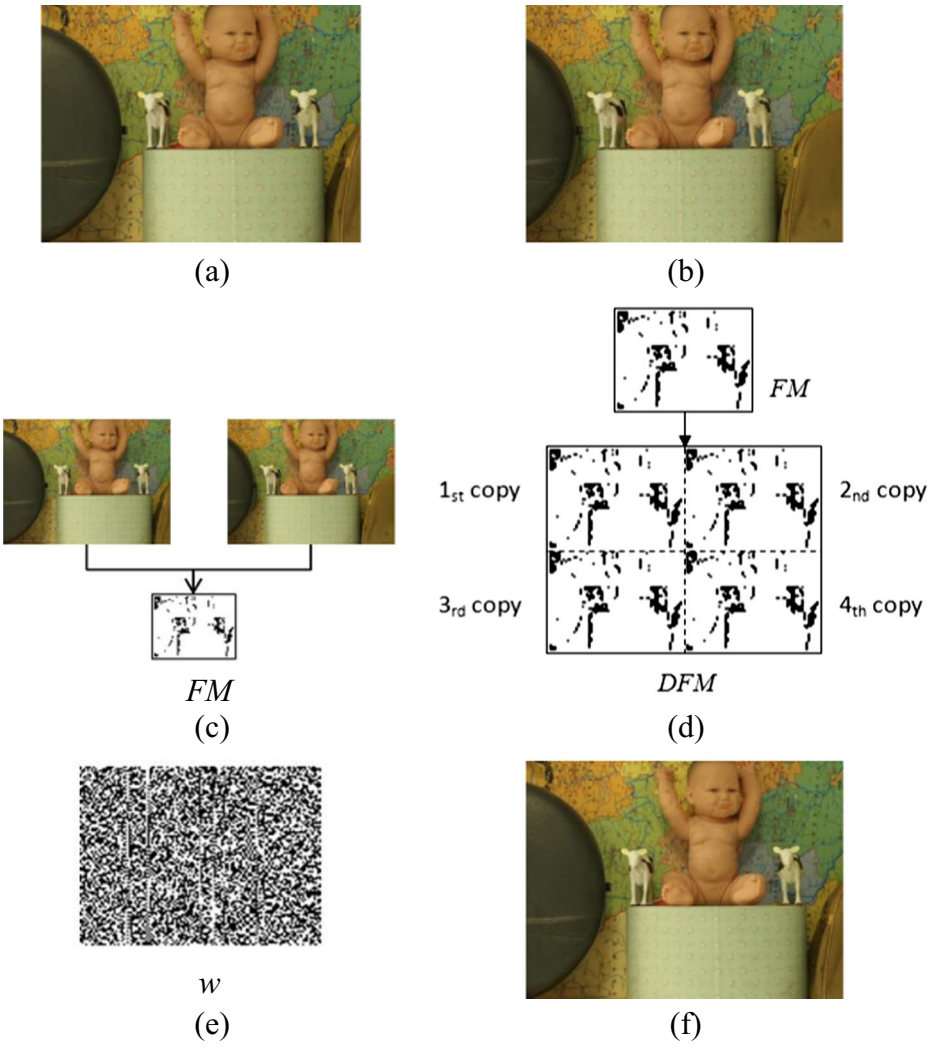

Fig. 11 Example of the proposed watermarking method. (**a**) Original left image. (**b**) Original right image. (**c**) Extracted feature map (*FM*). (**d**) Duplicated feature map (*DFM*). (**e**) Watermark (*w*). (**f**) Watermarked left image

using an HTC Evo 3D camera. The final set contained 15 nature image pairs captured using a Nintendo 3DS camera. Figure 13 illustrates image pairs from the various image sets.

Attacks from the Stirmark benchmark program [7, 8] were applied to evaluate the robustness of the proposed method. These attacks included convolution filtering, JPEG compression, median filtering, affine transform, rescaling, rotation, and cropping. Table 2 shows the corresponding parameters. The 3×3 convolution filter mask coefficients [2] are defined as

$$\begin{bmatrix} C_a & C_b & C_c \\ C_d & C_e & C_f \\ C_g & C_h & C_i \end{bmatrix}, \tag{8}$$

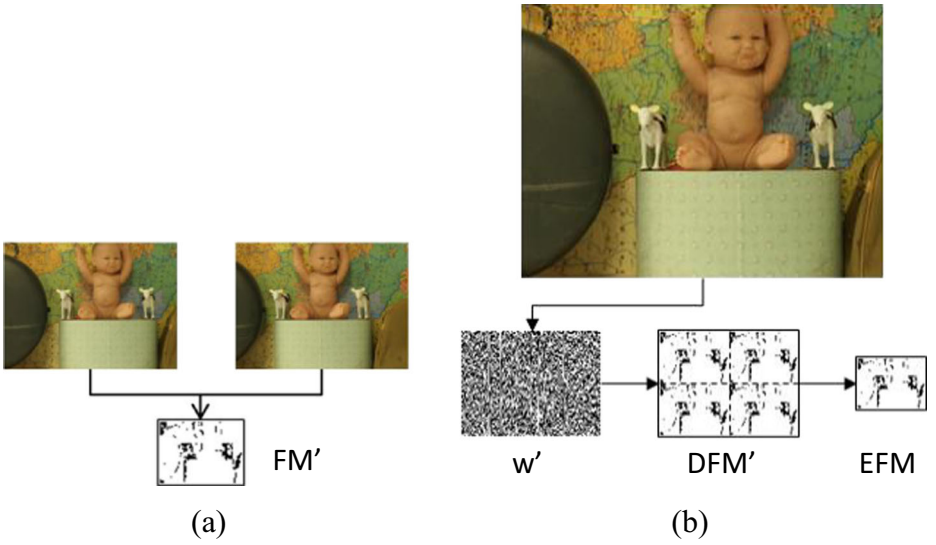

Fig. 12 Example of the proposed verification method. (**a**) Extracted feature map (*FM'*). (**b**) Estimated feature map (*EFM*)

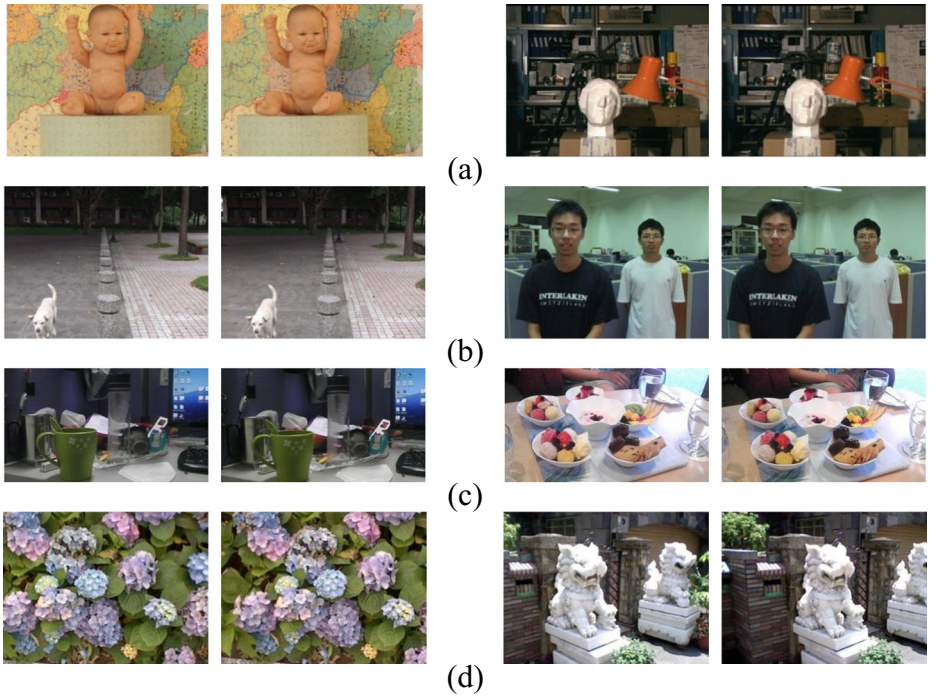

Fig. 13 Image pairs from the various sets. (**a**) From the database established by Scharstein et al. (**b**) Captured using a Fujifilm FinePix Real 3D W1 camera. (**c**) Captured using an HTC Evo 3D camera. (**d**) Captured using a Nintendo 3DS camera

Table 2 Attacks with various testing parameters

Attacks	Different testing parameters
3×3 Convolution Filtering	Test1: C_a=1/9, C_b=2/9, C_c=1/9, C_d=2/9, C_e=4/9, C_f=2/9, C_g=1/9, C_h=2/9, C_i=1/9 Test2: C_a=0, C_b=-1/9, C_c=0, C_d=-1/9, C_e=5/9, C_f=-1/9, C_g=0, C_h=-1/9, C_i=0
JPEG compression	Quality factor=30, 50, 70, 90
Median Filtering	Mask size=3×3, 5×5
Affine Transform	Test1: A_a=1, A_b=0, A_c=0.01, A_d=1 Test2: A_a=1, A_b=0, A_c=0.05, A_d=1 Test3: A_a=1, A_b=0.01, A_c=0.01, A_d=1 Test4: A_a=1.013, A_b=0.008, A_c=0.011, A_d=1.008
Rescaling	Ratio (on each side)=0.9, 1.1, 1.5, 2
Rotation	Degree=1°, 5°, 30 °, 45 °
Cropping	Retain ratio (on each side)=0.71, 0.78, 0.84, 0.9, 0.95

and the affine transform matrix coefficients [3] are defined as

$$\begin{bmatrix} A_a & A_b & 0 \\ A_c & A_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The attacks were classified into three groups. The first group comprised convolution filtering, JPEG compression, and median filtering. This group can be verified blindly, meaning that verification does not require extra information. The second group comprised affine transform, rescaling, and rotation. The size of the original watermarked image is required to verify the images attacked using the operations in this group. The third group contained cropping. The original *FM* is required to verify the images attacked using cropping. In the experiments, each attack was applied to both the left and right images of each stereo pair. Figure 14 depicts selected attacked left images.

3.1 Errors in feature maps

The error rates between *FM* and *FM'*, which were extracted from the original images and the attacked watermarked images, respectively, were calculated. The error rate was defined as

$$ErrorRate\,1 = \frac{DifferentFeatureBits}{TotalBits}, \tag{10}$$

where *Different* Feature *Bits* is the number of bits with $FM(i) \neq FM'(i)$. Figure 15 shows the error rates caused by various attacks. The Y axis represents the percentage of images, and the X axis represents the *ErrorRate*1. For example, there are 120 images with *ErrorRate*1=0 after the attack of JPEG with quality factor 50, as shown in Fig. 15a, then the percentage of images is 80 % (120/150).

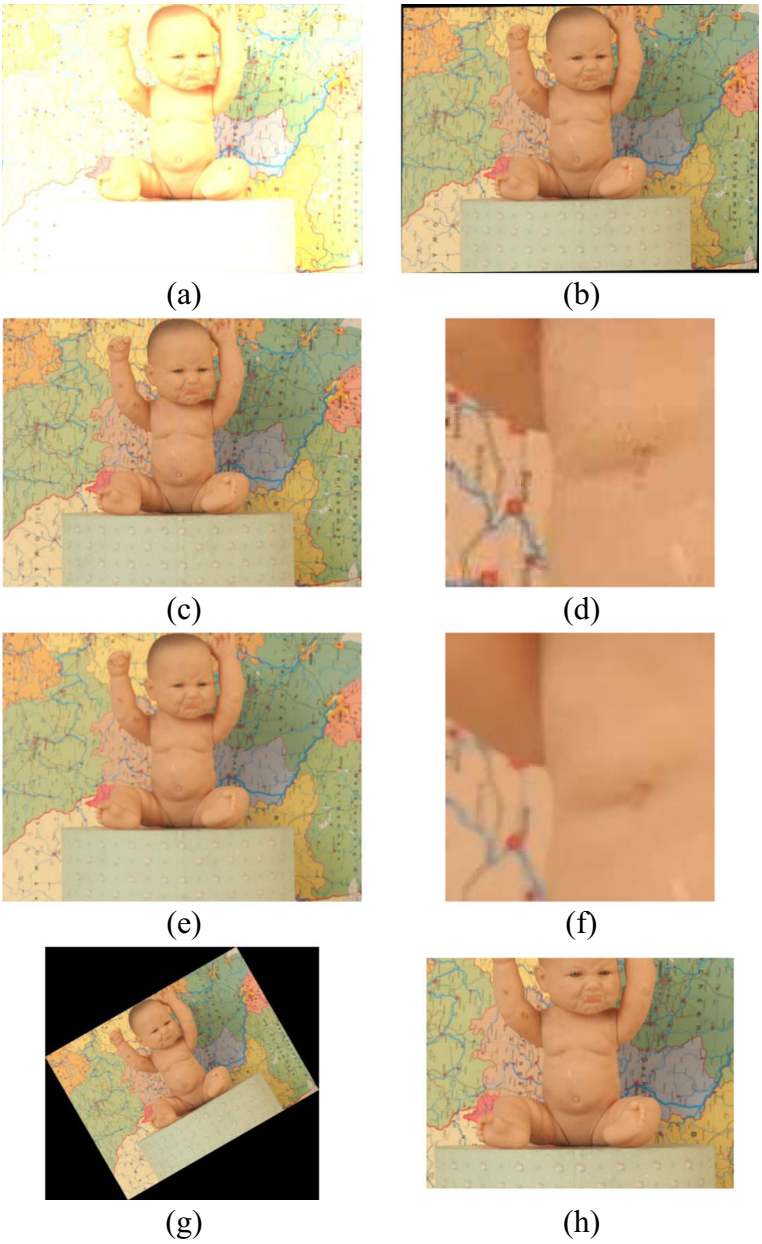

Fig. 14 Selected attacked results. (**a**) Convolution Filtering Test 1. (**b**) Affine Transform Test 4. (**c**) JPEG compression with quality factor 30. (**d**) Magnified part of (**b**). (**e**) 5×5 median filtering. (**f**) Magnified part of (**e**). (**g**) Rotation at 30°. (**h**) Cropping with retain ratio=0.71

Figure 15 indicates that only images attacked using convolution exhibited error rates greater than 1 % and less than 10 %, and the error rates of the other attacked images were less than 1 %. These results demonstrated the robustness of the proposed feature map extractor against most malicious attacks.

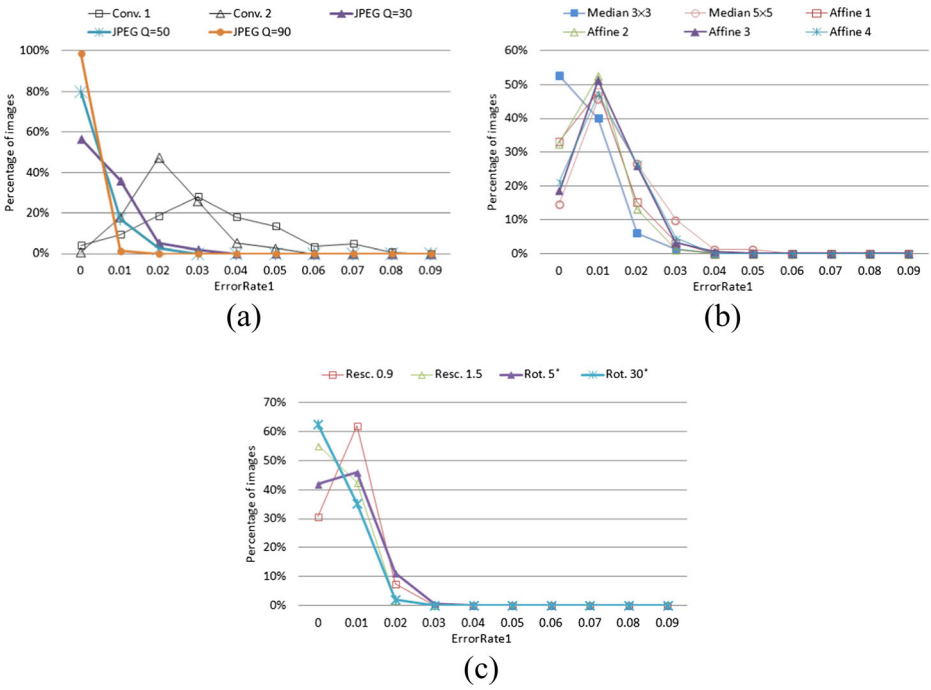

Fig. 15 Rates of error in the extracted feature maps caused by various attacks. (**a**) Convolution and JPEG compression attacks. (**b**) Median filtering and affine transform attacks. (**c**) Rescaling and rotation attacks

3.2 Errors in estimated feature maps

The errors in the estimated feature maps were caused by the loss of embedded information after the attacks. This type of error is due to the difference between *EFM* and *FM,* and the rate was defined as

$$ErrorRate2 = \frac{DiffEstimatedBits}{TotalBits - TieBits}, \tag{11}$$

where *DiffEstimatedBits* is the number of bits with $EFM(i) \neq -1$ and $EFM(i) \neq FM(i)$. Figure 16 shows the rates of error in *EFM* caused by various attacks. The Average *ErrorRate*1 and *ErrorRate*2 are also calculated to demonstrate the robustness, as shown in Table 3.

The rates of error caused by JPEG compression attacks (Fig. 16a) decreased with an increase in the quality factor. Convolution and JPEG compression attacks with a quality factor of 30 or 50 produced similar error rates. Figure 16b and c indicate that some error rates were close to 0.5, resulting in failure of the final result verification.

3.3 Verification results and comparison with other methods

A verification error rate less than 0.4 enables preserving the watermark because a random guess results in an error rate of approximately 0.5. A threshold value of 0.35 was used to compare the methods proposed by Campisi [1] and Wu et al. [12].

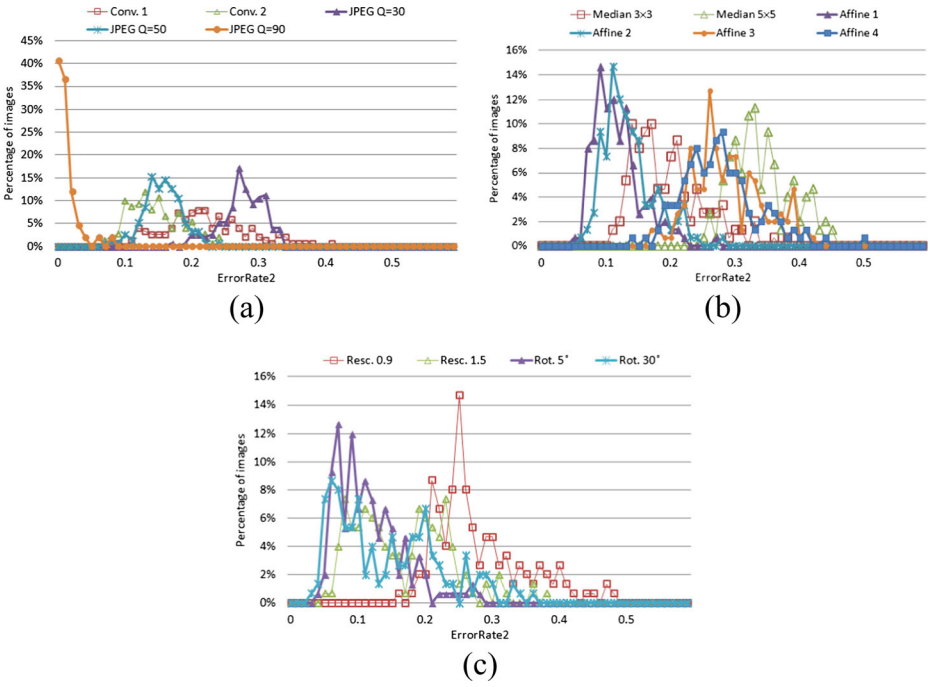

Fig. 16 Rates of error in estimated feature maps caused by various attacks. (**a**) Convolution and JPEG compression attacks. (**b**) Median filtering and affine transform attacks. (**c**) Rescaling and rotation attacks

Verification fails when an error rate is greater than this threshold. Based on this assumption, Tables 4 and 5 illustrate the percentages of images for which verification failed after they underwent various attacks.

In the proposed method, the 95-percentile and 90-percentile are used as thresholds for creating the feature map. All methods are robust against the cropping attack; therefore, the results of this attack are not described.

Table 4 lists the results of the attacks belonging to the first group; watermarks were extracted from the attacked images blindly. Most verifications performed using the proposed method were successful after the convolution attacks; however, the methods proposed by Campisi [1] and Wu et al. [12] almost failed in verifications.

The methods proposed by Campisi [1] and Wu et al. [12] were robust against JPEG compression attacks with quality factors>40 and>90, respectively. The proposed method was robust against a JPEG compression attack with a quality factor≥ 30. The proposed method was robust against a 3×3 median filter attack, but the methods proposed by Campisi [1] and Wu et al. [12] did not withstand median filter attacks.

Table 5 lists the results of the attacks belonging to the second group, for which the size of the original image is required for the verification process, and indicates that the method proposed by Wu et al could not withstand the attacks in this group. The method proposed by Campisi [1] withstood only rotation attacks with a rotation degree<1; however, the proposed method withstood more attacks.

Table 3 Average *ErrorRate*1 and *ErrorRate*2 of 150 images after various attacks

Attack Methods	Parameters	Average *ErrorRate*1	Average *ErrorRate*2
Convolution Filtering	Test 1	3.78 %	22.61 %
	Test 2	2.78 %	14.95 %
JPEG	Q=30	1.04 %	27.98 %
	Q=50	0.73 %	16.01 %
	Q=70	0.54 %	8.41 %
	Q=90	0.31 %	1.57 %
MEDIAN Filtering	Mask 3×3	1.09 %	20.03 %
	Mask 5×5	1.9 %	34.56 %
Affine Transform	Test 1	1.4 %	12.16 %
	Test 2	1.32 %	13.98 %
	Test 3	1.62 %	29.19 %
	Test 8	1.66 %	27.96 %
Rescaling	Ratio=0.9	1.26 %	28.44 %
	Ratio=1.1	1.02 %	20.86 %
	Ratio=1.5	0.97 %	17.15 %
	Ratio=2	0.7 %	6.36 %
Rotation	Degree=1	1.4 %	7.25 %
	Degree=5	1.2 %	12.1 %
	Degree=30	0.96 %	14.86 %
	Degree=45	1.5 %	33.11 %

3.4 Quality of the watermarked images

In this study, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) were used to evaluate the image quality after the watermark had been embedded.

The PSNR is defined as

$$PSNR = 10 \times \log_{10}\left(\frac{W \times H \times 255^2}{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}[I(i,j)-I'(i,j)]^2}\right), \quad (12)$$

Table 4 Percentages of 150 images for which verification failed after various attacks (Group 1)

Attack Methods	Parameters	Proposed Method (95-percentile)	Proposed Method (90-percentile)	Method proposed by Campisi [1]	Method Proposed by Wu et al. [12]
Convolution Filtering	Test 1	6.67 %	8.67 %	16 %	100 %
	Test 2	0 %	0 %	100 %	85.33 %
JPEG	Q=30	0 %	0 %	9.33 %	98 %
	Q=50	0 %	0 %	0 %	81.33 %
	Q=70	0 %	0 %	0 %	50 %
	Q=90	0 %	0 %	0 %	4 %
MEDIAN Filtering	Mask 3×3	1.33 %	2 %	100 %	100 %
	Mask 5×5	44.67 %	46.67 %	100 %	100 %

Table 5 Percentages of 150 images for which verification failed after various attacks (Group 2)

Attack Methods	Parameters	Proposed Method (95-percentile)	Proposed Method (90-percentile)	Method proposed by Campisi [1]	Method Proposed by Wu et al. [12]
Affine Transform	Test 1	0 %	0 %	12.67 %	100 %
	Test 2	0 %	0 %	12 %	100 %
	Test 3	17.33 %	18.67 %	98.67 %	100 %
	Test 8	14 %	15.33 %	85.33 %	100 %
Rescaling	Ratio=0.9	18 %	18 %	99.33 %	100 %
	Ratio=1.1	0.67 %	2 %	29.33 %	100 %
	Ratio=1.5	2 %	2 %	30.67 %	100 %
	Ratio=2	0 %	0 %	0.67 %	100 %
Rotation	Degree=1	0 %	0 %	0 %	100 %
	Degree=5	0 %	0 %	3.33 %	100 %
	Degree=30	1.33 %	2 %	38.67 %	100 %
	Degree=45	32 %	36 %	100 %	100 %

where $I(i,j)$ and $I'(i,j)$ are the gray values of pixel (i,j) in the original left image and the watermarked left image, and W and H are the image width and height, respectively. Figure 17 illustrates the PSNR results of images watermarked using different embedding methods, indicating that the proposed method produces higher PSNR values than do those proposed by Campisi [1] and Wu et al. [12].

The SSIM is defined as

$$SSIM(I, I') = \frac{(2\mu_I\mu_{I'} + c_1)(2\sigma_{I,I'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)}, \tag{13}$$

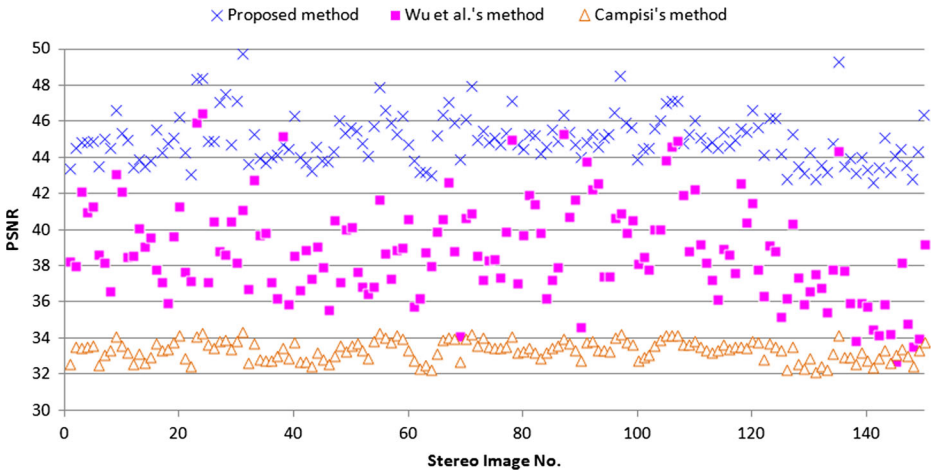

Fig. 17 PSNR results of various embedding methods

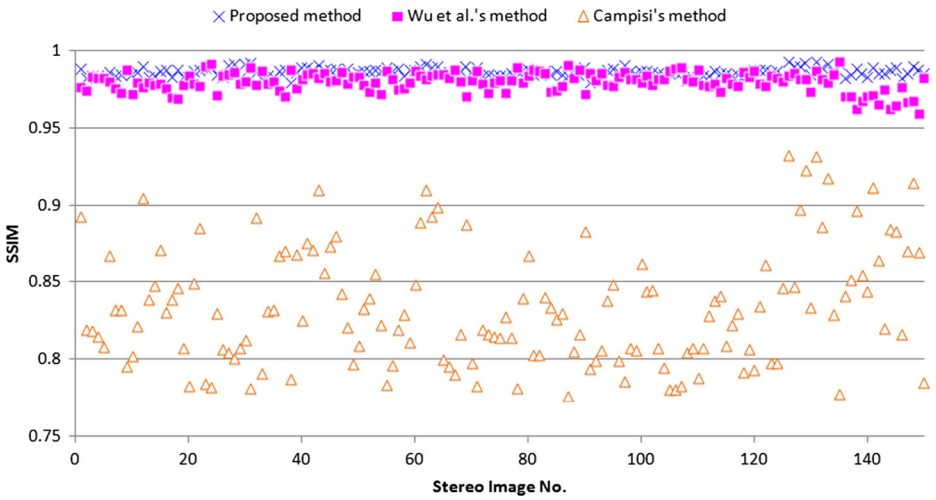

Fig. 18 SSIM results of various embedding methods

where μ_I and $\mu_{I'}$ are the averages of image I and I', respectively; σ_I and $\sigma_{I'}$ are the variances of image I and I', respectively; $\sigma_{I,I'}$ is the covariance of I and I'; and $c_1=(0.01\times255)^2$ and $c_2=(0.03\times255)^2$ are two constant values. Figure 18 shows the SSIM results of images watermarked using different embedding methods, indicating that the SSIM results of the proposed method and that proposed by Wu et al [12] were similar; however, most of the images watermarked using the proposed method exhibited higher SSIM values.

3.5 Computational analysis

In the proposed method, the feature extraction is a time consuming process due to that a full search is applied in block matching. This process needs $\frac{W\times H\times r_h\times r_v}{4}$ block matching. To demonstrate the feasibility of the proposed method, a program was written in C under the Microsoft Visual Studio 2010 environment, using an Intel i5-2500, 3.30 GHz personal computer with 8.0 GB memory. Assume that the image size is 1024×768 and r_v=3, the computational time under different r_h is shown in Table 6.

Table 6 The computational time under various r_h

		Embedding			Verification		
r_h		32	64	128	32	64	128
Computational time (second)	Feature extraction	1.34	2.46	4.61	1.34	2.46	4.59
	The whole process	3.34	4.43	6.65	3.57	4.71	6.89

According to this table, even the search area covers half of the downsampled image, the proposed method needs only few seconds to do embedding and verification.

4 Conclusion

This paper proposes a robust watermarking method for protecting copyrights of stereo-pair images. A feature map is extracted based on the similar properties of stereo image pairs, and a watermark is created and embedded in the left image. In addition, a verification process is conducted to ensure that the copyright is protected. The feature map records the positions of the unmatched blocks and is robust against malicious attacks. Furthermore, the embedding method entails selecting from the randomly grouped AC pairs the pair with the smallest swapping distortion to embed a watermark bit. This process increases the PSNR and SSIM values. The experimental results indicate that the proposed method is robust against various malicious attacks such as JPEG compression, filtering, affine transform, rescaling, rotation, and cropping. The experimental results also indicate that the proposed method protects stereo images and minimizes distortion more efficiently than other stereo image watermarking methods do.

Acknowledgments This work is supported in part by National Science Council of Republic of China under grant NSC-103-2221-E-009-121-MY2

References

1. Campisi P (2008) Object-oriented stereo-image digital watermarking. J Electron Imaging 17(4):043024
2. Gonzalez RC and Woods RE (2008) "Digital image processing," *third edition*, Pearson Prentice Hall, pp.144-152
3. Gonzalez RC and Woods RE (2008) "Digital image processing," *third edition*, Pearson Prentice Hall, pp.85-92
4. Hirschmüller H and Scharstein D (2007) "Evaluation of cost functions for stereo matching," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*
5. Kim HD, Lee JW, Oh TW, Lee HK (2012) Robust DT-CWT watermarking for DIBR 3D images. IEEE Trans Broadcast 58(4):533–543
6. Kingsbury NG (1998) "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters," *Proc. Eighth IEEE DSP Workshop*
7. Petitcolas F (2000) "Watermarking schemes evaluation," *IEEE Signal Processing Magazine*, pp. 58-64
8. Petitcolas F, Anderson R and Kuhn M (1998) "Attacks on copyright marking systems," *Proc. Int. workshop on Information Hiding*, pp. 218-238
9. Scharstein D and Pal C (2007) "Learning conditional random fields for stereo," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*
10. Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vis 47(1):7–42
11. Scharstein D and Szeliski R (2003) "High-accuracy stereo depth maps using structured light," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. I-195-I-202
12. Wu AH, Yu M, Peng ZJ, Shao F and Jiang GY (2011) "Relationship modulation based blind stereoscopic image watermarking algorithm for 3D media," *Proc. IEEE Int. Conf. on Internet Technology and Applications*, pp. 1-4

Zhan-He Ou received the B.S. and M.S. degrees in computer science and information engineering from Ming Chuan University, Taiwan, in 2005 and 2007. He is currently pursuing the Ph.D. degree at the Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include image processing and information security.

Ling-Hwei Chen received M.S. degree in Applied Mathematics from National Tsing Hua University, Taiwan in 1977 and Ph.D. degree in Computer Engineering from National Chiao Tung University, Taiwan in 1987. From 1977 to 1979 she worked in Chung-Shan Institute of Science and Technology. From 1979 to 1981 she worked in Electronic Research and Service Organization, Industry Technology Research Institute. From 1981 to 1983 she worked in Institute of Information Industry. She is now a Professor of the Department of Computer Science at National Chiao Tung University. Her current research interests include pattern recognition, Multimedia compression, content-based retrieval and Multimedia Steganography.