

Asymmetric self-recovery oriented stereo image watermarking method for three dimensional video system

Ting Luo^{1,2} · Gangyi Jiang^{1,3} · Mei Yu^{1,3} · Haiyong Xu^{1,2}

Received: 4 November 2014 / Accepted: 20 July 2015 / Published online: 25 August 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract To authenticate integrity of the stereo image for three dimensional video systems, a new asymmetric self-recovery oriented stereo image watermarking method is proposed by considering inter-correlations between the left and right views of the stereo image. An asymmetric self-recovery mechanism is presented to conduct allocation of watermarking capacity for asymmetrically embedding recovery references of each view. The presented mechanism is also used to recover tampered left and right views asymmetrically to obtain high quality of the stereo image with the help of disparity when the tampered regions of the stereo image are authenticated. For security, a chaotic function is employed to generate authentication bits to detect tamper. The high-frequency energy of discrete wavelet transform is utilized to divide blocks of each view into smooth and complex types, so that the recovery reference with alterable bits is computed. Moreover, to obtain a trade-off between transparency and watermarking capacity, human visual characteristics are taken into account and then a just-noticeable difference model is exploited to classify the blocks into sensitive and insensitive types, which

defines two or three least significant bits of pixels are allocated for embedding watermark. Experimental results demonstrate that the proposed method can reconstruct tamper efficiently and outperform other stereo image watermarking methods, especially for extensive tamper.

Keywords Three dimensional video system · Stereo image watermarking · Just-noticeable difference · Asymmetric self-recovery · Disparity

1 Introduction

Three dimensional video (3DV) has been attracting more and more attention recently, since it is able to provide a visual experience with depth perception for new multimedia applications, such as three dimensional TeleVision (3DTV) and Free-viewpoint TeleVision (FTV) [1, 2]. The stereo image, as a main representation of the 3D image in 3DV system, can be easily modified and tampered by using a variety of sophisticated image processing tools during the transmission [3], and thus precision and integrity of the stereo image are greatly threatened. To solve this problem, the authentication watermarking techniques were developed by embedding watermark information into the stereo image [4, 5].

In the previous studies, the focus of authentication watermarking methods was mainly placed on whether an image has been modified or not [6, 7]. However, it was desirable that tampered regions can not be only located but also recovered. Watermarking methods with both tamper location and self-recovery could be classified into two types. In the first type, tampered regions can be restored completely [8]. In Zhang's method [8], difference expansion was used to embed the recovery reference, and it can be used to reconstruct tamper without any error. However,

Communicated by L. Zhou.

✉ Gangyi Jiang
jianggangyi@126.com

Ting Luo
luoting@nbu.edu.cn

¹ Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China

² College of Science and Technology, Ningbo University, Ningbo 315211, China

³ National Key Lab of Software New Technology, Nanjing University, Nanjing 210093, China

it cannot work when tampered regions are not small enough. In the second type, recovered images are tried to be approximate to the originals as long as the image degradation cannot be perceived by human vision [9–13]. Qin et al. [9] concatenated all recovery references and embedded them into the image, in which type indicators for the number of recovery reference bits are utilized to discriminate one reference from the others. However, if one of type indicators is modified, each recovery reference may not be extracted correctly, and it results in serious damage of recovery. To avoid the usage of the type indicator, in [10], recovery reference of each block was embedded into its unique mapping block, so that tampered blocks can be reconstructed via extracting corresponding recovery reference from their un-modified mapping blocks. Moreover, in order to increase chances of recovering tampered block, two copies of each recovery reference were embedded into two different mapping blocks, respectively [11–13]. Above authentication watermarking methods were all designed for the monocular image, and if they are directly extended to the left and right views of stereo image, watermarking performance will not be improved because of missing characteristics of stereo image.

Stereo image is captured by two cameras from the same scene, and presents two offset views with disparity map [14]. Thus, the disparity map can be used to reconstruct one view of the stereo image from the other view. Some disparity based stereo image watermarking methods were proposed to protect copyright of stereo images [15, 16]. Yu et al. employed the inter-relationship and intra-relationship modulation to embedding watermark information into stereo image for obtaining robustness [17]. Ou and Chen recorded unmatched blocks as the robust feature to protect copyright [18]. However, up to now, only a few authentication stereo image watermarking methods have been studied. Wang et al. employed singular value decomposition (SVD) and discrete wavelet transform (DWT) to authenticate the stereo image, but the method had no capability of recovery [19]. Yu et al. classified blocks into matchable and non-matchable types, and alterable bits were allocated for different types of blocks, so that it performed well on small tampered regions [20]. But, when the size of tampered regions is increased, the recovery quality of the stereo image will be degraded.

In this paper, considering inter-correlations between the left and right views of the stereo image, an asymmetric self-recovery oriented stereo image watermarking method is proposed for 3DV system. The asymmetric self-recovery mechanism is defined to embed recovery references of the left and right views asymmetrically. Based on the presented mechanism, two views of the tampered stereo image are asymmetrically restored to obtain high quality of the recovered stereo image with the help of disparity.

Moreover, smoothness of the block is calculated by using high frequency energy of the DWT decomposed block, and it is exploited to compute alterable bits of recovery reference. To obtain a trade-off between transparency and capacity of watermarking, the just-noticeable difference (JND) model [21] is utilized to define three or two LSBs of pixels to be allocated for embedding watermark. In the processes of detecting tamper, the inter-correlations between tampered regions of two views are used to improve the performance of tamper detection scheme. Experimental results show that the proposed method identifies tamper accurately, and its recovery performance is superior to the other methods, such as Lin's [10], Tong's [11], Huo's [13] and Yu's [20].

The remainder of paper is organized as follows. The proposed asymmetric self-recovery oriented stereo image watermarking method is presented in Sect. 2. Section 3 gives experimental results to demonstrate effectiveness of the proposed method. Section 4 concludes this paper.

2 Proposed asymmetric self-recovery oriented stereo image watermarking method

To authenticate integrity of the stereo image for 3DV system, a novel asymmetric self-recovery oriented stereo image watermarking method is proposed with a new asymmetric self-recovery mechanism. The presented asymmetric self-recovery mechanism consists of asymmetric embedding for recovery reference of two views and asymmetric tamper self-recovery, and they are used to allocate watermarking capacity and conduct procedures of tamper self-recovery, respectively. In the following sub-sections, the proposed algorithms of watermark embedding, tamper detection and self-recovery are depicted in detail.

2.1 Watermark generation and embedding

Let $\{I_{a,b}^L\}$ and $\{I_{a,b}^R\}$ be the left and right views of a stereo image, with the size of $N_1 \times N_2$, respectively, where $1 \leq a \leq N_1$ and $1 \leq b \leq N_2$. Suppose the left view is regarded as the reference view, a pixel-wise disparity map $\{d(a, b)\}$ is calculated with the disparity estimation algorithm in [22], where if $I_{a,b}^L$ is not matched with any pixel in the right view, the value of $d(a, b)$ is set to 255. Although $d(a, b)$ is computed for the left view, pixels in the right view can also find matched pixels in the left view by using $d(a, b)$. Main procedures of the proposed watermark generation and embedding algorithm as illustrated in Fig. 1 are depicted as follows.

Step 1 The left and right views are divided into non-overlapping blocks with the size of 4×4 , and the blocks

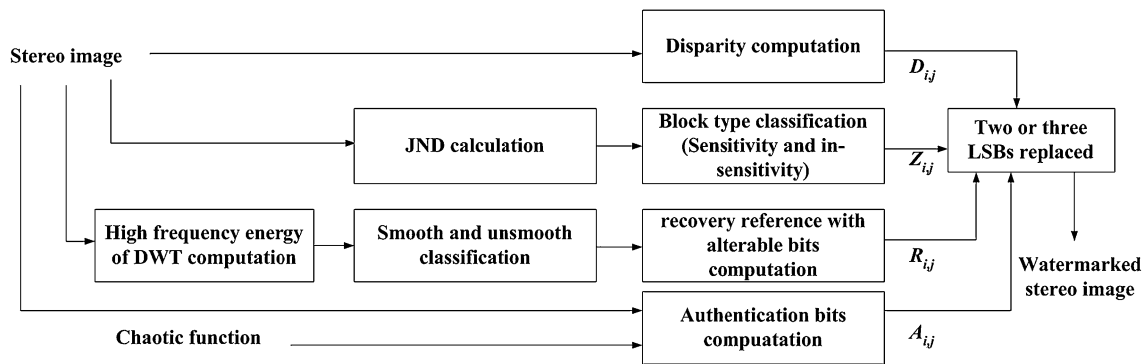


Fig. 1 Block diagram of watermark embedding

of the left and right views are denoted as $B_{i,j}^L$ and $B_{i,j}^R$, respectively, where $1 \leq i \leq N_1/4$ and $1 \leq j \leq N_2/4$.

Step 2 The pixel-wise disparity map is divided into non-overlapping blocks with the size of 4×4 , and is transformed to a block-wise disparity map. The block-wise disparity $D_{i,j}$ is computed that if pixel-wise disparities in the same block are all equal, $D_{i,j}$ is $d(a, b)$, otherwise, it is set to 255.

Step 3 JND value is computed for each block to determine the block type $Z_{i,j}$, that is, sensitive or insensitive. Compared with the sensitive block, large modification of the insensitive block can not be discovered by human vision. Thus, two and three LSBs of sensitive and insensitive blocks are supplied for embedding watermark, respectively. It is concretely described in Sect. 2.1.1.

Step 4 Since DWT coefficients with high frequency energies represent detail information of blocks, they are used to classify blocks into smooth and unsmooth. Recovery reference $R_{i,j}$ are computed based on block's smoothness. It will be depicted specifically in Sect. 2.1.2.

Step 5 Authentication bits $A_{i,j}$ are computed and described in Sect. 2.1.3.

Step 6 Watermarking capacity is allocated for asymmetric embedding, and $D_{i,j}$, $Z_{i,j}$, $R_{i,j}$ and $A_{i,j}$ are embedded into the left and right views of the original stereo image to obtain its watermarked stereo image. In detail, they are depicted in Sects. 2.1.4 and 2.1.5, respectively.

2.1.1 Block type classification

To obtain a trade-off between watermarking's transparency and capacity, blocks of the left and right views are classified into sensitive or insensitive ones according to JND value which is related to background luminance masking and texture masking. Background luminance masking is taken into account that luminance contrast is more sensitive

than absolute luminance, and texture masking is built on that textured regions can hide more information than smooth or edge regions [21]. Let $JND_{a,b}^p$ denote the JND value of the pixel p at the position (a, b) and then it is computed by using Eq. (1) [21].

$$JND_{a,b}^p = U_l + U_t - v_{l,t} \times \min(U_l, U_t) \quad (1)$$

where U_l and U_t are visibility thresholds of background luminance masking and texture masking, respectively, and they are specifically defined in [21]. $v_{l,t}$ is the parameter accounting for the overlapping effect in masking between 0 and 1. Since JND value of each pixel is computed for detecting tamper, in the process of computing texture masking, the threshold for image edge detection with Sobel operator is a secret key. Let $JND_{i,j}^B$ denote the JND value of each block, and it is computed as follows

$$JND_{i,j}^B = \frac{1}{16} \left(\sum_{a=(i-1) \times 4 + 1}^{i \times 4} \sum_{b=(j-1) \times 4 + 1}^{j \times 4} JND_{a,b}^p \right) \quad (2)$$

Then, let $Z_{i,j}$ denote block type, and it is calculated by

$$Z_{i,j} = \begin{cases} 1, & JND_{i,j}^B < T_s \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where 1 and 0 indicates that the block is sensitive and insensitive, respectively. When three LSBs of a pixel are substituted, the largest difference between the original pixel and the modified pixel is 7. Thus, T_s is set to 7 for controlling LSB substitution without exceeding JND values. As illustrated in Fig. 1, the pixels of the sensitive block only offer two LSBs for embedding watermark, while the pixels of the insensitive block offer three LSBs [23].

2.1.2 Recovery reference generation

Alterable bits are adopted to represent recovery reference $R_{i,j}$ so as to recover the corresponding block when the

block is tampered. Since difference among all pixels of a smooth block is small, fewer recovery reference bits are enough for smooth blocks compared with the unsmooth ones. Smoothness of a block at position (i, j) of the left or right view is judged by computing its high frequency energy in DWT domain [24]. Firstly, the left or right view is transformed with one level DWT, and the three high frequency subbands HL, LH, and HH are further divided into blocks with the size of 2×2 . Let $H_{i,j}^z(x, y)$ denote DWT coefficient at the position (x, y) of 2×2 block of the z -th subband corresponding to the block at the position (i, j) of the left or right view, where $z = 1, 2$ and 3 representing the HL, LH and HH, respectively, and $1 \leq x, y \leq 2$. High frequency energy $E_{i,j}$ of 4×4 block at (i, j) is computed by using Eq. (4).

$$E_{i,j} = \sum_{x=1}^2 \sum_{y=1}^2 \left(H_{i,j}^1(x, y) \right)^2 + \sum_{x=1}^2 \sum_{y=1}^2 \left(H_{i,j}^2(x, y) \right)^2 + \sum_{x=1}^2 \sum_{y=1}^2 \left(H_{i,j}^3(x, y) \right)^2 \tag{4}$$

Then, smoothness $S_{i,j}$ of the (i, j) -th block in the left or right view is computed as

$$S_{i,j} = \begin{cases} 1, & \text{if } E_{i,j} < \gamma \times E_{\text{avg}} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where 1 or 0 denotes that the corresponding block is smooth or unsmooth, and E_{avg} is average high frequency

$$A_{i,j}(z) = \begin{cases} \left(g\left(\left\lfloor H_{i,j}^z(1, 1) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(1, 2) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 1) \right\rfloor\right) \right) \bmod 2 & p(m) \geq 0 \ \& \ q(m) \geq 0 \\ \left(g\left(\left\lfloor H_{i,j}^z(1, 1) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(1, 2) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 2) \right\rfloor\right) \right) \bmod 2 & p(m) \geq 0 \ \& \ q(m) < 0 \\ \left(g\left(\left\lfloor H_{i,j}^z(1, 2) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 1) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 2) \right\rfloor\right) \right) \bmod 2 & p(m) < 0 \ \& \ q(m) \geq 0 \\ \left(g\left(\left\lfloor H_{i,j}^z(1, 1) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 1) \right\rfloor\right) + g\left(\left\lfloor H_{i,j}^z(2, 2) \right\rfloor\right) \right) \bmod 2 & p(m) < 0 \ \& \ q(m) < 0 \end{cases} \tag{7}$$

energy of all blocks in left and right views, and $\gamma \in [0, 1]$. Recovery references with alterable bits are computed for smooth and unsmooth blocks. The blocks with the size of 4×4 are further divided into four sub-blocks with the size of 2×2 , and average intensity of each sub-block from left to right and top to bottom is calculated and denoted as $\text{avg}_1, \text{avg}_2, \text{avg}_3$ and avg_4 , respectively.

1. For unsmooth blocks, 20 or 15 bits are used as $R_{i,j}$. Since 5 most significant bits (MSB) can be mainly used to represent a pixel, thus 5 MSBs of $\text{avg}_1, \text{avg}_2, \text{avg}_3$ and avg_4 , that is, total 20 bits are used as $R_{i,j}$. However, due to the limited watermarking capacity,

20 bits may be too many to be allocated in this case. Thus, it is decreased to 15 bits for representing $R_{i,j}$, which is formed by 5 MSBs of $\text{avg}_1, \text{avg}_2$ and avg_5 , where $\text{avg}_5 = (\text{avg}_1 + \text{avg}_3)/2$. Watermarking capacity allocation will be mainly discussed in Sect. 2.1.4.

2. For smooth blocks, 10 or 5 bits are used as $R_{i,j}$. $\text{avg}_6 = (\text{avg}_2 + \text{avg}_4)/2$, and 5 MSBs of avg_5 and 5 MSBs of avg_6 are used as $R_{i,j}$ with 10 bits. Or, 5 MSBs of avg are used as $R_{i,j}$, where $\text{avg} = (\text{avg}_5 + \text{avg}_6)/2$ for limited watermarking capacity.

2.1.3 Authentication bits generation

A block serial number m is assigned to each block of the left or right view, and $m = (j-1) \times (N_1/4) + i$. To improve security of watermarking, a chaotic function [25] is used to guide authentication bits generation, and expressed by

$$\begin{cases} p(m) = 1 - \alpha \times q(m-1)^2 \\ q(m) = \cos(\beta \times \arccos(p(m-1))) \end{cases} \tag{6}$$

where α and β are two control parameters and set to 2 and 6, respectively [25], in order that $p(m), q(m) \in [-1, 1]$ are chaotic. $p(0)$ and $q(0)$ are initially set to the values between -1 and 1 . Since DWT high frequencies represent detail information of the image, they are easy to be modified. Three authentication bits $A_{i,j}$ of each block are computed by using three coefficients of HL, LH and HH, respectively.

where $\&$ denotes the logical AND operator, mod returns the remainder of division, and $g(\cdot)$ is defined by Eq. (8).

$$g(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{8}$$

2.1.4 Watermarking capacity allocation for asymmetrically embedding watermark

Watermarking capacity allocation is realized by using an asymmetric self-recovery mechanism that two or more copies of recovery reference in one view are embedded into different mapping blocks, and two or less copies of reference in the other view. Suppose that copies of recovery reference

Fig. 2 Illustration of mapping relationships

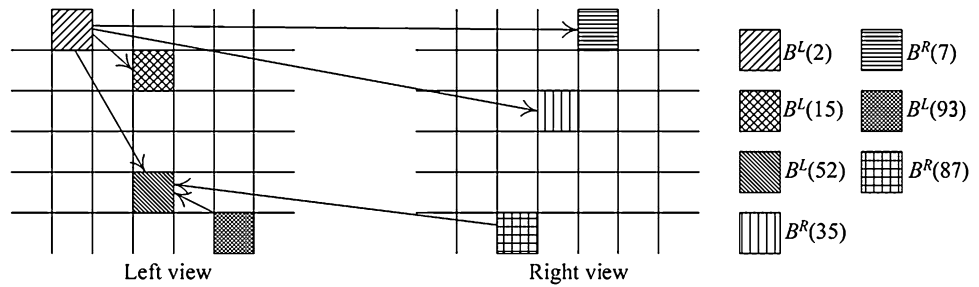


Table 1 $G_{i,j}$ corresponding to $H_{i,j}$

L										
$Z_{i,j} = 1$	$V_{i,j} = 1$	$G_{i,j}$	00	01	10	11				
		$H_{i,j}$	D, R_1 (5, 20)	D, R_2	D, R_3	D, R_1, R_2 (5, 10, 10)				
$Z_{i,j} = 1$	$V_{i,j} = 0$	$G_{i,j}$	00	01	10	11				
		$H_{i,j}$	R_1, R_3 (20, 5)	R_2, R_3	R_3, R_1	R_1, R_2, R_3 (10, 10, 5)				
$Z_{i,j} = 0$	$V_{i,j} = 1$	$G_{i,j}$	000	001	010	011	100	101	110	111
		$H_{i,j}$	D, R_1, R_2 (5, 20, 15)	D, R_2, R_1	D, R_1, R_2, R_3 (5, 20, 10, 15)	D, R_1, R_3, R_2	D, R_2, R_1, R_3	D, R_2, R_3, R_1	D, R_3, R_1, R_2	D, R_3, R_2, R_1
$Z_{i,j} = 0$	$V_{i,j} = 0$	$G_{i,j}$	000	001	010	011	100	101	-	-
		$H_{i,j}$	R_1, R_2 (20, 20)	R_2, R_3	R_1, R_3	R_1, R_2, R_3 (20, 10, 10)	R_3, R_1, R_2	R_2, R_1, R_3	-	-
R										
$Z_{i,j} = 1$		$G_{i,j}$	000	001	010	011	100	101	110	-
		$H_{i,j}$	R_1, R_3 (20, 15)	R_2, R_3	R_3, R_1	R_3, R_2	R_1, R_2	R_2, R_1	R_1, R_2, R_3 (20, 10, 5)	-

of the left view are to be embedded more, and more embedding positions are prepared for recovery reference of the left view compared with the right view. Specifically, four and two embedding positions are provided by each block of the left and right views, respectively. Mapping relationships are designed to define embedding positions of recovery reference, and six one-to-one mapping sequences are generated to represent the mapping relationships. Firstly, let $e(m)$ represent an embedding position for recovery reference of the m -th block and it can be created by

$$e(m) = (k \times m) \bmod L + 1, \quad m = 1, 2, \dots, L \tag{9}$$

where k is a prime number between 1 and $L-1$, and $L = (N_1 \times N_2)/16$. Thus, if k is set to k_1, k_2, k_3 or k_4 (prime numbers), four sequences are generated in the same way and denoted as $\{e_1(m)\}, \{e_2(m)\}, \{e_3(m)\}$ and $\{e_4(m)\}$, respectively.

Secondly, other two sequences, denoted as $\{e_5(m)\}$ and $\{e_6(m)\}$, are computed by

$$e(m) = \begin{cases} m + L/2, & \text{if } m \leq L/2 \\ m - L/2, & \text{otherwise} \end{cases} \tag{10}$$

Let $B^L(m)$ and $B^R(m)$ denote the m -th block with the size of 4×4 in the left and right views, respectively. Using $\{e_1(m)\}, \{e_2(m)\}, \{e_3(m)\}, \{e_4(m)\}, \{e_5(m)\}$ and $\{e_6(m)\}$, mapping relationships for blocks of the two views are defined as follows. $B^L(e_1(m)), B^L(e_5(m)), B^R(e_2(m))$ and $B^R(e_3(m))$ are four mapping blocks of $B^L(m)$, and $B^L(e_4(m))$ and $B^R(e_6(m))$ are two mapping blocks of $B^R(m)$. In other words, each block is the mapping block of two blocks in the left view and one block in the right view, and those three blocks are labeled as B_1, B_2 , and B_3 , respectively. For example, suppose $L = 100, k_1 = 7, k_2 = 17, k_3 = 53$ and $k_4 = 73, B^L(15), B^L(52), B^R(7)$ and $B^R(35)$ are mapping blocks of $B^L(2)$ as illustrated in Fig. 2. Figure 2 also shows that block $B^L(52)$ is the mapping block of the blocks $B^L(93), B^L(2)$ and $B^R(87)$ (denoted as B_1, B_2 , and B_3 , respectively).

Due to limited watermarking capacity of each block, not all recovery reference of each block can be embedded into its mapping blocks and recovery references of the blocks B_1, B_2 , and B_3 will be chosen to be embedded. Thus, watermarking capacity should be allocated properly for embedding enough recovery references. Besides recovery references, watermarking capacity of each block

is also allocated for block type $Z_{i,j}$, authentication bits $A_{i,j}$ and disparity bits $D_{i,j}$. Let $C_{i,j}^L$ and $C_{i,j}^R$ denote watermark bits of the blocks $B_{i,j}^L$ and $B_{i,j}^R$, respectively. Since disparity can be used to recover both of views, one copy of the disparity is embedded for saving watermarking capacity. In the proposed method, disparity is only embedded into blocks of the left view, and thus $C_{i,j}^L$ is represented as

$$C_{i,j}^L = Z_{i,j}||A_{i,j}||V_{i,j}||G_{i,j}||H_{i,j} \tag{11}$$

where ‘||’ denotes concatenation operation, $V_{i,j}$ denotes whether the block B_1 has its matched block in the right view or not, and its corresponding value is 1 or 0. $G_{i,j}$ denotes the block type of recovery reference embedding, and $H_{i,j}$ is specific recovery reference embedding corresponding to $G_{i,j}$ as shown on the row L of Table 1. In Table 1, D represents a disparity value of the block B_1 and it is supposed to range from 0 to 31 represented by 5 bits. R_1, R_2 , or R_3 is a recovery reference of the block B_1, B_2 , or B_3 , and digits in the parenthesis are the number of bits of them. For instance, if $Z_{i,j} = 1, V_{i,j} = 1$ and $G_{i,j} = 00$, 32 bits of $B_{i,j}^L$ are allocated for embedding watermark, R_2 and R_3 are not embedded, and $H_{i,j}$ represents D and R_1 with 5 and 20 bits, respectively.

1. Watermark embedding of unsmooth blocks is superior to that of smooth blocks, and watermark embedding of the left view’s blocks is superior to that of the right view’s blocks.
2. Recovery reference of the unsmooth block in the right view must be embedded at least once.
3. 15 or 20 bits are allocated for the unsmooth block, and 5 or 10 bits are for the smooth block on the contrary. But 5 or 10 bits may represent the recovery reference of the unsmooth block if watermarking capacity is not enough. On the contrary, if there is enough capacity, more than 10 bits can be used to represent the smooth block.

Based on the rules, when recovery reference of the same block is embedded into different mapping blocks, the number of embedding bits may be different. Specifically computing $G_{i,j}$, firstly for $B_{i,j}^L$ if $Z_{i,j} = 1, G_{i,j}$ is computed by using sub-algorithm 1. Following the example of Fig. 2, suppose $(j-1) \times (N_1/4) + i = 52, D = 10100$ for block $B^L(93), R_2 = 10100 11010 10101 10010$ for unsmooth block $B^L(2)$ and $A_{i,j} = 101$. Therefore, $V_{i,j}$ is 1 because of the valid disparity value for $B^L(93), G_{i,j}$ is 01 by using the sub-algorithm 1, and finally $C_{i,j}^L = 1 101 1 01 10100 10100 11010 10101 10010$ by using Eq. (10).

<p>sub-algorithm 1:</p> <p>1: If block B_2 is unsmooth</p> <p>2: $G_{i,j}=01$</p> <p>3: else if block B_1 is unsmooth</p> <p>4: $G_{i,j}=00$</p> <p>5: else if block B_3 is unsmooth</p> <p>6: $G_{i,j}=10$</p> <p>7: else</p> <p>8: $G_{i,j}=11$</p>	<p>sub-algorithm 2:</p> <p>1: If block B_1 and block B_2 are unsmooth</p> <p>2: $G_{i,j}=funcE(B_1, B_2)$</p> <p>3: else</p> <p>4: $G_{i,j}=funcE(B_1, B_2, B_3)$</p>
---	--

For $C_{i,j}^R$, disparity bits are not included, $C_{i,j}^R$ is formed by

$$C_{i,j}^R = Z_{i,j}||A_{i,j}||G_{i,j}||H_{i,j} \tag{12}$$

where $G_{i,j}$ and $H_{i,j}$ are shown on the row R of Table 1 for $Z_{i,j} = 1$. If $Z_{i,j} = 0, G_{i,j}$ and $H_{i,j}$ are the same as the row L for $Z_{i,j} = 0$ and $V_{i,j} = 0$.

With regard to computing $G_{i,j}$, taking account of the proposed asymmetric self-recovery mechanism, the rules are listed as follows.

<p>sub-algorithm 3:</p> <p>1: If block B_1 and block B_2 are unsmooth</p> <p>2: $G_{i,j}=000$</p> <p>3: else if block B_1 and block B_3 are unsmooth</p> <p>4: $G_{i,j}=010$</p>	<p>5: else if block B_2 and block B_3 are unsmooth</p> <p>6: $G_{i,j}=001$</p> <p>7: else</p> <p>8: $G_{i,j}=funcE(B_1, B_2, B_3)$</p>
---	--

If $Z_{i,j} = 0$ and $V_{i,j} = 1$, the sub-algorithm 2 is used to compute $G_{i,j}$, where $funcE(\cdot)$ returns the value of $G_{i,j}$ according to comparison of high frequency energies of different blocks. The greater high frequency energy gets more number of bits for their copies of recovery reference. For instance, if high frequency energy of the block B_1 is greater than that of the block B_2 , $funcE(B_1, B_2)$ returns 000, and $H_{i,j}$ consists of D, R_1 and R_2 with 5, 20 and 15 bits, respectively. Moreover, if $Z_{i,j} = 0$ and $V_{i,j} = 0, G_{i,j}$ is calculated by using the sub-algorithm 3.

Secondly, for B_{ij}^R , if $Z_{ij} = 0$, G_{ij} is computed similarly as the sub-algorithm 3. Only, the exception is that if the block B_3 is unsmooth and its reference is not embedded, $G_{ij} = 010$ or $G_{ij} = 001$ is superior to $G_{ij} = 000$. If $Z_{ij} = 1$, G_{ij} is defined by using the sub-algorithm 4.

Step a-3 Mapping relationships for embedding positions of block recovery references are built by using Eqs. (9) and (10).

Step a-4 C_{ij}^L or C_{ij}^R is computed for each block by Eq. (11) or Eq. (12), where G_{ij} is computed by using the

sub-algorithm 4:	
1: If block B_3 is unsmooth	14: If block B_3 is smooth
2: If reference of block B_3 is not embedded yet	13: if block B_1 is unsmooth
3: If block B_1 has lower high frequency energy	14: $G_{ij}=000$
4: $G_{ij}=010$	15: else if block B_2 is unsmooth
5: else	16: $G_{ij}=001$
6: $G_{ij}=011$	17: else
7: if reference of block B_3 is embedded	18: $G_{ij}=110$
8: if block B_1 is unsmooth	
9: $G_{ij}=100$	
10: else if block B_2 is unsmooth	
11: $G_{ij}=101$	
12: else	
13: $G_{ij}=010$	

2.1.5 Watermark embedding

Procedures of embedding watermark are described with the following four main steps.

Step a-1 The left and right views of stereo image are divided into non-overlapping blocks with the size of 4×4 . High frequency energy of each block E_{ij} is computed, and the corresponding S_{ij} is set to 1 or 0 by using Eq. (5). $avg_1, avg_2, avg_3, avg_4, avg_5, avg_6$ or avg of each block is computed for R_{ij} .

Step a-2 Authentication bits A_{ij} are calculated by using Eq. (7), and D_{ij} is computed as well. The third LSB of each pixel of the left and right views are preserved, and three LSBs of each pixel are set to 0. Each block's JND value and Z_{ij} are computed by Eqs. (2) and (3), respectively. The preserved third LSBs are put back to the original positions.

sub-algorithm 1, 2, 3, or 4. C_{ij}^L or C_{ij}^R replaces two or three LSBs of pixels in B_{ij}^L or B_{ij}^R . Repeat this step until all C_{ij}^L or C_{ij}^R of each block is embedded, and the watermarked stereo image is obtained.

2.2 Tamper detection

At the receiving side, received stereo image is identified whether it is tampered or not. The left and right views of stereo image have content correlations, and thus they are often symmetrically tampered, otherwise, tamper is perceived visually. Besides authentication bits extraction, similarity between tampered regions of the left and right views is used to detect tamper. Let $F^L = \{F_{ij}^L\}$ and $F^R = \{F_{ij}^R\}$ denote tamper masking of the left and right views' blocks, respectively. Main steps of tamper detection are described as follows.

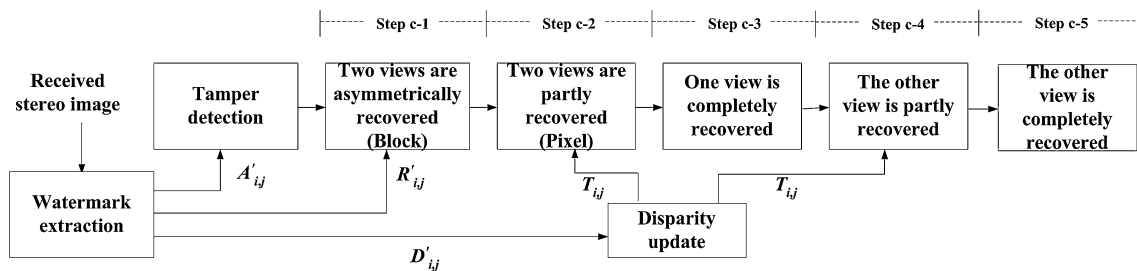


Fig. 3 Block diagram of asymmetric tamper self-recovery

Fig. 4 Original left views of stereo images. **a** Bowling; **b** Akko; **c** Laptop; **d** Alt Moabit; **e** Laundry; **f** Puppy

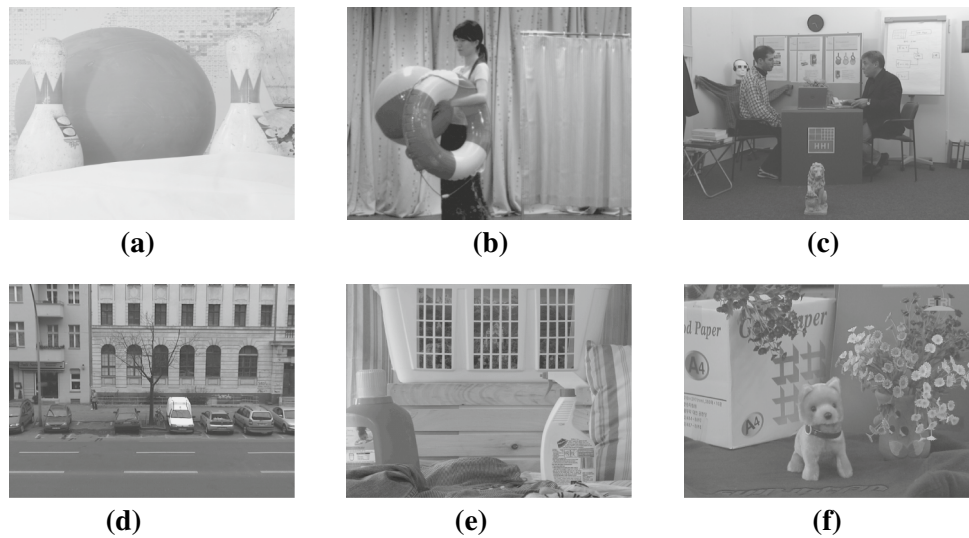


Table 2 PSNR and PSPNRs comparison of watermarked stereo image [dB] (PSNR, PSPNR)

	Bowling	Akko	Laptop	Alt Moabit	Laundry	Puppy
Tong's						
Left	(40.77, 68.82)	(40.73, 67.30)	(40.74, 72.98)	(40.70, 72.48)	(40.72, 67.58)	(40.71, 69.64)
Right	(40.76, 68.58)	(40.74, 67.85)	(40.72, 73.16)	(40.71, 72.36)	(40.74, 67.50)	(40.74, 70.15)
Proposed						
Left	(43.95, 73.79)	(42.32, 69.57)	(39.94, 73.36)	(40.28, 73.23)	(41.46, 68.79)	(40.51, 74.64)
Right	(43.79, 73.89)	(42.21, 70.56)	(40.23, 73.58)	(40.41, 73.33)	(41.58, 68.43)	(40.49, 73.69)

Step b-1 The received stereo image is divided into non-overlapping blocks with the size of 4×4 . Sensitive or insensitive type $Z'_{i,j}$ is directly extracted from each block, and three authentication bits, denoted as $A'_{i,j}$, are extracted as the reverse of embedding watermark. Moreover, recovery reference $R'_{i,j}$ is extracted as well for tamper recovery, which will be described in Sect. 2.3.

Step b-2 Three LSBs of pixels in left and right views are set to 0. $A''_{i,j}$ are computed by using Eq. (7), and are compared with $A'_{i,j}$. If any pair is not equal, $F^L_{i,j}$ or $F^R_{i,j}$ is set to 1, otherwise, $F^L_{i,j}$ or $F^R_{i,j}$ is set to 0. Repeat the step, until all $A''_{i,j}$ of each block are compared.

Step b-3 Each block's JND value and $Z''_{i,j}$ are computed by Eqs. (2) and (3), respectively. If $Z'_{i,j}$ is not equal to $Z''_{i,j}$, the block is identified as tampered, and $F^L_{i,j}$ or $F^R_{i,j}$ is set to 1. Repeat the step, until all $Z'_{i,j}$ of each block are compared.

Step b-4 Morphological erosion and dilation operations are performed to F^L and F^R [20].

Step b-5 Connected regions of F^L and F^R are extracted and denoted as $F^{L,n}$ and $F^{R,n}$, respectively, where n denotes the n -th tampered region. The n -th tampered regions of left and right views are merged as

$$F^n = F^{L,n} \cup F^{R,n} \quad (13)$$

where ' \cup ' is the binary 'OR' operation. F^n replaces both of $F^{L,n}$ and $F^{R,n}$, and consequently, F^L and F^R are updated. Tampered blocks are detected with F^L and F^R finally.

2.3 Asymmetric tamper self-recovery

If $F^L_{i,j}$ or $F^R_{i,j}$ is 1, $B^L_{i,j}$ or $B^R_{i,j}$ is tampered and the asymmetric tamper self-recovery scheme is used to reconstruct tampered blocks. Five asymmetric self-recovery steps are described as shown in Fig. 3.

Step c-1 Recovery reference $R'_{i,j}$ is extracted as the reverse of watermark embedding. More than one valid copy for one tampered block may be extracted, and the valid recovery reference with the maximum bits is used to recover the tamper. In this step, left view is recovered better than the right view.

Step c-2 Extracted disparity $D'_{i,j}$ is used to recover tamper. $D'_{i,j}$ may not be the same as $D_{i,j}$. An assumption is given that disparity is continuous [26], and $D'_{i,j}$ is updated as $T_{i,j}$ by using Eq. (14).

$$T_{i,j} = \begin{cases} D'_{i,j-1}, & \text{if } D'_{i,j-1} = D'_{i,j+1} \ \& \ D'_{i,j+1} \neq 255 \\ D'_{i-1,j}, & \text{if } D'_{i-1,j} = D'_{i+1,j} \ \& \ D'_{i+1,j} \neq 255 \end{cases} \quad (14)$$

Fig. 5 Tampered left view of Akko with different tamper ratios. **a** 21.57 %; **b** 38.52 %; **c** 57.09 %

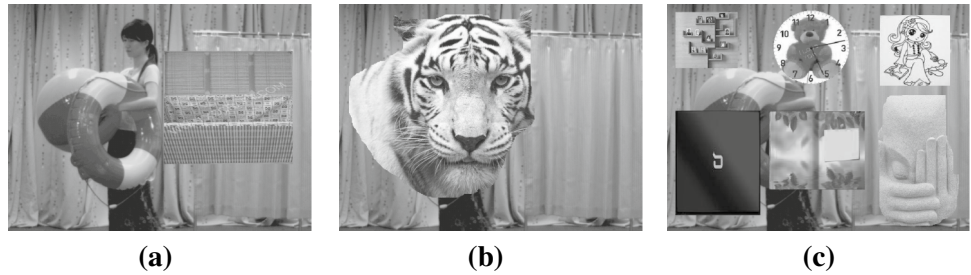
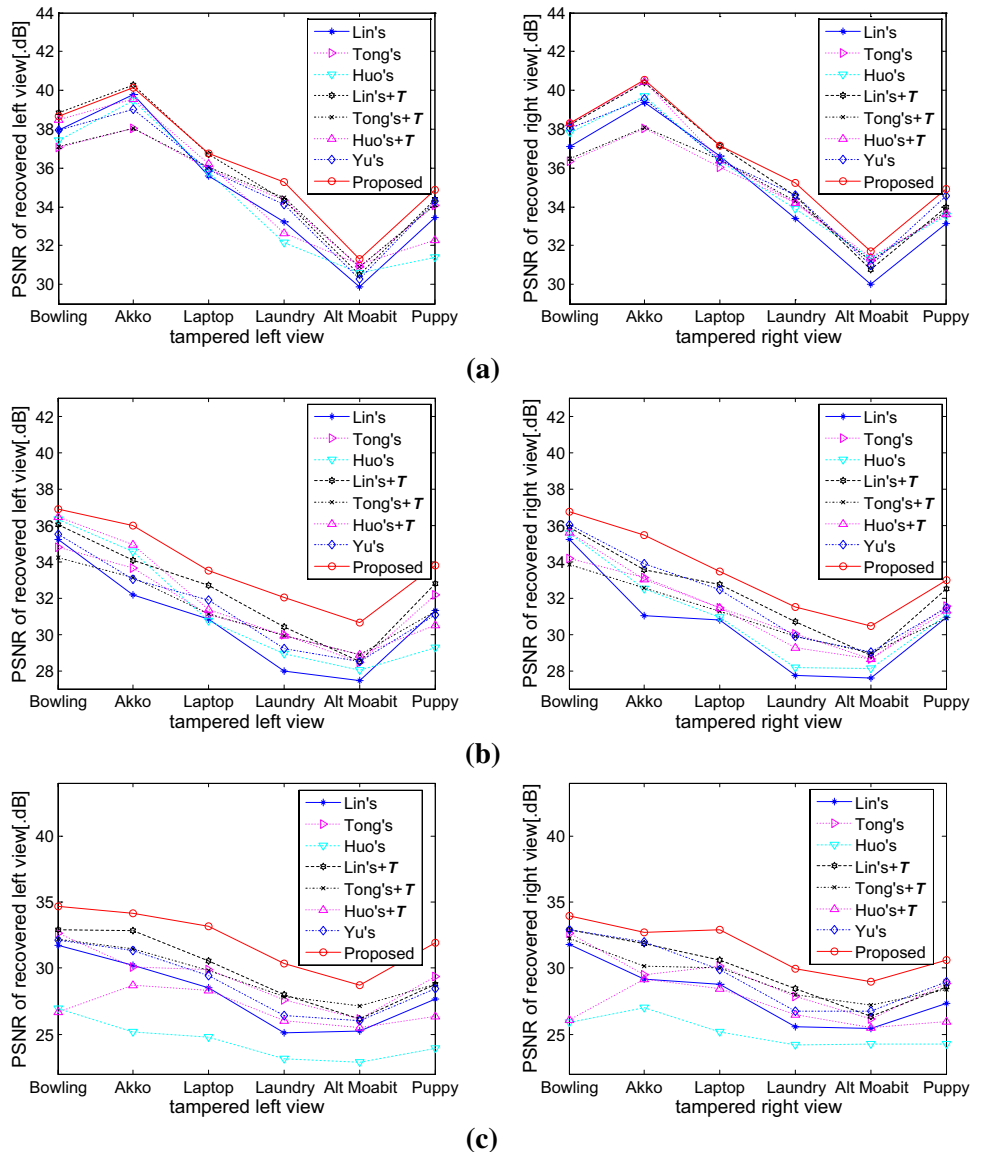


Fig. 6 PSNR comparison of different watermarking methods for different size of pasting tamper. **a** 21.57 %; **b** 38.52 %; **c** 57.09 %

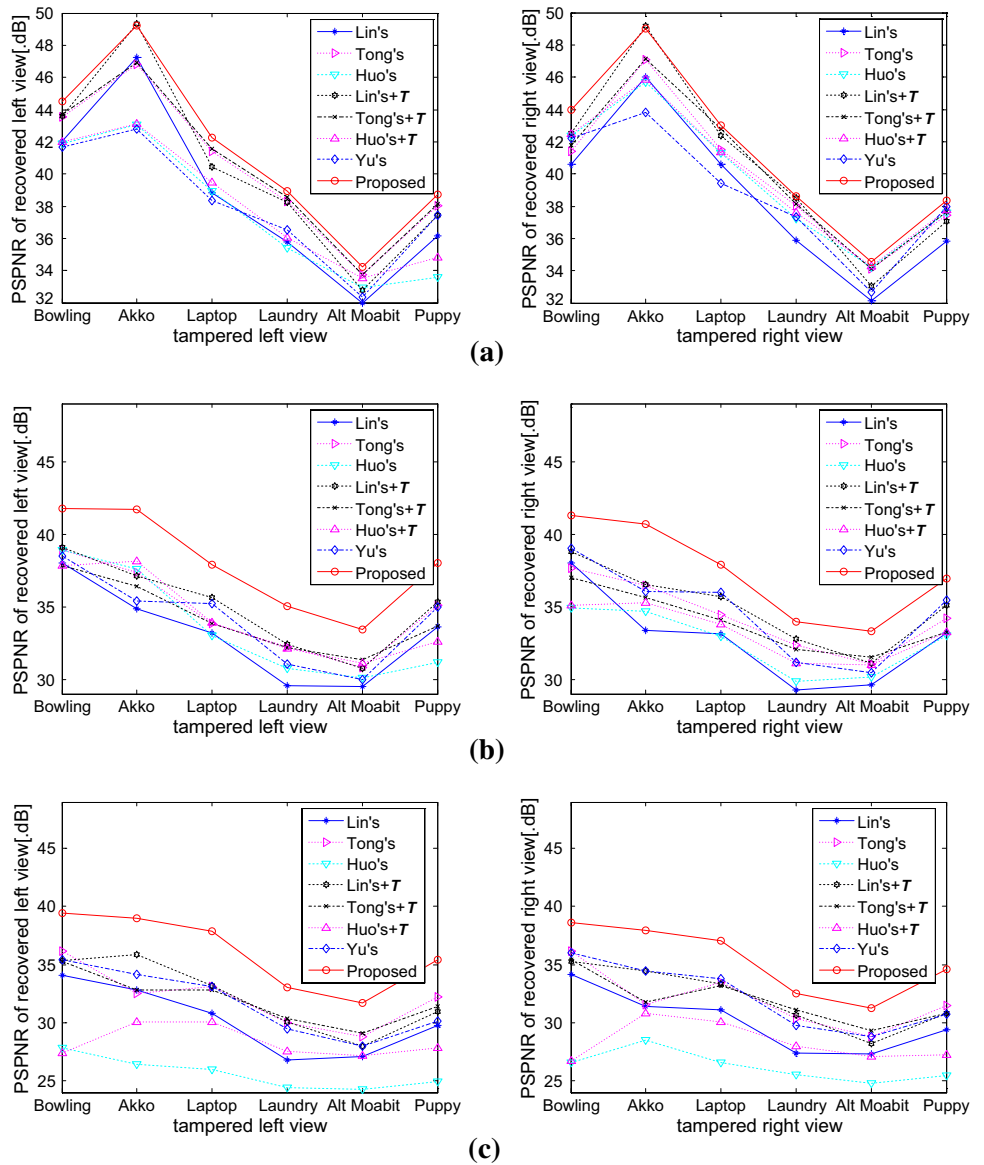


Tampered pixels in both views are recovered by using $T_{i,j}$, such as, valid $I_{a,b}^L$ replaces the tampered $I_{a-d(a,b),b}^R$ or valid $I_{a-d(a,b),b}^R$ replaces the tampered $I_{a,b}^L$.
 Step c-3 One view with better recovery is completely reconstructed by using the inpainting method [27].
 Step c-4 The other view is partly recovered by using $T_{i,j}$.

Step c-5 The other view is completely recovered using the inpainting method [27]. Finally, tampered stereo image is recovered.

In the above self-recovery steps, if two views are completely recovered before their due steps, the remaining

Fig. 7 PSPNR comparison of different watermarking methods for different size of pasting tamper. **a** 21.57 %; **b** 38.52 %; **c** 57.09 %



steps will be omitted. For example, if both of views are completely recovered in the fourth step, the final step will not be carried out.

3 Experimental results and discussions

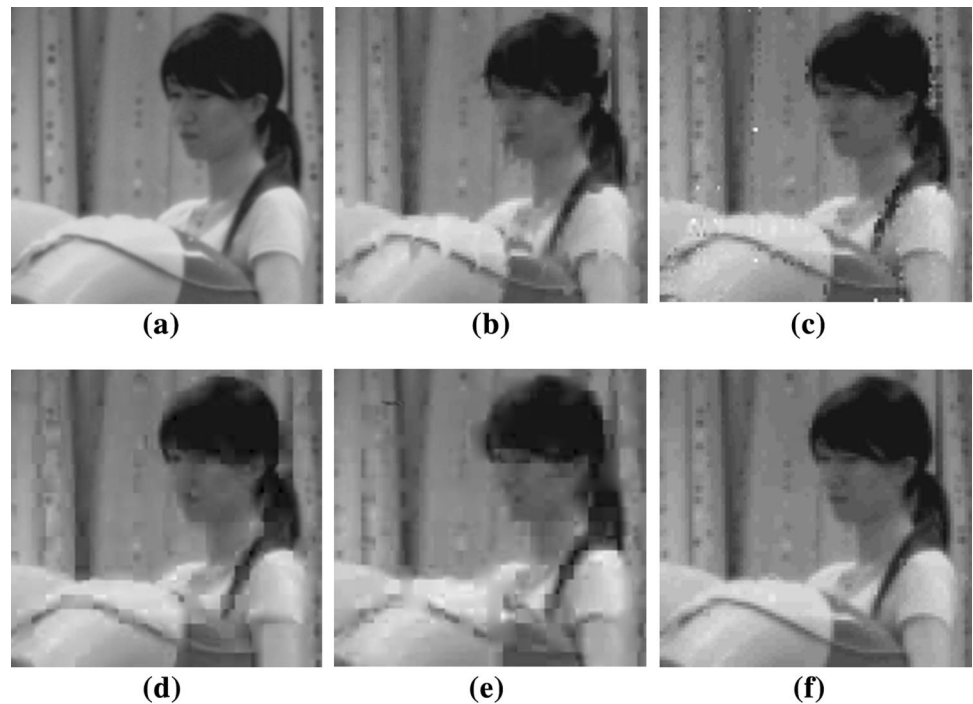
To verify the efficiency of the proposed method, six stereo images with the size of 640×480 are used to evaluate performance of the proposed method as illustrated in Fig. 4. Peak Signal to Noise Ratio (PSNR) and Peak Signal to Perceptual Noise Ratio (PSPNR) [28] are used to evaluate quality of watermarked stereo images and recovered stereo images, where PSPNR takes into account the distortion exceeds JND and reflects perceptual quality. Tamper Detection Ratio (TDR) [12] is used to show capability of tamper detection.

$$TDR = N_d/N \times 100 \% \tag{15}$$

where N_d and N are the number of valid blocks detected and the number of tampered blocks, respectively. In the proposed method, the parameters v_{p_t} in Eq. (1) and γ in Eq. (5) are empirically set to 0.5 and 0.1, respectively. For comparison of self-recovery capability, Lin's [10], Tong's [11] and Huo's [13] methods are extended to stereo image directly, that is, the left and right views are considered as two independent views. Moreover, they are extended further to stereo image that the procedure of tamper recovery is modified by using disparity $T = \{T_{i,j}\}$. T is the side information to recover tamper after recovery using extracted recovery references. These three revised methods are named as Lin's + T , Tong's + T and Huo's + T , respectively. Moreover, Yu's method [20] is also used to be compared.

Table 3 Comparison of recovery for 21.57 % of pasting tamper using PSNR [.dB]

Method	Lin's	Tong's	Huo's	Lin's + T	Tong's + T	Huo's + T	Yu's	Proposed
Bowling								
L	37.94	37.05	37.41	38.83	37.11	38.50	37.95	38.67
R	37.11	36.35	37.83	38.24	36.49	38.33	38.03	38.32
Akko								
L	39.77	38.02	39.47	40.26	38.05	39.58	39.03	40.12
R	39.40	38.06	39.67	40.40	38.08	40.47	39.55	40.53
Laptop								
L	35.57	35.95	35.77	36.50	36.03	36.19	35.90	36.73
R	36.61	36.05	36.34	36.94	36.44	36.41	36.37	37.13
Laundry								
L	33.24	34.32	32.14	34.34	34.47	32.63	34.10	35.26
R	33.38	34.19	33.90	34.54	34.28	34.19	34.61	35.21
Alt								
L	29.86	30.90	30.57	30.51	30.90	31.03	30.29	31.31
R	30.00	31.20	31.33	30.76	31.19	31.39	30.99	31.70
puppy								
L	33.42	34.06	31.39	34.40	34.12	32.28	34.28	34.89
R	33.11	33.66	33.52	33.99	33.76	33.62	34.56	34.92

Fig. 8 Local recovery comparison for left view of Akko. **a** original; **b** Lin's + T (27.99 dB); **c** Tong's + T (29.36 dB); **d** Huo's + T (28.76 dB); **e** Yu's (27.78 dB); **f** the proposed (31.74 dB)

PSNR and PSPNR of watermarked stereo images are shown in Table 2. For smooth stereo images, such as Bowling and Akko, low capability of watermark is hidden, and therefore their PSNRs are high. On the contrary, for textured stereo images, such as Laptop, Alt Moabit, Laundry and Puppy, more information can be embedded, and PSNR is a little low but PSPNR is still high to show the high visual quality. Although PSNRs of the proposed method

for Laptop, Alt Moabit and Puppy are lower than those of Tong's, PSPNRs of the proposed method are higher than those of Tong's. Compared with Tong's, it reveals that visual quality of the proposed method is better. It is mainly because two or three LSBs of each pixel are adaptively allocated for embedding watermark based on the JND model for the proposed method. But, in Tong's method, three LSBs of each pixel are fixed to embed watermark

Fig. 9 Left view of Laptop is tampered with different ratios. **a** 20 %; **b** 30 %; **c** 40 %; **d** 50 %; **e** 60 %; **f** 70 %

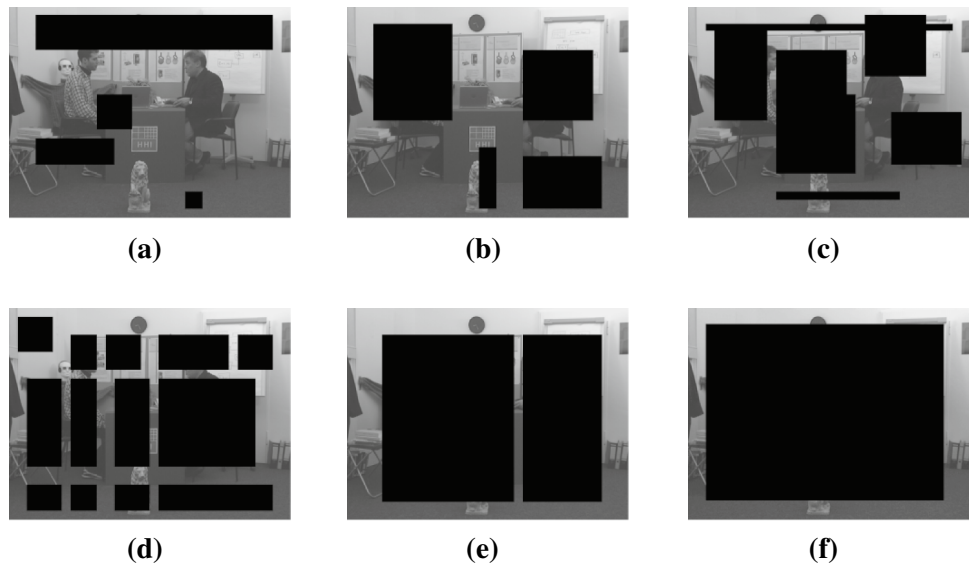
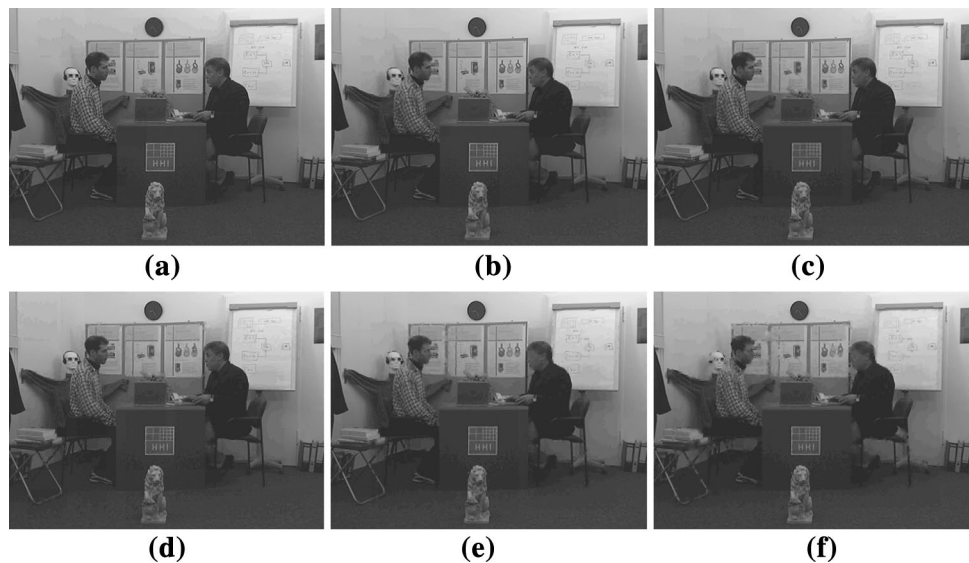


Fig. 10 Recovered left view of Laptop with different cropping ratios. **a** 20 %; **b** 30 %; **c** 40 %; **d** 50 %; **e** 60 %; **f** 70 %



without considering visual perception. It proves the superiority of watermark embedding with the guidance of the JND model.

3.1 Pasting tamper

In the following experiments, the left and right views are symmetrically tampered. Figure 5 shows that the left view of Akko is tampered at different number of locations. The first one is pasted with a clothes basket, the second one is modified by a tiger, and the last one is modified at six locations. Tamper ratios are expanding from 21.57 to 57.09 %. Other five stereo images are tampered in the same way as Akko. The proposed method nearly identifies tamper with TDR of more than 99.93 %. Five steps of asymmetric self-recovery are performed on the identified tampered blocks,

and both of views for each stereo image are reconstructed imperceptibly.

Figures 6 and 7 show PSNR and PSPNR comparisons of different eight stereo image watermarking methods for recovery of the tampered stereo image with different tamper ratios. PSNRs and PSPNRs of Lin's + T , Tong's + T , Huo's + T are almost higher than those of Lin's, Tong's and Huo's, respectively, and it shows the effectiveness of T . When the tamper ratio is 21.57 %, Fig. 6a shows that PSNRs of the proposed method are almost higher than those of other seven methods for both of views, and the specific data is shown in Table 3. But only PSNRs of Lin's + T are highest for left views of Bowling and Akko, and there are two main reasons. Firstly, T improves recovery much. Secondly, most of tampered pixels are located at smooth regions, neighbor pixels are used to recover tamper well. Although

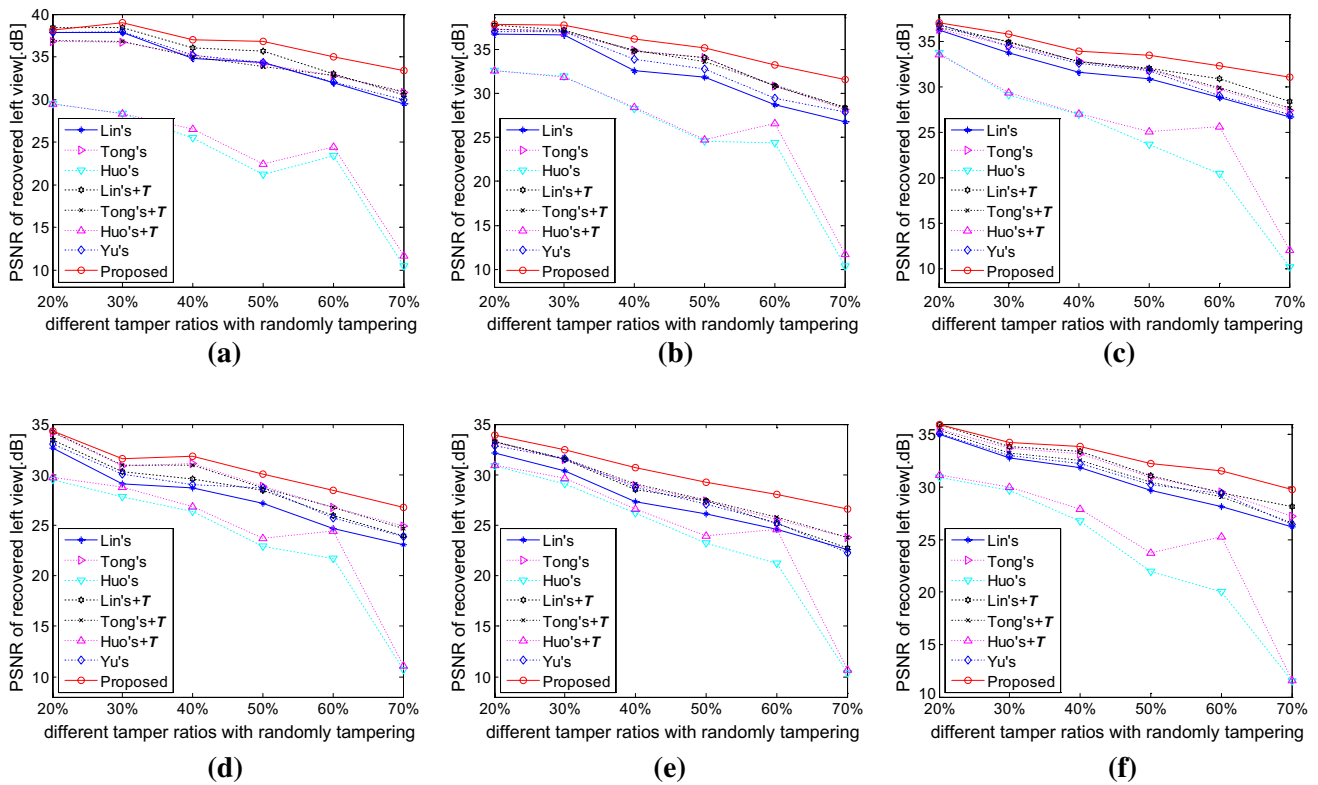


Fig. 11 PSNR of different watermarking methods for left views of different stereo images. **a** Bowling; **b** Akko; **c** Laptop; **d** Laundry; **e** Alt Moabit; **f** Puppy

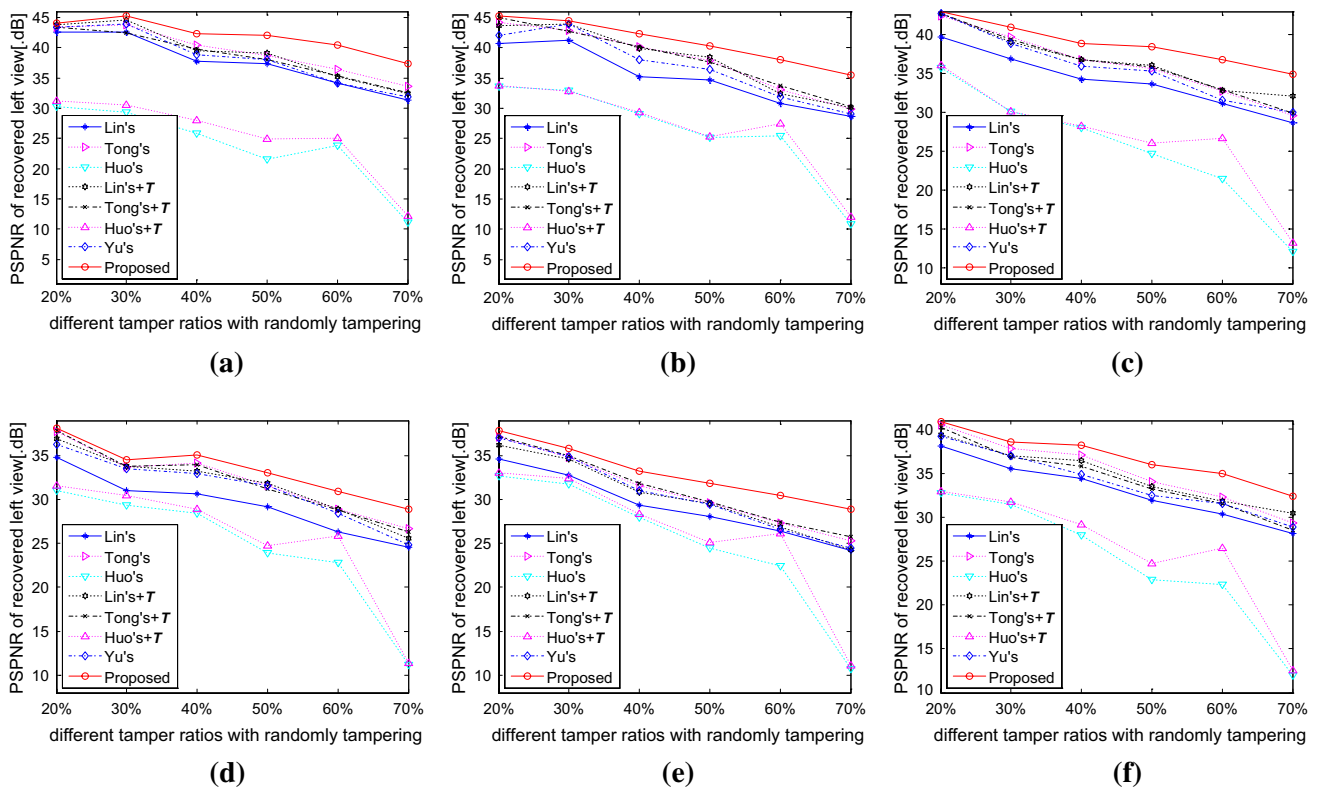


Fig. 12 PSPNR of different watermarking methods for left views of different stereo images. **a** Bowling; **b** Akko; **c** Laptop; **d** Laundry; **e** Alt Moabit; **f** Puppy

PSPNRs of the proposed method is still a little lower than those of Lin's + T for Akko as illustrated in Fig. 7a, but, Lin's + T should transmit T with extra bands, while the proposed method does not need to transmit it. In addition, PSPNRs of the proposed method are higher than those of those of other methods for different stereo images. It proves that the proposed method has good visual qualities of recovery. Although correlations between left and right views are used in Yu's method, watermarking capacity is not allocated properly, so that PSNRs and PSPNRs of Yu's are much lower than those of the proposed method. Thus, taking into account self-recovery capability and transmission load, the proposed method is superior to other seven watermarking methods.

As tamper is extended to 38.52 % as shown in Fig. 5b, PSNRs of the proposed method are nearly 1 dB higher than those of other seven methods for all tested stereo images. Especially for the left views of Laundry and Alt Moabit, PSNRs of the proposed method are nearly 2 dB higher than those of other seven methods. Moreover, the recovered quality of Lin's + T is not better compared with the proposed method. Due to extensive tamper, most of pairs of matched pixels may not be recovered after recovery using only one copy of embedded reference, and T is consequently declined in the role of recovery for Lin's + T . Compared with recovery of 21.57 % of tamper, recovered quality of the proposed method is not degraded as much as that of other seven methods. When tamper is increased to 57.09 %, PSNRs of the proposed method are 1.5 dB higher than those of other seven methods for six stereo tampered images, and especially 2.5 dB higher for corresponding complex stereo images Laptop, Laundry, Alt Moabit and Puppy as illustrated in Fig. 6c. It is because that matched pixels are better than neighbor pixels for recovery of complex regions. Compared with Fig. 6b, the proposed method is much superior on extensive tamper.

From Fig. 7b, c, PSPNRs of the proposed method are 1.5 dB higher than those of other seven methods, and most are 3 dB higher. It denotes that visual recovered quality of the proposed method is much better. Besides PSPNR, in order to prove subjective visual quality, local recovered images of Lin's + T , Tong's + T , Huo's + T , Yu's and the proposed watermarking methods for 38.52 % tamper of Akko are compared as illustrated in Fig. 8. Recoveries of Lin's + T , Tong's + T , Huo's + T and Yu's are vague, such as "head", "ball" and "life buoy". Tong's + T performs better than other five methods, and however, recovery is not as clear as that of the proposed method, such as "rope" around "life buoy" in Akko. Visual quality of the proposed method is the closest to the original image. Moreover, PSNR of local recovery is also computed as shown in Fig. 8, and PSNR of the proposed method is more than 2 dB higher than those of other four methods. Above experiments prove that the proposed method performs best on different ratios and locations of pasting tamper compared with other methods.

3.2 Cropping tamper

Cropping is another general tamper. Laptop is cropped from 20 to 70 % symmetrically with randomly tampering, and Fig. 9 shows tampered left view. In the experiments, pixels cropped are set to 0, and other five stereo images are cropped in the same ways. The proposed method can totally detect tamper with TDR of 100 %, and the recovered left views of Laptop are imperceptible as illustrated in Fig. 10. Lin's, Lin's + T , Tong's, Tong's + T , Yu's and the proposed method can totally detect tamper with TDR of 100 % as well. However, Huo's and Huo's + T methods miss detecting some tampered blocks, and especially for the tamper ratios of 70 % with TDR of nearly 80 %. Figures 11 and 12 show PSNRs and PSPNRs of different watermarking methods for recoveries of tampered left views, respectively, and similar recovered results are obtained for the right views. PSNRs and PSPNRs of Huo's and Huo's + T are worst because of the low TDR. PSNRs and PSPNRs of the proposed method are mostly higher than those of other six methods for different stereo images for different tamper ratios, and only are a little less than those of Lin's + T for 20 % tamper of Bowling. Small ratios of tamper on smooth regions and T help the performance of Lin's + T . However, Lin's + T still has the disadvantage that T needs extra bands to be transmitted. PSPNR of the propose method is a little higher than those of Lin's + T as illustrated in Fig. 12a and it denotes that recovered visual quality of the proposed method is better again. When tamper ratios are from 30 to 70 %, PSNRs of the proposed method are from around 0.1 dB to 1.5 dB higher than those of other six methods. For instance, it is 0.1 dB and 3.46 dB higher than those of Tong's + T for tamper ratios of 30 and 70 %, respectively. It proves that the proposed asymmetric self-recovery mechanism works better on extensive tamper again compared with small ratios of tamper. It is mainly because that difference between recoveries of two views using embedded reference is enlarged for large percents of tamper, and extracted disparity T helps self-recovery efficiently. For visual quality comparison, PSPNRs of the proposed method are higher than those of other seven methods as illustrated in Fig. 12, and the PSPNR increment is more obvious for large ratios of tamper. All cropping experiments prove that the proposed method has high capability of recovering extensive tamper.

4 Conclusions and future work

In this paper, the asymmetric self-recovery oriented stereo image watermarking method has been proposed. Making use of the inter-correlations between two views of stereo images, the proposed asymmetric self-recovery mechanism conducts not only asymmetric embedding for recovery references of

two views, but also asymmetric tamper self-recovery. Consequently, high quality of recovered stereo images can be obtained when stereo image is tampered with different ratios and locations. Moreover, to embed more recovery references, alterable bits are adopted to represent smooth and unsmooth blocks based on computing high frequency energy of discrete wavelet transform decomposed blocks. To obtain visual quality of the watermarked stereo image, just-noticeable difference (JND) is used to guide watermark embedding for reaching a trade-off between watermarking capacity and transparency. Experimental results prove that the proposed asymmetric self-recovery method outperforms other seven stereo image watermarking methods, and its superiority is more apparent if tamper is extensive.

The JND model only considers monocular vision properties, and however besides monocular masking the stereo image consisting of two views have binocular vision properties. Binocular vision can be further exploited to compute stereo masking, such as binocular combination and rivalry. Future work will focus on stereo vision based stereo image watermarking, where watermarking capacity will be increased further.

Acknowledgments This work was supported by Natural Science Foundation of China (Grant No. U13012572, 61171163, 61271270, 61311140262, 61175054, 61501270), Zhejiang Provincial Natural Science Foundation of China (Grant No. LY14F010004, LY15F010005), open fund of Zhejiang Provincial Key Academic Project (first level) (No. xkx1150 and F01739144200), and the Twelfth Five-Year Plan of Zhejiang province key discipline (No. 20121114). It was also sponsored by the K.C.Wong Magna Fund in Ningbo University.

References

1. Son, J.Y., Son, W.H., Kim, S.K., Lee, K.H., Javidi, B.: Three-dimensional imaging for creating real-world-like environments. *Proc. IEEE* **101**, 190–205 (2013)
2. Shao, F., Lin, W., Gu, S., Jiang, G., Srikanthan, T.: Perceptual full-reference quality assessment of stereoscopic images by considering binocular visual characteristics. *IEEE Trans. Image Process.* **22**, 1940–1953 (2013)
3. Hur, N., Lee, H., Lee, G.S., Lee, S.J., Gotchev, A., Park, S.I.: 3DTV broadcasting and distribution systems. *IEEE Trans. Broadcast.* **57**, 395–407 (2011)
4. Dragoi, I., Coltuc, D.: Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* **23**, 1779–1790 (2014)
5. Chen, W., Shahid, Z., Stütz, T., Atrusseau, F., LeCallet, P.: Robust drift-free bit-rate preserving H. 264 watermarking. *Multimed. Syst.* **20**(2), 179–193 (2014)
6. Al-Otum, H.M.: Semi-fragile watermarking for grayscale image authentication and tamper detection based on an adjusted expanded-bit multiscale quantization-based technique. *J. Vis. Commun. Image Represent.* **25**, 1064–1081 (2014)
7. Chang, C.C., Chen, K.N., Lee, C.F., Liu, L.J.: A secure fragile watermarking scheme based on chaos-and-hamming code. *J. Syst. Softw.* **84**, 1462–1470 (2011)
8. Zhang, X., Wang, S.: Fragile watermarking with error-free restoration capability. *IEEE Trans. Multimed.* **101**, 490–1499 (2008)
9. Qin, C., Chang, C.C., Chen, P.Y.: Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. *Sig. Process.* **92**, 1137–1150 (2012)
10. Lin, P.L., Hsieh, C.K., Huang, P.W.: A hierarchical digital watermarking method for image tamper and recovery. *Pattern Recogn.* **38**, 2519–2529 (2005)
11. Tong, X., Liu, Y., Zhang, M., Chen, Y.: A novel chaos-based fragile watermarking for image tampering detection and self-recovery. *Sig. Process. Image Commun.* **28**, 301–308 (2013)
12. Li, C., Wang, Y., Ma, B., Zhang, Z.: A novel self-recovery fragile watermarking method based on dual-redundant-ring structure. *Comput. Electr. Eng.* **37**, 927–940 (2011)
13. Huo, Y., He, H., Chen, F.: Alterable-capacity fragile watermarking scheme with restoration capability. *Opt. Commun.* **285**, 1759–1766 (2012)
14. Chammem, A., Mitrea, M., Prêteux, F.: Stereoscopic video watermarking: a comparative study. *Ann. Telecommun. Ann. Télécommun.* **68**, 673–690 (2013)
15. Hwang, D.C., Base, K.H., Kim, E.S.: Stereo image watermarking method based on discrete wavelet transform and adaptive disparity estimation. *Proc. SPIE* **5208**, 196–205 (2004)
16. Chammem, A., Mitrea, M., Prêteux, F.: DWT-based stereoscopic image watermarking. *Proc. SPIE* **7863**, 786326 (2011)
17. Yu, M., Wu, A., Luo, T., Jiang, G., Li, F., Fu, S.: New block-relationships based stereo image watermarking algorithm. *ICSNC* **2011**, 171–174 (2011)
18. Ou, Z.H., Chen, L.H.: A robust watermarking method for stereo-pair images based on unmatched block bitmap. *Multimed. Tools Appl.* 1–22 (2015). doi:10.1007/s11042-014-2433-0
19. Wang, J., Yu, M., Luo, T., Zhou, W., Jiang, G.: SVD-DWT based semifragile watermarking method for stereo image content authentication. *Adv. Mater. Inf. Technol. Process.* **87**, 381–384 (2014)
20. Yu, M., Wang, J., Jiang, G., Peng, Z., Shao, F., Luo, T.: New fragile watermarking method for stereo image authentication with localization and recovery. *AEU-Int. J. Electr. Commun.* **269**, 361–370 (2015)
21. Yang, X., Lin, W., Lu, Z., Ong, E., Yao, S.: Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile. *IEEE Trans. Circuits Syst. Video Technol.* **15**, 742–750 (2005)
22. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1124–1137 (2004)
23. Yang, C.H., Weng, C.Y., Wang, S.J., Sun, H.M.: Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Trans. Inf. Forens. Secur.* **3**, 488–497 (2008)
24. Arivazhagan, S., Ganesan, L.: Texture classification using wavelet transform. *Pattern Recogn. Lett.* **24**(9), 1513–1521 (2003)
25. Wang, L., Ye, Q., Xiao, Y., Zou, Y., Zhang, B.: An image encryption scheme based on cross chaotic map. *Int. Congr. Image Signal Process.* **3**, 22–26 (2008)
26. Zitnick, C.L., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 675–684 (2000)
27. Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**, 1019–1043 (2002)
28. Yang, X.K., Ling, W.S., Lu, Z.K., Ong, E.P., Yao, S.: Just noticeable distortion model and its applications in video coding. *Sig. Process. Image Commun.* **20**, 662–680 (2005)