Efficient Multilevel Reversible Data Hiding for Video Sequences Using Temporal and Spatial Approach

Kuo-Liang Chung*,[1], Wei-Jen Yang†, Ting-Chin Chang*, and Hong-Yuan Mark Liao‡,[2]
* National Taiwan University of Science and Technology, Taipei 10672 Taiwan R. O. C.
E-mail: {k.l.chung, M9615065}@mail.ntust.edu.tw Tel: +886-2-2737-6771
† National Taiwan University, Taipei 10617 Taiwan R. O. C.
E-mail: f93035@csie.ntu.edu.tw
‡ Academia Sinica, Taipei 11529 Taiwan R. O. C
E-mail: liao@iis.sinica.edu.tw Tel: +886-2-2788-3799 #1811, 1519

***Abstract*—Reversible data hiding can guarantee that the original image can be recovered from the marked image without any distortion. In this paper, an efficient multilevel reversible data hiding algorithm for video sequences is presented. Since the gray level distribution of the difference map is Laplacian, the peak point of the distribution thus leads into high data hiding capacity and good image quality. We also show that the peak signal to noise ratio (PSNR) lower bound of our proposed algorithm outperforms one by applying the previous best reversible data hiding algorithm the to each image frame in the video sequence directly. Based on four popular test video sequences, experimental results demonstrate the data hiding capacity and image quality advantages of our proposed data hiding algorithm.**

I. INTRODUCTION

Data hiding is an important technique for authentication, identification, annotation, and copyright protection [11]. In the past decade, for still images, many kinds of data hiding algorithms have been proposed, such as the least significant bit plane (LSB)-based algorithm [20], [22], [35], the spread-spectrum-based data hiding algorithm [10], [13], [16], [17], [24], [25], [26], [28], the singular value decomposition (SVD)-based algorithm [3], [6], [8], [23], the vector quantization (VQ)-based algorithm [2], [4], [18], and color ordering and mapping function (COMF)-based algorithm [34]. Among these developed data hiding algorithms, although the hidden data can be embedded into the cover image without causing visual degradation, it is very hard to recover the original image from the marked image once the original image has been modified during the data hiding process.

In order to overcome the above problem, some reversible data hiding algorithms [12], [15], [27], which can recover the original image without any distortion from the marked image after extracting the hidden data, were proposed. Then, in order to increase the embedding capacity, some capacity-efficient reversible data hiding algorithms [1], [5], [14], [36] were proposed. Although the above capacity-efficient reversible data hiding algorithms have higher embedding capacity, they suffer from the degradation of the PSNRs between the original image and the marked image.

Recently, based on some peak-valley pairs in the image histogram, Ni *et al.* [31] made a breakthrough for reversible data hiding on still images. Because of only modifying the gray values of concerned pixels by one, the PSNR lower bound of Ni *et al.*'s algorithm is much higher than those of the previous reversible data hiding algorithms. Currently, Chung *et al.* [9] presented a dynamic programming approach to determine the most suitable peak-valley pairs in order to maximize the embedding capacity. However, it mainly works well for road maps in practical applications. In [21], Lin *et al.* first partition the still images into a set of equal-size blocks. For each block, the difference operations are performed on the horizontally adjacent pixels. Then, based on the highest peak point, an efficient multilevel reversible data hiding algorithm was presented to increase the embedding capacity.

After examining the above previously published reversible data hiding algorithms, it is observed that they only focus on still images but do not refer to video sequences. One video sequence can be treated as a sequence of still images and any one of the above reversible data hiding algorithms can be adopted to embed the hidden data into each image frame independently. However, in this research, we plan to develop a new reversible data hiding algorithm for video sequences with better embedding capacity and PSNR when compared with such an approach.

In this paper, a new efficient reversible data hiding algorithm for video sequences is presented. Since for the video sequence, usually, thirty image frames are captured in one second, the difference between two consecutive image frames is rather small. We first demonstrate that the gray level distribution of the difference map is Laplacian. This demonstration also indicates that the peak point of the distribution is rather high and it leads into high data hiding capacity and good

[1]Corresponding author. Supported by National Council of Science of R.O.C. under contract NSC98–2221–E–011–128.

[2]As a visiting scholar, this work was done when the author visited Dept. of Computer Science and Information Engineering at National Taiwan University of Science and Technology from Jan. 2009 to March 2009.

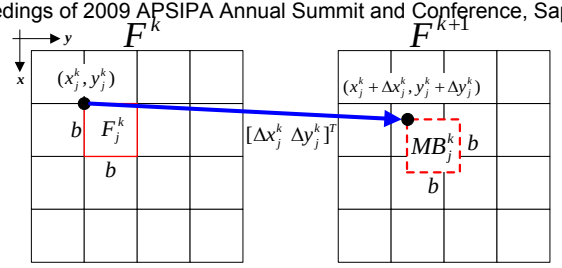


Fig. 1. The depiction of the relation between F_j^k and MB_j^k.

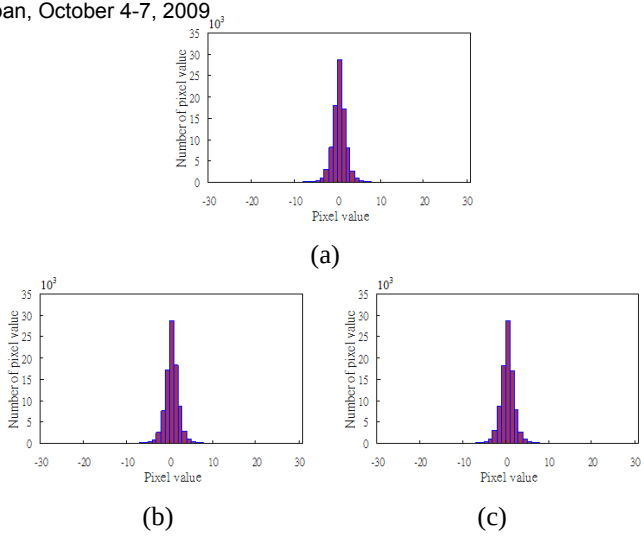


Fig. 2. For Mother and daughter video sequence, the three histograms of (a) D^1, (b) D^4, and (c) D^7.

image quality. Based on four popular test video sequences, experimental results demonstrate that our proposed data hiding algorithm has higher data hiding capacity and better image quality performance when compared with the one by applying the previous best reversible data hiding algorithm proposed by Lin *et al.* [21] to each image frame in the video sequence directly. In addition, we also show the PSNR lower bound superiority of our proposed algorithm; for the first embedding level, the PSNR lower bound is 48.13 dB.

The remainder of this paper is organized as follows. In Section II, the difference map between two adjacent image frames and the corresponding Laplacian distribution are discussed. In Section III, our proposed efficient multilevel reversible data hiding algorithm for video sequences and the PSNR lower bound analysis are presented. In Section IV, some experimental results are carried out to demonstrate the embedding capacity and image quality advantages of our proposed algorithm. Finally, some conclusions are addressed in Section V.

II. LAPLACIAN DISTRIBUTION OF THE DIFFERENCE MAP

This section demonstrates the difference map between two adjacent image frames and the corresponding Laplacian distribution. For convenience, the gray value of the pixel located at position (x, y) in the k-th image frame of the video sequence is denoted by $F^k(x, y)$ where the video sequence with n image frames is denoted by $V = \{F^i | \forall i \in \{1, 2, \ldots, n\}\}$.

Given two adjacent image frames F^k and F^{k+1}, we first divide F^k into a set of the blocks, each with size $b \times b$, and the block set is denoted by $B_{F^k} = \{F_j^k | \forall j \in \{1, 2, \ldots, m\}\}$ where F_j^k denotes the j-th block in F^k and m denotes the number of the total blocks in F^k. For each block $F_j^k \in B_{F^k}$, based on the motion estimation technique [7], [19], [29], [33], [37], its best matched block MB_j^k, which is in F^{k+1}, can be obtained by

$$MB_j^k(x', y') = F^{k+1}(x_j^k + \Delta x_j^k + x', y_j^k + \Delta y_j^k + y') \quad \text{for } x', y' \in \{0, 1, \ldots, b-1\} \tag{1}$$

where (x_j^k, y_j^k) denotes the corresponding coordinate of up-left corner of F_j^k on F^k and its two components, x_j^k and y_j^k, can be calculated by $x_j^k = b\lfloor \frac{j}{r_b} \rfloor$ and $y_j^k = b(j\%r_b)$, respectively; $\lfloor \cdot \rfloor$ denotes the floor operation; the symbol "%" denotes the modulus operator; r_b denotes the number of blocks in each row of F^k; Δx_j^k and Δy_j^k denote the two components of the motion vector $[\Delta x_j^k \ \ \Delta y_j^k]^T$. The depiction of the relation between F_j^k and MB_j^k is illustrated in Fig. 1. Next, the difference blocks $B_{D^k} = \{D_j^k | \forall j \in \{1, 2, \ldots, m\}\}$ can be generated by

$$D_j^k(x', y') = F_j^k(x', y') - MB_j^{k+1}(x', y') \quad \text{for } x', y' \in \{0, 1, \ldots, b-1\}$$

After generating all of the difference blocks in B_{D^k}, the difference map D^k can be constructed by $D^k = \bigcup_{j=1}^m D_j^k$. For Mother and daughter video sequence, Fig. 2 illustrates the three histograms of the difference maps and it is observed that the three histograms tally with the zero mean Laplacian distribution [32] which can be defined by

$$p(\tilde{D}^k) = \frac{\sqrt{2}}{2\sigma} \exp\left(-\frac{\sqrt{2}|\tilde{D}^k|}{\sigma}\right)$$

where the random variable $\tilde{D}^k$ is the random variable $D^k(x, y)$ which ignores the position parameter (x, y); σ is the standard deviation of the difference map.

Further, in order to increase the embedding capacity, for each pixel value in D_j^k, we apply the absolute value operation on D_j^k to obtain the absolute difference block $\hat{D}_j^k$, i.e.

$$\hat{D}_j^k(x, y) = |F_j^k(x', y') - MB_j^{k+1}(x', y')| \quad \text{for } x', y' \in \{0, 1, \ldots, b-1\}. \tag{2}$$

Then, the absolute difference map $\hat{D}^k$ can be constructed by $\hat{D}^k = \bigcup_{j=1}^m \hat{D}_j^k$. For Mother and daughter video sequence, Figs. 3(a)–(c) illustrate the three histograms of $\hat{D}^1$, $\hat{D}^4$, and $\hat{D}^7$, respectively. Comparing Fig. 2 with Fig. 3, it is observed that the peak points of the histograms in Fig. 3 are higher than the corresponding ones in Fig. 2. We therefore utilize the peak value, which is the corresponding gray value of the peak point in the histogram, in the absolute difference map to create the free space for embedding hidden data.

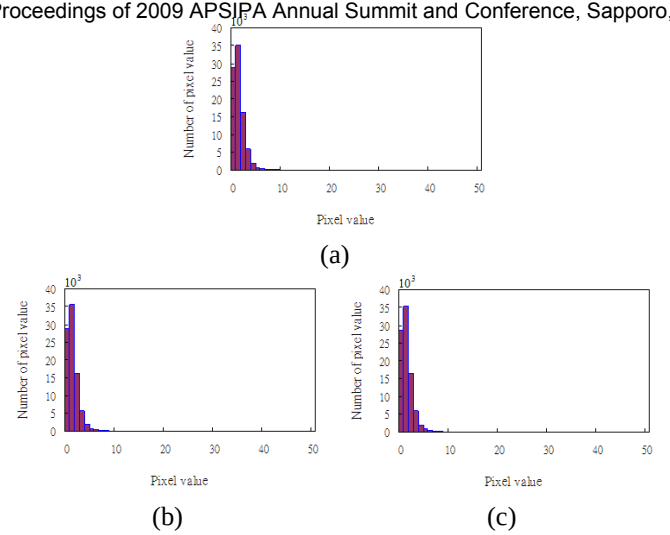


Fig. 3. For Mother and daughter video sequence, the three histograms of (a) $\hat{D}^1$, (b) $\hat{D}^4$, and (c) $\hat{D}^7$.

III. THE PROPOSED MULTILEVEL REVERSIBLE DATA HIDING ALGORITHM FOR VIDEO SEQUENCES

In this section, our proposed two-phase efficient multilevel reversible data hiding algorithm for video sequences is presented. In what follows, the embedding phase of the proposed algorithm is first presented in Subsection III.A. Then, the extracting and reversing phase is discussed in Subsection III.B.

A. The Embedding Phase

This subsection presents embedding phase of the proposed algorithm. In Subsection III.A.1, the procedure of the embedding phase is first presented. Then, the motion vector reuse scheme and the security strategy are discussed in Subsection III.A.2. Finally, the PSNR lower bound between the original video sequence and the marked video sequence is discussed in III.A.3.

1) The embedding procedure: Given a video sequence $V = \{F^i|\forall i \in \{1,2,\ldots,n\}\}$, the embedding phase consists of the following five steps:

Step 1: Initially, we set $k = 1$, and then go to Step 2.

Step 2: First, we divide F^k into a set of the blocks, each with size $b \times b$, and the block set is denoted by $B_{F^k} = \{F_j^k|\forall j \in \{1,2,\ldots,m\}\}$ where F_j^k denotes the j-th block in F^k and m denotes the number of the total blocks. Next, for each block $F_j^k \in B_{F^k}$, based on the motion estimation technique, its best matched block MB_j^k in F^{k+1} can be obtained by Eq. (1), and then the absolute difference block can be generated by Eq. (2). Consequently, we have the difference blocks $B_{\hat{D}^k} = \{\hat{D}_j^k|\forall j \in \{1,2,\ldots,m\}\}$.

Step 3: For each absolute difference block $\hat{D}_j^k \in B_{\hat{D}^k}$, the peak value P_j^k can be obtained easily. In order to embed the hidden bit h, based on the peak value P_j^k, the histogram modification technique [21], [31] is used to modify the pixel values in $\hat{D}_j^k$ by the follow rule:

$$M\hat{D}_j^k(x',y') = \begin{cases} \hat{D}_j^k(x',y')+1 & \text{if } \hat{D}_l^k(x',y') > P_j^k \\ \hat{D}_j^k(x',y')+h & \text{if } \hat{D}_j^k(x',y') = P_j^k \\ \hat{D}_j^k(x',y') & \text{otherwise} \end{cases}$$

for $j \in \{1,2,\ldots,m\}$ and $x',y' \in \{0,1,\ldots,b-1\}$

where $h \in \{0,1\}$. Consequently, we have the marked absolute difference blocks $B_{M\hat{D}^k} = \{M\hat{D}_j^k|\forall j \in \{1,2,\ldots,m\}\}$.

Step 4: Based on $M\hat{D}_j^k$ and its two related blocks, F_j^k and MB_j^k, the set of the marked blocks $B_{M^k} = \{M_j^k|\forall j \in \{1,2,\ldots,m\}\}$ can be generated by

$$M_j^k(x',y') = \begin{cases} F_j^{k+1}(x',y') + M\hat{D}_j^k(x',y') \\ \text{if } F_j^k(x',y') \geq MB_j^k(x',y') \\ \\ F_j^{k+1}(x',y') - M\hat{D}_j^k(x',y') \\ \text{otherwise} \end{cases}$$

for $j \in \{1,2,\ldots,m\}$ and $x',y' \in \{0,1,\ldots,b-1\}$.

After generating all the blocks in B_{M^k}, the k-th marked image frame M^k can be reconstructed by $M^k = \bigcup_{j=1}^{m} M_j^k$. Then, if the condition $k < n$ holds, $k = k+1$ and go to Step 2. Otherwise, go to Step 5.

Step 5: Since the image frame F^n is the last one, it implies that the image frame F^{n+1} does not exist in the video sequence. Thus, after dividing F^n into a set of the blocks $B_{F^n} = \{F_j^n|\forall j \in \{1,2,\ldots,m\}\}$, for each block $F_j^n \in B_{F^n}$, we obtain its best matched block MB_j^n from the (n-1)th marked image frame M^{n-1}, and then generate the absolute difference block by Eq. (2). Next, by the same way as Step 3 and Step 4, the hidden data can be embedded into F^n, and then we have the marked image frame M^n. Finally, the marked video sequence $V_m = \{M^i|\forall i \in \{1,2,\ldots,n\}\}$ is obtained.

Based on multilevel concept, our proposed reversible data hiding algorithm can be preformed iteratively to embed more hidden data. Further, after embedding the hidden data into image frames, the over/underflow problem may happen. Thus, for the overflow problem, when gray values of pixels in the marked image frames exceed 255, we set these gray values to be 255 and record the differences and the locations of these pixels. The similar strategy is applied to the underflow problem is the same. Based on the four test video sequences used in Section IV, experimental results indicate that even when the embedding levels are ten rounds, in average, only 0.039% pixels need to be corrected in the video sequence. Thus, the overhead information for overcoming the over/underflow problem is very small.

2) The motion vector reuse scheme and the security strategy: If we want to obtain the best matched block of each block

for each embedding level, the motion estimation process must be applied for each block to find the motion vector and motion vector must be record. However, recording the motion vectors would cause a large amount of overhead. For convenience, the above multilevel reversible data hiding approach is called the rollback data hiding scheme. Based on the four test video sequences used in the experiments, when the embedding levels are from 1 to 10, the average PSNR, the average bits per pixel (BPP), and the average overhead bits to hidden bits ratio (OHR) are 39.07, 1.394, and 49.50, respectively, where PSNR, BPP, and OHR for each video sequence including N frames, each with size $X \times Y$, are respectively defined by

$$\text{PSNR} = 10\log_{10}\frac{255^2}{\text{MSE}} \quad (3)$$

$$\text{BPP} = \frac{\#\{\text{hidden bits}\}}{NXY} \quad (4)$$

$$\text{OHR} = \frac{\#\{\text{overhead bits}\}}{\#\{\text{hidden bits}\}} \times 100\% \quad (5)$$

where $\text{MSE} = \frac{1}{NXY}\sum_{k=1}^{N}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[F^k(x,y) - M^k(x,y)]^2$; $F^k(x,y)$ and $M^k(x,y)$ denote the gray values of the pixels located at position (x,y) in the k-th frames of the original video sequence and the marked video sequence, respectively; $\#\{\text{hidden bits}\}$ and $\#\{\text{overhead bits}\}$ denote the numbers of the hidden bits and the overhead bits, respectively.

Although in average, the rollback data hiding scheme has good PSNR and BPP, however, the average OHR reaches 49.50%. In order to improve the poor OHR performance, we propose the motion vector reuse scheme to reduce the OHR caused by recording the motion vectors in the rollback data hiding scheme. The main concept of the motion vector reuse scheme is that we only apply the motion estimation process for each block to find its motion vector at the first embedding level, and then for other embedding levels, the motion vector obtained at the first embedding level is directly reused for each block to point out its matched block in the reference frame. Based on the same four test video sequences, when the embedding levels are from 1 to 10, Table I illustrates the average data hiding performance comparison between the rollback data hiding scheme and the motion vector reuse scheme. From Table I, it is observed that the average BPP and the average PSNR of the motion vector reuse scheme would be degraded by 8.45% ($= \frac{1.394-1.726}{1.394} \times 100\%$) and 1.83% ($= \frac{39.07-38.35}{39.07} \times 100\%$), respectively. However, the average OHR of the motion vector reuse scheme has 45.93% ($= \frac{49.50-26.76}{49.50} \times 100\%$) improvement ratio. Since the degraded BPP and PSNR of the motion vector reuse scheme are slight, but the OHR improvement is large, the motion vector reuse scheme is adopted in our proposed multilevel data hiding algorithm.

Next, the security strategy, which is used to enhance the security of the hidden data and allow only the authorized party to extract the hidden data, is discussed. Although our proposed data hiding algorithm embeds the hidden data sequentially rather than randomly, the peak value and motion vector of each

TABLE I

BASED ON THE FOUR TEST VIDEO SEQUENCES, THE AVERAGE DATA HIDING PERFORMANCE COMPARISON BETWEEN THE ROLLBACK DATA HIDING SCHEME AND THE MOTION VECTOR REUSE SCHEME.

	Comparison		
	BPP	PSNR	OHR
The rollback data hiding scheme	1.394	39.07	49.50
The motion vector reuse scheme	1.276	38.35	26.76

block and the over/underflow correcting information can be used as the secret key together. Adopting the security strategy in [21] we concatenate the all the peak values, motion vectors, and over/underflow correcting information are concatenated to be a secret key. In order to decrease the length of the overhead information, we choose the arithmetic coding [30] as the entropy coder to encode the secret key. Finally, the encoded secret key is transmitted to the receiver side for extracting the hidden data and recovering the original video sequence.

3) The PSNR lower bound: At present, the PSNR lower bound between the original video sequence and the marked video sequence is discussed. For a video sequences with N image frames, each with size $X \times Y$, the definition of PSNR has been given in Eq. (3). For each embedding level of the proposed algorithm, it is known that in the worst case, the peak value of each absolute difference block is 0 and all the hidden bits are 1. Thus, for each embedding level, in the worst case, all the pixel values in each absolute difference block should be increased by 1 and the MSE is accumulated by 1. It implies that when the embedding level is l, the MSE upper bound is l^2 and the PSNR lower bound is $10\log_{10}\frac{255^2}{l^2}$.

Theorem 1. *When the embedding level is l, the PSNR lower bound is* $10\log_{10}\frac{255^2}{l^2}$

For example, when the embedding level is 1, the PSNR lower bound is 48.13 dB ($= 10\log_{10}\frac{255^2}{1^2}$) and it is observed that the PSNR lower bound of our proposed algorithm is higher than the one by applying Lin *et al.*'s algorithm [21] to each image frame in the video sequence independently, and the PSNR lower bound of Lin *et al.*'s method is 42.69 dB. For simplicity, without confusion, the above Lin *et al.*'s algorithm for embedding hidden data into a video sequence is called Lin *et al.*'s method.

B. The extracting and reversing phase

This subsection presents the phase for extracting the hidden data and recovering the original video sequence from the marked video sequence. For convenience, the marked video sequence and the reversed original video sequence are denoted by $V_M = \{M^i | \forall i \in \{1,2,\ldots,n\}\}$ and $V_R = \{R^i | \forall i \in \{1,2,\ldots,n\}\}$, respectively. Unlike to the embedding phase mentioned above, the extracting and reversing phase is processed from the last image frame to the first image frame. The procedure of the extracting and reversing phase consists of the following six steps:

Step 1:The received secret key is decoded to correct the over/underflow pixels and to obtain the peak values

and motion vectors. Then, we set $k = n$ and go to Step 2.

Step 2: The marked image frame M^k is divided into a set of the blocks, each with size $b \times b$, and the block set is denoted by $B_{M^k} = \{M_j^k | \forall j \in \{1, 2, \ldots, m\}\}$ where M_j^k denotes the j-th block in M^k and m denotes the number of the total blocks.

Step 3: For each marked block $M_j^k \in B_{M^k}$, according to the decoded motion vector $[\Delta x_j^k \quad \Delta y_j^k]^T$, its best matched block MB_j^k can be obtained by

$$MB_j^k(x', y') = \begin{cases} M^{k-1}(x_j^k + \Delta x_j^k + x', y_j^k + \Delta y_j^k + y') \\ \text{if } k = n \\ \\ R^{k+1}(x_j^k + \Delta x_j^k + x', y_j^k + \Delta y_j^k + y') \\ \text{otherwise} \end{cases}$$

for $j \in \{1, 2, \ldots, m\}$ and $x', y' \in \{0, 1, \ldots, b-1\}$

where (x_j^k, y_j^k) denotes the corresponding coordinate of the up-left corner of M_j^k on M^k. Then, the set of the marked difference blocks $B_{M\hat{D}^k} = \{M\hat{D}_j^k | \forall j \in \{1, 2, \ldots, m\}\}$ can be constructed by

$$M\hat{D}_j^k(x', y') = |M_j^k(x', y') - MB_j^k(x', y')|$$

for $j \in \{1, 2, \ldots, m\}$ and $x', y' \in \{0, 1, \ldots, b-1\}$.

Step 4: According to the decoded peak value P_j^k, the hidden bits in $M\hat{D}_j^k$ can be extracted by

$$h = \begin{cases} 1 & \text{if } M\hat{D}_j^k(x', y') = P_j^k + 1 \\ 0 & \text{if } M\hat{D}_j^k(x', y') = P_j^k \end{cases}$$

for $j \in \{1, 2, \ldots, m\}$ and $x', y' \in \{0, 1, \ldots, b-1\}$.

Step 5: In order to reconstruct the original difference blocks, the pixel values in each marked difference block $M\hat{D}_j^k$ would be shifted and removed the hidden data by the following rule:

$$\hat{D}_j^k(x', y') = \begin{cases} M\hat{D}_j^k(x', y') - 1 \\ \text{if } M\hat{D}_j^k(x', y') \geq P_j^k + 1 \\ \\ M\hat{D}_j^k(x', y') \\ \text{otherwise} \end{cases}$$

for $j \in \{1, 2, \ldots, m\}$ and $x', y' \in \{0, 1, \ldots, b-1\}$.

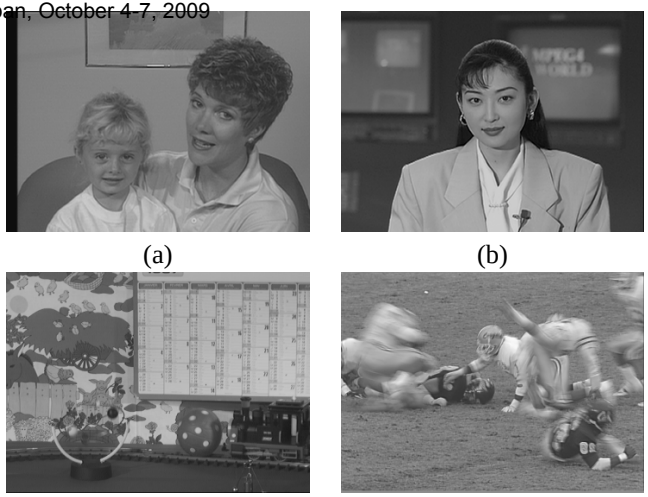
Consequently, we have the original difference blocks $B_{\hat{D}^k} = \{\hat{D}_j^k | \forall j \in \{1, 2, \ldots, m\}\}$.

Step 6: Based on $B_{\hat{D}^k}$, the reversed original block $B_{R^k} = \{R_j^k | \forall j \in \{1, 2, \ldots, m\}\}$ can be reconstruct by the following rule:

$$R_j^k(x', y') = \begin{cases} T_j^k(x', y') + \hat{D}_j^k(x', y') \\ \text{if } M_j^k(x', y') \geq T_j^k(x', y') \\ \\ T_j^k(x', y') - \hat{D}_j^k(x', y') \\ \text{otherwise} \end{cases}$$

for $j \in \{1, 2, \ldots, m\}$ and $x', y' \in \{0, 1, \ldots, b-1\}$

where $T_j^k = M_j^{k-1}$ if $k = n$; $T_j^k = R_j^{k+1}$, otherwise. After reconstructing all the blocks in B_{R^k}, the k-th reversed original image frame R^k can be reconstructed by $R^k = \bigcup_{j=1}^{m} R_j^k$. If the condition $k > 1$ holds, $k = k + 1$ and go to Step 2. Otherwise, all of the image frames in the reversed original video sequence V_R have been already reconstructed and stop the procedure.



(a) (b) (c) (d)

Fig. 4. The first image frames of (a) Mother and daughter video sequence, (b) Akiyo video sequence, (c) Calendar video sequence, and (d) Football video sequence.

After presenting our proposed multilevel reversible data hiding algorithm for video sequence, in next section, some experimental results will be carried out to demonstrate the embedding capacity and image quality advantages of our proposed reversible data hiding algorithm.

IV. Experimental results

In this section, based on four popular test video sequences, experimental results demonstrate that our proposed data hiding algorithm has higher data hiding capacity and better image quality performance when compared with the one by applying the previous best reversible data hiding algorithm proposed by Lin *et al.* [21] to each image frame in the video sequence directly. The concerned two algorithms are implemented on the IBM compatible computer with Intel Core 2 Duo CPU 1.8GHz and 2GB RAM. The operating system used is MS-Windows XP and the program developing environment is Borland C++ Builder 6.0.

Figs. 4(a)–(d) illustrate the first image frames of the four test video sequences, namely Mother and daughter video sequence, Akiyo video sequence, Calendar video sequence, and Football video sequence, respectively. For each video sequence, it includes thirty image frames, each with size 256×352. Based on the four test video sequences, the three data hiding performance measures, PSNR, BPP, and OHR, are utilized to evaluate the performance of the two concerned

TABLE II
FOR MOTHER AND DAUGHTER VIDEO SEQUENCE, THE DATA HIDING PERFORMANCE COMPARISON BETWEEN THE TWO CONCERNED ALGORITHMS.

	Embedding level									
Algorithm	1	2	3	4	5	6	7	8	9	10
(a) BPP comparison										
Lin *et al.*'s	0.286	0.494	0.664	0.810	0.939	1.056	1.161	1.259	1.349	1.434
The proposed	0.428	0.725	0.945	1.123	1.275	1.410	1.530	1.641	1.743	1.839
(b) PSNR comparison (dB)										
Lin *et al.*'s	45.11	39.54	36.30	34.01	32.21	30.74	29.51	28.44	27.49	26.65
The proposed	50.69	44.98	41.72	39.40	37.61	36.15	34.92	33.85	32.92	32.08
(c) OHR comparison (%)										
Lin *et al.*'s	11.48	16.06	19.57	22.76	25.76	28.64	31.44	34.16	36.81	39.39
The proposed	20.64	15.89	16.47	18.04	20.01	22.12	24.32	26.56	28.79	31.01

TABLE III
FOR AKIYO VIDEO SEQUENCE, THE DATA HIDING PERFORMANCE COMPARISON BETWEEN THE TWO CONCERNED ALGORITHMS.

	Embedding level									
Algorithm	1	2	3	4	5	6	7	8	9	10
(a) BPP comparison										
Lin *et al.*'s	0.306	0.528	0.709	0.864	1.000	1.123	1.233	1.335	1.429	1.517
The proposed	0.759	1.199	1.512	1.741	1.924	2.078	2.213	2.333	2.443	2.545
(b) PSNR comparison (dB)										
Lin *et al.*'s	45.37	39.76	36.55	34.25	32.44	30.96	29.71	28.64	27.69	26.84
The proposed	50.93	45.80	42.61	40.23	38.35	36.80	35.51	34.39	33.42	32.55
(c) OHR comparison (%)										
Lin *et al.*'s	10.19	14.60	17.96	20.96	23.77	26.47	29.10	31.69	34.24	36.75
The proposed	4.84	5.39	6.70	8.30	10.09	11.97	13.90	15.84	17.80	19.76

TABLE IV
FOR CALENDAR VIDEO SEQUENCE, THE DATA HIDING PERFORMANCE COMPARISON BETWEEN THE TWO CONCERNED ALGORITHMS.

	Embedding level									
Algorithm	1	2	3	4	5	6	7	8	9	10
(a) BPP comparison										
Lin *et al.*'s	0.244	0.425	0.572	0.700	0.812	0.913	1.006	1.091	1.170	1.244
The proposed	0.331	0.576	0.770	0.934	1.077	1.206	1.324	1.433	1.535	1.631
(b) PSNR comparison (dB)										
Lin *et al.*'s	45.48	40.06	36.79	34.46	32.67	31.20	29.97	28.90	27.96	27.12
The proposed	50.20	44.51	41.26	38.96	37.17	35.72	34.49	33.43	32.49	31.66
(c) OHR comparison (%)										
Lin *et al.*'s	13.70	19.06	23.33	27.19	30.83	34.34	37.74	41.05	44.27	47.40
The proposed	27.65	21.65	21.84	23.32	25.24	27.33	29.52	31.74	33.96	36.19

reversible data hiding algorithms. The definitions of PSNR, BPP and OHR have been given in Eqs. (3)–(5), respectively.

For fairness, the block size of the two concerned algorithms is set to 8×8 and the two generated secret keys are encoded by the arithmetic coding [30] to reduce the bit rate. Based on the four test video sequences, Tables II–V illustrate the data hiding performance comparisons in terms of BPP, PSNR, and OHR at various embedding levels among the two concerned algorithms. Tables II–V demonstrate the embedding capacity and image quality advantages of our proposed data hiding algorithm over the one by running Lin *et al.*'s data hiding algorithm on each image frame directly. For convenience, the latter method is called Lin *et al.*'s algorithm. Except embedding level 1, for most cases, our proposed data hiding algorithm has smaller OHRs. From the above comparative results, since Akiyo video sequence is a low motion video sequence, most peak values and motion vectors are 0's and it implies that the encoded secret key has very low bit rate. Thus, even when the embedding level is 1, the OHR in our proposed algorithm is still lower than that in Lin *et al.*'s method. Based on the same four test video sequences, Fig. 5 demonstrates that in average, our proposed data hiding algorithm has higher data hiding capacity and better image quality performance when compared with Lin *et al.*'s method. However, since at the first embedding level, the motion vector should be recorded, the advantage of lower OHRs in our proposed algorithm appears after the second embedding level.

Further, we compare the visual effect performance to

TABLE V

FOR FOOTBALL VIDEO SEQUENCE, THE DATA HIDING PERFORMANCE COMPARISON BETWEEN THE TWO CONCERNED ALGORITHMS.

	Embedding level									
Algorithm	1	2	3	4	5	6	7	8	9	10
(a) BPP comparison										
Lin *et al.*'s	0.163	0.292	0.403	0.502	0.592	0.675	0.752	0.825	0.893	0.958
The proposed	0.247	0.440	0.602	0.743	0.869	0.984	1.090	1.189	1.283	1.372
(b) PSNR comparison (dB)										
Lin *et al.*'s	44.35	38.84	35.59	33.28	31.49	30.03	28.80	27.73	26.79	25.96
The proposed	50.11	44.51	41.24	38.91	37.11	35.65	34.41	33.34	32.40	31.55
(c) OHR comparison (%)										
Lin *et al.*'s	29.83	35.35	40.26	44.70	48.86	52.81	56.58	60.21	63.70	67.11
The proposed	52.86	40.90	39.20	39.90	41.49	43.43	45.58	47.81	50.10	52.39

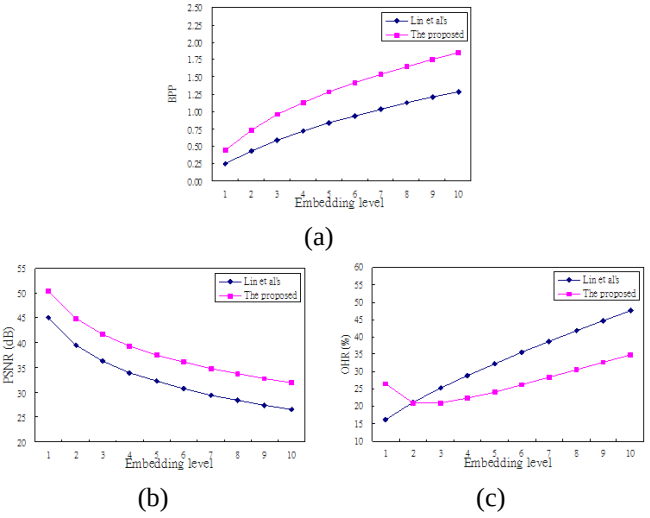


Fig. 5. The average hiding performance comparisons at various embedding levels between the two concerned algorithms. (a) BPP comparison. (b) PSNR comparison. (c) OHR comparison.

demonstrate the visual quality advantage of the proposed data hiding algorithm. We take the fifteenth (medium) image frame of Calendar video sequence as the test example and it is illustrated in Fig. 6(a). Fig. 6(b) and Fig. 6(c) illustrate the marked images obtained by running Lin *et al.*'s method and our proposed algorithm two rounds, respectively, i.e. the embedding level is 2. Fig. 6(d) and Fig. 6(e) illustrate the marked image frames by running both methods ten rounds. From Fig 6, it is observed that the large the number of rounds is, the better the visual effect of our proposed algorithm is. For Football video sequence, Fig. 7 confirms the similar visual advantage of our proposed algorithm.

Finally, we discuss the superiority of our proposed algorithm over the hybrid approach. In hybrid approach, for each block, we examine the embedding capacity by Lin *et al.*'s method and the embedding capacity by our proposed algorithm. We naturally select the maximal embedding capacity from the two capacities. However,for each block, it needs one extra bit to mark sure which method is adopted. Based on the same four video sequence, experimental results indicate that in average, the number of extra overhead bits is larger than the number of increasing embedding capacity. We therefore discard the hybrid approach.

V. CONCLUSIONS

In this paper, an efficient multilevel reversible data hiding algorithm for video sequences has been presented. Since the gray level distribution of difference map is Laplacian, the peak point of the distribution thus leads into high data hiding capacity and good image quality. Based on four popular test video sequences, experimental results demonstrate the embedding capacity and image quality advantages of the proposed algorithm when compared with the one by applying the previous best reversible data hiding algorithm proposed by Lin *et al* [21] to each image frame in the video sequence directly. In addition, our proposed algorithm has lower overhead bits to hidden bits ratio. The results of this paper can be applied to the field of sensitive video sequences, such as military video sequences, medical video sequences, artwork video sequence, etc, where the total reconstruction of the original video sequences ate demanded.

REFERENCES

[1] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in: *Proc. of the IEEE Int. Conf. on Image Processing*, Rochester, NY, vol. 2, pp. 157–160, 2002.

[2] C. C. Chang, G. M. Chen, and M. H. Lin, "Information hiding based on search-order coding for VQ indices," *Pattern Recognition Letters*, vol. 25, pp. 1253–1261, 2004.

[3] C. C. Chang, P. Y. Tsai, and M. H. Lin, "SVD-based digital image watermarking scheme," *Pattern Recognition Letters*, vol. 26, pp. 1577–1586, 2005.

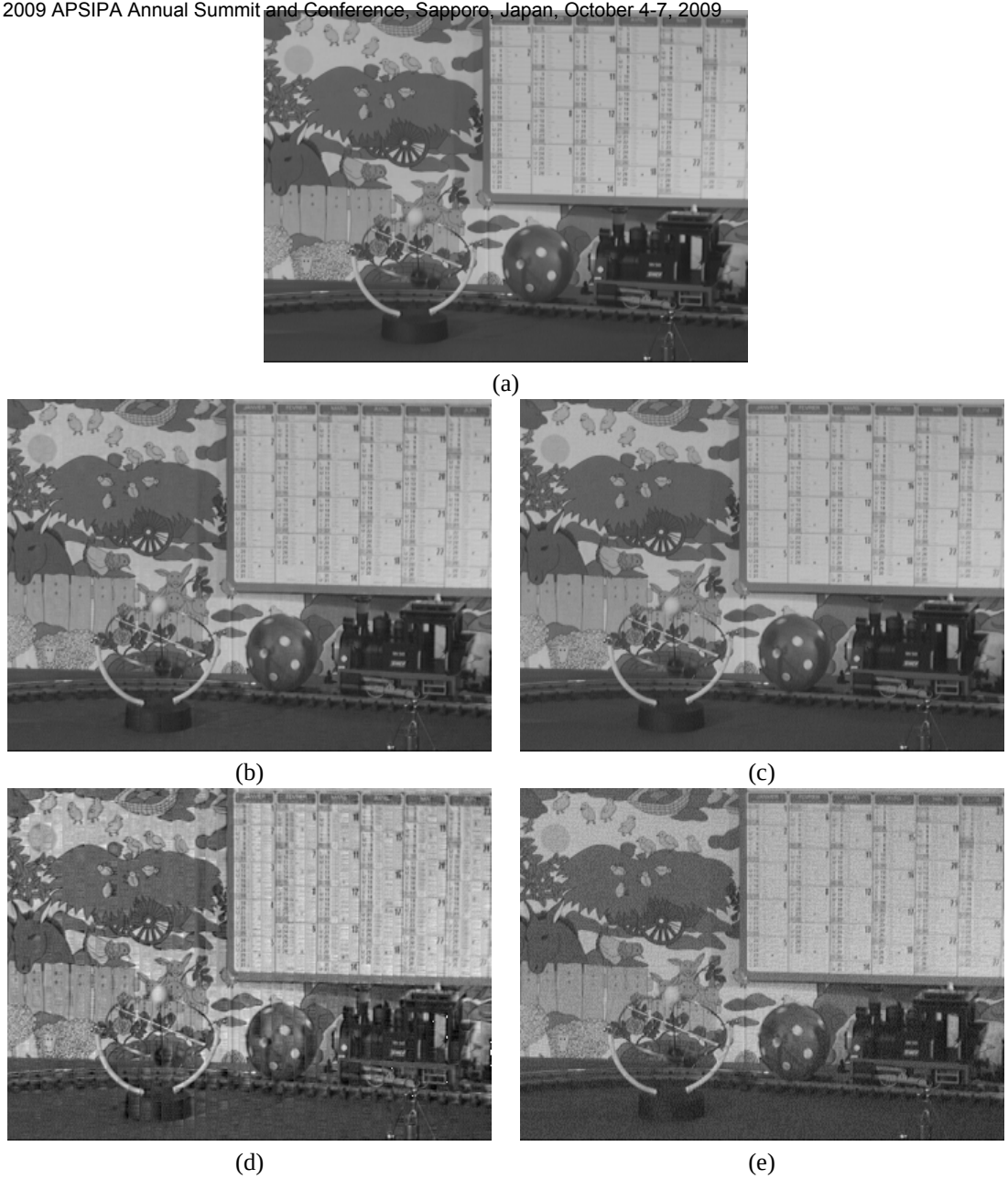
[4] C. C. Chang, W. L. Tai, and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, pp. 1301–1308, 2006.

[5] C. C. Chang, C. C. Lin, and Y. H. Chen, "A reversible data embedding scheme using differences between original and predicted pixel values," *IET Information Security*, vol. 2, pp. 35–46, 2008.

[6] K. L. Chung, C. H. Shen, and L. C. Chang, "A novel SVD- and VQ-based image hiding scheme," *Pattern Recognition Letters*, vol. 22, pp. 1051–1058, 2001.

[7] K. L. Chung and L. C. Chang, "A new predictive search area approach for fast block motion estimation," *IEEE Trans. Image Processing*, vol. 12, pp. 648–652, 2003.

[8] K. L. Chung, W. N. Yang, Y. H. Huang, S. T. Wu, and Y. C. Hsu, "On SVD-Based watermarking algorithm," *Applied Mathematics and Computation*, vol. 188, pp. 54–57, 2007.



(a) (b) (c) (d) (e)

Fig. 6. The fifteenth frame of Calendar video sequence. (a) The original image frame. The marked image frames obtained by running (b) Lin *et al.*'s method and (c) our proposed algorithm two rounds. The marked image frames obtained by running (d) Lin *et al.*'s method and (e) our proposed algorithm ten rounds.

[9] K. L. Chung, Y. H. Huang, W. N. Yang Y. C. Hsu, and C. H Chen, "Capacity maximization for reversible data hiding based on dynamic programming approach," *Applied Mathematics and Computation,* vol. 208, pp. 284–292, 2009.

[10] J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing,* vol. 6, pp. 1673–1687, 1997.

[11] I. J. Cox, M. Miller, and J. Bloom, *Digital Watermarking,* Morgan Kaufmann, San Francisco, CA, 2001.

[12] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in: *Proc. of the SPIE, Security and Watermarking of Multimedia Contents,* San Jose, CA, vol. 3971, pp. 197–208, 2001.

[13] M. Gkizeli, D. A. Pados, and M. J. Medley, "Optimal signature design for spread-spectrum steganography," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 16, pp. 391–405, 2007.

[14] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," in: *Proc. of the Fourth Int. Hiding Workshop,* Pittsburgh, PA, vol. 2137, pp. 27–41, 2001.

[15] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," US Patent# 6 278 791, 2001.

[16] C. T. Hsu and J. L. Wu, "Multiresolution Watermarking for Digital Images," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing,* vol. 45, pp. 1097–1101, 1998.

[17] C. T. Hsu and J. L. Wu, "Hidden Digital Watermarks in Images," *IEEE Trans. Image Processing,* vol. 8, pp. 58–68, 1999.

[18] M. Jo and H. D. Kim, "A digital image watermarking scheme based on vector quantization," *IEICE Trans. Information and System,* vol. E85-D, pp. 1054–1056, 2002.

[19] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-

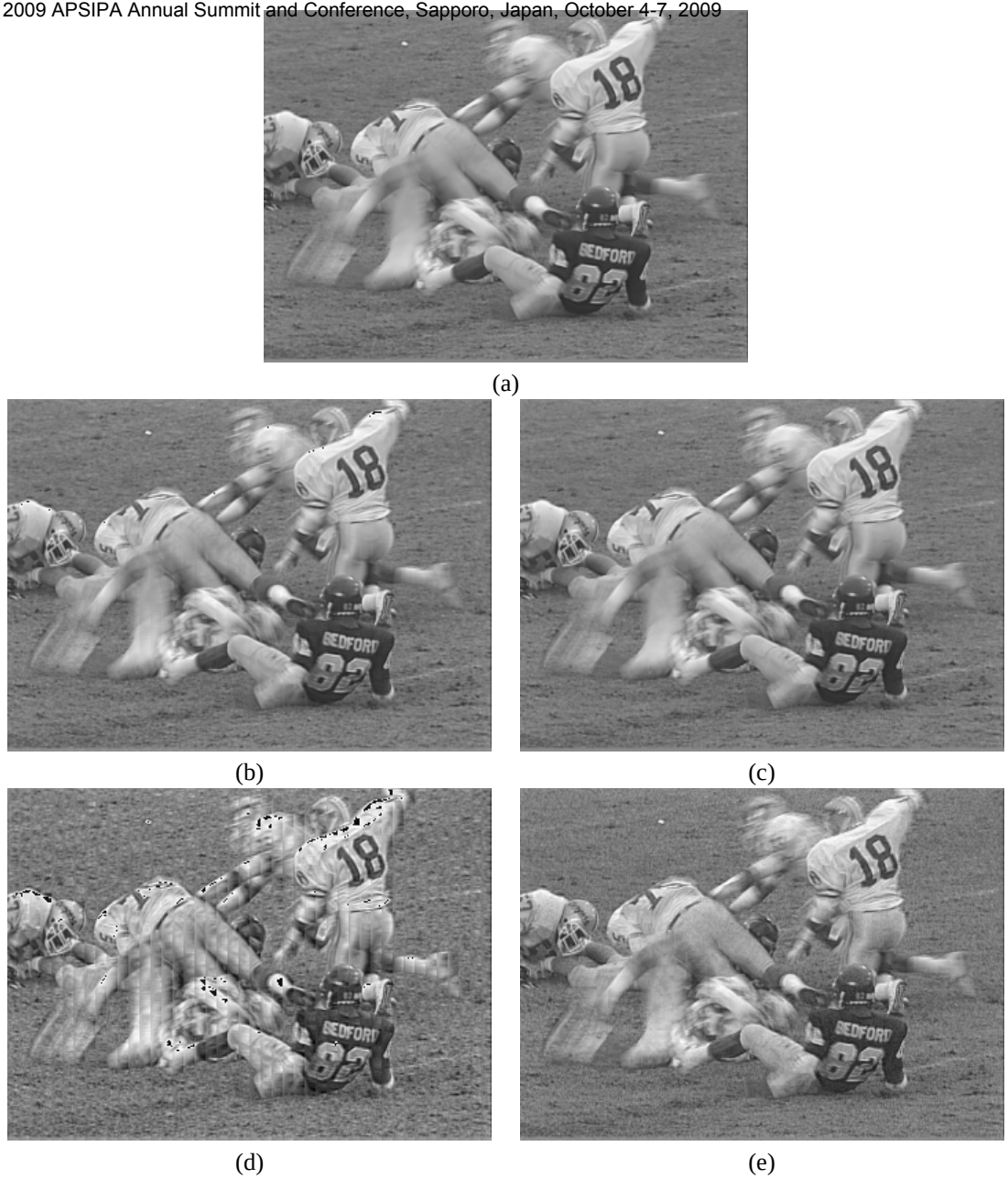


Fig. 7. The fifteenth frame of Football video sequence. (a) The original image frame. The marked image frames obtained by running (b) Lin *et al.*'s method and (c) the proposed algorithm for two rounds. The marked image frames obtained by running (d) Lin *et al.*'s method and (e) our proposed algorithm ten rounds.

compensated interframe coding for video conferencing," in *Proc. of Nat. Telecommunications Conf. (NTC)*, New Orleans, LA, vol. 81, pp. C9.6.1–9.6.5, 1981.

[20] S. L. Li, K. C. Leung, L. M. Cheng, and C. K. Chan, "Data hiding in images by adaptive LSB substitution based on the pixel-value differencing," in: *Int. Conf. on Innovative Computing, Information and Control*, vol. 3, pp. 58–61, 2006.

[21] C. C. Lin, W. L. Tai, and C. C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, pp. 3582–3591, 2008.

[22] P. L. Lin, C. K. Hsieh, and P. W. Huang, "Hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognition*, vol. 38, pp. 2519–2529, 2005.

[23] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimedia*, vol. 4, pp. 121–128, 2002.

[24] C. S. Lu, S. K. Huang, C. J. Sze, and H. Y. Mark Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. Multimedia*, vol. 2, pp. 209–224, 2000.

[25] C. S Lu, and H. Y. Mark Liao, "Multipurpose watermarking for image authentication and protection," *IEEE Trans. Image Processing*, vol. 10, pp. 1579–1592, 2001.

[26] C. S. Lu, J. R. Chen, and K. C. Fan, "Real-time frame-dependent video watermarking in VLC domain," *Signal Processing: Image Communication*, vol. 20, pp. 624–642, 2005.

[27] B. Macq and F. Deweyand, "Trusted headers for medical images," in: *Proc. of DFG VIII-DII Watermarking Workshop*, Erlangen, Germany, pp. 1–13, 1999.

[28] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image

steganography," *IEEE Trans. Image Processing*, vol. 8, pp. 1075–1083, 1999.

[29] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, New York: Chapman & Hall, 1997.

[30] A. Moffat, "Linear time adaptive arithmetic coding,"*IEEE Trans. Information Theory*, vol. 36, pp. 401–406, 1990.

[31] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, pp. 354–362, 2006.

[32] E. Parzen, *Modern Probability Theory and Its Applications*, New York: John Wiley and Sons, 1960.

[33] K. R. Rao and J. J. Hwang, *Techniques And Standards for Image, Video, and Audio Coding*, Englewood Cliffs, NJ: Prentice-Hall, 1996.

[34] C. H. Tzeng, Z. F. Yang, and W. H. Tsai, "Adaptive data hiding in palette images by color ordering and mapping with security protection," *IEEE Trans. Communications*, vol. 52, pp. 791–800, 2004.

[35] D. C. Wu and W. H. Tsai, "Spatial-domain image hiding using image differencing," in: *IEE Proc. Vision, Image and Signal Processing*, vol. 147, pp. 29–37, 2000.

[36] G. Xuan, J. Zhu, J. Chen, Y.Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEE Electronic Letters*, vol. 38, pp. 1646–1648, 2002.

[37] S. Zhu and K. K. Ma, "A new diamond searchalgorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287–290, 2000.