JOURNAL OF INFORMATION SECURITY AND APPLICATIONS 26 (2016) 39–50

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa

Detecting video frame-rate up-conversion based on periodic properties of edge-intensity

Yuxuan Yao [a], *Gaobo Yang* [a,*], *Xingming Sun* [b], *Leida Li* [c]

[a] *School of Information Science and Engineering, Hunan University, Changsha 410082, China*
[b] *School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 410082, China*
[c] *School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*

ARTICLE INFO

Article history:
Available online 6 January 2016

Keywords:
Video forensics
Frame-rate up-conversion (FRUC)
Video edge-intensity
Kaufman adaptive moving average (KAMA)

ABSTRACT

Video frame-rate up-conversion (FRUC) is one of the common temporal-domain operations. From the earlier frame repetition and linear interpolation, FRUC has been developed to motion compensated frame interpolation (MCFI), which effectively overcomes the temporal jerkiness and ghosting shadows. In a broad sense, FRUC can be regarded as a video forgery operation. By experiments, it is observed that FRUC still leads to edge discontinuity or over-smoothing artifacts around object boundaries. In this paper, an edge-intensity based passive forensics approach is proposed to detect the possible FRUC operation in candidate video. After computing the edge intensities of every frame, Kaufman adaptive moving average (KAMA) is exploited to define an adaptive threshold to distinguish the interpolating frames by FRUC from the original frames. Moreover, the original frame-rate of up-converted video can be inferred. Experimental results show that the proposed approach is not only effective for simple frame repetition and linear interpolation, but also valid for advanced FRUC techniques such as MCFI. The detection accuracy is up to 94.5% on average. Its computation is simple as well.

1. Introduction

The rapid development of various video editing tools such as VideoEdit facilitates the improvement of visual quality for digital video. However, it is becoming much easier for video tampering and forgery as well. As a consequence, the forensics techniques are anticipated to verify the authenticity and integrity of digital video (Edward et al., 2009; Li et al., 2015). Active forensics techniques require pre-embed auxiliary information such as digital watermark into videos (Tian et al., 2015), or pre-designed side information such as forensics hash (Wei et al., 2015) in advance, and then the tampering is determined then by detecting the integrity of pre-embedded auxiliary information or pre-designed side information. Passive forensics is to detect the inconsistent regularities or specific artifacts of digital video for forgery detection (Milani et al., 2012). For example, Wang and Farid (2009) proposed a video forgery detection approach by exposing double quantization artifacts. In addition, Wang and Farid (2007) proposed a forensic approach to detect traces of tampering in interlaced and deinterlaced videos. Subramanyam and Emmanuel (2012) presented a blind detection approach for the spatial and temporal copy paste tampering, which is based on Histogram of

* *Corresponding author.* School of Information Science and Engineering, Hunan University, Changsha 410082, China. Tel.: +86 0731 8882 3141; fax: +86 0731 8882 1907.
E-mail address: yanggaobo@hnu.edu.cn (G. Yang).
http://dx.doi.org/10.1016/j.jisa.2015.12.001

Oriented Gradients (HOG) feature matching and video compression properties. Subramanyam and Emmanuel (2013) also proposed a double quantization detection approach by exploiting the principles of estimation theory. Each pixel of a given frame is estimated from the spatially colocated pixels of all the other frames in a Group of Picture (GOP). The error between the true and estimated value is subjected to a threshold to identify the double compressed frame or frames in a GOP. Apparently, passive video forensics does not need any priori information, which makes it more suitable for practical applications. Thus, passive video forensics is recently becoming one of the hottest topics in the field of video information security (Rocha et al., 2011).

Compared with still image, digital video brings extra temporal dimension. Frame-based video manipulation is specific to digital video. Until now, there are some representative works about the blind forensics of frame duplication, frame deletion and frame-adding. Yang et al. (2014) presented a detection approach for frame duplication forgery using frame-level similarity analysis. Shanableh (2013) proposed a machine learning based detection approach for frame deletion. The discriminative features are based on prediction residuals, percentage of intra-coded macroblocks, quantization scales and reconstruction quality. Wang and Li (2014) presented an inter-frame forgery identification approach based on the consistency of correlation coefficients of gray values. Moreover, velocity field consistency and optical flow consistency are exploited to expose the video inter-frame forgery, respectively (Chao et al., 2013; Wu et al., 2014). Zernike opponent chromaticity moments and coarseness analysis are also exploited to expose video inter-frame forgery (Liu and Huang, 2015). Stamm et al. (2012) presented a theoretical model of the forensically detectable fingerprints that frame deletion or addition leaves behind. This model is further exploited in temporal forensics and anti-forensics for motion compensated video and better detection performances are achieved for frame deletion or addition than the approach by Wang and Farid (2007).

Frame-rate up-conversion (FRUC) is the procedure of increasing the frame-rate of a video by temporal interpolation of frames (Choi et al., 2000; Kang et al., 2007). From simple frame repetition and linear interpolation, FRUC has been developed to advanced motion compensated frame interpolation (MCFI) techniques. By introducing all kinds of assumptions to refresh the motion vectors and optimize the texture, MCFI can achieve better visual quality of the resultant video (Xue et al., 2015; Yoo et al., 2013). Apparently, FRUC is a special frame-adding operation, which is originally proposed to improve the visual quality of low frame-rate video. However, FRUC can also be used for video forgery purpose. For example, when two videos with different frame rates are needed to be spliced together, the low frame-rate video is usually up-converted by FRUC to match the relatively high frame-rate video. In recent years, the detection of video FRUC has attracted the attention from the community of video information security. Bian et al. proposed a similarity-analysis-based detection approach for frame duplication (Bian et al., 2014). After dividing the video sequence into overlapping sub-sequences, the similarities between the sub-sequences are calculated, which are exploited to identify those video sequences with high similarity as candidate duplication frames. However, it only reports the detection results of simple FRUC approaches such as frame repetition and linear weighting average. Moreover, Huang and Chen (2011) propose to a video forgeries detection approach based on bidirectional motion vectors. It does not investigate advanced FRUC techniques as well.

Actually, there are lots of advanced MCFI approaches in recent years (Choi et al., 2000; Kang et al., 2007; Stamm et al., 2012; Yoo et al., 2013). They consider the motion between successive frames by overlapped block motion compensation (OBMC) and adaptive motion compensation. These advanced FRUC techniques obtain more natural and realistic videos, which consequently bring extra technical challenges for their passive forensics. In essence, FRUC is a special type of frame-adding operation. The interpolated frames are obtained by block-based average, no matter whether the inter-frame motion is compensated or not. Therefore, FRUC inevitably leads to blurring artifacts to some extent, especially for those pixels near the edge. That is, the edge intensity might be decreased for those interpolated frames. Moreover, since the interpolated frames are periodically inserted into the original frames, the frame-level edge intensity of up-converted video will exhibit some periodicity along the temporal axis. Motivated by this, a novel passive forensics approach is proposed to detect FRUC by exploiting the temporal periodicity of frame edge intensity. A local adaptive threshold is determined by using Kaufman adaptive moving average (KAMA) (Kaufman, 1995). Since the adaptive threshold considers the dynamic change of video content, it can expose the abnormal change of edge intensity caused by frame interpolation and differentiate interpolated frames from the original frames. Moreover, since FRUC inserts the interpolation frames into the original frame periodically, the frame rate of original video sequence can also be inferred.

The rest of this paper is organized as follows. Section 2 briefly describes the FRUC techniques, and comparisons are made among them in terms of the visual qualities of interpolated frames. Section 3 presents the proposed blind detection approach. Section 4 discusses the experimental results and analysis. We conclude this paper in Section 5.

2. Preliminaries of video FRUC techniques

The simplest FRUC techniques are frame repetition (FR) and frame averaging (FA). They do not consider the motion between successive frames, which easily lead to temporal jerkiness and Ghosting shadow for non-static regions. To improve the visual quality, advanced MCFI techniques have been proposed in recent years. The basic idea behind MCFI is to estimate the motions as close as possible to the true motions by introducing various assumptions. Thus, more complex searching pattern is designed or the estimated motion vectors are refined to make the resultant up-converted video more realistic.

2.1. *Simple FRUC techniques*

Simple video FRUC approaches include frame repetition and linear interpolation (Tekalp, 1995), which can be modeled with a weighted linear averaging of forward and backward reference frames. That is,

$$f_n'(i,j) = \omega f_{n-1}(i,j) + \varphi f_{n+1}(i,j) \quad (1)$$

where (i, j) is the pixel position in the horizontal and vertical directions, respectively. $f_n(i, j)$ and $f_{n+1}(i, j)$ are the forward and backward reference frames, and $f(i, j)$ is the estimated interpolation frame. ω and φ are the weighting coefficients, and the sum of them is 1. When $\omega = 1$ and $\varphi = 0$ or $\omega = 0$ and $\varphi = 1$, it is apparent that Eq. (1) will be simplified to frame repetition. When ω and φ are not zeros, it indicates a linear frame interpolation. Especially, when $\omega = \varphi = 0.5$, it is a simple frame average. That is, frame average is actually a special case of linear frame interpolation.

2.2. MCFI

MCFI uses motion information between successive video frames to generate the interpolated frames. The accuracy of motion vectors, which is obtained by block-matching based motion estimation (ME), determines the visual quality of interpolated frame. However, the ME involved in MCFI is to estimate true motion for interpolated frame, instead of minimal prediction residual in the conventional ME of video encoder. The existing MCFI techniques can be divided into four categories as follows.

2.2.1. *Linear average compensation*

For every pixel in the interpolated frame, its grayscale value is computed by the linear averaging of corresponding pixels in the previous and successive frames, which are defined by the estimated motion vectors. This is the basic MCFI method.

2.2.2. *Motion compensation with median filter*

This approach is derived from linear average compensation. That is, the estimated motion vectors are firstly filtered with median filter. Then, the interpolated frame is generated similar to linear average compensation.

2.2.3. *OBMC*

By extending traditional ME, OBMC is employed in FRUC for its superior performance in reducing the blocking artifacts (Kang et al., 2007; Xue et al., 2015; Yoo et al., 2013). OBMC can generate much smoother interpolated frame. However, OBMC may result in blurring or over-smoothing artifacts in case of non-consistent motion regions since fixed weights are assigned for neighboring blocks.

2.2.4. *Adaptive OBMC (AOBMC)*

To better adjust the weights of OBMC, AOBMC is proposed to adjust the weights of different blocks in terms of the confidence of neighboring motion vectors (Lee et al., 2003). It actually integrates the four compensation methods including frame averaging without motion compensation, linear average compensation, OBMC and linear average compensation with multiple candidate vectors. The selection of motion compensation methods depends on the motion pattern of current block, which is determined by a motion analyzer. The motion analyzer divides image blocks into four types including static block, local moving block, boundary moving block and global moving block, which corresponds to the above four compensation modes, respectively. AOBMC can achieve better visual quality since it inherits the advantage of various compensation methods.

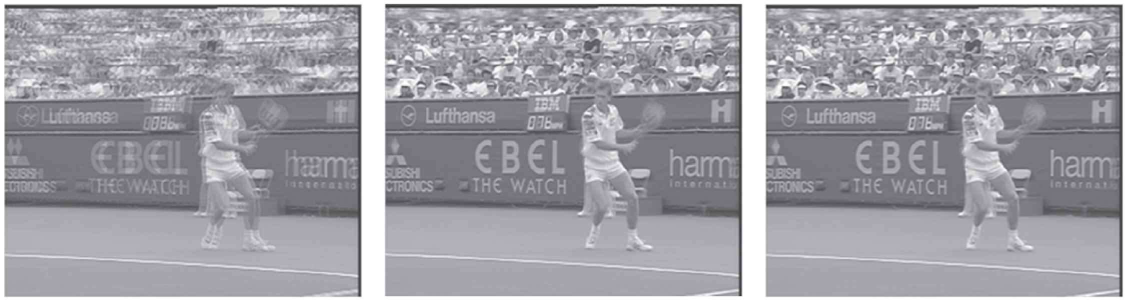
Fig. 1 compares the interpolated frame of *Foreman* sequence by FA and MCFI, where Fig. 1a shows the forward and backward frames for interpolation, Fig. 1b shows the interpolated frames by linear interpolation and frame averaging, and Fig. 1c shows the interpolated frames of the advanced MCFI

(a)Forward and backward frames

(b)Interpolated frame by linear interpolation and FA

(c)Interpolated frame by Yoo *et al* [20] and Xue *et al* [21]

Fig. 1 – Interpolated frame generated by advanced MCFI approaches.



(a) FA (b)Yoo *et al* [20] (c) Xue *et al* [21]

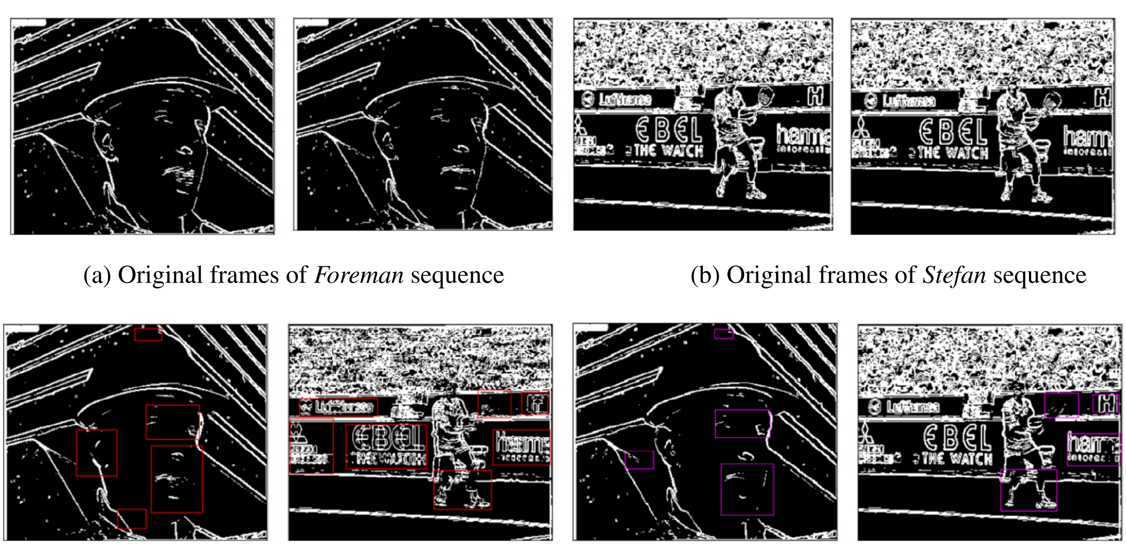
Fig. 2 – Interpolated frames of *Stefan* sequence by FA and MCFI.

approaches by Yoo et al. (2013) and Xue et al. (2015), respectively. To highlight the differences among different FRUC approaches, a block is selected from every frame and highlighted at the right side. In general, FR and FA can achieve desirable visual qualities when there is only slight motion in the original video sequence. However, for video sequences with complex motion, FR and FA will lead to temporal jerkiness and motion blur, respectively. Since MCFI exploits the motion information, it achieves better visual quality of interpolated frame than FA and FR. However, there still exist some artifacts such as motion blur and edge reduction in the interpolated frames. Fig. 2 shows the interpolated frames for *Stefan* sequence, which is a typical sequence with acute motion. It is apparent that there are serious motion blur and edge reduction. In the following, further analysis is made to find out the reasons behind the decrease of edge and texture details.

Both FA and MCFI generate the interpolated frame by pixel prediction from forward and backward frames. Specifically, they contain the mechanism of average and weighted average. This shows some similarity with image smoothing in spatial domain. Consequently, this leads to the loss of image edge information. Fig. 3 shows the edge detection results by Kirsch edge detector. The same original frame and interpolated frames in Fig. 1 are used. It can be observed that the edges of interpolated frames are less than those of adjacent original frames. The differences between them are shown in Fig. 3c and d, which are marked with red and purple boxes, respectively. It is apparent that there are more serious edge losses in Fig. 3c than Fig. 3d. These preliminary experimental results motivate us to differentiate the interpolated frames from the original video frames by utilizing the abnormal changes of edge-intensity along temporal axis as the clue for blind forensics. Actually, this is the main idea behind the proposed passive forensics approach for video FRUC.

3. Proposed blind forensics method

For digital video, there are high temporal correlations among adjacent frames. The edge-intensity of video frames is also consistent for the original sequence. After FRUC, the edge-intensity of interpolated frame will be significantly less than that of its adjacent original frame. That is, frame



(a) Original frames of *Foreman* sequence (b) Original frames of *Stefan* sequence

(c) Interpolated frames by FA (d) Interpolated frames by Yoo *et al* [20]

Fig. 3 – Edge detection results by Kirsch operator.

interpolation breaks the consistency of edge-density along the temporal axis. Moreover, since FRUC performs frame interpolation along the motion trajectory periodically to increase the frame rate, the discontinuity of edge-intensity also shows some periodicity. Thus, this kind of periodicity can be used to estimate the original frame rate of video sequence.

3.1. *Edge-intensity calculation*

Firstly, the original video sequence is converted into successive frames. Then, the luminance component of every frame is extracted, and Kirsch edge detector is used to detect the edges. Finally, the edge-intensity is calculated by counting the number of edge pixels. The steps of edge-intensity calculation are summarized as follows.

(1) Convert candidate video into consecutive frames $\{I_k\}$, where k is the frame index and m is the total number of frames, $k = 1, 2, \ldots, m$.
(2) Extract the luminance component of each frame I_k, which forms a new sequence $\{Y_k\}$.
(3) Extract the edge of luminance sequence $\{Y_k\}$ by Kirsch detector.
(4) Count the number of edge pixels $\{N_k\}$ in each frame.

Let S be the number of pixels in a frame, the edge-intensity is defined as follows.

$$\rho_k = \frac{N_k}{S} \tag{2}$$

Since frame interpolation contains the mechanism of averaging and weighted averaging, the calculation of edge intensity does not need any smoothing filter to remove image noise before edge detection. Moreover, image smoothing is contradictory to edge detection to some extent, since the averaging involved in frame interpolation leads to the losses of edge information. Considering the fact that the conventional edge detectors such as Sobel, Prewitt, Canny and Log are easy to smooth image edges, Kirsch operator is exploited for the calculation of edge intensity in this paper (Tekalp, 1995). Kirsch detector is a non-linear edge detector which finds the maximum edge magnitude in eight predetermined directions. Specifically, it takes a single kernel mask (as shown below) and rotates with an angular increment of 45.

$$g^{(1)} = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad g^{(2)} = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$g^{(3)} = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad g^{(4)} = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$g^{(5)} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad g^{(6)} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$g^{(7)} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad g^{(8)} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

The edge magnitude of traditional Kirsch operator is defined as the maximum magnitude across the above eight directions. That is,

$$h_{n,m} = \max_{z=1,\ldots,8} \sum_{i=-1}^{1} \sum_{j=-1}^{1} g_{ij}^{(z)} * p_{n+i,m+j} \tag{3}$$

where z enumerates the eight direction kernels, $p_{n+i,m+j}$ is a 3×3 pixel block and * is the convolution operation. For example, if p_A is the central pixel of a 3×3 pixel block, and p_i is its adjacent pixels $(i = 1, 2, \ldots, 8)$. Then, the first edge detection result d_1 is the convolution of the first Kirsch operator $g^{(1)}$ and the pixel block as follows.

$$\begin{aligned} d_1 = g^{(1)} * \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_A & p_5 \\ p_6 & p_7 & p_8 \end{bmatrix} &= 5p_1 + 5p_2 + 5p_3 - 3p_4 - 3p_5 - 3p_6 - 3p_7 - 3p_8 \\ &= 5(p_1 + p_2 + p_3) - 3(p_4 + p_5 + p_6 + p_7 + p_8) \end{aligned} \tag{4}$$

Similarly, $d_2, d_3, \ldots, d_8$ are obtained by utilizing the rest seven Kirsch operators $g^{(2)}$, $g^{(3)}$, …, $g^{(8)}$, respectively. Then, the maximum magnitude is found among $d_2, d_3, \ldots$ and d_8 as the output of original Kirsch operator. In this paper, the modulus of these eight edge magnitudes is defined as the final value of edge pixel *A*. That is,

$$d_A = \sqrt{d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 + d_7^2 + d_8^2} \tag{5}$$

Then, a threshold *T* is needed to judge whether pixel *A* is an edge pixel or not. If $d_A \geq T$, pixel *A* is judged as an edge pixel, and vice versa. Fig. 4 shows the edge detection results of *Flower* sequence after converting its frame rate from 15 fps (frames per second) to 30 fps by Yoo et al. (2013). Here, we manually set the threshold $T = 1200$ by experiments. It is apparent that Kirsch operator achieves the most significant difference of edge-intensity between the interpolated frames and the original frames. Thus, Kirsch edge operator is most suitable for our proposed forensics approach.

It is well-known that the video captured by different devices has different frame rates. For instance, the video captured mobile cameras are usually 15 fps or 20 fps, those taken by digital cameras are 25 fps, whereas some professional digital video recorders can capture videos with the frame rate as high as 30 fps or 60 fps. With the aid of FRUC approaches, the captured videos can be easily converted to high frame rate by inserting frames in different ways. Table 1 summarizes some typical ways of frame interpolation, where *num*_1 denotes that a frame generated by FRUC is inserted into the original video sequence every *num* frames. For instance, 3_1 means that a frame is inserted every 3 frames. (num_1, num_1) denotes that there are twice conversions. For example, (3_1, 4_1) means that a frame is firstly inserted into the original sequence every 3 frames, and then a frame is inserted every 4 frames.

Let ρ_k be the edge-intensity of the *k*th frame. It is previously claimed that the edge-intensity of interpolated frame is less than that of the original frame. Since the generated frames by FRUC are inserted into the original video sequence in a periodical way summarized in Table 1, the edge-intensities of all the frames will exhibit local minimum in a way similar to the

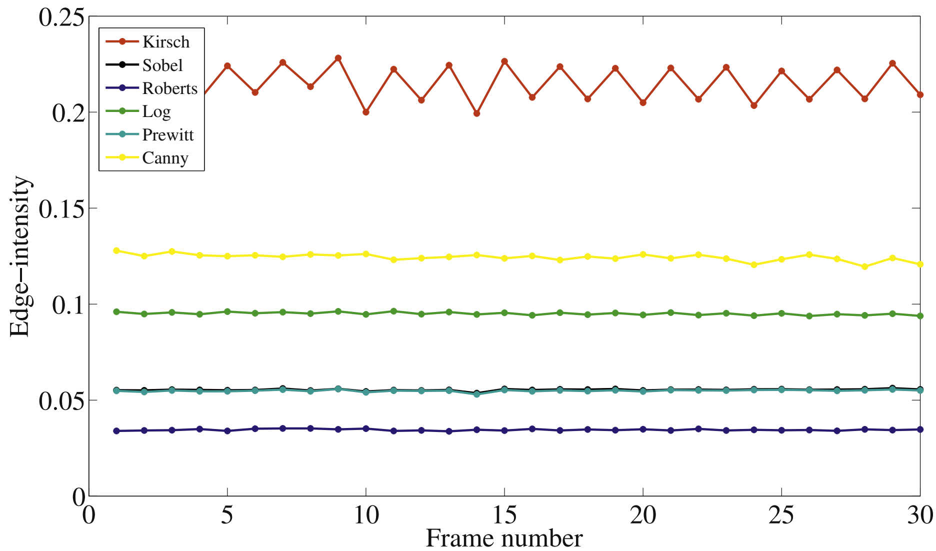


Fig. 4 – Edge detection by different operators.

Table 1 – Patterns of frame interpolation.

Original frame rate (fps)	Up-conversion patterns	Resultant frame rate (fps)
15	3_1	20
15	(3_1,4_1); (4_1,3_1); (3_2)	
20	4_1	25
15	1_1	30
20	2_1	30
25	5_1	30
15	(1_1,1_1); (1_3)	60
20	(2_1,1_1); (1_1,2_1); (1_2)	60
25	(5_1,1_1); (1_1,5_1)	60
30	1_1	60

pattern of frame interpolation in video FRUC. Fig. 5 shows the tendency of edge-intensity change for *Flower* sequence, which is up-converted from 20 fps to 25 fps using Yoo et al. (2013). The way of frame of interpolation is (4_1). The blue curve is the edge-intensity of the original video sequence, and the green curve is the edge-intensity of resultant video sequence after FRUC. (For interpretation of the references to color in this text, the reader is referred to the web version of this article.) Apparently, the edge-intensity of the original video sequence shows some continuity, but there is dis-continuity in the positions of interpolated frames. Specifically, the frame indexes of interpolated frames are 5, 10, 15 and so on. In those positions, there are local minimums for their edge-intensities, which seem like downward spikes. Moreover, these spikes exhibit obvious periodicity.

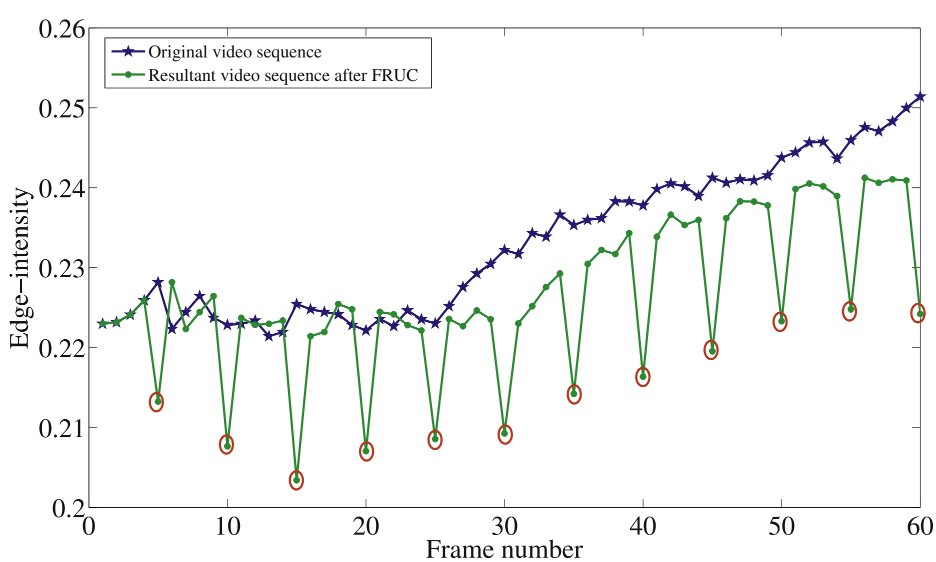


Fig. 5 – The comparison of edge-intensity for the original sequence and up-converted sequence.

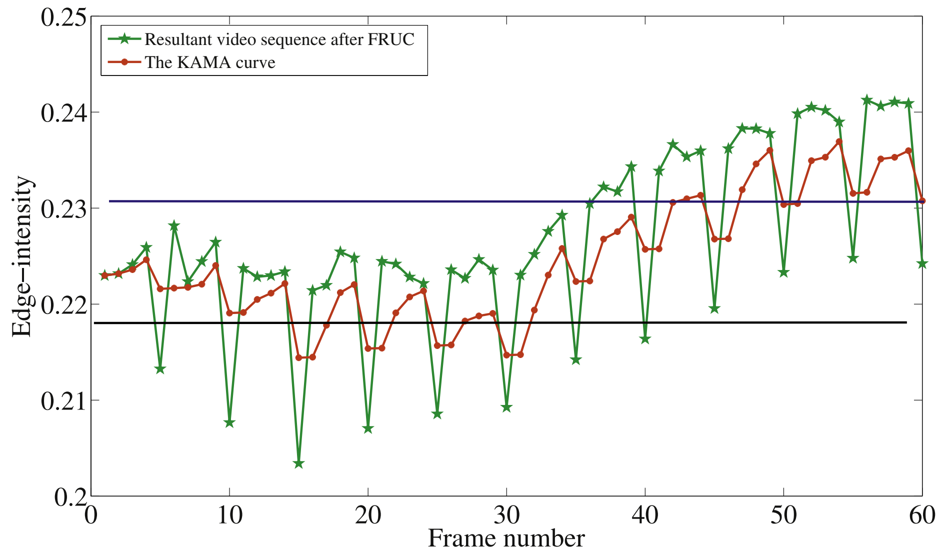


Fig. 6 – Threshold selection for distinguishing the interpolated frames from the original frames.

3.2. *Adaptive threshold for the localization of interpolated frames*

The local minimum of edge intensity provides significant clue for the detection and further location of interpolated frames. However, the downward spikes in Fig. 5 are actually local minimums. Because of the diversity of video content, especially the possible shot switching in video capturing, a fixed threshold is not suitable for the detection of these local minimums, which is closely related with the classification of original frames and interpolated frames for FRUC forensics. We also use *Flower* sequence as an example. Its frame rate is up-converted from the original 20 fps to 25 fps by Yoo et al. (2013), with the interpolation pattern of (4_1). Fig. 6 shows its edge-intensities before and after FRUC, where the red curve is the KAMA curve and the green curve is the resultant video after FRUC. Apparently, a relatively fixed threshold, no matter it is relatively high (blue line) or low (black line), inevitably leads to false detection. (For interpretation of the references to color in this text, the reader is referred to the web version of this article.) Therefore, a local adaptive threshold is a requisite for the detection of local minimums.

Let $\{\rho_k\}$ be the edge-intensity sequence, where $k = 1, 2, 3, \ldots, m$. Since $\{\rho_k\}$ varies significantly among different frames, the concept of "Moving Average", which is widely used in statistics analysis such as stock information, is borrowed to define an adaptive threshold. In general, there are two types of moving average, i.e., long-term moving average and short-term moving average. They are suitable for the analysis of long-term and short-term data trends, respectively. In this paper, Kaufman adaptive moving average (KAMA) (Kaufman, 1995) is exploited to define the adaptive threshold because it combines the long-term moving average and short-term moving average. The KAMA value for the kth frame with edge-intensity value $\{\rho_k\}$, which is referred as AMA_k, is defined in an iterative way as follows.

$$AMA_k = AMA_{k-1} + c_k \cdot (\rho_k - AMA_{k-1}) \tag{6}$$

where c_k is the smoothing factor for each iteration. When the variation of edge-intensity is small, the value of c_k should be adjusted to achieve long-term moving average. When the edge-intensity varies greatly among successive frames, the value of c_k should be adjusted to achieve short-term moving average. Thus, it can reflect the local trend of edge-intensity and detect the local abnormity of edge-intensity due to frame interpolation.

Let E_k be the *efficiency coefficient* of edge-intensity, which is defined as the ratio between the direction and the volatility of edge-intensity. The edge-intensity direction is the net variation of edge-intensity in the interval of *len* and the volatility of edge-intensity is the sum of edge-intensity variation in the interval of *len*. Thus, E_k is defined as follows:

$$E_k = \frac{|\rho_k - \rho_{k-len}|}{\sum_{i=k-len-1}^{k} |\rho_i - \rho_{i-1}|} \tag{7}$$

Then, c_k is defined as follows.

$$c_k = \left[E_k \cdot \left(\frac{2}{fastlen+1} - \frac{2}{slowlen+1}\right) + \frac{2}{slowlen+1}\right] \tag{8}$$

where *fastlen* and *slowlen* are the adjustment coefficients, which can follow the slow trend for the slow variation of edge-intensity and also follow the fast trend for the high variation of edge-intensity.

In this paper, the local minimums of frame edge-intensity after those videos after FRUC are utilized by combining KAMA to highlight the discontinuity and inconsistency of edge-intensity. Thus, the local average value of AMAs is decreased, which is bigger than those of interpolated frames but less than those of original frames. Fig. 6 shows the adaptive threshold curve, which is depicted as the red curve. (For interpretation of the references to color in this text, the reader is referred to the web version of this article.) Apparently, it is becoming easier to discriminate the interpolated frames from the original frames. Specifically, if the edge intensity ρ_k of the kth frame

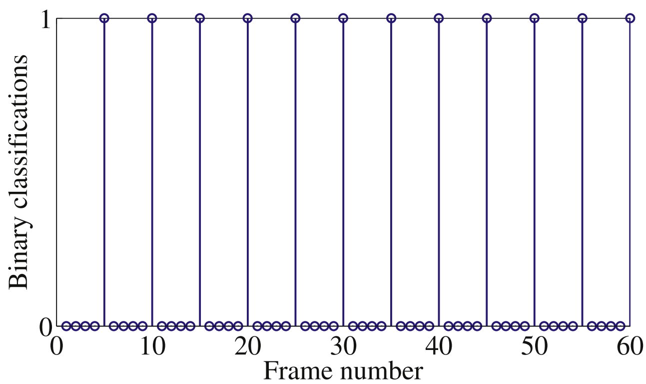


Fig. 7 – Binary classification of candidate video frames.

is less than the local threshold AMA_k, this frame is determined as the interpolated frame. On the contrary, if the edge intensity ρ_k of the kth frame is bigger than the local threshold AMA_k, this frame is determined as the original frame. Thus, we can obtain the binary classifications of all the frames in a candidate video sequence. Fig. 7 shows such an example after this kind of binary classification of interpolated frames and original frames. It is intuitive to observe the periodicity of interpolated frames for video FRUC.

However, due to the diversities of video FRUC approaches, it is difficult to always obtain such intuitive and ideal classification results. Actually, the complexity of video content, especially its motion complexity, is also closely related with the detection accuracy. Figs. 8 and 9 show the edge intensity along the temporal axis for *Stefan* and *Foreman* sequences after FRUC, respectively. FRUC adopts the approach by Yoo et al. (2013). The periodicity of interpolated frame is (3_1) in Table 1. That is, there is an interpolated frame every three original frames. In these two figures, the green curves are the edge intensities and the blue curves are their AMA values. (For interpretation of the references to color in this text, the reader is referred to the web version of this article.) It is obvious that for most frames, the difference between the edge intensity and its corresponding AMA value is not small. For some frames, their edge intensities are quite near to their corresponding AMA values, which are highlighted at the left side.

Fig. 10 shows the results of binary classification by comparing the edge intensity and its corresponding AMA value in Figs. 8 and 9, respectively. From it, we can easily observe the intervals between those frames which are classified as

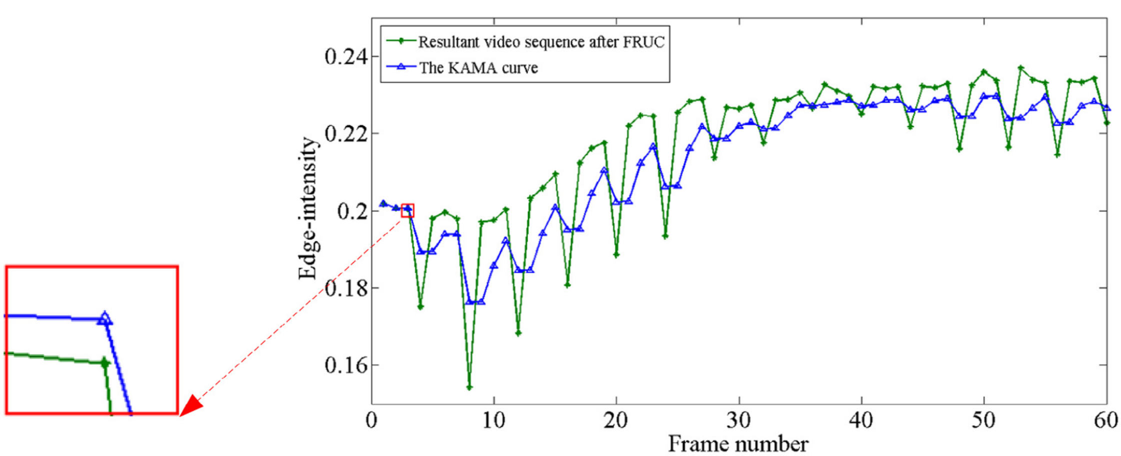


Fig. 8 – The edge-intensity and AMA curves of *Stefan* sequence after FRUC.

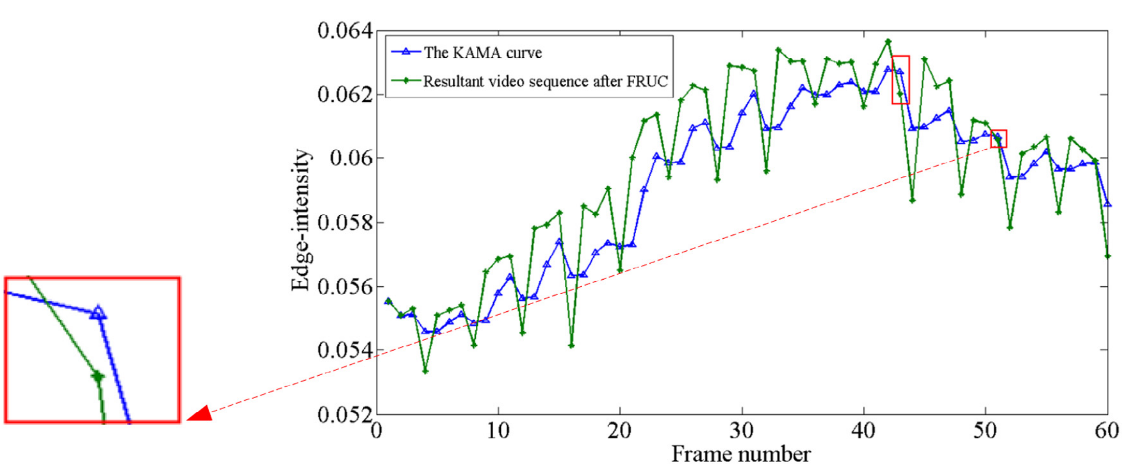


Fig. 9 – The edge-intensity and AMA curves of *Foreman* sequence after FRUC.

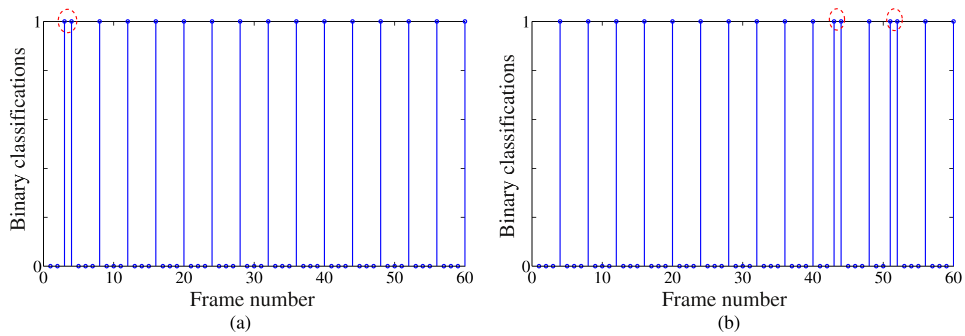


Fig. 10 – Binary classification of *Stefan* and *Foreman* sequences after FRUC.

interpolated frames. The majority of these intervals are the same, which actually reflect the periodicity of frame interpolation. However, there are also some exceptions, which are marked with red ovals. (For interpretation of the references to color in this text, the reader is referred to the web version of this article.) The intervals between two interpolated frames are too close. Actually, these are the cases where the differences between the edge intensity and its corresponding AMA value are too small, which leads to mis-classification.

3.3. *Estimation of the original frame rate*

After the binary classification of frames in candidate video, the interpolated frames are discriminated from the original frames. That is, the interpolated frames are located. Then, the original frame rate can be estimated from the up-converted video after FRUC. Let M be the total number of frames and N be the number of interpolated frames. If R_2 is the frame rate after FRUC, then the original frame rate R_1 can be inferred as follows.

$$R_1 = R_2 \cdot \left(1 - \frac{N}{M}\right) \tag{9}$$

Please note that the frame rates of digital video are usually fixed for some applications. The commonly-used frame rates are 15 fps, 20 fps, 25 fps, 30 fps and 60 fps, respectively. Therefore, we can choose a frame rate which is mostly close to the commonly-used frame rates.

4. Experimental results and analysis

To demonstrate the performance of the proposed approach, experiments are done on typical test video sequences. The hardware configuration is a personal computer (Intel(R) Pentium(R) CPU G630@2.70 GHz, 2.0 GB RAM), and the forensics detector is implemented with MATLAB R2014a. Until now, there is still no open video database of FRUC. Thus, we build a set of test video sequences for FRUC forensics by ourselves. In this experiment, we collect 15 uncompressed YUV sequences with different contents, motion and texture complexities. There are four typical categories of videos: single object with intense motion, single object with slight motion, more objects with intense motion, more objects with slight motion. They include *Flower*, *Bus*, *Stefan*, *Mobile*, *Foreman*, *Mother & Daughter*, *Coastguard* and so on. They are directly downloaded from http://trace.eas.asu.edu/yuv/. Their spatial resolutions vary from 176×144 to 1920×1080 pixels. Five commonly used frame rates including 15 fps, 20 fps, 25 fps, 30 fps and 60 fps have been tested. Then, these original video sequences are up-converted to different higher frame rates using the approach by Yoo et al. (2013). In summary, we obtain $15 \times 9 = 135$ video sequences after FRUC. Together with the 15 original sequences, there are 150 video sequences for experimental tests. Moreover, to show the detection performance against lossy video compression, H.264/AVC reference code JM10.2 is used to encode these video sequences into video streams. Thus, we also obtain 150 video sequences after video compression for test.

For the performance evaluation metrics, the well-known False Negative Rate (FNR) and the False Positive Rate (FPR) are used simply because they are widely used for pattern classification tasks. In this paper, the correct detection ratio (DR) is defined as follows.

$$DR = 1 - \frac{FNR + FPR}{2} \tag{10}$$

4.1. *Parameters selection*

From the definition of AMA_k in Eq. (6), it is easy to understand that this adaptive threshold depends on the selection of three parameters *len*, *fastlen* and *slowlen*. That is, these three parameters have great impacts on the detection performance of the proposed approach. *len* is a parameter used to define E_k. It has influences on the threshold selection when it takes different values. Fig. 11 shows the four KAMA curves when *len* is set with 1, 2, 3 and 4, respectively. It is apparent that when *len* takes different values, the degrees of edge-intensity curve's approximation to these KAMA curves are quite different. In general, with the increase of *len*, the approximation degree is reduced. *fastlen* and *slowlen* are the adjustment coefficients for the definition of smooth factor c_k, which achieves the adaptive moving average of video edge-intensity. And *fastlen* should be smaller than *slowlen*. In this paper, we choose $len = 2$, $fastlen = 2$ and $slowlen = 30$ by experiments.

4.2. *Experimental results*

Tables 2 and 3 show the detection results for uncompressed videos and H.264/AVC compressed videos in different

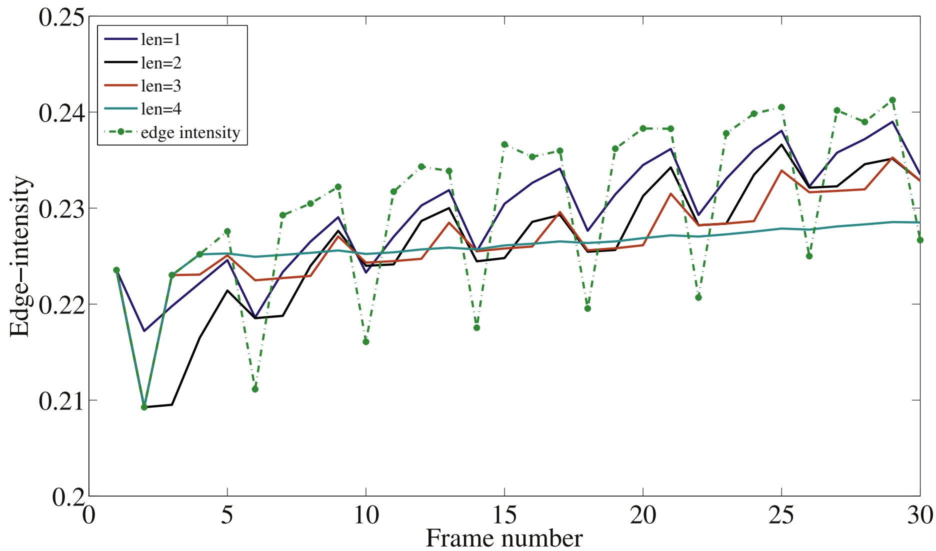


Fig. 11 – The influences of different values of *len* toward threshold selection.

interpolation patterns of FRUC (the 300 video mentioned above). From Table 2, we can see that all the DRs are above 93% and the average DR is 94.05%. We also note that for those test sequences with relatively lower motion, their DRs are also lower. For example, there is only slight motion around the woman's lip for *News* sequence. Thus, the difference of edge-densities between the original frame and the interpolated frames is also small. As a consequence, the local minimums for their edge-intensities are not obvious, which might confuse the binary classification of interpolated frames and original frames. Furthermore, it will also significantly confuse the inference of the interpolation period, simply because the estimated interpolation frames are not always correct. From Table 3, it can be known that all the DRs are above 92% and the average DR is 93.2%. Although they are a little lower that those in Table 2, they are still desirable for most practical applications. Actually, the slight decrease of DR is caused by the lossy video compression. Specifically, the lossy video compression by H.264/AVC will lead to a loss of edge and texture details. Thus, the accuracy of measured edge intensity is slightly decreased. In summary, the experimental results in Tables 2 and 3 show the effectiveness of the proposed approach.

Moreover, to further verify the effectiveness of the proposed forensics approach, different FRUC approaches are used to generate the high frame-rate videos. Specifically, five different FRUC methods include three motion compensation based FRUC approaches (MCFI) (Choi et al., 2000; Kang et al., 2007; Xue et al., 2015), linear frame interpolation and frame averaging are used in the experiments. The later two approaches are typical non-motion compensation approaches. Similar to the above experiments, these FRUC approaches are used to generate high frame-rate videos with different interpolation patterns. We found that the edge-intensities obtained by these FRUC approaches are coincident with each other. Especially for those MCFI methods (Choi et al., 2000; Kang et al., 2007; Xue et al., 2015; Yoo et al., 2013), their edge-intensities are very close to each other, which are shown in Fig. 12. In this figure, the MC-FRUC-EI is denoted as the edge-intensity obtained by MCFI, whereas NMC-FRUC-EI is denoted as the edge-intensity obtained by linear frame interpolation and frame averaging.

Table 2 – DRs of uncompressed video sequences (%).

(fps) Original frame rate	Resulting frame rate	Interpolation patterns	Accuracy (%) FPR	FNR	DR
15	20	(3_1)	0	13	93.5
15	25	(3_2)	13	0	93.5
20	25	(4_1)	0	13	93.5
15	30	(1_1)	0	13	93.5
20	30	(2_1)	0	15	92.5
25	30	(5_1)	0	13	93.5
15	60	(1_3)	13	0	93.5
20	60	(1_2)	0	13	93.5
30	60	(1_1)	0	13	93.5
Original video			0	0	100
Average			2.6	9.3	94.05

Table 3 – DRs for encoded video sequences by H.264/AVC(%).

(fps) Original frame rate	Resulting frame rate	Interpolation patterns	Accuracy (%) FPR	FNR	DR
15	20	(3_1)	0	15	92.5
15	25	(3_2)	15	0	92.5
20	25	(4_1)	0	15	92.5
15	30	(1_1)	0	15	92.5
20	30	(2_1)	0	16	92
25	30	(5_1)	0	15	92.5
15	60	(1_3)	15	0	92.5
20	60	(1_2)	0	15	92.5
30	60	(1_1)	0	15	92.5
Original video			0	0	100
Average			3.0	10.6	93.2

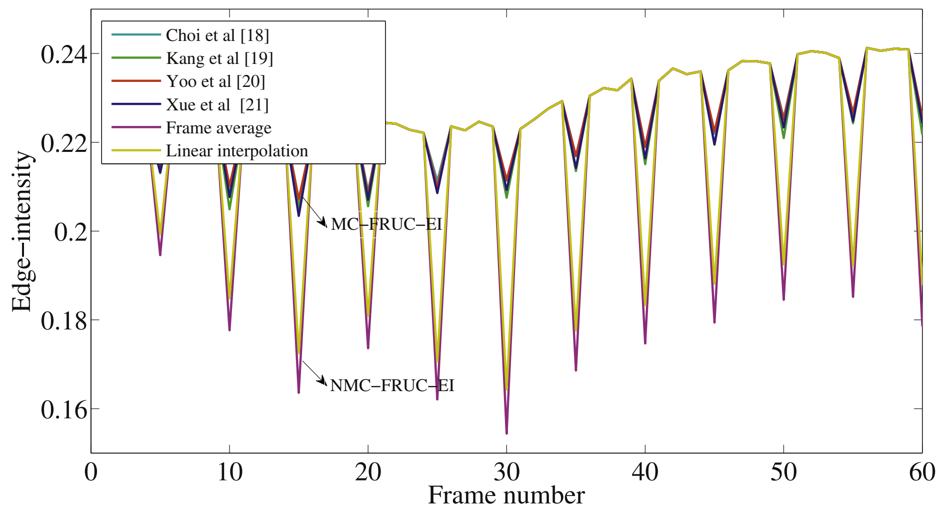


Fig. 12 – The edge-intensity obtained by different FRUC methods.

The performance of detection rates is summarized in Table 4. From Table 4, we can see that most accuracies are above 93%, and the average detection accuracy for these six FRUC methods is 94.5%. It is apparent that the detection ratios for frame averaging and linear interpolation are a little higher than the rest four FRUC approaches. This is simply because for frame averaging and linear interpolation, they do not consider the motion between video frames, thus the local minimums of edge-intensities are highlighted. Choi et al. (2000) utilize the overlapped block motion compensation (OBMC) technique to reduce the blocking artifacts. Kang et al. (2007) use the weighted index-based bidirectional MCFI and the OBMC scheme to reduce block artifacts. Yoo et al. (2013) and Xue et al. (2015) also use the popular bidirectional overlapped block motion compensated interpolation (OBMCI) method to generate the interpolated frames. The methods reduce the blocking artifacts, but over-smoothing artifacts around object boundaries still exist. Therefore, the proposed approach is effective for both motion compensation and non-motion compensation based FRUC approaches.

4.3. *Execution time*

Table 5 shows the execution times of the proposed algorithm for five test videos used in the experiments. Clearly, from Table 5, our algorithm exhibits an outstanding performance in time efficiency, and the average runtime of each frame is less than 0.1 s. The key reason for this result is that our algorithm uses the edge-intensity as the forensics solution of each frame and the KAMA algorithm is effective, which greatly reduces the amount of computation.

5. Concluding remarks and future works

FRUC is a popular temporal operation for digital video to improve the frame rate, which might be used for video forgery as well. From the earlier frame repetition and linear interpolation, FRUC has been developed to advance MCFI technique, which effectively reduces the temporal jerkiness and ghosting shadows. Based on our extensive experiments, we observe that by analyzing the edge-intensity variations among successive frames for a candidate video sequence, it is possible to unveil the interpolated frames and further estimate its periodicity. In this paper, an edge-intensity based passive forensics approach is proposed to detect the presence of FRUC in a candidate video sequence. The proposed approach is not only effective for simple frame averaging and linear frame interpolation, but also valid for advanced MCFI techniques. Moreover, the original frame-rate of up-converted video can be inferred as well. The detection accuracy is up to 94.5% on average. In future work, we will further investigate the

Table 4 – Detection accuracy (%) of different FRUC methods.

FRUC method	Accuracy (%)		
	FPR	FNR	DR
Choi et al. (2000)	4.1	8.3	93.80
Kang et al. (2007)	0	12.5	93.75
Yoo et al. (2013)	3.9	8.5	93.80
Xue et al. (2015)	0	12.3	93.85
Frame average	0	8.2	95.90
Linear interpolation	0	10.0	95.00
Average	1.3	9.7	94.50

Table 5 – Execution times of the test videos under the proposed algorithm.

Test videos	The length of video	Resolution	Execution times	
			Total time (s)	Time(s)/frame
Flower	60	352 × 288	5.191	0.087
Stefan	60	352 × 288	5.551	0.093
Foreman	60	352 × 288	4.918	0.082
Bus	60	352 × 288	4.998	0.083
Mobile	60	352 × 288	5.243	0.087

inherent mechanisms of advanced MCFI approaches and attempt to discriminate them from each other. If possible, we will also attempt to estimate the key parameters in various MCFI approaches.

Furthermore, we will investigate the least significant bit (LSB) features of interpolated frames for forensics. The reasons are two-folds: First, video forgery shows similarities with video steganagraphy, and thus the passive video forensics can benefit from the advancement of video steganalysis. Second, LSB features are frequently exploited by image and video steganalysis (Xia et al., 2014a, 2014b).

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (61379143, 61232016, U1405254), the Program for New Century Excellent Talents in University (NCET-11-0134), the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under grant 20120161110014, the S&T Program of Xuzhou City (XM13B119) and the PAPD fund.

REFERENCES

Bian S, Luo WQ, Huang JW. Detecting video frame-rate up-conversion based on periodic properties of inter-frame similarity. Multimedia Tools Appl 2014;72(1):437–51.

Chao J, Jiang X, Sun T. A novel video inter-frame forgery model detection scheme based on optical flow consistency. Digital forensics and watermaking. Springer.; 2013. p. 267–81.

Choi BT, Lee SH, Ko SJ. New frame rate up-conversion using bi-directional motion estimation. IEEE Trans Consum Electron 2000;46(3):603–9.

Edward D, Nasir M, Min W. Digital forensics (from the guest editors). IEEE Signal Process Mag 2009;26(2):14–15.

Huang TQ, Chen ZW. Digital video forgeries detection based on bidirectional motion vectors. Shandong Da Xue Xue Bao Li Xue Ban 2011;41(4):13–19.

Kang SJ, Cho KR, Kim YH. Motion compensated frame rate up-conversion using extended bilateral motion estimation. IEEE Trans Consum Electron 2007;53(4):1759–67.

Kaufman PJ. Smarter trading: improving performance in changing markets. McGraw-Hill; 1995. p. 138–51.

Lee SH, Kwon O, Park RH. Weighted-adaptive motion-compensated frame rate up-conversion. IEEE Trans Consum Electron 2003;49(3):485–92.

Li J, Li XL, Yang B, Sun XM. Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forens Secur 2015;10(3):507–18.

Liu Y, Huang T. Exposing video inter-frame forgery by Zernike opponent chromaticity moments and coarseness analysis. Multimedia Systems 2015;1–16.

Milani S, Fontani M, Bestagini P, et al. An overview on video forensics. APSIPA Trans on Signal and Inf Proc 2012;1(e1):1–18.

Rocha A, Scheirer W, Boult T, et al. Vision of the unseen: current trends and challenges in digital image and video forensics. ACM Comput Surv 2011;43(4):26–52.

Shanableh T. Detection of frame deletion for digital video forensics. Digital Investigation 2013;10(4):350–60.

Stamm MC, Lin WS, Liu KJ. Temporal forensics and anti-forensics for motion compensated video. IEEE Trans Inf Forens Secur 2012;7(4):1315–29.

Subramanyam AV, Emmanuel S. Video forgery detection using HOG features and compression properties. In: IEEE 14th international workshop on multimedia signal processing (MMSP). 2012. p. 89–94.

Subramanyam AV, Emmanuel S. Pixel estimation based video forgery detection. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). 2013. p. 3038–42.

Tekalp AM. Digital video processing. Englewood Cliffs, NJ: Prentice Hall; 1995.

Tian L, Zheng N, Xue J, et al. Authentication and copyright protection watermarking scheme for H. 264 based on visual saliency and secret sharing. Multimedia Tools Appl 2015;74(9):2991–3011.

Wang Q, Li ZH. Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. Journal of Comput and Comm 2014;2:51–7.

Wang W, Farid H. Exposing digital forgeries in interlaced and deinterlaced video. IEEE Trans Inf Forens Secur 2007;2(3):438–49.

Wang W, Farid H. Exposing digital forgeries in video by detecting double quantization. In: Proceedings of ACM 11th workshop on multimedia and security. 2009. p. 39–48.

Wei F, Yang G, Li L, Xia M, Zhang D. Detection of seam carving based video retargeting using forensics hash. Secur Commun Netw 2015;8(6):2102–13.

Wu Y, Jiang X, Sun T, et al. Exposing video inter-frame forgery based on velocity field consistency. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). 2014. p. 2674–8.

Xia ZH, Wang XH, Sun XM, Wang BW. Steganalysis of least significant bit matching using multi-order differences. Secur Commun Netw 2014a;7(8):1283–91.

Xia ZH, Wang XH, Sun XM, Liu QS, Xiong NX. Steganalysis of LSB matching using differences between nonadjacent pixels. Multimedia Tools Appl 2014b;1–16.

Xue CL, Li R, Zhu XC. Multiple features matching based bidirectional motion estimation for frame rate up-conversion. Video Engineering 2015;39(1):19–23.

Yang JM, Huang TQ, Su LC. Using similarity analysis to detect frame duplication forgery in videos. Multimedia Tools Appl 2014;1–19.

Yoo DG, Kang SJ, Kim YH. Direction-select motion estimation for motion-compensated frame rate up-conversion. Journal of Display Tech 2013;9(10):840–50.