Detecting video frame rate up-conversion based on frame-level analysis of average texture variation

Min Xia, Gaobo Yang, Leida Li, Ran Li & Xingming Sun

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501
Volume 76
Number 6

Multimed Tools Appl (2017)
76:8399-8421
DOI 10.1007/s11042-016-3468-1

Springer

Multimed Tools Appl (2017) 76:8399–8421
DOI 10.1007/s11042-016-3468-1

Detecting video frame rate up-conversion based on frame-level analysis of average texture variation

Min Xia[1,2] · Gaobo Yang[1] · Leida Li[3] · Ran Li[4] · Xingming Sun[5]

Received: 8 May 2015 / Revised: 2 March 2016 / Accepted: 16 March 2016 /
Published online: 4 April 2016

Abstract Frame rate up-conversion (FRUC) refers to frame interpolation between adjacent video frames to increase the motion continuity of low frame rate video, which can improve the visual quality on hand-held displays. However, FRUC can also be used for video forgery purposes such as splicing two videos with different frame-rates. We found that most FRUC approaches introduce visual artifacts into texture regions of interpolated frames. Based on this observation, a two-stage blind detection approach is proposed for video FRUC based on the frame-level analysis of average texture variation (ATV). First, the ATV value is computed for each frame to obtain an ATV curve of candidate video. Second, the ATV curve is further processed to highlight its periodic property, which indicates the existence of FRUC operation and further estimates the original frame rate. Thus, the positions of interpolated frames can be inferred as well. Extensive experimental results show that the proposed forensics approach is efficient and effective for the detection of existing typical FRUC approaches such as linear frame averaging and motion-compensated interpolation (MCI). The detection performance is superior to the existing approaches in terms of time efficiency and detection accuracy.

✉ Gaobo Yang
yanggaobo@hnu.edu.cn

[1] School of Information Science and Engineering, Hunan University, Changsha, 410082, China

[2] College of Electrical and Information Engineering, Southwest University for Nationalities, Chengdu, 610041, China

[3] School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, 221116, China

[4] School of Computer and Information Technology, Xinyang Normal University, Xinyang, 464000, China

[5] School of Computer and Software, Nanjing University of Information Science and Technology, Nangjing, 210044, China

Keywords Digital video forensics · Frame-rate up-conversion (FRUC) · Motion compensated interpolation (MCI) · Average texture variation (ATV)

1 Introduction

With the proliferation of advanced video editing tools such as VideoEdit Magic, video tampering is becoming much easier even for amateur users. However, malicious users can change the content that digital video delivers by various forgery operations such as frame-based manipulation. This makes video tampering detection a challenging and cumbersome task for video forensic analyst [11, 16]. Passive video forensic is to recover the artifacts, which are left in the tampered video without using any auxiliary data such as digital watermark or signature [21, 22]. In recent years, it has been a hot topic in the field of video information security community.

Compared with still image, digital video has an extra temporal dimension [15]. Frame-based video manipulations are special video forgeries which include frame deleting, frame adding or GOP reorganization. In the literature, there exist a lot of works on the detection of frame-based tampering [6, 7, 12, 14, 24–26]. The most representative works are summarized as follows. Wang et al. proposed a detection approach for frame adding/deleting based on the double compression artifacts [24]. Dong et al. also proposed a blind detection approach for frame adding/deleting, which utilizes the motion-compensated edge artifacts (MCEA) [6]. Liu et al. exploit a temporal-domain feature based on the prediction errors of P-frame to detect frame deletion [14]. Gironi et al. proposed a blind forensics approach to detect frame adding and deleting by exploiting the double compression artifacts [7]. It is applicable even when different codecs are used for the first and second compression, and performs well when the second compression is as strong as the first one. For frame duplication, Wang et al. presented a detection approach by exploiting the spatial and temporal correlations of video frames [25]. Lin et al. presented a coarse-to-fine detection approach for frame duplication based on spatial and temporal analysis [12]. Moreover, velocity field consistency is exploited to expose video inter-frame forgery including consecutive frame deletion and consecutive frame duplication [26].

Frame-rate up-conversion (FRUC) increases the frame rate of video sequence by periodically interpolating new frames between original frames to enhance visual quality of low frame-rate video [8]. Frame repetition and frame averaging are the most common FRUC approaches. In recent years, advanced motion-compensated interpolation (MCI) has been proposed, which is also referred as MC-FRUC. It significantly reduces temporal jerkiness, and leaves no visually annoying artifacts. Forgers might use these FRUC techniques to generate faked high frame rate videos, especially for videos over Internet. For example, when two video sequences with different frame rates are needed to be spliced together, the lower frame-rate video is usually up-converted to the desired frame rate by FRUC in advance. Actually, the quality of resultant videos might have not been enhanced since no additional information is provided about video content. Therefore, it is necessary to develop forensics techniques to detect the presence of FRUC.

Until now, there is still few work reported for exposing the presence of FRUC. Bian et al. [4] exploit the periodicity properties of inter-frame similarity for FRUC detection. However, it only reports the detection results of frame repetition, which is the simplest case. Later, they proposed another forensics approach to expose those fake high bitrate videos which might be up-converted by FRUC techniques [3]. Moreover, this work was further extended to investigate the specific artifacts of those fake bitrate videos in both frequency and spatial domains [5]. However, these two works are not specially designed for the forensics of FRUC. Bestagini et al. [2] presented a detector capable of revealing the use of MC-FRUC in a video sequence and further estimating the original frame rate. The detector computes an estimation of each frame from its neighboring frames, and the errors between the estimated and interpolated frames exhibit some periodicity. By further post-processing of this periodic signal, the original frame rate is inferred as well. However, since the estimation of each frame involves the whole process of motion estimation and compensation, its computational complexity is very high.

For typical MCI techniques, the interpolated frame is constructed by either simple frame averaging or block-based weighting average. For MC-FRUC, motion vectors of interpolation frames are filtered by mean filter or median filter to maintain the temporal continuity [8]. Apparently, these weighting and filtering mechanisms improve the visual quality of resultant videos after FRUC, but they lead to texture variation of interpolated frames to some extent. Motivated by this, an efficient and effective blind forensics approach is proposed to detect the presence of FRUC. The difference of texture variation between the original frames and the interpolated frames is used as the clue for blind forensics. Specifically, the periodic change of average texture variation is used to expose the interpolated frames in candidate video. Compared with the existing approaches [2, 4], the proposed approach can accurately locate the positions of interpolated frames and infer the original frame rate. Meanwhile, computational complexity is significantly reduced since no computation-intensive motion estimation/compensation is involved.

The rest of this paper is organized as follows. Section 2 briefly introduces the typical FRUC techniques. Section 3 presents the proposed FRUC detection approach. Experimental results and comparisons are reported in Section 4. We conclude this paper in Section 5.

2 Related works

The existing FRUC techniques can be divided into two categories. The approaches in the first category include frame repetition and linear frame interpolation, which do not consider object motion. The second category approaches consider object or block based motion, which are generally known as MC-FRUC.

2.1 Frame repetition and frame averaging

Frame repetition and linear frame averaging (FA) are the simplest FRUC approaches, which are defined in a straightforward way as follows:

$$f_n = \sum_{j=-l_1}^{l_2} k_{n+j} f_{n+j} \tag{1}$$

where f_n is the nth frame to be interpolated, f_{n+j} is its adjacent frame, k_{n+j} is the weighting coefficient ($j \neq 0$), and $[-l_1, l_2]$ is the temporal window for interpolation. l_1 and l_2 are non-negative integers. If $l_1 = 1, l_2 = 0$ and $k_{n-1} = 1$, then formula (1) is simplified as

$$f_n = f_{n-1} \tag{2}$$

Apparently, frame repetition is a special case of frame averaging. Because frame repetition and frame averaging do not consider the motion between successive frames, they are only effective for videos without motion or just slight motion. If there are noticeable motions in video sequences, they will suffer from temporal jerkiness, temporal blurring and sometimes annoying ghosting artifacts.

Some popular video software such as ImTOO [20] and AVS video converter [18] adopt frame repetition to increase the frame rate. Bian et al. exploit the fact that frame repetition will inevitably introduce some near-duplication (interpolated frames) into the resultant video. It is expected that these interpolated frames show much bigger similarities with their adjacent frames. Moreover, this kind of abnormal similarity shows periodicity because frame repetition is periodically used to increase the frame rate. Based on this observation, SSIM (structural similarity index measurement) is used to measure this kind of similarity and a SSIM curve is generated for suspicious video. However, this approach does not consider frame averaging and advanced MC-FRUC.

2.2 MC-FRUC

To overcome the drawbacks of frame repetition and frame averaging, a typical MC-FRUC scheme consists of two key elements: motion estimation and motion-compensated frame interpolation. Motion estimation is to estimate the spatial displacement of pixels in neighboring frames, i.e., motion vectors (MVs). These motion vectors are then refined to generate the interpolation frames. Existing MC-FRUC approaches can be categorized into three categories, namely, motion compensation interpolation (MCI) [8], overlapped block motion compensation (OBMC) [9], and multiple hypotheses Bayesian FRUC (MHB-FRUC) [13].

Let f_n, f_{n-1} and f_{n+1} be the frame to be interpolated, the previous frame and the successive frame, respectively. MCI generates the interpolated frame as follows:

$$f_n(x, y) = \frac{1}{2}(f_{n-1}(x + v_x, y + v_y) + f_{n+1}(x - v_x, y - v_y)) \tag{3}$$

where (x, y) is the spatial location, v_x and v_y are the motion vectors between f_{n-1} and f_{n+1}. From (3), it is apparent that an assumption of translational motion is made for MCI. Since video object might be irregular and adjacent blocks within an object might have quite different motion vectors, this assumption easily leads to perceivable blocking artifacts in the interpolation frames.

To further reduce the blocking artifacts, OBMC introduces a concept of window which is larger than a block and the blocks can overlap with each other. Thus, the transition near block boundaries is much smoother in the interpolated frame. Moreover, a weighting mechanism is adopted in the frame interpolation process. When the motion activity is low, OBMC can effectively reduce the blocking artifacts and yield desirable visual quality. However, OBMC still does not consider the spatial consistency of neighboring pixels and object motion trajectory. When adjacent blocks have obviously different motions, OBMC still leads to blur or over-smoothing artifacts. MHB-FRUC incorporates both temporal motion model and spatial image model into the optimization criterion of interpolated frame with maximum *a posteriori* probability [13]. The spatial image model describes the spatial structure of neighboring pixels whereas the temporal motion model describes the temporal correlation of pixels along motion trajectories. Instead of a single optimal motion, multiple optimal motion trajectories are exploited to form a group of motion hypotheses. To obtain an accurate estimation of those pixels in intermediate frames, the interpolated frames generated by these motion trajectory hypotheses are adaptively fused in terms of the reliability of each hypothesis.

Bestagini et al. unify frame averaging and typical MC-FRUC techniques including MCI, OBMC and MHB-MFRUC with a weighted interpolation as follows.

$$X_{ij}^{\phi}(\phi t) = \sum_{k=-K}^{K} h_k \cdot X_{m_{t,i,j} n_{t,i,j}}^{0}(\phi t + \phi k) \tag{4}$$

where h is the interpolation filter (i.e., a one-dimensional low-pass filter), X^0 is the original sequence defined on the support of the interpolated one, and while now the spatial indices $m_{t,i,j}$ and $n_{t,i,j}$ change in each frame for each pixel position (i, j), in order to follow motion estimated trajectories in time.

Bestagini et al. analyzed the correlation introduced by the frame interpolation filter to design a detector, which is capable of identifying the interpolation factor used and revealing the original frame rate. However, complex motion estimation and compensation is involved in the forensics approach, its computational complexity is very high. Moreover, it only reports the experimental results on up-converted videos by popular software such as ISTWZCOdec, Medianet and MSU.

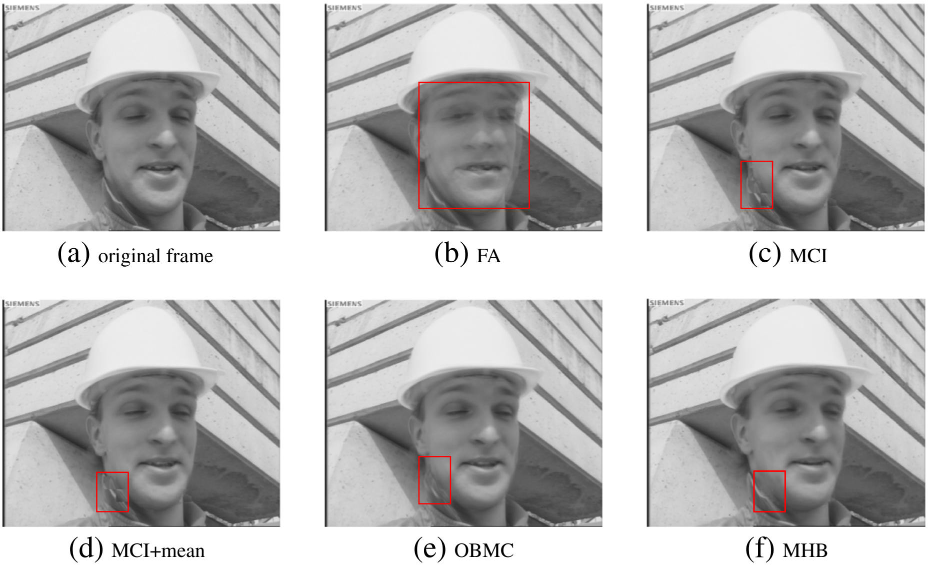
2.3 Visual comparison of different FRUC approaches

Figure 1 compares the visual quality of interpolated frames generated by different FRUC approaches. *Foreman* sequence is used as an example. The builder rotates his head, which is a typical rotational motion. Fig. 1a is the original frame, and Fig. 1b is the interpolated frame of frame averaging (FA). Apparently, there are blur artifacts simply because the assumption of translational motion involved in frame averaging is not suitable for the rotational motion in *Foreman* sequence. Figure 1c is the interpolated frame of MCI, which is much better than Fig. 1b, since MCI exploits the motion displacement within successive frames. However, there are still some blocking artifacts, especially those blocks around the cheek and collar area. Figure 1d is the result of both MCI+mean, which is better than Fig. 1c. The interpolated frames of OBMC and MHB-FRUC are shown in Fig. 1e and f, respectively. Their subjective qualities are greatly improved, since the blocking artifacts and blur are greatly reduced.

3 Proposed FRUC detection approach

3.1 Motivation

The idea behind the proposed FRUC detection approach is motivated by the forensics of spatial image subsampling, which exploits the spectrum periodicity of resulting image due to periodic interpolation. From equations (1, 3, 4), we know that the interpolated frame depends on the criterion of frame averaging among adjacent frames and the most similar block searching for motion estimation. Actually, the weighting mechanism involved in FRUC is similar to the averaging of multi-images for noise elimination. It not only removes the high frequency components, but also improves the visual quality. This phenomenon is particularly obvious in the texture-rich regions. That is, the frame averaging or motion estimation involved in MC-FRUC will lead to some texture variation artifacts in the interpolated frames. Moreover, it is expected that the texture variation artifacts will exhibit temporal periodicity as well.



(a) original frame (b) FA (c) MCI

(d) MCI+mean (e) OBMC (f) MHB

Fig. 1 Subjective visual quality of the interpolated frames by different FRUC approaches

3.2 The average texture variation (ATV) metric

We exploit the texture variation artifacts for the detection of video FRUC, which is based on the observation that the local texture variation is relatively high in the original frame. The low pixel variations in texture area are difficult to be perceived by human visual system. However, the low variation in texture area is a desirable indicator of FRUC forgery. In the motion compensation stage of typical MC-FRUC techniques, a weighting mechanism is adopted to find most similar blocks in adjacent frames. Thus, the texture regions in the interpolated frame will have slight blur. This kind of blur can be measured by frame-level average texture variation (ATV). Moreover, it is expected that the interpolated frames have smaller ATV values than the original frames.

Similar to the definition of local variation in [1], we firstly define the maximum local variation (MLV) as the maximum intensity variation of a pixel with respect to its 8-connected neighbors given by

$$v(f_n(i,j)) = \{\max |f_n(i,j) - f_n(x,y)|; i-1 \leq x \leq i+1, j-1 \leq y \leq j+1\} \tag{5}$$

where $f_n(x,y), i-1 \leq x \leq i+1, j-1 \leq y \leq j+1$ are the 8-neighboring pixels of $f_n(x,y)$. For a gray-scale image, MLV changes within the range of [0,1], where 0 means no variation between a pixel and its 8-neighbors while 1 indicates the biggest variation. Given an image f_n with size $M \times N$, the MLV of all the pixels are calculated using (6), which generates a texture variation map $v(f_n(i,j))$ for this image as follows:

$$v(f_n) = \left\{ \begin{array}{ccc} v(f_{1,1}) & \cdots & v(f_{1,N}) \\ \vdots & \ddots & \vdots \\ v(f_{M,1}) & \cdots & v(f_{M,N}) \end{array} \right\} \tag{6}$$

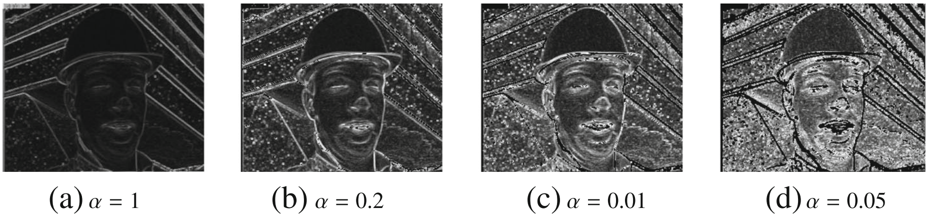

(a) $\alpha = 1$ (b) $\alpha = 0.2$ (c) $\alpha = 0.01$ (d) $\alpha = 0.05$

Fig. 2 Texture variation maps of the original frame

A threshold α is defined to discriminate the texture regions and the edge regions as follow:

$$v_T(v(f_n)) = \begin{cases} v(f_n), & v(f_n) \leq \alpha \\ 0, & v(f_n) > \alpha \end{cases} \tag{7}$$

where α is an adjustment factor, which ranges from 0 to 1. The smaller is α, the more edge regions will be removed. Let the second frame of *Foreman* sequence be an example. Figures 2 and 3 are texture variation maps of the original frame and the interpolation frame by MCI, respectively. It is apparent that a smaller α implies a more obvious texture region. Even those texture details which can be perceived by human eyes are highlighted as well. It can also be observed from Fig. 1c that there are obvious blocking artifacts in the face and collar regions. With the decrease of α, the edges of these blocks are removed. Apparently, too small α will remove those regions with relatively bigger texture variations. In the following subsection, the selection of α is determined by experiments.

Finally, the ATV value of the frame is obtained as follows:

$$ATV = \frac{1}{num} \cdot \sum_{(i,j)\in T} v_T(v(f_n(i,j))) \tag{8}$$

where (i, j) is the pixel in the texture region and *num* is the number of pixels in the frame. A larger value of *ATV* means that the average variation of texture region is larger, and the texture details are richer. For the interpolated frames by FRUC, frame interpolation will destroy some texture details, and thus the *ATV* value will be smaller. Moreover, since the interpolated frames are inserted into the original frames in a periodic way, the change of *ATV* values will also be periodical.

3.3 The proposed FRUC detection algorithm

The proposed FRUC detection algorithm is made up of two stages: (1) generate the ATV curve for suspicious video; (2) postprocess the ATV curve processing and make binary decision.

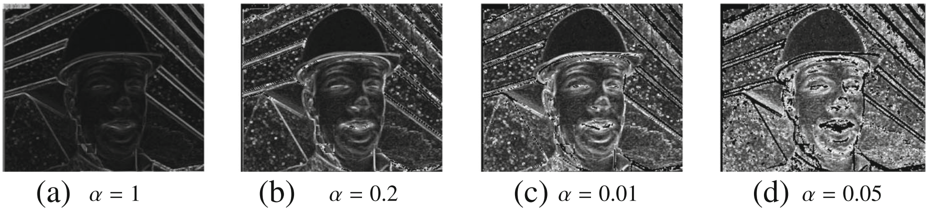

(a) $\alpha = 1$ (b) $\alpha = 0.2$ (c) $\alpha = 0.01$ (d) $\alpha = 0.05$

Fig. 3 Texture variation maps of the interpolated frame by MCI

(1) The generation of ATV curve

Step I. For a suspicious encoded video sequence, decode it into an uncompressed video sequence $\{f_n; n = 1, 2, \cdots L\}$;

Step II. Extract the luminance component of each frame to obtain $\{Y_n; n = 1, 2, \cdot\cdot \cdot L\}$

Step III. Compute the ATV metric from the luminance component of each frame.

Step IV. Repeat step II to step III, until *n*=*L*. Then, the ATV curve is obtained for each frame in the suspicious video sequence.

Let *Foreman* sequence be an example. Figure 4 compares three ATV curves, which are the original video and its up-converted versions by MCI and FA, respectively. It is apparent that the ATV curve of the original sequence is relatively smooth, where the ATV values vary between 0.0228 and 0.0265 for each frame. Moreover, there is no periodic variation of the ATV curve (either rise or fall). However, the ATV curve of the up-converted video by MCI has acute fluctuation, which varies from 0.0084 to 0.0265 and shows strong periodicity. The green curve is the ATV value of up-converted video by FA, which varies between 0.0206 and 0.0265. Apparently, this variation is much slighter than MCI. However, it still exhibits some periodicity even though it is not strict and apparent. In the following, these ATV curves are post-processed to highlight the periodic artifacts, and then to determine whether the suspicious video is up converted or not, and to further estimate its original frame rate.

(2) Postprocessing of the ATV curve and binary decision

Step I. The the ATV curve is pre-processed with a sliding window so as to separate different periods (FRUC) and non-periodic segment (original video). Specifically, the sliding window is processed as follows:

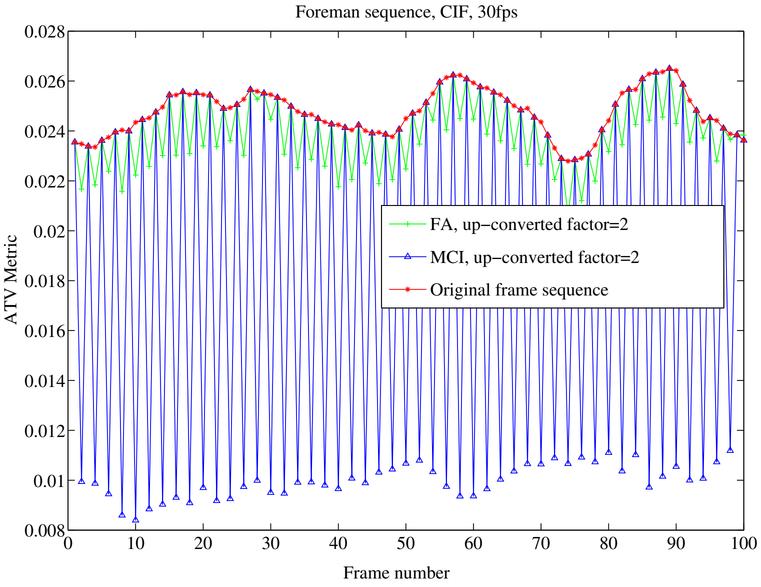


Fig. 4 Comparison of ATV curves between the original sequence and its up-converted versions

Firstly, select a short segment in candidate video sequence. Let its length be E. After Discrete Fourier Transform (DFT), its spectrum amplitude is computed to obtain the location of the maximum high frequency component. The whole calculation process is defined by (9) to (12) as follows.

$$F_{k,i} = \sum_{n=i}^{L_i} ATV_n \cdot e^{-j\frac{2\pi nk}{E}} \tag{9}$$

$$H_k = \begin{cases} 1, & d < k < E - d + 1 \\ 0, & otherwise \end{cases} \tag{10}$$

$$G_{k,i} = H_k \cdot |F_{k,i}| \tag{11}$$

$$K_{\max,i} = \{k; \arg\max(G_{k,i}; k = 1, 2...E)\} \tag{12}$$

where $L_i = E + i - 1, i = 1, 2...L - E + 1$, and d is the cutoff frequency. E and d are initialized as $E = 18, d = 3$. Then, make these small pieces consisting of E points move along the ATV curve point by point, as shown in Fig. 5. Then, the position $K_{\max,i}$ where the maximum value of high-frequency components occurs is obtained. If $K_{\max,i} \neq K_{\max,1}$, it means that the periodicity after the point i is different from the rest video. In this way, the videos of different frame rates are automatically separated. Repeat this process until the whole video is decomposed into a number of segments with different periodicity. Thus, the nonperiodic original video is separated. In the experiment, the location of spliced frames can be determined by continuously adjusting E and d.

Step II. Estimate the original frame rate according to the periodic part of the ATV curve. We firstly transform the curve into frequency domain using Discrete Fourier Transform (DFT) and obtain the amplitude spectrum. In Fig. 5, the X-axis indicates the distribution of frequency components as the original frame-rate is 30 fps. Then we use an ideal high-pass filter to

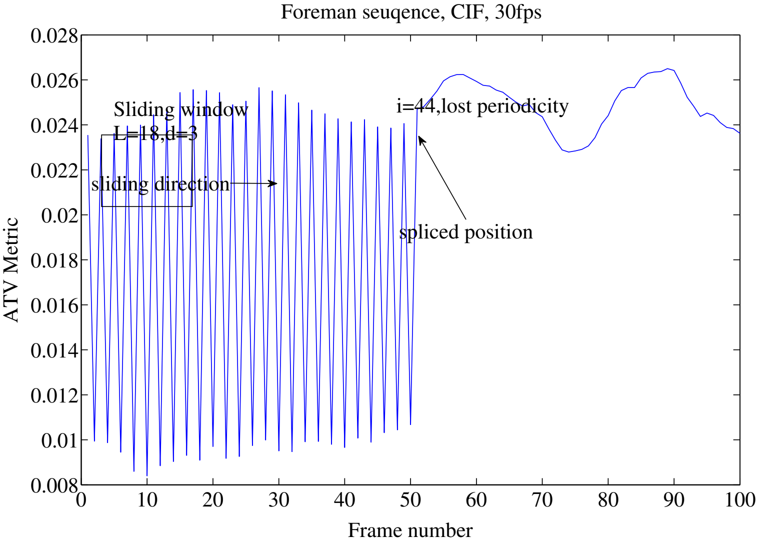


Fig. 5 Illustration of sliding windows

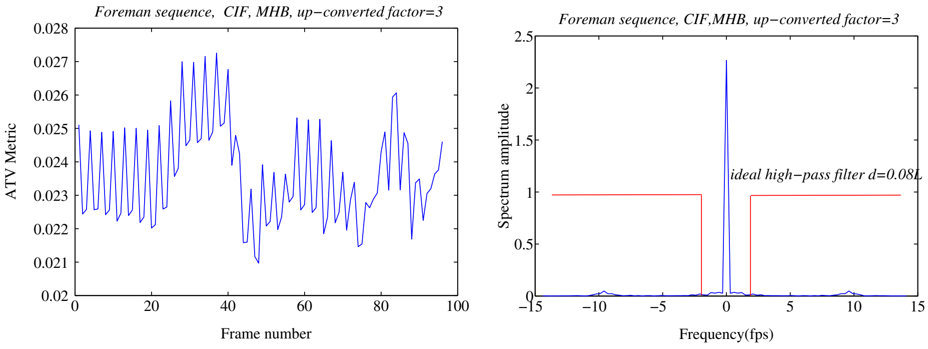
make it filtered, the formulas used are the same as the formulas in *Step I*, $d = 0.08L$, $L = 50$. Due to the symmetry property of DFT, the first value is the DC component, the second to the $\lceil L/2 \rceil$ values are positive frequencies, and the values of $\lceil L/2 \rceil + 1$ to L are the corresponding negative frequencies. Let L be the sequence length. In the detection method, we need to compute the k and ω when the maximum magnitude is achieved. ω is the ratio between its magnitude and the average magnitude of the non-zero spectrum after removing the DC and low frequency components. if it meets $\omega > \mu$, where μ is a threshold, then the candidate video is decided as a forgery one, and vice versa. In the literature [4], μ is set to 2.5, and ω is defined as follows:

$$\omega = \frac{\arg\max(G_k;\, k = 1, 2 \ldots L)\}}{\frac{1}{L-2d} \sum_{i=d+1}^{L-d} G_i} \tag{13}$$

Furthermore, we can infer the original frame rate of the up-converted video from the computed k. Let x be the frame rate of faked high frame-rate video, its original frame rate y is calculated as follows:

$$y = \tfrac{kx}{L} \;\; phor \;\; y = x - \tfrac{kx}{L} \tag{14}$$

Figure 6 shows that the detection process of video sequence which is up-converted by MHB-FRUC (the up-conversion factor is 3). Figure 6a is the ATV curve. Figure 6b is the spectral amplitude curve, where the interpolated frame rate is 30 *fps*. From Fig. 6b, it can be observed that there is small impulse near the position where their frequency factor is $10\,fps$ except the DC and low frequency components. To eliminate the influence of DC and low frequency components, an ideal high-pass filter ($d = 0.08L$, $L = 100$) is utilized for filtering. The final result is shown in Fig. 7. It is apparent that there is a spike in the position where the frequency factor is $10\,fps$, which actually indicates its original frame rate is $10\,fps$.



(a) ATV curve of the forgery video by MHB-FRUC

(b) Spectrum of Fig.8(a)

Fig. 6 ATV curve and its Fourier spectrum for the *Foreman* sequence

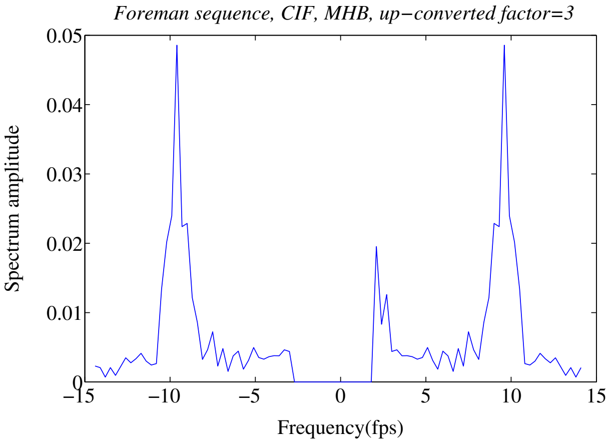


Fig. 7 Illustration of removing low frequency component in Fig 6b

Step III. The localization of interpolated frames

After estimating the original frame rate of faked video, the periodicity T of up-converted video and the number of original frames N_T can be computed as follows:

$$\frac{T}{N_T} = \frac{x}{y} \tag{15}$$

where T/N_T is the simple fraction of x/y.

Similar to the moving average method in [10], an adaptive threshold is defined in terms of T and N_T as follows.

$$\tau_i = \begin{cases} (ATV_i + ATV_{i+1})/2 & N^I{}_i \geq T - N_T \;\; or \;\; N^O{}_i \geq N_T \\ \tau_{i-1} + S_i \cdot (ATV_i - \tau_{i-1}) & otherwise \end{cases} \tag{16}$$

where $\{i = 1, ..., L\}$, $N^O{}_i$ and $N^I{}_i$ represent the numbers of original frames and interpolated frames in the $T-1$ frames before the ith frame, respectively. The initialization conditions are as follows:

$$\{N^O{}_i = 0, N^I{}_i = 0, \tau_i = 0, i = 1, 2, ..., T-1\}.$$

Besides, S_i is the smoothing factor corresponding to τ_i. Similar to the derivation process in [13], we have

$$S_i = \lambda_1 \cdot \frac{|ATV_i - ATV_{i-T+1}|}{\sum\limits_{j=i-T+2}^{i} |ATV_j - ATV_{j-1}|} + \lambda_2 \tag{17}$$

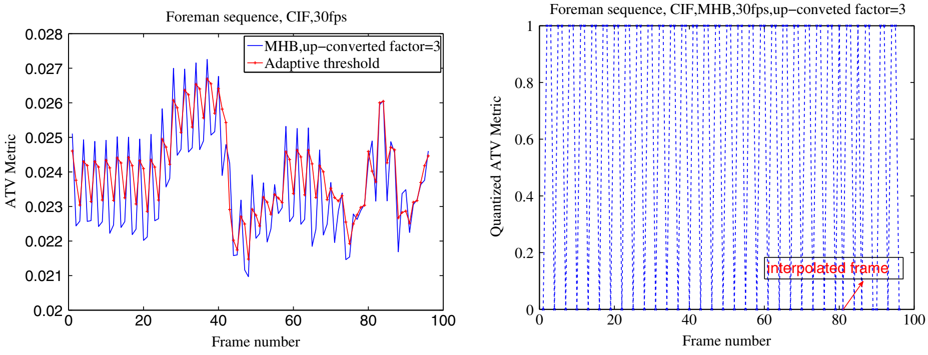
where $\lambda_1 = 0.6022$, $\lambda_2 = 0.0645$ (Please refer to [10] for details.)

Finally, the frame-level ATV values are binarized with the adaptive threshold as follows:

$$C_i = \begin{cases} 1 & ATV_i < \tau_i \\ 0 & otherwise \end{cases} \tag{18}$$

when it meets $C_i = 1$, it implies that the$i$th frame is an interpolated frame, or else an original frame.

Following equations (15–18), Fig. 8 is obtained from Fig. 6a. From Fig. 8a, it can be observed that for the ATV curve of up-converted sequence by MHB-FEUC, there is great



(a) The adaptive threshold selection

(b) Binary classification

Fig. 8 Adaptive threshold selection and the corresponding binary classification of candidate video sequences

fluctuation. It is still difficult for human eyes to infer the original frames and locate the interpolated frames. By setting adaptive thresholds τ (red curve), the ATV values above the red curve correspond to the original frames, and the ATV values below the red curve correspond to the interpolated frames. Figure 8b further shows the classification results by utilizing the adaptive threshold τ. It can be observed that for most cases, there are two interpolated frames between every three consecutive frames. However, there are still some special cases such as the 81st frame. It should be an interpolated frame, instead of the original frame which is wrongly classified. Actually, it is caused by the rapid change of video scene, which might lead to ATV value abnormity for those interpolated frames. However, it does not influence the overall judgement that the candidate video is up-converted. Moreover, it does not seriously influence the estimation of the original frame rate.

4 Experimental results and analysis

4.1 Experimental settings

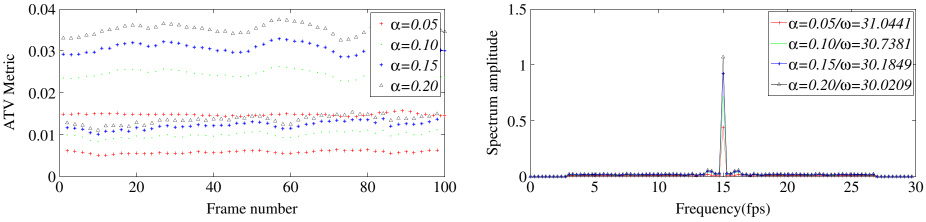
To evaluate the performance of the proposed FRUC detection approach, 30 uncompressed YUV video sequences are selected for experiments. These sequences are downloaded from open websites [23, 27], which include videos captured under different scenarios composing sports, parties, news, video surveillance, traffic scene and so on. Their spatial resolutions are CIF and the original frame rates are 10 fps, 15 fps, 20 fps, 25 fps and 30 fps, respectively. For each YUV sequence, it is up-converted to the specified target frame rates including 15 fps, 20 fps, 25 fps, 30 fps and 60 fps with different FRUC approaches. In this experiments, the FRUC approaches are either most representative or state-of-the-art. Specifically they include FA, MCI without smoothing motion vectors [2], MCI with mean filtering of motion vectors, OBMC with simple smoothing of motion vectors [9], advanced MHB-FRUC [13], and commercial video software MSU [17] and Mvtools [19]. Moreover, both un-compressed and compressed videos with H.264/AVC are tested in this experiment. In summary, there are 4290 video sequences obtained as positive samples by these FRUC techniques. Thirty original video sequences with five different frame rates in both uncompressed and compressed formats are used as negative samples. That is, there are 300 original videos.

Experimental software and hardware configuration is summarized as follows: Intel CORE i5 2.0 GHZ CPU; 4 GB Memory; Geforce HD graphical card; Microsoft Windows 7 professional and Matlab Version 7.12.0635 (R2011a).

4.2 Parameters setup

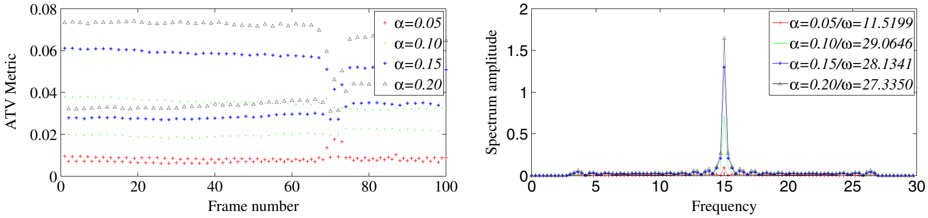
As described previously, α, d and μ are three key parameters. α is the adjusting factor , d is the cutoff frequency, while μ is the threshold to differentiate the original and faked videos. Three typical video sequences are selected to select these key parameters. *Foreman* is a sequence with relatively static background and rapid motion object, whereas *Coastguard* has more motion objects, and both the object and background have rapid motion. *News* sequence is a typical head-shoulder sequence. Only the region around the woman's lips has obvious motion. Each sequence has 100 frames in uncompressed format. As claimed in Section 3.3, we know that a bigger α means more edge components, which makes the smoothing effects in the texture regions of up-converted video less obvious. The cutoff frequency d is used to determine the low-frequency component of the ATV curve after DFT transform, which is to be removed in postprocessing. A less d will incur more low-frequency component, which has negative effects towards the periodicity judgement of the ATV curve. ω is the ratio of its magnitude and the average magnitude of non-zero spectrum after removing the DC and low frequency components. It actually reflects the periodicity of ATV curve. A larger ω implies a stronger periodicity of video and a higher possibility of up-converted video. Thus, we need to select an appropriate threshold μ. When it meets $\omega > \mu$, it is determined as a faked video. The frame rates of these three video sequences are up-converted by MCI, and the up-conversion factor is 2. That is, there is an original frame between every two consecutive frames.

From Fig. 9a, we can see that for *Foreman* video sequence with a relatively bigger human face motion, the value of α varies between 0.05 and 0.20 (the stepsize is 0.05). It leads to minor fluctuation of ATV curve, which corresponds to the decrease of ω in frequency domain from 31.04 to 30.02 (Fig. 9b). It is apparent that ω is approximately steady. When α is larger than 0.05, it does not have any influences on the detection accuracies of those videos with moderate motion. For *Coastguard* sequence with fast-moving scene, it can be observed from Fig. 10a and b that the change of α leads to a rigorous fluctuation of the ATV curve. When α increases from 0.05 to 0.10, ω increases from 11.52 to 29.06, and then decreases slowly. This implies that for video sequences with rapid motion, α should not be too small. From Fig. 11a and b , we know that for *News* sequence, the increase of α leads to the periodic variation of the ATV curve, which is apparently smaller that *Foreman*



(a) ATV curve (b) Magnitude spectrum

Fig. 9 ATV curve of *Foreman* sequence and its corresponding DFT spectrum



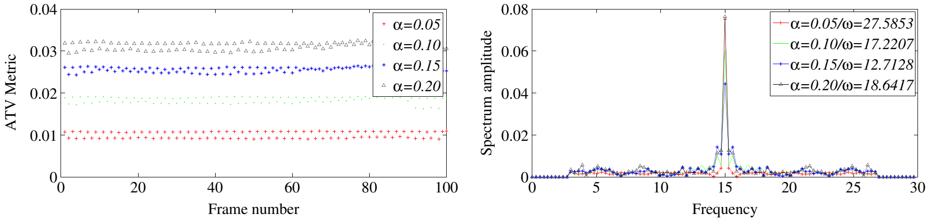
(a) ATV curve (b) Magnitude spectrum

Fig. 10 ATV curve of *Coastguard* sequence and its corresponding DFT spectrum

and *Coastguard* sequences, respectively. When α increases from 0.05 to 0.20, ω rapidly decreases from 27.58 to 12.71, and then it steadily increases to 18.64.

Figure 12 shows the relationship between ω and α for these three sequences. It is apparent that for a bigger ω, which implies stronger periodicity, a higher detection accuracy is achieved. While α rises, the ω of *Foreman* sequence rises as well. When α is 0.04, ω reaches the maximum value of 31.08 and then gradually falls. For *Coastguard* sequence with fast moving objects, the smaller is the value α, the smaller is the corresponding value of ω as well. However, with the rapid increase of α up to 0.10, ω will gradually turn into a stable state (near to 28). For *News* sequence in which the announcer is almost motionless and only some background area has slight motion, its ω is bigger than that of *Coastguard* sequence. Moreover, with the increase of α, ω also gradually increases. When $\alpha = 0.04$, there is an intersection point with the decline of α for *Foreman* sequence. That implies that for video sequences with slowing motion such as *Foreman*, a smaller α can be chosen to reduce computational cost. For those video sequences with rapid motion such as *Coastguard* , a relatively bigger α should be chosen. To improve the detection accuracy , α is chosen to vary within the range of 0.06 to 0.08.

Figure 13 shows the influence of the cutoff frequency d. For *Foreman* sequence, it is apparent that an increase of d leads to the decrease of ω. However, for *News* and *Coastguard* sequences, ω firstly increases with the increase of d, and it reaches its maximum value when d =8 and 10. Then, ω decreases gradually. By extensive experiments, we found that $d = 0.06L$ to $0.10L$ is a suitable choice, where L denotes the length of suspicious video sequence. From Figs. 12 and 13, we can observe that with the range of $\alpha(0.06 - 0.08)$ and $d(0.06L - 0.08L)$, ω is much greater than 2.5. We still use the same threshold $\mu = 2.5$ [2] for detection.



(a) ATV curve (b) Magnitude spectrum

Fig. 11 ATV curve of *News* sequence and its corresponding DFT spectrum

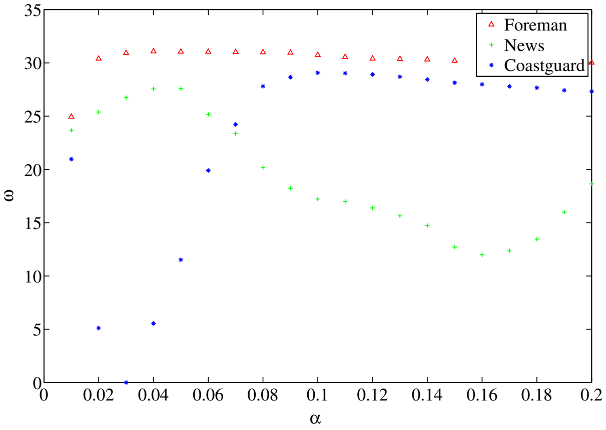


Fig. 12 The relationship of α and ω

4.3 Analysis of experimental results

Based on previous analysis, we set $\alpha = 0.07, d = 0.08L$ and $\mu = 2.5$ in the following experiments. Tables 1 and 2 are the detection results for uncompressed videos after FRUC operations with different interpolation patterns. In Table 1, OR and RR are the frame rates of the original video clips and the up-converted videos, respectively. FNR and FPR are False Negative Rate and False Positive Rate, respectively. Then, the average detection accuracy is computed as follows:

$$DR = 1 - \frac{FNR + FPR}{2} \tag{19}$$

From Table 1, we can observe that the proposed method is effective for the detection of video FRUC. For the up-converted videos, most detection accuracies are over 90 % on average. For the original videos, the FPR is very small, especially those videos with frame rate bigger than 25 fps can be accurately detected. For the up-converted videos, if the frame rate

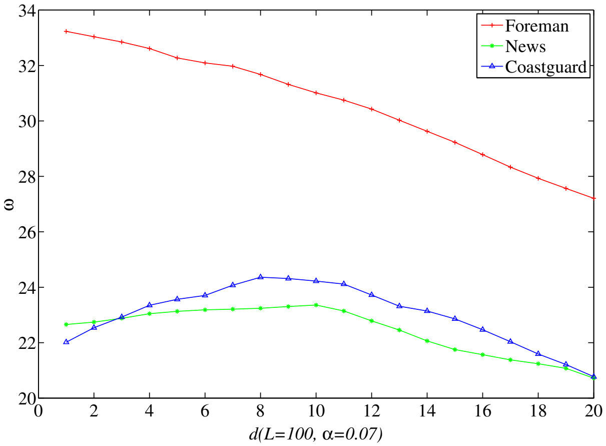


Fig. 13 Relationship of d and ω when $\alpha = 0.07$

Table 1 Detection results (%) for un-compressed video after FRUC with different interpolation patterns

OR	RR		FA		MCI [8]		MCI+mean		OBMC [9]		MHB [13]	
		FPR	FNR	DR	FNR	DR	FNR	DR	FNR	DR	FNR	DR
10	15	3	20	88.5	13	92	13	92	10	93.5	0	98.5
10	20	3	10	93.5	6	95.5	6	97	3	97	3	97
15		3	10	93.5	6	95.5	6	95.5	6	95.5	20	88.5
10	25	3	10	93.5	6	95.5	6	95.5	6	95.5	30	83.5
15		3	13	92	10	93.5	10	93.5	6	97	10	93.5
20		3	23	87	16	90.5	16	90.5	16	90.5	26	85.5
10	30	0	10	95	6	97	3	98.5	3	98.5	3	98.5
15		0	6	97	3	98.5	6	97	6	97	0	100
20		0	10	98.5	6	97	6	97	6	97	3	98.5
25		0	23	88.5	20	90	20	90	16	92	27	86.5
15	60	0	10	95	30	85	16	92	16	92	3	98.5
20		0	10	98.5	6	97	6	97	6	97	6	97
25		0	16	92	13	93.5	10	95	13	93.5	23	88.5
30		0	6	97	6	97	3	98.5	3	98.5	3	98.5

after FRUC is integer multiples of the original frame rate and the up-conversion factor is less than 4, the detection accuracy is bigger than 93.5 %. However, when the up-conversion factor is bigger than 4, its detection performance is not steady. Luckily, the up-conversion factor is usually less than 4 for video FRUC. Among them, FNRs are 3 % and 30 % for MHB and MCI, respectively. For MCI+mean and OBMC with smoothing, their FNRs decrease to 16 %. This implies that MCI+mean and OBMC improve the image quality and also incur unperceived blur artifacts in texture regions. Moreover, when the up-converted frame rate is not integer multiples of the original frame rate, FNR is bigger than 6 %.

Table 2 summarizes the average detection accuracies and their standard deviations. For those up-converted videos by OBMC, the highest average detection accuracy and the lowest standard deviation (2.5 %) are achieved. That is, the proposed approach achieves the best performance for the detection of OBMC. For those videos up-converted by FA, its detection accuracy is the lowest (93.5 %). This is caused by the fact that only simple averaging is involved in two neighboring original frames without further operation. Compared with MCI, MCI+mean further processes the estimated motion vectors with mean filter. This does benefits to the improvement of detection accuracy. For MHB, its detection accuracy is the the second lowest among the five FRUC techniques. The reasons are summarized as follows:

Table 2 Average detection rate and standard deviation for up-converted videos without compression (%)

Video sequences	FRUC methods				
	FA	MCI [8]	MCI+mean	OBMC [9]	MHB [13]
DR	93.5	94.1	94.9	95.2	93.7
Standard deviation	3.7	3.7	2.9	2.5	6

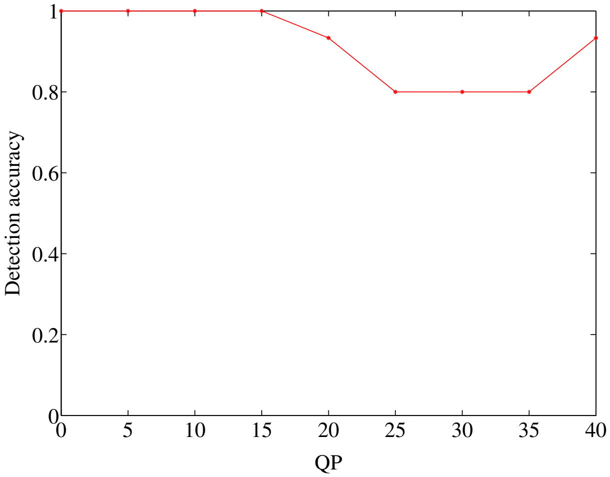


Fig. 14 Relationship between QP and detection accuracy

its frame interpolation incorporates both temporal motion model and spatial image model into the optimization criterion of interpolated frame estimation with maximum *a posteriori*. Instead of a single uniquely optimal motion, multiple optimal motion fields are employed to form a group of motion trajectory hypotheses. Thus, the blur of texture region in the interpolated frame is easily influenced by the neighboring frames. The standard deviation of its detection accuracy is the biggest, which is up to 6 %.

To further evaluate the performance, further experiments are made to detect those up-converted videos with lossy compression. Thirty original video sequences are up-converted from 15 fps to 30 fps by MHB, and the resultant videos are further encoded with H.264/AVC encoder. For video compression, the compression ratio is controlled by the quantization parameters (QP) for P frames and I frames, respectively. A bigger QP indicates a higher compression ratio. In consequence, there are more serious visual distortions after compression, which will bring extra challenge for the detection of FRUC. Figure 14 shows the relationship between QP and detection accuracy. It can be observed that with the increase of QP, the detection accuracy decreases gradually. When the QP reaches 25, the detection accuracy keeps about 80 %. When the QP increases to 40, the detection accuracy increases to 93.3 % as well. This implies that the detection accuracy is not seriously influenced by video compression, especially when the compression ratio is moderate ($QP < 15$).

Table 3 Detection accuracy (%) and runtime (s) compared with the approach by Bestagini et al. [2]

Detection method	30(fps)		45(fps)		60(fps)	
	DR	runtime	DR	runtime	DR	runtime
Bestagini et al. [2]/MSU	66.7	172.93	66.7	180.54	70	172.06
Proposed approach/MSU	96.7	1.35	96.7	1.28	90	1.27
Bestagini et al. [2]/MVTools	83.3	184.41	86.7	165.37	83.3	165.92
Proposed approach/MVTools	93.3	1.25	93.3	1.28	96.7	1.26

Table 4 Comparison of existing FRUC detection approaches (+: valid; -: invalid)

Detection	Luo et al. [4]			Bestagini et al. [2]			Proposed approach		
FRUC	FR	FA	MC-FRUC	FR	FA	MC-FRUC	FR	FA	MC-FRUC
Validity	+	−	−	−	+	+	−	+	+

To further verify the effectiveness of the proposed FRUC detector, two popular software MVTools [27] and MSU [23] are also used to up-convert the frame rate and the up-converted videos are also tested. The same thirty original videos with frame-rate 15fps are up-converted to 30 fps, 45 fps and 60 fps by MVTools and MSU (without further compression), respectively. The experimental results are reported in Table 3. The average detection accuracies are up to 90 % for the up-converted videos by both MVTools and MSU. For the approach by Bestagini, its highest detection accuracy is 86.7 %. Moreover, there is big fluctuation for its detection accuracy. Especially, the detection accuracy is less than 70 % for the up-converted videos by MSU. From the computational complexity point of view, the time consumption of the proposed approach is less than 2 seconds for video clips with more than 100 frames, whereas the approach by Bestagini takes more than 3 minutes. The comparison of computational complexity is further analyzed in Section 4.4.

Table 4 compares the applicability between the proposed approach and the existing works. Luo's method [4] is suitable for frame repetition, and the Bestagini's approach [2] is reported to be effective for the detection of those videos up-converted by commercial software such as ISTWZCOdec, Medianet, MSU and MVTools2. However, it does not provide sufficient details. The proposed approach investigates the mechanism of frame repetition, FA and the state-of-the-art MC-FRUC approaches, and can detect simple and advanced FRUC approaches such as MCI, MCI+mean, OBMC and MHB.

4.4 Analysis of computational complexity

In this section, we compare the time complexities of the Luo's approach [4], the Bestagini's approach [2] and the proposed approach. Let the spatial resolution of candidate video sequence be $W{\cdot}H$ and N be the frame number. For the Bestagini's approach, the size of macroblock is $M{\cdot}M$ and the search range is S. Tables 5 and 6 are the pseudo-codes and the corresponding time complexity for the proposed approach and the Bestagini's approach, respectively.

Table 5 Time complexity computation of the proposed method

Step	Algorithm pseudo-code	Time complexity

	for index=1: N
	  compute
	  ATV(frame(index))→atv(index);
	end
		$phN{*}O(ATV){=}O(WHN)$

	compute DFT(atv(1: N))→ F(1: N);	O(Nlog$_2$(N))
	compute Ma x(F(d+1: N-d+1)), (Mean(F(d+1: N-d+1));	

Table 6 Time complexity computation of P. Bestagini et al. method

Step	Algorithm pseudo-code	Time complexity

1	for index=1: 2: N-1
	  for number=1:(W*H/M^2)
	    compute
	    Macro(frame(number,index),frame(number,index+2))
	    → I(index+1);
	  end
	end
		(N-2)*(W*H/M^2)*O(Macro)=O($WHSSN$)

2	for index=2: 2: N
	  for number=1:(W*H/M^2)
	    compute
	    MCI(frame(number, index), frame(number, index+2))
	    → I(index+1);
	  end
	end	

	for index=2: N-1
	  frame(index)-I(index)→ Error(index-1)
	  Sum(Error(index-1)^2)→ e(index-1)
	end
		(N-2)*O(e)=O(WH(N-2))

3	compute DFT(e(1: N-2))→ F(1: N-2);
	compute Max(F(d+1: N-d-1)),(Mean(F(d+1: N-d-1)));
		O((N-2)$\log_2$(N-2))

From Table 5, we know that the time complexity of the proposed method mainly depends on the computation of ATV metric and DFT in step one and step two, respectively. In step one, the ATV value is computed for each frame in the suspicious video, and then its complexity is $O(WHN)$. While the time complexity of step two is $O(N\log_2(N))$. Thus, the total time-complexity is $O(WHN)+O(N\log_2(N))$ for the proposed method. This indicates that the computation time is closely related with the spatial resolution and frame number of candidate video. From Table 6, the overall time complexity of the Bestagini's approach is $O(WH^2(N\text{-}2))$ + $O(WH(N\text{-}2))$ +$O((N\text{-}2)\log_2(N\text{-}2)$. For the computational complexity of the Luo's approach [4], it is reported to be $O(WHN)+O(N\log_2(N)$, which is very near to that of the proposed approach. However, the computational complexity of the proposed approach outperforms the Bestagini's approach. Table 7 summarizes the actual time consumptions of these three detection approaches. The time consumption of the proposed approach is somewhat lower than that of the Luo's approach, simply because its similarity computation is slightly more computation-intensive than the proposed approach. Moreover, the quantization processing of similarity consumes extra time [4]. However, the Bestagini's approach consumes much more time because each frame is required to regenerate an interpolation frame for comparison, which is really a very computation-intensive process. Moreover, the proposed approach can automatically separate the up-converted videos from the original frames, but the other two approaches need human intervention.

Table 7 Performance comparison of the existing FRUC detection algorithms

Detection method	Separation of different frame rate	Averaging computation time (100 frames)
Luo et al. [4]	Human intervention	2.80 s
Bestagini et al. [2]	Human intervention	184.40 s
Proposed approach	Automatic	1.25 s

5 Conclusions

In this paper, an efficient blind forensic approach is proposed to detect the presence of video FRUC, which is usually used to generate faked high frame-rate video or splice two videos with different frame rates. The candidate video is investigated by the frame-level analysis of local texture variation. The frame interpolation will exhibit some periodicity, which can be used to infer the original frame rate. The proposed approach is not only effective for simple frame repetition and FA, but also is effective for advanced video FRUC approaches such as MCI, OBMC and MHB. Moreover, experimental results on those up-converted videos which are further encoded with H.264/AVC show that the proposed approach is also robust to lossy video compression with moderate compression ratio. In future research, we will further investigate the mechanisms of various video FRUC techniques and attempt to further decide the specific FRUC approach when a candidate is detected as a faked video by FRUC.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China (61379143, 61232016, 61572183, U1405254), the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) (20120161110014) and the S&T Program of Xuzhou City (XM13B119) and the PAPD fund. This paper is also supported in part by Southwest University for Nationalities for the Fundamental Research Funds for the Central Universities (82000742). The authors appreciate the nice help from Mr Moses Odero for improving the English usages.

References

1. Bahrami K, Kot AC (2014) A fast approach for no-reference image sharpness assessment based on maximum local variation. IEEE Signal Process Lett 21(6):751–755
2. Bestagini P, Battaglia S, Milani S, Tagliasacchi M, Tubaro S (2013) Detection of temporal interpolation in video sequences. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 3033–3037
3. Bian S, Luo W, Huang J (2013) Exposing fake bitrate video and its original bitrate. In: Proceedings of IEEE international conference on image processing (ICIP), pp 4492–4496
4. Bian S, Luo W, Huang J (2014) Detecting video frame-rate up-conversion based on periodic properties of inter-frame similarity. Multimed Tools Appl 72(1):437–451
5. Bian S, Luo W, Huang J (2014) Exposing fake bit rate videos and estimating original bit rates. IEEE Trans Circ Syst Video Tech 24(12):2144–2154
6. Dong Q, Yang G, Zhu N (2012) A MCEA based passive forensics scheme for detecting frame-based video tampering. Digit Invest 9(2):151–159
7. Gironi A, Fontani M, Bianchi T, Piva A, Barni M (2014) A video forensic technique for detecting frame deletion and insertion. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6226–6230

8. Ha T, Lee S, Kim J (2004) Motion compensated frame interpolation by new block-based motion estimation algorithm. IEEE Trans Consumer Elec 50(2):752–759
9. Kang SJ, Cho KR, Kim YH (2007) Motion compensated frame rate up-conversion using extended bilateral motion estimation. IEEE Trans Consumer Elec 53(4):1759–1767
10. Kaufman P (1995) A guide to smarter trading-perry Kaufman on market analysis. Tech Anal Stock Commod 13(6)
11. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Info Foren Sec 10(3):507–518
12. Lin GS, Chang JF (2012) Detection of frame duplication forgery in videos based on spatial and temporal analysis. Inter J Pattern Recog Artif Intell 26(7):1250017
13. Liu H, Xiong R, Zhao D, Ma S, Gao W (2012) Multiple hypotheses Bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations. IEEE Trans Circ Syst Video Tech 22(8):1188–1198
14. Liu H, Li S, Bian S (2014) Detecting frame deletion in H.264 Video. In: Information security practice and experience, Springer, pp 262–270
15. Pan ZQ, Zhang Y, Kwong S (2015) Efficient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Trans Broadcasting 61(2):166–176
16. Rocha A, Scheirer W, Boult T, Goldenstein S (2011) Vision of the unseen: current trends and challenges in digital image and video forensics. ACM Comput Surveys 43(4):26–50
17. Software (2011) Available on http://avisynth.org.ru/mvtools/mvtools2.html. Accessed Dec 2015
18. Software (2011) Available on http://www.avs4you.com/AVS-Video-Converter.aspx. Accessed Dec 2011
19. Software (2011) Available on http://compression.ru/video/frame_rate_conversion/index_en_msu.html. Accessed Dec 2015
20. Software (2011) Available on http://www.imtoo.com/video-converter.html. Accessed Dec 2011
21. Xia ZH, Wang XH, Sun XM, Liu QS, Xiong NX (2014) Steganalysis of LSB matching using differences between nonadjacent pixels. Multimed Tools Appl. doi:10.1007/s11042-014-2381-8
22. Xia ZH, Wang XH, Sun XM, Wang BW (2014) Steganalysis of least significant bit matching using multi-order differences. Secur Commun Networks 7(8):1283–1291
23. Xiph.org Video Test Media [derf's collection]. http://media.xiph.org/video/derf/. Accessed Nov 2014
24. Wang W, Farid H (2006) Exposing digital forgeries in video by detecting double MPEG compression. In: Proceedings of the 8th workshop on multimedia and security, pp 37–47
25. Wang W, Farid H (2007) Exposing digital forgeries in video by detecting duplication. In: Proceedings of the 9th workshop on multimedia & security, pp 35–42
26. Wu Y, Jiang X, Sun T, Wang W (2014) Exposing video inter-frame forgery based on velocity field consistency. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 2674–2678
27. YUV Video Sequences. http://trace.eas.asu.edu/yuv/. Accessed Nov 2014

Min Xia is a PhD student in Hunan University, China. He is also a lecture in the Southwest University for Nationalities, China. His research interests is passive video forensics.

Gaobo Yang is a professor in Hunan University, China. He is also a key member of Hunan Provincial Key Laboratory of Networks and Information Security. He received the Ph.D. degree in Communication and Information System from Shanghai University in 2004. He is the PI of several projects such as Natural Science Foundation of China (NSFC), Special Pro-phase Project on National Basic Research Program of China (973) and program for New Century Excellent Talents (NCET) in university. Currently, his research interests are in the area of image and video signal processing, digital media forensics.

Leida Li is an associate professor in the School of Information and Electrical Engineering, China University of Mining and Technology. He obtained his PhD degree from Xidian University, China in 2010. During Jan 2014 to Jan 2015, he made his academic visit to Nanyang University of Technology, Singapore. His research interests include image quality assessment and image forensics.

Ran Li is a lecture in Xinyang Normal Unviersity, China. He obtained his PhD degree from Nanjing University of Posts and Telecommunications, China in 2013. His research interests include video frame rate up-conversiton.

Xingming Sun is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. He received the B.S. degree in mathematics from Hunan Normal University, Hunan, China, in 1984, the M.S. degree in computing science from the Dalian University of Science and Technology, Dalian, China, in 1988, and the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2001. His research interests include network and information security, digital watermarking, cloud computing security, and wireless network security.