

Detecting video frame-rate up-conversion based on periodic properties of inter-frame similarity

Shan Bian · Weiqi Luo · Jiwu Huang

Published online: 8 February 2013
© Springer Science+Business Media New York 2013

Abstract Video frame-rate up-conversion is one of the common operations for tampering digital videos in the temporal domain, such as creating fake high-quality videos and splicing two video clips with different frame rates. However, few existing works have been proposed for detecting this form of tampering operation. Based on the analysis of extensive experiments, we found that frame-rate up-conversion algorithms employed in most current video editing softwares will inevitably introduce some periodic artifacts into inter-frame similarity in the resulting video frame sequence. By analyzing such artifacts, we propose a simple yet very effective method to expose video after frame-rate up-conversion, and further estimate its original frame rate. The experimental results evaluated on 100 original videos at different frame rates have shown the effectiveness of the proposed method. The average detection accuracy can achieve as high as 99 % on noise-free videos in uncompressed and H.264/AVC formats. Besides, the proposed method is robust to noise as the detection accuracy could reach over 85 % and 95 % on noised videos with Gaussian white noise when SNR is 33 db and 36 db respectively.

Keywords Digital forensics · Video forensics · Frame-rate up-conversion · Fake quality videos

S. Bian · J. Huang
School of Information Science and Technology, Sun Yat-sen University,
Guangzhou 510006, People's Republic of China

S. Bian
e-mail: biانشan.sysu@gmail.com

J. Huang
e-mail: isshjw@mail.sysu.edu.cn

W. Luo (✉)
School of Software, Sun Yat-sen University, Ghuangzhou,
510006, People's Republic of China
e-mail: weiqi.luo@yahoo.com

1 Introduction

With the rapid development of video editing techniques, it is becoming much easier to tamper videos without leaving any visual clues for common users. Nowadays, digital video forgeries can be found everywhere in our daily life, making videos no longer reliable in many situations. Obviously, the abuse of those forged videos would potentially bring about many serious moral, ethical and legal consequences. Therefore, the corresponding forensic techniques face great challenges. Typically, forensic methods can be classified into two categories, that is, active and passive ways. The active technique needs some proactive operations [7, 8, 14, 24], such as inserting an imperceptible digital watermark or attaching a digital signature to the multimedia data at the time of data generation, and then uses this side information for tampering detection at a later time. While the passive technique has no need of side information, it is possible to provide forensics information on how multimedia data is acquired and processed via analyzing some inherent properties of digital multimedia. Recently, passive forensics have attracted more and more attention.

Up to now, most of passive forensic works are mainly focused on digital images [4, 11], while just a few literature pay attention to digital video (and audio) forensics. In [6, 9], the noise correlation in video has been explored to detect video region forgery intra a frame. Wang and Farid [20, 22] proposed a method to detect forgeries in double MPEG compressed videos based on double quantization artifacts, and Liao et al. [10] employed the similar features in detecting recompressed H.264/AVC videos. Based on the periodic pattern of blocking artifact strengths, Luo et al. [12] proposed a method to detect forged MPEG videos after frame removal or double compression with different GOP (group of pictures) structures. In [21], the inherent relationship between interlaced and de-interlaced videos has been studied and a forensic technique has been proposed by Wang and Farid. Bestagini et al. [1] proposed a method for identifying the type of video codec used in the first compression via analyzing the coding-based footprints in double compressed videos. Furthermore, Stamm et al. [19] have proposed new video frame deletion/addition forensic and anti-forensic techniques along with a new framework for evaluating the interplay between a forger and forensic investigator.

Video frame-rate up-conversion is one of commonly used operations in tampering digital videos in the temporal domain. For example, since videos at higher bitrate usually obtain higher click ratios and thus bring more revenue from advertisements, some uploaders of video-sharing websites possibly convert lower bitrate videos to higher ones by means of increasing frame rate before uploading. It is similar to the forensic scene of fake-quality MP3 detection as mentioned in Yang's work [25]. For another example, the actual video quality of some bargain-price DV, such as from TV direct sales or telemarketing, is not so satisfying as boasted "high frame rate" in the promotion, thus the actual frame rate is deeply doubted. Besides, when splicing two videos at different frame rates, the lower frame rate one is probably converted to higher frame rate in order to maintain the coherence of the frame rate in the resulting video. Therefore, we need some forensic techniques to authenticate video frame rate and defeat those fake-quality videos. To the best of our knowledge, there are few relative works in the literature specializing in the detection of fake-quality videos after frame-rate up-conversion.

In this paper, we focus on the detection of video frame-rate up-conversion in a passive way. Based on extensive experiments, we found that most of frame-rate up-conversion algorithms will introduce some periodic properties into inter-frame similarity in the resulting video. By analyzing these periodic artifacts, we firstly propose a simple yet very effective method to expose forged videos in frame rate, and then estimate their original frame rates. The experimental results evaluated on 100 original videos at different frame rates have shown the effectiveness of the proposed method. The average detection accuracy can achieve as high as 99 % on noise-free videos in uncompressed and H.264/AVC formats. Besides, the proposed method is robust to noise as the detection accuracy could reach over 85 % and 95 % on noised videos with Gaussian white noise when SNR is 33 db and 36 db respectively.

The rest of the paper is organized as follows. Section 2 describes the model of frame-rate up-conversion. Section 3 shows the details of the proposed detection method. Section 4 shows the experimental results and discussions. Finally, conclusions and future works will be given in Section 5.

2 Video frame-rate up-conversion

It is well-known that video frame rate varies across different capturing devices and/or real applications. For instance, videos taken by mobile cameras are usually at 15 fps (frames per second) or 20 fps, and those taken by digital cameras are at 24 fps or 25 fps, while some professional digital video recorders can shoot films with the frame rate as high as 30 fps or 60 fps. For some practical applications, however, we could easily convert video frame rate into desired higher one with the aid of video conversion softwares. In such a case, some extra frames have to be inserted into the resulting video. To this end, many interpolation algorithms are available for the frame insertion. The inserted frame $f[i]$ is typically modeled as the linear combinations of its adjacent frames as follows:

$$f[i] = \sum_{j=-k_1}^{k_2} w_j \cdot f[i + j], \quad j \neq 0 \quad (1)$$

where $f[i]$ denotes the i -th frame in the video, w_j is the interpolation weight, and the considered time window is $[-k_1, 0) \cup (0, +k_2]$, where k_1 and k_2 are positive integers.

To meet the real-time requirement in practice, we found that in most of popular video editing softwares, such as ImTOO video converter [15], AVS video converter [16] and Any video converter [17] from the TopTenReview site [18], only one adjacent frame is used to create the inserted frame, which means that in formula (1), $k_1 = 1$, $k_2 = 0$ and $w_{-1} = 1$. The positions of these inserted frames are mainly dependent on the relationship between the original frame rate and the resulting one. In this way, the video frame rate can be increased into the desired one very quickly using these video conversion softwares. Please note that there is no visual artifacts in the resulting video, especially when its original frame rate is higher than 20 fps.

In the following Section 3, we will analyze the statistical artifacts after the above-mentioned interpolation. We should note that there are some other advanced interpolations in the literature, such as motion-based algorithms [2, 3]. However, these

algorithms are usually time-consuming, and they are not employed in the existing softwares [18]. If these advanced algorithms are applied, the proposed method should be modified accordingly. Similar experimental results can be obtained by analyzing the corresponding interpolation algorithms, e.g. investigating the correlations of motion data between neighbor frames to detect the motion-based frame interpolations [2, 3].

3 Proposed method

As described previously, several near-duplications will be inserted into the resulting video after frame-rate up-conversion. It is expected that the similarity between the inserted frame and the corresponding neighbor will be much larger than that of two original frames, since the content (pixel values) between two original adjacent frames will change much more even in a very short time (e.g. 1/60 s), compared to those near-duplicated inserted ones (the corresponding experimental results and analysis are given in Section 4.1). Furthermore, since these inserted frames are presented periodically, the key issue of the proposed method is to determine whether there exists periodicity or not for those higher similarities in a questionable video clip. If there is, we will further estimate the period and the original frame rate for the up-converted video. Therefore, our method includes three steps: (1) inter-frame similarity measurement; (2) quantization of similarities and (3) estimation of the original frame rate. More detailed descriptions about the algorithm and time complexity analysis are given in the following subsections.

3.1 Inter-frame similarity measurement

In our work, we use the SSIM (structural similarity index measurement) [23] to measure the similarities $s[i]$ between two adjacent frames ($f[i]$, $f[i + 1]$) as follows:

$$s[i] = SSIM(f[i], f[i + 1]) \quad (2)$$

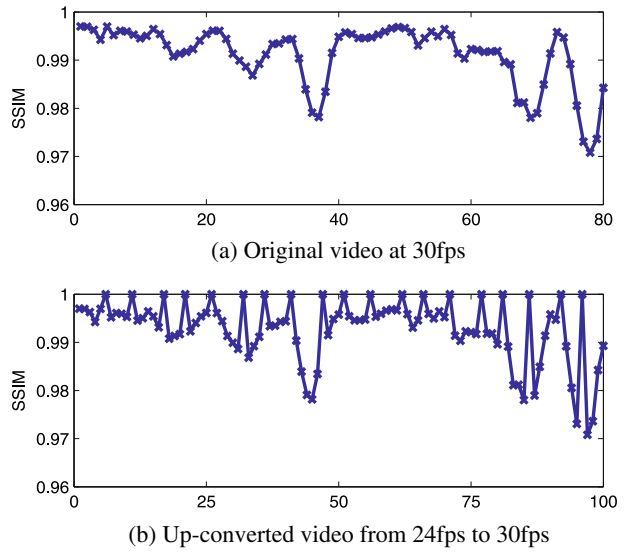
where $i = 1 \dots N - 1$, $f[i]$ is the i -th frame in the video and N is total frame number of a given video.

We use the set of $\{s[i] | i = 1, 2, \dots, N - 1\}$ to denote the inter-frame similarity of a given video. If i -th frame is an interpolated frame by video frame-rate up-conversion, the corresponding $s[i]$ is expected to be larger than others. What is more, it is also observed that such larger values would occur periodically as shown in Fig. 1b. In Fig. 1, we illustrate the similarities for both original and up-converted videos. In Fig. 1a, we obtain the test video clip by encoding the raw YUV sequence “akiyo” at the frame rate 30 fps, while in Fig. 1b, we firstly encode the raw YUV sequence at the frame rate 24 fps, and then up-converted it into 30 fps. Therefore, both test videos in Fig. 1 are at the same frame rate of 30 fps.

3.2 Quantization of similarities

For some slow-moving videos, however, the similarities of some adjacent frames in the original video are very close to those of interpolated ones. In such a case, the period introduced by the true inserted frames would be significantly confused. In this

Fig. 1 Illustrations of inter-frame similarities $s[i]$ for both original and up-converted videos. **a** Original video at 30 fps without up-conversion; **b** the up-converted video from 24 fps to 30 fps. The *horizontal axis* denotes frame index



step, we aim to find a proper threshold to differentiate those frames in the original video clip and the interpolated frames. To this end, we divide the frame similarities $s[i]$ into two non-overlapping subsets, that is, the similarity set P due to interpolation and the similarity set O of original frames. As mentioned in Section 3.1, it is expected that the values in P would occur periodically and are usually larger than those in O . In the ideal case, therefore, there exists a threshold τ_1 , subjecting to

$$o < \tau_1 < p, \forall o \in O, \forall p \in P \tag{3}$$

In the quantization step, values less than τ_1 , namely set of O , are quantized to zeros, while values greater than τ_1 , namely P , are set as ones. For all video clips with different contents, however, it is difficult or impossible to obtain such an ideal threshold due to the strong similarities between those adjacent original frames. For instance, Fig. 2 shows the quantized results for Fig. 1a and b with the same threshold $\tau_1 = 0.995$. It is observed that after quantization, most of the original similarities (see Fig. 2b) will become zeros, and most of the similarities due to interpolation (comparing Fig. 2a and c) will become ones. In Section 4.2, we will show that the quantization step will significantly increase the detection accuracies. Please note that some values supposed to be zeros in O have been quantized to ones and vice versa. Therefore, those ones after quantization operation in Fig. 2d will not occur periodically exactly. In next step, we try to estimate the corresponding period from the “noised” ones.

3.3 Estimation of original frame rate

In order to estimate the period of those quantized similarities in Fig. 2b and d. We firstly transform them into frequency domain using Discrete Fourier Transform (DFT) and get the normalized frequency spectrums as shown in Fig. 3a and b, respectively. Similar to the interpolation properties in digital images [5, 13], the

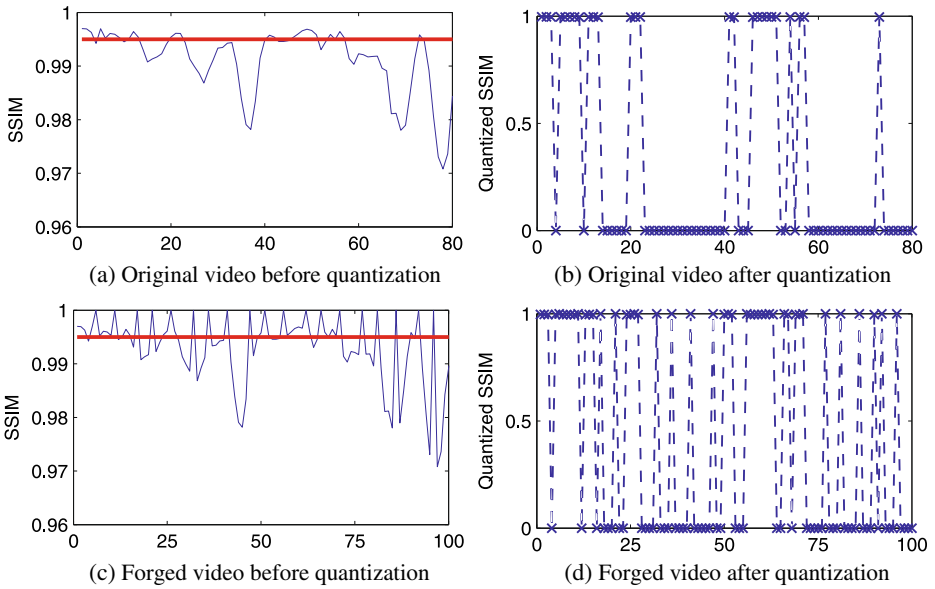
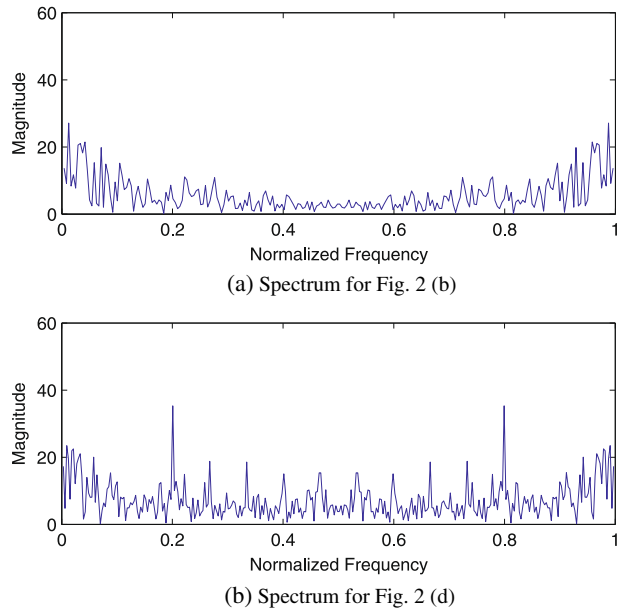


Fig. 2 Illustrations of quantized similarities $s[i]$ for ‘Akiyo’ in Fig. 1a and b, respectively. The threshold τ_1 in this example is set as 0.995

relationship between the interpolation factors r_1/r_2 and the positions of peaks f_p in the spectrum domain is as follows:

$$\frac{r_1}{r_2} = 1 - f_p \quad \text{or} \quad \frac{r_1}{r_2} = f_p \tag{4}$$

Fig. 3 Illustrations of the Fourier spectrum for ‘Akiyo’ in Fig. 2b and d, respectively. The horizontal axis indicates normalized frequencies



where r_1 and r_2 represent the frame rate before and after frame-rate up-conversion respectively, subjecting to $r_1 < r_2$. In our method, a frequency is regarded as a candidate peak f_p if its magnitude is τ_2 times greater than the average magnitude of the whole spectrum. Therefore, the threshold τ_2 can be used as the criterion to distinguish the original video and those up-converted ones, namely, if there exists such peaks whose magnitudes are τ_2 times greater than the average, the video is classified as a tampered one, vice versa.

Based on our experiments (please refer to Section 4.1 for more details), τ_2 is set as 2.5. In this example, there is no peak in Fig. 3a, and there are two peaks (the corresponding positions are 0.2 and 0.8) in Fig. 3b, which means that the video shown in Fig. 3a is an original one, while the video shown in Fig. 3b has been converted from some lower frame rate r_1 . In this case, we will further estimate the original frame rate r_1 based on the formula (4):

$$r_1 = (1 - f_p) \cdot r_2 \quad \text{or} \quad r_1 = f_p \cdot r_2 \quad (5)$$

Please note that video frame rates are usually fixed in practical applications. The commonly-used frame rates are 15 fps, 20 fps, 24 fps, 25 fps, 30 fps and 60 fps, respectively. In our method, therefore, we will select the consistent one from the estimated values r_1 with less than 10 % rounding error.

3.4 Time-complexity analysis

As described above, the proposed method includes three steps, that is, inter-frame similarity measurement, quantization of similarities and estimation of original frame rate. We will discuss the time-complexity of the proposed method in this section. We assume that the video resolution is W by H and the frame numbers of a given video clip is N . Table 1 demonstrates the pseudo-code for each step in the proposed method, and the corresponding time complexity. From this table, the time-complexity of the proposed method is mainly dependent on the calculation of SSIM and DFT in step one and step three respectively. In step one, we calculate the similarities between every two frames using SSIM [23], and the time complexity is $O(WHN)$ according to [23]. While the time complexity of step three (i.e. computation of DFT) is $O(N \log_2(N))$. Therefore, the total time-complexity of the proposed method is $O(WHN) + O(N \log_2(N))$, which means the computation time is related to the video resolution and frame numbers.

Table 1 Time complexity analysis of the proposed method

Step	Algorithm pseudo-code	Time complexity
1	for index:= 1:N-1 compute SSIM(frame(index), frame(index+1)) → s(index); end	$N * O(\text{SSIM}) = O(WHN)$
2	for index:= 1:N-1 quantize s(index); end	$N * O(1)$
3	compute DFT(s{1:N-1}) → d{1:N-1}; compute MEAN(d{1:N-1}), MAX(d{1:N-1});	$O(N \log_2(N))$

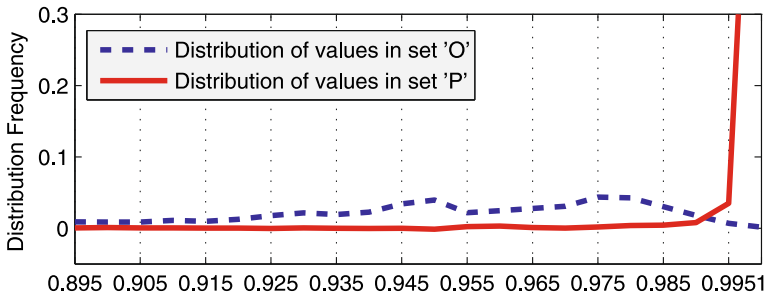


Fig. 4 Distributions for the values in sets of O and P

4 Experimental results and analysis

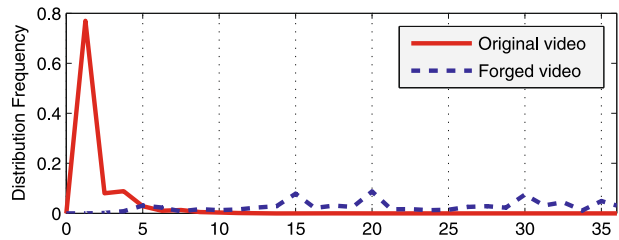
In our experiments, we randomly collect 100 uncompressed YUV sequences¹ with different contents, including news, sports, surveillance, vehicles and party and so on. Their resolutions are ranging from 176×144 to 1920×1080 pixels. Six commonly used frame rates, that is, 15 fps, 20 fps, 24 fps, 25 fps, 30 fps and 60 fps, have been tested. For each original YUV sequence, we firstly convert it into a video with a certain frame rate, and then convert the resulting video with another higher frame rate using ImTOO video converter[15]. Both uncompressed and H.264/AVC compressed video clips are employed in our experiments. In all, we obtain 3,000 up-converted videos with 15 combinations of six different frame rates as the positive instances. In addition, we compress each original YUV sequence into five different frame rates (from 20 fps to 60 fps) in both uncompressed and H.264/AVC formats. In all, there are 1,000 original videos as the negative instances.

4.1 Parameter selection

As described previously, τ_1 and τ_2 are two important parameters. In the proposed method, τ_1 is used to differentiate the similarities due to original frames and similarities due to interpolated frames (see Section 3.2), while τ_2 serves as a criterion of determining whether a video has been tampered with frame rate up-conversion (see Section 3.3). To obtain the two proper thresholds, we show the distributions of original similarities O and the interpolated ones P in Fig. 4 and the distributions of the ratios of the largest magnitude against the mean magnitude of the whole spectrum for both original and up-converted videos in Fig. 5, respectively. From Fig. 4, it is clearly observed that the values of P concentrate over 0.99, while those of O are spread around the range of $[0.003, 0.997]$ (we cannot show the whole range in the figure due to the page limitation) and most of them are smaller than values of P . So it is expected that the proper τ_1 should be around 0.99. In addition, from Fig. 5, it is observed that nearly 80 % ratio values for original videos are centered on the value of 1, but those values of up-converted videos are distributed over 10. Thus we expect that the proper τ_2 should be less than 10.

¹The test video clips are coming from the public website <http://trace.eas.asu.edu/yuv/> and AVS Workgroup (<http://www.avs.org.cn/>).

Fig. 5 Distributions of the ratios of the largest magnitude against the mean magnitude of the whole spectrum for both original and converted videos, respectively



Based on the previous analysis, we set τ_1 and τ_2 with different values in our experiments, where τ_1 ranges from 0.97 to 1 with a step size of 0.005, τ_2 ranges from 1.5 ($1.5 > 1$) to 10 with a step size of 1 (Please note that smaller step sizes, e.g. 0.001 or 0.1, would increase the detection accuracy with the sacrifice of the time complexity significantly. Based on our experiments, we found that the proposed steps i.e. 0.005 and 1, can obtain very satisfying detection results.). In all, there are 63 pairs of (τ_1, τ_2) . To find the best threshold pair, we randomly split the video clips into two non-overlapping subsets equally, namely, one subset is used for training and another is used for testing. After that, we apply the 63 pairs of (τ_1, τ_2) on the training data to train a reliable classifier under the principle of minimizing detection error rate, and then evaluate the classifier on the testing data. We repeat this process ten times and find that the best threshold pair (τ_1, τ_2) (when $\tau_1 = 0.995$ and $\tau_2 = 2.5$) is steady for every iteration. The ten detection accuracies evaluated on the testing data are demonstrated in Fig. 6. It is observed from the figure that the average detection accuracies are all above 99.6 % with a small deviation, which means the proposed method is effective for those videos with different contents.

4.2 Results and analysis

Based on previous experimental analysis, therefore, we set $\tau_1 = 0.995$ and $\tau_2 = 2.5$ in the following experiments. Tables 2 and 3 shows the detection results for uncompressed videos and H.264 compressed videos in different cases of frame-rate up-conversion. The average detection accuracy is computed as $1 - (FNR + FPR)/2$, where FNR denotes the False Negative Rate and FPR denotes False Positive Rate. From the tables, it is clearly observed that the proposed method works very well.

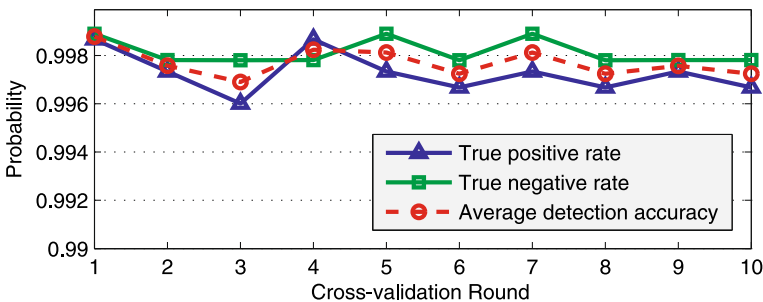


Fig. 6 Detection accuracies for ten iterations of testing. The x -axis presents the ten iterations of tests and y -axis is the detection accuracy on testing set

Table 2 Average detection accuracies for uncompressed videos (%)

Original frame rate	Resulting frame rate	False positive rate	False negative rate	Average detection accuracy
15	20	0	0	100
15	24	0	0	100
20			0	100
15	25	1	0	99.5
20			0	99.5
24			1	99
15	30	0	0	100
20			0	100
24			0	100
25			1	99.5
15	60	0	0	100
20			0	100
24			0	100
25			0	100
30			0	100

Most average accuracies are over 99 %. We should note that the detection rate is relatively lower when detecting up-converted videos converted from 24 fps to 25 fps. The main reason is that there is only one inserted frame into every 24 frames per second. Usually, such an inserted frame is easily quantized to zero with the threshold τ_1 , especially for the compressed videos (please compare the corresponding results in Tables 2 and 3). Furthermore, the original frames with high similarities (that is, those values in set O are quantized to ones) will also significantly confuse the period introduced by the true inserted frames in such cases.

In the following, therefore, we want to evaluate whether it is necessary to quantify the similarities (i.e. Step 2 described in Section 3.2) with a threshold τ_1 , namely, is it better to estimate the frame rate from Fig. 2c directly rather than Fig. 2d? To

Table 3 Average detection accuracies for H.264/AVC videos (%)

Original frame rate	Resulting frame rate	False positive rate	False negative rate	Average detection accuracy
15	20	0	0	100
15	24	2	0	99
20			0	99
15	25	1	0	99.5
20			0	99.5
24			5	97
15	30	0	0	100
20			0	100
24			0	100
25			0	100
15	60	0	0	100
20			0	100
24			0	100
25			0	100
30			0	100

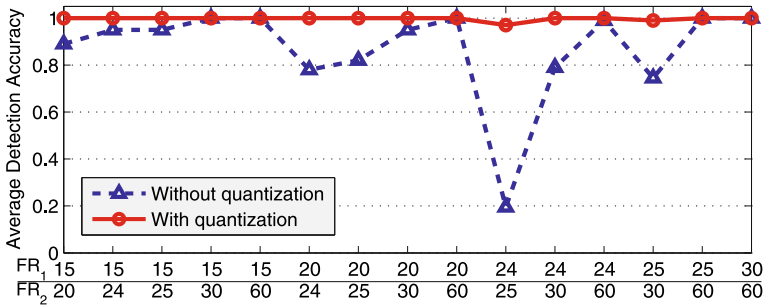


Fig. 7 Detection accuracies with and without quantization

this end, we evaluate the method without quantization on different cases of frame rate up-conversion and obtain the average detection accuracies, which are shown in Fig. 7 compared to results of the proposed method with quantization. Here, the x-axis indicates the original frame rate FR_1 and the resulting frame rate FR_2 after frame-rate up-conversion. The red line and the dash blue line denote the average detection accuracies for the methods with and without quantization operation. It is clearly observed that the quantization operation can improve the detection performances significantly in most cases, especially in the case of converting 24 fps to 25 fps, where the average improvement is as high as 77.5 %.

4.3 Robustness against noise contamination

In the previous section, we just consider the robustness against lossy H.264 compression, which is the most popular operation in digital video. Besides of this, we also take noise contamination into consideration. In doing so, we may perform some de-noising operations before using the proposed method, for example, we may apply a

Table 4 Average detection accuracies (%) for noised videos where SNR=33 db

Original frame rate	Resulting frame rate	False positive rate	False negative rate	Average detection accuracy
15	20	1	28	85.5
15	24	1	28	85.5
20			24	87.5
15	25	1	27	86
20			32	83.5
24			30	84.5
15	30	1	24	87.5
20			26	86.5
24			33	83
25			31	84
15	60	1	27	86
20			25	87
24			31	84
25			29	85
30			25	87

Table 5 Average detection accuracies (%) for noised videos where SNR=36 db

Original frame rate	Resulting frame rate	False positive rate	False negative rate	Average detection accuracy
15	20	1	7	96
15	24	1	5	97
20			24	87.5
15	25	1	7	96
20			12	93.5
24			10	94.5
15	30	1	1	99
20			3	98
24			11	94
25			14	92.5
15	60	1	10	94.5
20			6	96.5
24			7	96
25			7	96
30			3	98

mean filter with a 4×4 kernel to reduce the noise. In the experiment, we evaluate the proposed method on those noised videos with different strengths, i.e. 33 db, 36 db. The experimental results are shown in Tables 4 and 5 respectively. From these tables, it is observed that the results are satisfactory when the noise strength is 36 db. However, when the noise is strong such as 33 db, the average detection accuracy drops to 85.5 %. Please note that we employ the same thresholds i.e. $\tau_1 = 0.995$ and $\tau_2 = 2.5$ in these tables. The performances of the proposed method could be improved if we adjust the thresholds. For example, we could reduce the threshold τ_2 slightly, such as 0.99 to reduce the *FNR* (False Negative Rate) as shown in the following Table 4. However, there is a tradeoff between the *FPR* (False Positive Rate) and *FNR*. To achieve more proper thresholds, those noised videos with different strengths should be included in the training stage. Besides, other advanced de-noising methods can be applied before using the proposed method.

5 Concluding remarks and future works

Video frame-rate up-conversion is one of the commonly used operations for tampering digital videos in the temporal domain. Based on our extensive experiments, we found that such operation in most of popular video editing softwares [18] has to insert some frames into original frames periodically. By analyzing the similarities between adjacent frames for a questionable video sequence, it is possible to find out the inserted frames and estimate their corresponding period. In this paper, we first present a simple yet very effective method to expose such tampered videos after frame-rate up-conversion based on the periodic properties of inter-frame similarity, and further to estimate the original frame rate. The experimental results evaluated on 100 original videos at different frame rates have shown the effectiveness of the proposed method. The average detection accuracy can achieve as high as 99 % on noise-free videos in uncompressed and H.264/AVC formats. Besides, the proposed method is robust to noise as the detection accuracy could reach over 85 % and 95 %

on noised videos with Gaussian white noise when SNR is equal to 33 db and 36 db respectively.

In our future work, we will extend our method to identify tampered videos using more advanced frame interpolated algorithms reported in the literature, such as [2] and [3]. If these advanced interpolation algorithms are employed, the first step of the proposed method has to be modified according to the specific interpolation algorithm under investigation. For example, a motion-compensation based frame interpolation [2, 3] may leave some traces in the motion information of the inserted frames, such as motion vectors or residuals. In such a case, we may measure the frame similarity based on the motion vectors in the step one. Please note that since the inserted frames would also occur periodically, the last two steps in the proposed method could also be applied similarly. Besides of this, other advanced de-noising methods would be taken into consideration to further improve the robustness of the proposed method. Furthermore, we will investigate whether it is possible to expose videos after frame-rate down-conversion in our future work.

Acknowledgements This work is supported by the 973 Program (2011CB302204), NSFC (61272191,61003243,61173081), Zhujiang Science & technology (2011J2200091), and Guangdong Natural Science Foundation (S2011020001215).

References

1. Bestagini P, Allam A, Milani S, Tagliasacchi M, Tubaro S (2012) Video codec identification. In: Proceedings of ICASSP, pp 2257–2260
2. Choi BT, Lee SH, Ko SJ (2000) New frame rate up-conversion using bi-directional motion estimation. *IEEE Trans Consum Electron* 46:603–609
3. Choi BD, Han JW, Kim CS, Ko SJ (2007) Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Trans Circuits Syst Video Technol* 17:407–416
4. Farid H (2009) Image forgery detection. *IEEE Signal Process Mag* 26(2):16–25
5. Gallagher AC (2005) Detection of linear and cubic interpolation in JPEG compressed images. In: Proceedings of 2nd Canadian conference on computer and robot vision, pp 65–72
6. Hsu CC, Hung TY, Lin CW, Hsu CT (2008) Video forgery detection using correlation of noise residue. In: Proceedings of IEEE 10th workshop multimedia signal processing, pp 170–174
7. Huang HC, Chang FC (2013) Hierarchy-based reversible data hiding. *Expert Syst Appl* 40(1): 34–43
8. Huang HC, Fang WC (2010) Metadata-based image watermarking for copyright protection. *Simulation Modelling Practice and Theory* 18(4):436–445
9. Kobayashi M, Okabe T, Sato Y (2010) Detecting forgery from static-scene video based on inconsistency in noise level functions. *IEEE Trans Inf Foren Sec* 5(4):883–892
10. Liao D, Yang R, Liu H, Li J, Huang J (2011) Double h.264/avc compression detection using quantized nonzero ac coefficients. *Proc. SPIE* 7880:78800Q–78800Q-10
11. Luo W, Qu Z, Pan F, Huang J (2007) A survey of passive technology for digital image forensics. *Front Comput Sci Chin* 1(2):166–179
12. Luo W, Wu M, Huang J (2008) Mpeg recompression detection based on block artifacts. *Proc SPIE* 6819:68190X–68190-12
13. Mahdian B, Saic S (2008) Blind authentication using periodic properties of interpolation. *IEEE Trans Inf Foren Sec* 3:529–538
14. Ritchey PC, Rego VJ (2012) A context sensitive tiling system for information hiding. *JIH–MSP* 3(3):212–226
15. Software (2011) Available on <http://www.imtoo.com/video-converter.html>. Accessed Dec 2011
16. Software (2011) Available on <http://www.avs4you.com/AVS-Video-Converter.aspx>. Accessed Dec 2011
17. Software (2011) Available on <http://www.any-video-converter.com/>. Accessed Dec 2011

18. Softwares (2011) Available on <http://video-converter-software-review.toptenreviews.com/>. Accessed Nov 2011
19. Stamm MC, Lin WS, Liu KJR (2012) Temporal forensics and anti-forensics for motion compensated video. *IEEE Trans Inf Foren Sec* 7(4):1315–1329
20. Wang W, Farid H (2006) Exposing digital forgeries in video by detecting double mpeg compression. In: *Proceedings of ACM 8th workshop on multimedia and security*, pp 37–47
21. Wang W, Farid H (2007) Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Trans Inf Foren Sec* 2:438–449
22. Wang W, Farid H (2009) Exposing digital forgeries in video by detecting double quantization. In: *Proceedings of ACM 11th workshop on multimedia and security*, pp 39–48
23. Wang Z, Bovik A, Sheikh H, Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13:600–612
24. Weng S, Pan JS, Gao X (2012) Reversible watermark combining pre-processing operation and histogram shifting. *JJIH-MSP* 3(4):320–326
25. Yang R, Shi YQ, Huang J (2009) Defeating fake-quality mp3. In: *Proceedings of ACM 11th workshop on multimedia and security*. NY, USA, pp 117–124



Shan Bian received the B.S. degree from Sun Yat-sen University, China, in 2009. She is currently pursuing the Ph.D. degree in School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China.

Her research interests include digital image forensics and video forensics.



Weiqi Luo received the Ph.D. degree from Sun Yat-sen University, China, in 2008. He is currently an associate professor at the School of Software, Sun Yat-sen University, China, and a researcher at the Guangdong Key Laboratory of Information Security Technology, Guangzhou, China.

His research interests include digital multimedia forensics, pattern recognition, steganography, and steganalysis.



Jiwu Huang received the B.S. degree from Xidian University, China, in 1982, the M.S. degree from Tsinghua University, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, in 1998.

He is currently a professor with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. His current research interests include multimedia forensics and security.

Dr. Huang received the China National Funds for Distinguished Young Scientists from NSF of China in 2003. He received the first-grade award in science research from the National Ministry of Education, China, in 2004. He and his students won the best student paper award in the 8th International Workshop on Digital Watermarking (Pusan, Korea) in 2008. He has served as a technical program committee member for many international conferences. He serves as a member of IEEE CASS Multimedia Systems and Applications Technical Committee, and a member of IEEE SPS Information Forensics and Security Technical Committee. He is an Associate Editor for IEEE Transactions on Information Forensics and Security, LNCS Transactions on Data Hiding and Multimedia Security (Springer), and EURASIP Journal on Information Security (Hindawi).