A Novel Video Inter-frame Forgery Model Detection Scheme Based on Optical Flow Consistency

Juan Chao[1], Xinghao Jiang[1,2], and Tanfeng Sun[1,2,3,*]

[1] School of Information Security Engineering Shanghai Jiao Tong University, Shanghai 200240, China
{xhjiang,tfsun}@sjtu.edu.cn
[2] National Engineering Lab on Information Content Analysis Techniques, GT036001, Shanghai 200240, China
[3] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark 07102, USA

Abstract. In this paper, a novel video inter-frame forgery detection scheme based on optical flow consistency is proposed. It is based on the finding that inter-frame forgery will disturb the optical flow consistency. This paper noticed the subtle difference between frame insertion and deletion, and proposed different detection schemes for them. A window based rough detection method and binary searching scheme are proposed to detect frame insertion forgery. Frame-to-frame optical flows and double adaptive thresholds are applied to detect frame deletion forgery. This paper not only detects video forgery, but also identifies the forgery model. Experiments show that our scheme achieves a good performance in identifying frame insertion and deletion model.

Keywords: optical flow consistency, frame deletion, frame insertion, forgery model identification.

1 Introduction

The development of digital equipment has made surveillance videos important evidences in court. How to detect the integrity and authenticity of videos has become an importance field in information security [1].

Video forgery detection includes active detection and passive detection. Active video forgery detection based on watermark and digital signature has been researched for years and has got much progress [2]. Active detection depends on watermark or signature. However, most cameras don't have such functions, making it impossible.

Passive video forgery detection extracts internal features of videos and has caught much attention nowadays. Prof. A. De proposed a method to uncover video forgery based on readout noise introduced by the Readout from camera CCD [3]. M. Kobayashi tried to detect suspicious surveillance videos with noise characteristics [4]. D.D. Liao proposed a method to detect double H.264/AVC compression detection

* Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Pérez-González (Eds.): IWDW 2012, LNCS 7809, pp. 267–281, 2013.

using quantized nonzero AC coefficients [5]. W.H. Wang and H. Farid proposed a video tampering tracing technique in de-interlaced and interlaced video [6].

Digital video forgery includes inter-frame forgery and intra-frame forgery. Intra-frame forgery is similar to image forgery. Intra-frame forgery detection is much easier comparing with inter-frame forgery detection, since image forgery detection research has got many achievements. M. K. Johnson proposed an image forgery detection method based on the lighting inconsistencies [7]. A. Swaminathan detected image forgery via intrinsic fingerprints of both inside and outside processing operations [8].

From above discussion, this paper focuses on detecting forgery and identifying forgery model, which has not been considered by other researchers. They mainly focus on if a video has been tampered but don't analyze the forgery model. In section 2, the optical flow generation and how inter-frame forgery affects it are introduced. Section 3 gives the framework and detailed procedures of our scheme, experiments are shown in section 4 and a brief conclusion will be conducted in the last section.

2 Optical Flow Analysis for Inter-frame Forgery Video Frames

The Lucas Kanade optical flow is proposed by B.D. Lucas and T. Kanade, it has been widely used in layered motion, mosaic construction and face coding, but has never been used in video forgery detection. In this paper, we found that the optical flow is sensitive to inter-frame forgery and analysis will be given to prove it. Based on this finding, we innovatively use it in our inter-frame forgery detection.

2.1 Optical Flow Generation in the Video

The main steps to extract the Lucas Kanade optical flow are as follows:

1. Given two images, a spatial sampling is conducted to reduce the computational complexity. For each image, take its odd rows to form an odd image and their even rows to form an even image as is shown in figure 1.

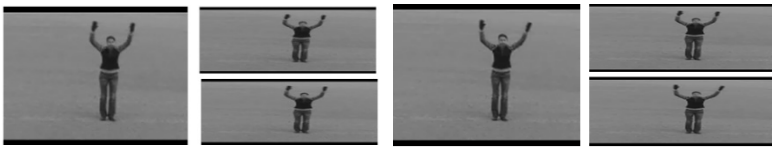

Fig. 1. Spatial sampling of image1 and image2

2. A pyramid is built for each image. In the figure below, the picture in the bottom is the original image, and the fours above it are the pyramid built on it.

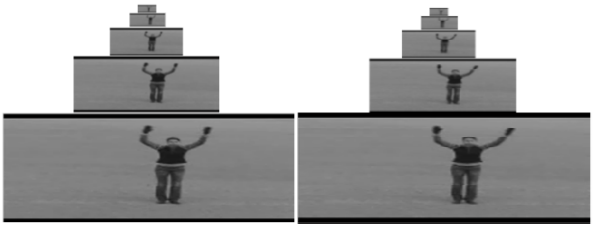

Fig. 2. 5-layer pyramid of odd image1 and odd image 2

3. Estimate motion vectors in both X and Y directions of each layer in figure 2 from top to bottom. The first frame in figure 3 is the motion vector between the top layers. Accordingly, the last one is the motion vector between the bottom layers.

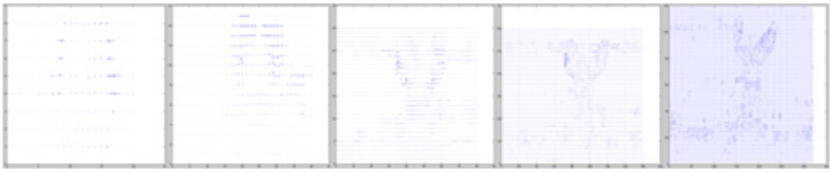

Fig. 3. Motion vectors of the five pyramid layers

4. Expand the motion vector in the top layer twice in both X and Y directions and add it to its lower layer, then smooth the sum of this two layers. Repeat these steps until the bottom layer. In the top row below, the first figure is the first motion vector in figure 3; the second is the sum of the first expanded motion vector and second motion vector in figure 3; accordingly, the last figure is the sum of all motion vectors in figure 3. The five pictures in the bottom row are the smoothing results of relative figures in the top row; the last is the final optical flow between the odd image 1 and odd image 2 in figure 1.

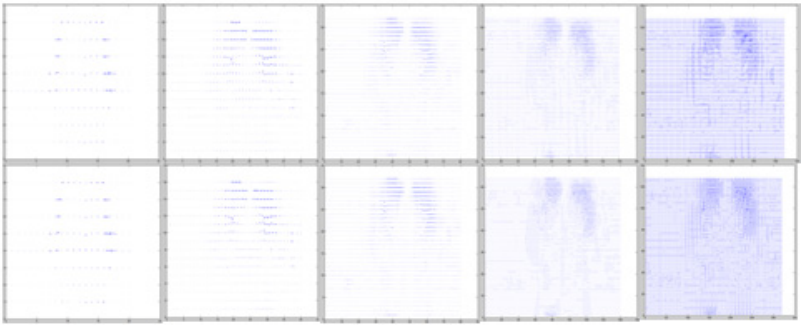

Fig. 4. Sum of motion vectors of the five pyramid layers before and after smoothing

5. Below is the optical flow figures extracted with image1 and image2, odd image1 and odd image 2, even image 1 and even image 2. The odd optical flow and the even one are almost the same, while the original one is the sum of the two, reducing the computing complexity while keep the optical flow feature.

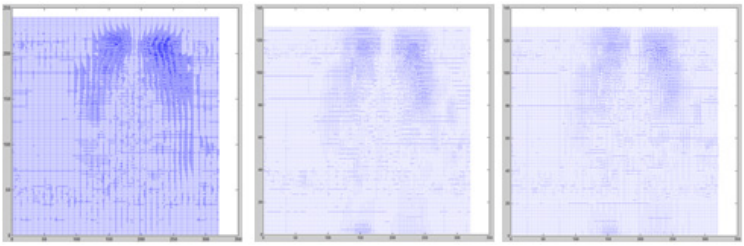

Fig. 5. Optical flow of original images, odd images and even images

6. For two frames *M* and *N*, two optical flow figures $OFX_{(m,n)}$ and $OFY_{(m,n)}$ which are the optical flow vectors in the 2D space are computed by adding the absolute values of the optical flow in each pixel *(i, j)*with equation (1).

$$S_{(m,n)}(x) = \sum_{i=1}^{width} \sum_{j=1}^{height} OFX_{(m,n)}(i,j) \tag{1}$$

Where $S_{(m,n)}(x)$ is the sum of optical flow values between frame M and frame N in the X direction, and X can be replace with y to calculate $S_{(m,n)}(y)$, which is the sum of optical flow values between frame M and frame N in the Y direction, width and height are the number of pixels in each row and each column of the optical flow figure.

2.2 Analyzing Video Features of Inter-frame Forgery by Optical Flow

In this section, several examples based on the KTH video database are given to demonstrate how the inter-frame forgery affects the optical flow consistency.

Optical Flows of Original Video

In above figure, these six adjacent frames in the top row are extracted from an original video, and the five figures in the second row are the Lucas Kanade optical flows between adjacent frames in the top flow, that's to say, the K^{th} optical flow is computed with the K^{th} frame and the $(K+1)^{th}$ frame. In above figure, the optical flow in each figure mainly focuses on the upper half. The two histograms in the third row are the total optical flow values in X and Y directions of the five optical flow figures. From the histograms, the five bars in both X and Y directions are almost the same, which means their optical flows are consistent.

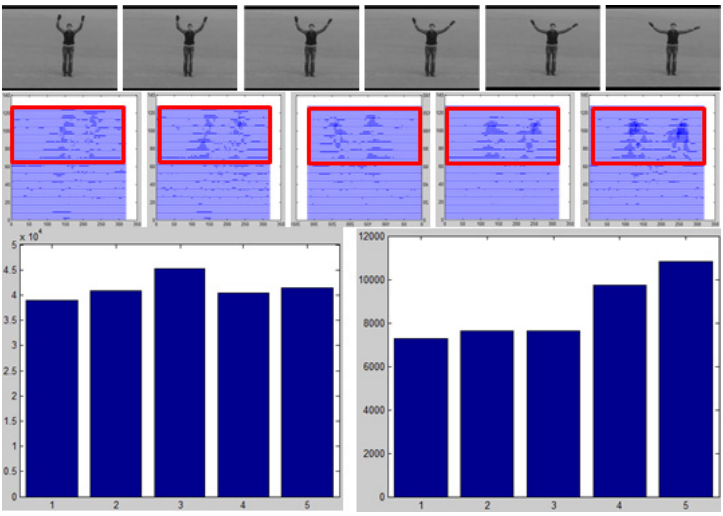

Fig. 6. Original video frames and their optical flows

Optical Flows of Frame Insertion Video

In above figure, these first three frames and last three frames in the top are extracted from two different videos. Among these five Lucas Kanade optical flow figures in the middle row, the third one is computed by the third and fourth frames and it is quite different from the others. From the optical value histogram in the bottom row, the third optical flow bar is in X direction and Y direction are much higher than others, which means that the optical flow are no longer consistent due to the insertion.

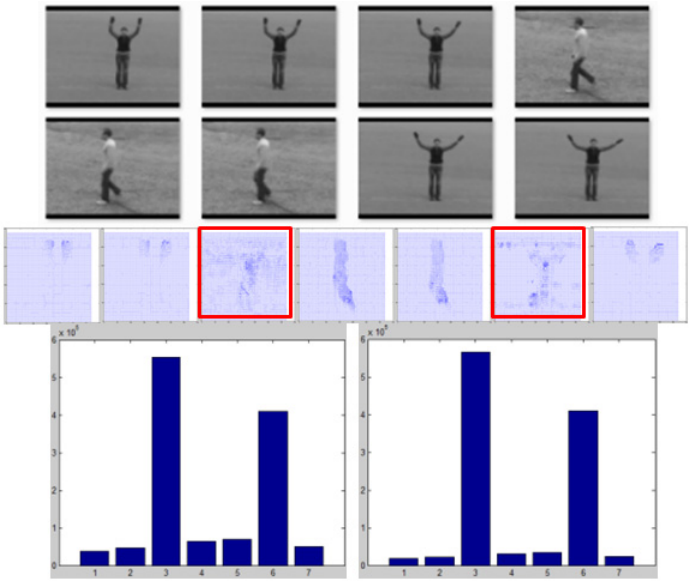

Fig. 7. Video frame insertion and their optical flows

Optical Flows of Frame Deletion Video

In above figure, these six frames in the top row are extracted from the same video, but they are not adjacent in the original video, the first three frames are the 6th, 7th, 8th frames and the last frames are the 151th, 152th, 153th frames. Though 142 frames are deleted, there is not much visible difference. However, in the optical flow figures, the third optical flow figure spreads almost the whole figure, compared to the other four where the optical flows focus mainly in the upper half. From the two optical flow histograms, the optical flow value at the deletion point is about 3 times than other points in the X direction and about 5 times larger than others in the Y direction.

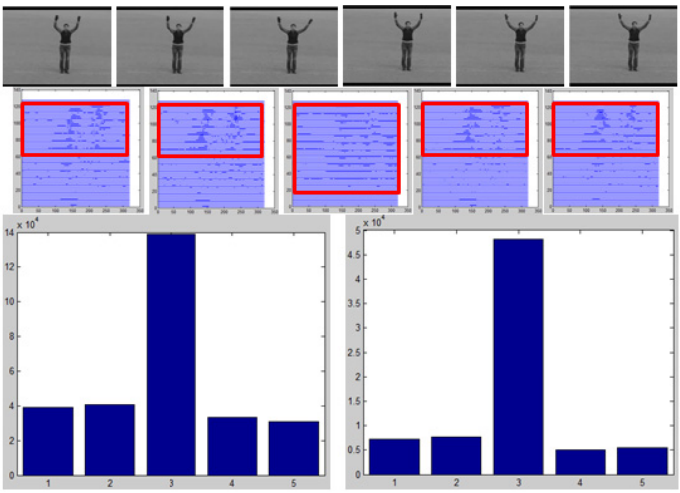

Fig. 8. Video frame deletion and their optical flows

From these three figures above, it can be seen that the optical flow is sensitive to frame insertion and frame deletion forgery. Thus in this paper, the optical flow is used as the main feature for inter-frame tampering detection.

3 Insertion and Deletion Forgery Detection Procedure

3.1 Framework of Inter-frame Forgery Detection Scheme

In this paper, an optical flow consistency based inter-frame forgery detection scheme is proposed. The main framework of our algorithm is as follows:

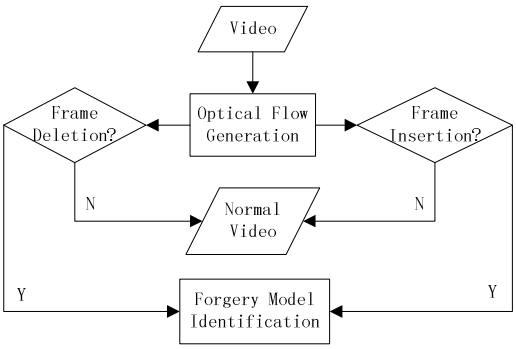

Fig. 9. Framework of inter-frame forgery detection scheme

The detailed scheme of our algorithm is:

1. For a given video, first generate the optical flow as is given in Section 2.
2. The optical flow is used to detection frame insertion and frame deletion forgery separately according to the procedures in Section 3.2 and 3.3.
3. If forgery is detected, identity their tampering model according to the difference between frame insertion and frame deletion.
4. If no frame insertion or deletion is detected, than mark a video as normal.

With this scheme, our algorithm can not only detect inter-frame forgery, but also identify forgery model, i.e. identify if it's frame insertion or frame deletion.

3.2 Frame Insertion Forgery Detection Procedure

For frame insertion forgery, a window based rough detection is proposed. By dividing a video into windows and computing the optical flow between the first frame and the last frame in each window, the detection time has been largely reduced. From figure 7 it's clear that the optical flow at an insertion point is hundreds of times larger than others, and the optical flow between two non-adjacent frames in figure 8 is only several times larger than that of adjacent frames. Even though the window mechanism will compute the optical flow of two non-adjacent frames and increase the optical flow value by several times, compared to the hundreds of times for insertion forgery, the window mechanism won't influence the optical flow much, but greatly fasten the detection speed. So the window mechanism is proposed for fast rough insertion detection.

In this paper, window mechanism is not used in frame deletion since inconsistency of deletion forgery is smaller comparing with insertion forgery. If window mechanism is applied, it will narrow this inconsistency further and result in more missed detections.

Below is the detailed process of the video frame insertion detection method.

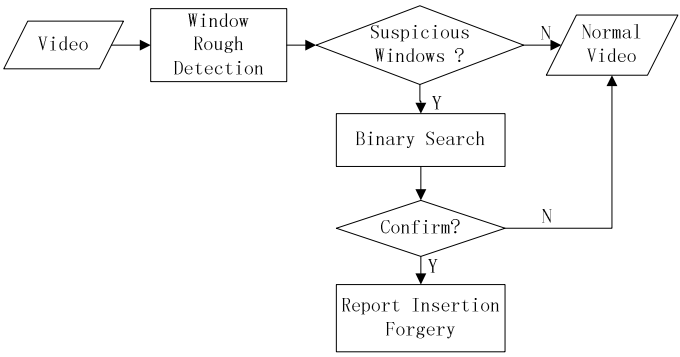

Fig. 10. Frame insertion forgery detection procedure

1. Each video is divided into equal-size windows. Here the window size is 16, because the optimal size for binary search is 2^n, and 16 is less than the frame rate (25Fps), ensuring that there won't be an integrate insertion in a window.
2. Then compute the Lucas Kanade optical flows $S_k(x)$ in X and $S_k(y)$ in Y direction between the first and the last frame in each window with equation 1 where X can be replaced with Y, but *m* and *n* are replaced with the first frame and last frame in a window.
3. Compute the average optical flow in X and Y directions with below equation (where X can be replaced with Y). *NOW* is the number of windows in a video; *AOF(x)* is the average optical flows in the X direction :

$$AOF(x) = \frac{1}{NOW} \sum_{k=1}^{NOW} S_k(x) \tag{2}$$

4. When the optical flow in a window *K* meets equation (3) either in X direction or Y direction where *T=2,* it means that the optical flow in this window is larger than the threshold and this window should be further detected.

$$S_k(x) \geq T * AOF(x) \tag{3}$$

5. Suspicious windows are binary researched to locate shift point. Each window is divided into two equal sub windows, optical flows between the first frame and the last frame in each sub window are computed in both X and Y direction. If the optical flows in left sub window and right sub window meet equation (4) where X can be replaced with Y, then a shift point may exist in the left sub window; go on with binary search in the left sub window until window size is 1. Here *L* means the left sub window, and *R* is the right sub window.

$$S_L(x) \geq T * S_R(x) \tag{4}$$

Else if the optical flow in X direction or Y direction meets equation (5) where X can be replaced with Y for the Y direction, then a shift point may exist in the right sub window, go on binary searching it until the sub window size is 1.

$$S_R(x) \geq T * S_L(x) \tag{5}$$

If a shift point is detected, then frames before and after this point are quite different in their optical flows. If optical flow between frame *K* and *(K+1)* and that between frame *(K+1)* and *(K+2)* meet (4), then *(K+1)* is a shift point. Else if two optical flows meet (5); then *(K+2)* is a shift point.

If the optical flows meet none of the two equations (4) and (5), then it means that there is no shift point in this suspicious window.

6. After the binary research in all suspicious windows, further detection is conducted to identify insertion part. For each point *I* and *J*, three optical flows will be computed: optical flow between *(I-2)* and *(I-1)*, optical flow between *(I-1)* and *(J+1)*, optical flow between *(J+1)* and *(J+2)*. If they are similar, it means that the video frames before the shift point *I* and after the shift point *J* come from the same video, so frames between these two shift points are inserted.

3.3 Frame Deletion Forgery Detection Procedure

For frame deletion forgery, the differences of optical flows are much smaller than frame insertion forgery. So in our method, optical flows between all adjacent frames are computed. Below is the detailed procedure of the frame deletion forgery detection.

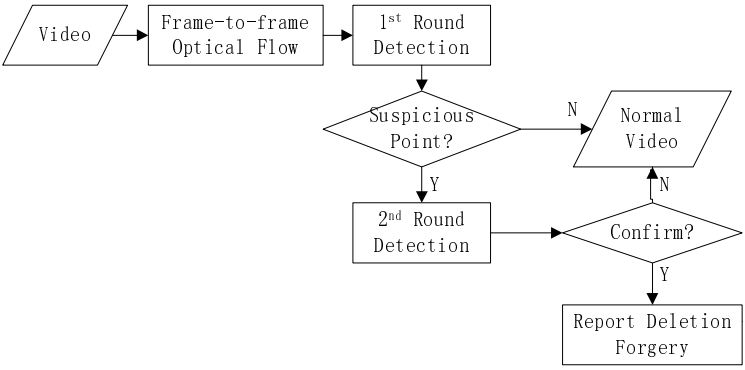

Fig. 11. Frame deletion forgery detection procedure

1. Each video is firstly divided into individual frames.
2. Then compute the Lucas Kanade optical flow between the I^{th} frame and the $(I+1)^{th}$ frame.

3. Compare each optical flow with its adjacent frames. If it meets equation (6) in X direction or Y direction where X can be replaced with Y, then it's a suspicious deletion position. Where T_1 is 2 in this paper.

$$S_{(k,k+1)}(x) \geq \frac{T_1}{4} \sum_{i=-2,i\neq 0}^{2} S_{(k-i,k-i+1)}(x) \tag{6}$$

4. Compute the average optical flow in both X and Y directions with below equation, where $S_{(k,k+1)}(x)$ is computed with equation (1); *NOF* is the number of optical flow figures in a video; *AOF(x)* is the average optical flow in the X. Replace the X in this equation with Y to get the average optical flow:

$$AOF(x) = \frac{1}{NOF-1} \sum_{k=1}^{NOF-1} S_{(k,k+1)}(x) \tag{7}$$

5. When the optical flow of the suspicious point in X direction or Y direction meets equation (8), it means that the optical flow between these two frames is much larger than average optical flow and some frames have been deleted between them. Where T_2 is 3.5 in this paper.

$$S_{(k,k+1)}(x) \geq T_2 * AOF(x) \tag{8}$$

6. Compare all $S_{(k,k+1)}(x)$ in the video, if none of the optical flow meets the equation (8) in neither X direction nor Y direction, then this video hasn't been tampered with frame deletion.

4 Simulation Experiments and Results

Video forgery model detection research just began in recent years and there are no other optical flow based video forgery detection methods, so here in this paper we didn't do comparison with other researchers' work.

4.1 Evaluation Standards

To evaluate the detection efficiency, the recall rate (R_r) and precision rate (R_p) are used, which are common standard in video and image related detection and classification research. The recall rate is the percentage of correctly detected videos among all tampered videos; a high recall rate can well prove the detection accuracy. The precision rate is the percentage of correctly detected videos among all the detected ones; a high precision rate can well demonstrate a low false alarm rate.

$$R_r = \frac{N_c}{N_c + N_m} \times 100\% \tag{9}$$

$$R_p = \frac{N_c}{N_c + N_f} \times 100\% \ . \tag{10}$$

Where N_c is the number of correctly detected video forgeries; N_m is the number of missed video forgeries; N_f is the number of falsely detected video forgeries.

4.2 Test Video Databases

Original Video Database
In this section, a large number of simulation experiments based on KTH database are conducted to evaluate our algorithm. The frame rate of videos in our test is 25Fps.

Frame Insertion Video Database Built with CBCD Scripts
The first test video database is generated with TRECVID Content Based Copy Detection (CBCD) scripts. To use the script, we first generate two folders: Reference and Non-reference, and then this script can automatically generate frame insertion videos by insertion a randomly chosen Reference video segment to a randomly chosen Non-reference video with random length.

Frame Insertion Video Database with OpenCV Library
To demonstrate that our frame insertion forgery detection method is robust to different kinds of frame insertion tools, we tried to generate another test video database with the OpenCV function library in C and C# coding language on visual studio 2010. Below is the frame insertion forgery database generation procedure with OpenCV.

For each video *I*, we insert *N* frames from video *(I+1)* to it. These *N* frames are selected randomly from video *(I+1)* and they are inserted to video *I*. In our test, we test different values of *N* (*N=100, N=25*) to evaluate the robustness to different insertion length. *N=100* (4s) is chosen as a representation of long insertion, and *N=25* is chosen to represent a short insertion based on the assumption that only video segments with meaningful activities are inserted and a meaningful activity will last for at least 1s.

Frame Deletion Video Database with OpenCV Library
Since the CBCD scripts can only generate frame insertion videos, we generated frame deletion forgery database with the OpenCV function on visual studio 2010.

For each video *I*, we delete *N* frames in it from *K* to *(K+N-1)*. In our test, we test different values of *N (N=100, N=25*) so as to evaluate the robustness of the proposed frame deletion forgery detection algorithm.

4.3 Experiment on Frame Insertion Forgery Model's Detection

Test Results of Frame Insertion Database Built with CBCD Scripts
From the experiment in table 1, 3000 frame insertion videos are tested. The frame insertion detection recall rate reaches 95.43% and the precision rate reaches 95.34%.

Table 1. Test Results with Frame Insertion Database 1

N_c	N_m	N_f	$R_r(\%)$	$R_p(\%)$
2863	137	140	95.43	95.34

Test Results of Frame Insertion Database Built with OpenCV Library

In the table above, different frame insertion numbers are tested. The detection recall rate is 94.33% with 100 frames insertion, while with 25 frames insertion, it drops to 92.67%. The detection precision rate with 100 frames insertion is 97.92%, but for 25 frames insertion, it reaches 98.58%. From the results, it can be found that when inserting fewer frames, the detection recall rate will drop, which is consistent to the theory that our visual system can find large changes easily but will skip the small changes. Accordingly, the low recall rate means that in the first round of window detection, less suspicious are detected, and in binary search round, more suspicious windows are rejected. Accordingly, less false detections occur in the first round and more false detections are rejected in the binary search, so precision rate increases with fewer frames inserted.

Table 2. Test results Frame Insertion Database 2

	N_c	N_m	N_f	$R_r(\%)$	$R_p(\%)$
N=100	566	34	12	94.33	97.92
N=25	556	44	8	92.67	98.58

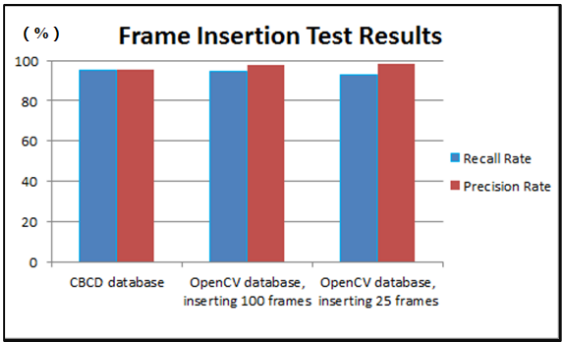

Fig. 12. Frame insertion detection results comparison with different test database

From figure 12, frame insertion detection results of three databases are similar. So we can conclude that our video frame insertion detection method achieves a well robustness with different tampered videos and different frame insertion lengths.

From the three groups of recall rates and precision rates, we can conclude that for frame insertion detection, the higher the recall rate, the lower the precious rate. This is because a fixed threshold is used among all databases, and when the recall rate is high for a database, it means the threshold is low for this insertion type, which will cause more false detections in this database, so the precision rate will increase.

4.4 Experiment on Frame Deletion Forgery Model's Detection

Test Results of Frame Deletion Database Built with OpenCV Library

From the experiment result above, the frame deletion is a little worse than frame insertion. For frame insertion, the lowest recall rate is above 92% and the lowest precision rate is above 95%, but for frame deletion, the recall rate and precision rate are both lower than 90%. It is because in frame insertion, frames are inserted from other videos, while different videos are usually recorded with different camera or different parameters, so frames from different videos have bigger optical flow inconsistency.

Table 3. Test Results with Frame Deletion Database 3

	N_c	N_m	N_f	$R_r(\%)$	$R_p(\%)$
N=100	514	86	61	85.67	89.39
N=25	507	93	90	84.50	84.92

From the results, we can find that the recall rate and precision rate of deleting 25 frames are lower than those of deleting 100 frames, which is consistent to the fact that if fewer frames are deleted, it will be harder to be detect, so the recall rate is lower when deleting fewer frames. As with the precision rate, it's different with frame insertion. For frame insertion, a fixed threshold T is used in the first round detection, and with a fixed threshold, the false detection will be reduced with fewer frames inserted, so the precision rate will increase. But for frame deletion, adaptive thresholds are used, and when fewer frames are deleted, frames around the shift point are much similar. So the adaptive threshold mechanism will drop down the threshold to guarantee the recall, but lower threshold will induce more false detection caused by fast movements such as jogging, so the precision will also increase with fewer frames deleted.

From above analysis, we can conclude that our detection scheme can achieve a relatively stable performance to different frame deletion forgery.

From experiments in 4.3 to 4.4, we can conclude that out method achieves a good performance in identifying both frame insertion forgery model and frame deletion forgery model. In addition, the proposed method is robust to different tampering tools and different inserted and deleted frame numbers.

4.5 Analyzing Detection Effect on Threshold Parameter Selection

It our experiment, we used adaptive threshold for the *T1* and *T2* in equation (6) and (8), below figures give the test results with different *T1* and *T2*.

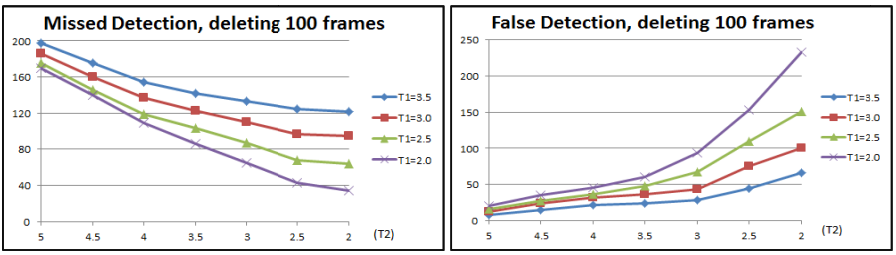

Fig. 13. Testing results of deleting 100 frames with different threshold *T1* and *T2*

From the two figures above, for a fixed *T1*, when *T2* increases, the missed detection number increases, but the false detection number decreases. This is because the threshold *T2* will filter false deletion after detecting suspicious deleting shift points. The bigger the threshold *T2* is, more false detections will be filtered, thus the false detection rate will decrease, but at the same time, bigger *T2* will increase the danger to filter true detections, causing the missed detection rate to increase.

With a fixed *T2*, the missed detection number will increase as *T1* increases, while the false detection number drops. This is because the smaller *T1* is, more frames will be detected as suspicious deletion shift points, thus it will decrease missed detection rate. Accordingly, it will increase the danger of regarding normal frames as frame deletion shift points, thus increasing the false detection number.

As is analyzed above, the missed detection number is inversely proportional to the false detection number; however, we can still achieve a balance recall rate and precision rate of frame deleting detection. We can conclude that our adaptive threshold mechanism can obtain an optimal value for *T1* and *T2* to balance the recall rate and precision rate.

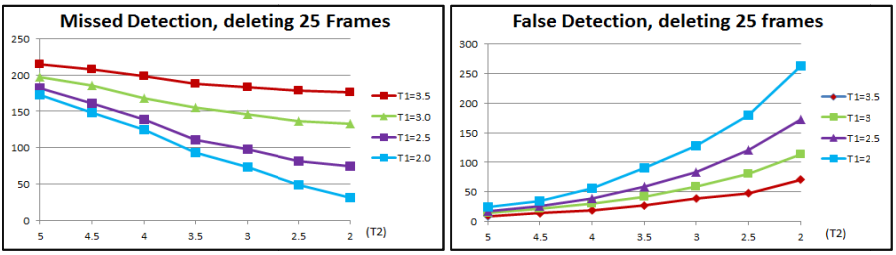

Fig. 14. Testing results of deleting 25 frames with different threshold *T1* and *T2*

From the comparison of figure 13 and 14, the influence of *T1* and *T2* while deleting 25 frames is similar with deletion 100 frames, but still there is some difference. With the same *T1* and *T2*, the missed detection number and false detection number of deleting 25 frames are both a little larger than deleting 100 frames, so the detection recall rate and precision rate of deleting 25 frames are lower than 100 frames. This result is consistent to the analysis and conclusion in 4.4.

5 Conclusion

In this paper, a novel video inter-frame forgery detection scheme based on Lucas Kanade optical flow consistency is proposed. It is based on the assumption that for adjacent frames in original videos, their optical flows are consistent, and inter-frame forgery will disturb this optical flow consistency. In this paper, a window based rough detection and binary search based precise detection is proposed for frame insertion forgery detection. And for frame deletion forgery detection, frame-to-frame mechanism and double adaptive thresholds are proposed to detect the tiny difference in optical flow, so as to detect deletion forgery. By detecting different forgery models separately according to their difference, our algorithm can not only well detect video inter-frame forgery, but also identify the forgery model. Experiments show that the recall rate reaches 95% and the precision rate reaches 98% of frame insertion forgery detection. For frame deletion forgery detection, the detection rates are a little lower than insertion forgery, but still the recall rate reaches 85% and precision rate reaches 89%. Future work will focus on detecting frame duplication and improve the recall rate and precision rate of detecting frame deletion.

Acknowledgement. We would like to thank Prof. Y. Q. Shi at NJIT in U.S.A. for the fruitful technical discussions and selfless help. The work of this paper is sponsored by the National Natural Science Foundation of China (No. 61071153, No. 61272249), the National New Century Excellent Talents Support Plan of Ministry of Education, China (No. NECT-10-0569). It is also under the Project of International Cooperation and Exchanges supported by Shanghai Committee of Science and Technology (No. 12510708500).

References

1. Rocha, A., Scheirer, W., Boult, T., et al.: Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics. ACM Computing Surveys 43(5), article number:26 (2011)
2. Zhou, Z.Y., Tang, X.H.: Integrity Authentication Scheme of Color Video Based on the Fragile Watermarking. In: International Conference on Electronics, Communications and Control (ICRCC), Ningbo, China, pp. 4354–4358 (2011)
3. De, A., Chadha, H., Gupta, S.: Detection of forgery in digital video. In: The 10th World Multi Conference on Systemics Cybernetics and Informatics, vol. V, pp. 229–233 (2006)
4. Kobayashi, M., Okabe, T., Sato, Y.: Detecting Video Forgeries Based on Noise Characteristics. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 306–317. Springer, Heidelberg (2009)
5. Liao, D.D., Yang, R., Liu, H.M., et al.: Double H.264/AVC compression detection using quantized nonzero AC coefficients. In: Conference on Media watermarking, Security, and Forensics, San Francisco, CA, vol. 7880, article number: 78800Q (2011)
6. Wang, W.H., Farid, H.: Exposing Digital Forgeries in Interlaced and De-Interlaced Video. IEEE Transactions on Information Forensics and Security 2(3), 438–449 (2007)
7. Johnson, M.K., Farid, H.: Exposing Digital Forgeries in Complex LightingEnvironments. IEEE Transaction on Information Forensics and Security 2(3), 450–461 (2007)
8. Swaminathan, A., Wu, M., Ray Liu, K.J.: Digital Image Forensics via IntrinsicFingerprint. IEEE Transaction on Information Forensics and Security 3(1), 101–117 (2008)