Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **3**, May **2018**

Computer-Aided Annotation for Video Tampering Dataset of Forensic Research

Ye Yao[1,2], Cai-Xia Yu[3,*], Li Yang[4] and Chin-Chen Chang[5]

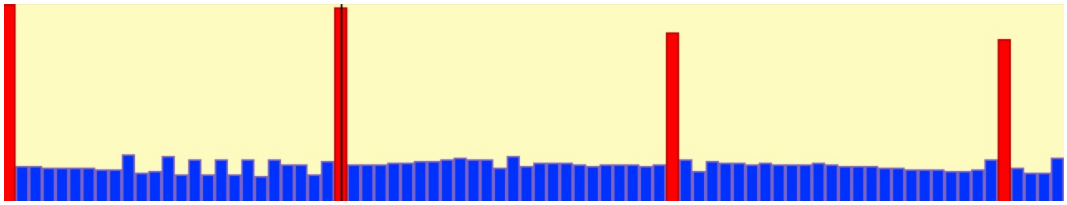
[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou, 310018, China
[2]Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China
[3]School of Management, Hangzhou Dianzi University, Hangzhou, 310018, China
[4]College of Information Engineering, China Jiliang University, Hangzhou, 310018, China
[5]Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan
*Corresponding author: yucaixia@hdu.edu.cn

Received February, 2018; revised April, 2018

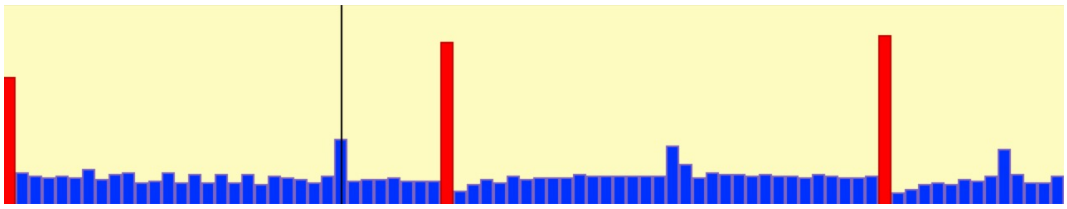
Abstract. *The annotation of video tampering dataset is a boring task that takes a lot of manpower and financial resources. At present, there is no published literature which is capable to improve the annotation efficiency of forged videos. We presented a computer-aided annotation method for video tampering dataset in this paper. This annotation method can be utilized to label the frames of forged video sequences. By means of comparing the original video frames with the forged video frames, we can locate the position and the trajectory of the forged areas of the forged video frames. Then, we select several key points on the temporal domain according to the trajectory of the forged areas, and mark the forged area of the forged frames in the key point with a mouse. Finally, we use the linear prediction algorithm based on the coordinates of the key positions in the temporal domain to generate the annotation information of forged areas in other video frames without manually labeled. If the bounding box generated by the computer-aided algorithm deviates from the actual location of the forged area, we can use the mouse to change the position of the bounding box during the preview period. This method combines the manual annotation with computer-aided annotation. It solves the problems of the inaccuracy of annotation by computer-aided as well as the low efficiency of annotation manually, and meet the needs of annotation for an enormous amount of forged videos in the research of video passive forensics.*
Keywords: Video tampering dataset, Computer-aided annotation, Bounding boxes, Temporal prediction

1. **Introduction.** Deep learning has been extensively used in the field of image and video processing, and has achieved a great success. At present, many researchers have begun to solve the problems of multimedia security [1] [2] such as steganalysis [3] and digital image tampering detection [4] in the field of multimedia security based on deep learning methods. Video tampering detection has drawn more and more attention in recent years. Some researchers have begun to use deep learning based methods to detect and locate the forged areas in the forged video sequences. However, detection and localization of video forged areas based on deep learning require a large number of labeled video samples for training. The video tampering dataset that is already available to researchers all over the world contains a small number of forged video sequences or does not provide the annotation information of forged areas. It can not meet the quantity requirement of the training data required for deep learning.

Automatic annotation for images and video sequences is a promising solution to build a large labeled dataset [5] [6]. However, the computer-aided automatic annotation [7] [8] for video tampering dataset cannot merely adopt the algorithms in the fields of computer vision and image processing. Firstly, the forged areas may be larger than the areas where the moving object is removed or inserted. Therefore, by means of comparing the difference between the moving objects in the original video frames and in the forged video frames, we can not get the location information of forged areas on the temporal domain and on the spatial domain accurately. Secondly, after the moving object in the video sequence has been forged, the video frames have to be compressed again, which is named as double compression. This will bring noise and may also change the GOP structure of the video sequence. As shown in Fig. 1, the GOP size in the original video sequence is 25, but it changes to 33 in the forged video sequence after double compression. Because of the noise introduced by the compression, subtracting the original video frames from the forged video frames cannot get the location information of the forged areas. Therefore, an algorithm has to be designed for the annotation of video tampering dataset. This algorithm should be able to eliminate the noise brought by video compression and output location information of forged areas accurately.



(a) GOP structure of original video sequence (GOP size=25).



(b) GOP structure of forged video sequence (GOP size=33).

FIGURE 1. Different GOP structure and frame size between original areas and forged areas.

A computer-aided annotation method for video tampering dataset is presented in this paper. This annotation method can be utilized to label the frames of forged video sequences. By means of comparing the original video frames with the forged video frames, we can locate the position and the trajectory of the forged areas in the forged video frames. Then, we select several key points on the temporal domain according to the trajectory of the forged areas, and mark the forged area of the forged frames in the key point with a mouse. Finally, we use the linear prediction algorithm based on the coordinates of the key positions in the temporal domain to generate the annotation information of forged areas in other video frames without manually labeled. If the bounding box generated by the computer-aided algorithm deviates from the actual location of the forged area, we can use the mouse to change the position of the bounding box during the preview period. This method combines the manual annotation with computer-aided annotation. It solves the problems of the inaccuracy of computer-aided annotation as well as the low efficiency annotation of manually and meets the need of annotation for an enormous amount of forged videos in the research of video passive forensics.

2. **Related Work.**

2.1. **Dataset.** There are several video tampering datasets for video forensics research. As far as we know, the first video tampering dataset is SULFA(Surrey University Library for Forensic Analysis) [9], which was designed and built by the forensics research group at Surrey University. SULFA contains several original and forged video sequences, which are freely available through the website of research group[10]. There are five pairs of video sequences with copy-move forgery. Each forged video includes information of the forged area, namely as annotation, in this paper.

Another tampering dataset, which was named as REWIND[11], was built by Bestagini *et al.*. This dataset is composed of 10 original video sequences and 10 forged ones. Each sequence has a resolution of 320x240 pixels and a frame rate of 30 fps. Some of the original sequences come from the SULFA tampering dataset. REWIND also contains the differences between the frames of the original sequences and the forged sequences, which serve as the annotation information.

Al-Sanjary *et al.* developed a video tampering dataset for forensic research[12]. It includes 10 video sequences with copy-move tampering. Some information such as the video length, the total number of frames and the ID number of forged frames, is provided in the ground truth, but there is no information about the tampering areas in the temporal domain.

Although SULFA and REWIND include the annotation information, the quantity of forged video sequences in these datasets is too low. It can not meet the requirement of deep learning based forensic methods. SYSU-OBJFORG (Sun Yat-sen University object forgery dataset) is the biggest video tampering dataset according to the report in [13]. It consists of 100 original video sequences and 100 forged video sequences. All video sequences are of 11 seconds, 1280×720 resolution sequences with a bitrate of 3 Mbit/s and a frame rate of 25 fps (frame per second). But there is no annotation information provided with the SYSU-OBJFORG dataset. In order to extract positive and negative samples from SYSU-OBJFORG dataset for the training of deep learning, an annotation algorithm should be presented to help to mark the forged areas in the forged video frames.

2.2. **Software and tool.** With the wide availability of user-friendly media editing software such as Premiere, After Effects and Final Cut, it becomes much easier to insert video objects to video sequences or remove video objects from video sequences. However, these pieces of software cannot export coordinate information of the forged areas in each video frame.

Ardizzone and Mazzola presented a tool for forensics researchers to create forged videos[14]. The authors also created a forged video dataset from 6 video sequences. This dataset is available at[15] with the information of bounding boxes for the modified objects. This tool can be utilized to clone video objects from a video frame to another and generate a forged video, but it can not be used to calculate the difference between the original video sequences and the forged video sequences. That is to say, we can not use this tool to create an annotation for a existing tampering dataset such as SYSU-OBJFORG.

As a result, we present a computer-aided annotation method for video tampering dataset in this paper. The proposed can be utilized to label the frames of a forged video sequence. It is a solution to the annotation of video tampering dataset which can significantly improve the accuracy and efficiency.

3. **Method.**

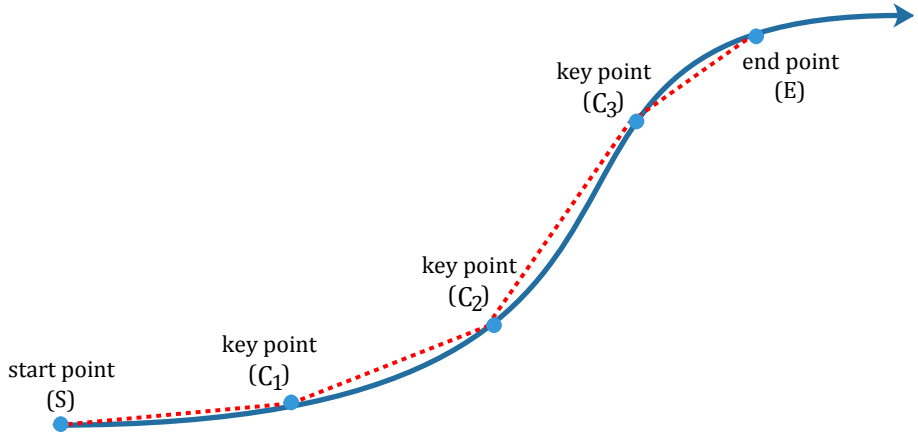


FIGURE 2. The top view for the actual trajectory of the forged areas (Solid blue line) and the manually selected annotation points (key point).

3.1. **Select the key points.** The first step in our method is to select the key point in the trajectory of the forged object. This method requires us to choose the key points manually. We are required to preview the original video sequence and the forged video sequence to obtain the rough trajectory based on our estimation. By means of comparing the differences between the original video frames and the forged video frames with our naked eyes, we can find out the forged areas in each forged video frame. If we take each forged area as a point on the temporal domain, all of the forged areas in the forged video sequence compose a trajectory of the forged object.

As shown in Fig. 2, the solid blue line is the actual trajectory of a forged object. Between the start point and the end point of the trajectory, we can select some points as key points. At each key point, there is a forged video frame. We need to mark out the forged areas on the forged video frame with a mouse. In order to make the prediction results more close to the actual trajectory, we need to select out more key points. On the other hand, the more key points we select, the more time we need to spend on. So it has to be a trade-off between prediction accuracy and manpower cost. We have given an example of the selection of the key points in Fig. 2. In which, we select a point as the key point when the forged object changes its trajectory in an unexpected speed or to an uncertain direction. It can greatly reduce the manpower cost for annotation as well as keep the prediction accuracy in a good condition.

Algorithm 1 Prediction of other points between start point S and end point E

Input: start point S, key points $\{C_1, C_2, ..., C_n\}$, end point E
Output: prediction results $C_k, k \in [S, E]$
  start key point $m \leftarrow S$
  for idx in $\{C_1, C_2, ..., C_n, E\}$ **do**
    end key point $n \leftarrow idx$
    predict all of the other points C_k between $[m, n]$
    start key point $m \leftarrow idx$
  end for

3.2. **Predict other points.** After we finish the selection of the key points, we need to predict the tampering area in other forged frames between any two of the key points. As shown in Algorithm 1 and Fig.2, we divide the trajectory into several segments. For each

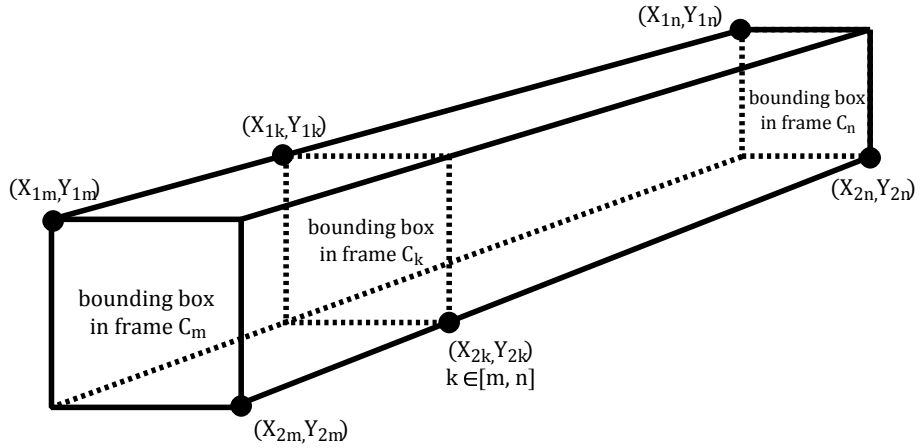


FIGURE 3. Prediction of bounding box in frame C_k based on bounding boxes in frames C_m and frame C_n.

segment, it is starting with a key point and ending with a consecutive key point. We apply our prediction algorithm to each segment one by one.

Between the key points m and n, we use Formula (1) to get the coordinate of the bounding box in the forged frame C_k. Note that according to Fig.3, the frame C_k is between the frame C_m and the frame C_n. Therefore, the prediction results (X_{1k}, Y_{1k}) and (X_{2k}, Y_{2k}) can be regarded as the coordinate of the bounding box in frame C_k, which represent the forged area in the forged video frame C_k as follows:

$$\begin{aligned} X_{1k} &= X_{1m} + (X_{1n} - X_{1m}) \times (k-m)/(n-m) \\ Y_{1k} &= Y_{1m} + (Y_{1n} - Y_{1m}) \times (k-m)/(n-m) \\ X_{2k} &= X_{2m} + (X_{2n} - X_{2m}) \times (k-m)/(n-m) \\ Y_{2k} &= Y_{2m} + (Y_{2n} - Y_{2m}) \times (k-m)/(n-m) \end{aligned} , \tag{1}$$

where m, n, k are the frame numbers in the forged video sequence, (X_{1m}, Y_{1m}) and (X_{2m}, Y_{2m}) are the left top coordinate and the right bottom coordinate of the bounding box in frame C_m, where (X_{1n}, Y_{1n}) and (X_{2n}, Y_{2n}) are the left top coordinate and the right bottom coordinate of the bounding box in frame C_n.

4. **Results.**

4.1. **Tool.** The source code corresponding to the annotation method has been freely available through github [16]. We implement the algorithm with python and OpenCV. The graphical user interface is shown in Fig.4. There are four windows. Three of them display the original video sequence, the forged video sequence and the differences between the original and the forged frames respectively. The last one displays the information of the bounding box outputted by the annotation tool.

To let users more easily to find out the forged area, we show the difference between the original and the forged frames on the third window. The difference is the result of subtraction between the two video frames. Based on the images shown in these windows, the user can mark out all the forged areas in the forged video sequence with a mouse.

We also add some codes to support the modification of the bounding box. The user can delete one bounding box by clicking on the bounding box through the right button of the mouse. After removing this bounding box, a new bounding box can be drawn on the correct area through the left button of the mouse. All of the operations are responded

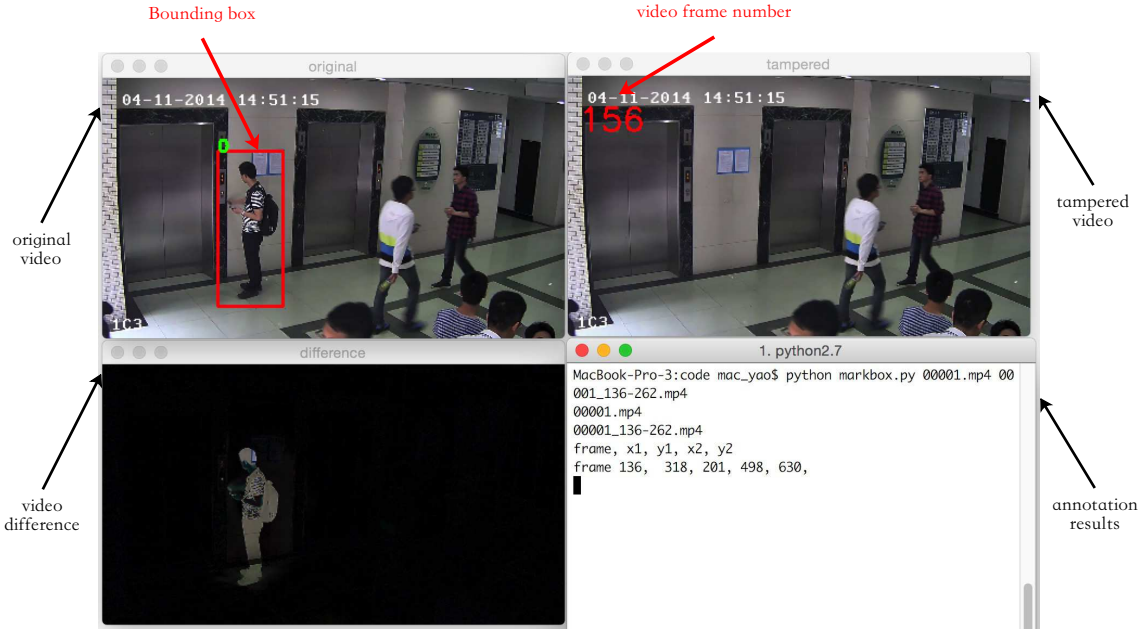


FIGURE 4. Graphical user interface.

to the first window, namely, to be processed through the window of the original video sequence.

There are several short keys for users to have a faster and smoother experience. For example, users can save the bounding box to the annotation list by pressing the "S" key. Users also can go back to the last video frame by pressing the "R" key, or display the next video frame by pressing any other keys.

TABLE 1. Statistics of the annotation results for the SYSU-OBJFORG

Item	Number	Ratio
Tampered video sequences	100	
Original video sequences	100	
Total tampered frames (F_1)	11074	$F_1/F_2 =$
Total original frames (F_2)	47368	23.38%
Bounding box marked (B_1)	586	$B_1/B_2 =$
Bounding box total (B_2)	11837	4.95%
Total time of marking (in Hours)	3.5 hours	

4.2. **Evaluation.** We test our algorithm and tool on the SYSU-OBJFORG dataset, which is the biggest video tampering dataset till now. The annotation result is also available at github [16] to researchers who want to train their tampering detection models. The statistics of the annotation results of the SYSU-OBJFORG dataset are shown in Table 1. It only takes 3.5 hours to mark all of the 100 pairs video sequences. There is no more than 5% of the bounding box that needs to be marked manually. That is to say, all of the other bounding boxes are generated by the computer-aided annotation algorithm. This algorithm is important for the researchers to annotate the forged video sequence quickly.

5. **Conclusions.** In this paper, a computer-aided annotation method for video tampering dataset has been presented. This method is developed for the purpose of marking the forged areas of the forged video sequence. With the aid of computer algorithms, we can increase the efficiency of annotation greatly. For the future work, we will continue to develop our method to support pixel-level marking. It will have the capability to annotate the forged area more precisely. The availability of such tools and algorithms will be extremely useful for forensics researchers. It can provide them with better training samples and improve the detection results of the algorithms based on deep learning.

Acknowledgment. This work is partially supported by the Zhejiang Provincial Natural Science Foundation (LQ16G030010), partially supported by the Public Technology Application Research Project of Zhejiang Province (2017C33146), partially supported by Education Department of Zhejiang Province (Y201534025).

REFERENCES

[1] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, Vision of the unseen: current trends and challenges in digital image and video forensics, *ACM Computing Surveys*, vol.43, no.4, pp.26–40, 2011.
[2] M. C. Stamm, M. Wu, and K. R. Liu, Information forensics: an overview of the first decade, *IEEE Access*, vol.1, pp.167–200, 2013.
[3] G. Xu, H. Z. Wu, and Y. Q. Shi, Structural design of convolutional neural networks for steganalysis, *IEEE Signal Processing Letters*, vol.23, no.5, pp.708–712, 2016.
[4] M. A. Qureshi and M. Deriche, A bibliography of pixel-based blind image forgery detection techniques, *Signal Processing: Image Communication*, vol.39, part A, pp.46–74, 2015.
[5] D. Tian, Extended Probabilistic Latent Semantic Analysis for Automatic Image Annotation, *Journal of Information Hiding and Multimedia Signal Processing*, Vol.8, No.4, pp.903–915, July 2017.
[6] D. Tian, Exploiting Semantic Context Relationships for Automatic Image Annotation, *Journal of Information Hiding and Multimedia Signal Processing*, Vol.8, No.6, pp.1203-1217, November 2017.
[7] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, pp.7464–7473, July 2017.
[8] C. Cuevas, E. M. Yáñez, and N. García, Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA, *Computer Vision and Image Understanding*, LNCS 9280, vol.152, pp.103–117, 2016.
[9] G. Qadir, S. Yahaya, and A. T. S. Ho, Surrey university library for forensic analysis (sulfa) of video content, in *IET Conference on Image Processing (IPR 2012)*, pp. 1–6, July 2012.
[10] (2012) Sulfa dataset. [Online]. Available: *http://sulfa.cs.surrey.ac.uk/.*
[11] (2013) Rewind dataset. [Online]. Available: *http://sulfa.cs.surrey.ac.uk/forged_1.php.*
[12] O. I. Al-Sanjary, A. A. Ahmed, and G. Sulong, Development of a video tampering dataset for forensic investigation, *Forensic Science International*, vol.266, pp.565–572, 2016.
[13] S. Chen, S. Tan, B. Li, and J. Huang, Automatic detection of object-based forgery in advanced video, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.26, no.11, pp.2138–2151, 2016.
[14] E. Ardizzone and G. Mazzola, A tool to support the creation of datasets of tampered videos, in *Image Analysis and Processing, ICIAP 2015*, Springer International Publishing, pp.665–675, 2015.
[15] (2015). [Online]. Available: *http://www.diid.unipa.it/cvip.*
[16] (2018). [Online]. Available: *http://github.com/yeffreyyao/MarkTamperedRect.*