Journal of XXXXX ©2009 ISSN xxxx-yyyy
Ubiquitous International Volume **1**, Number **1**, April **2009**

Computer-Aided Annotation for Video Tampering Dataset of Forensic Research

Ye Yao, X. XXX and X. XXX

School of Cyberspace, Hangzhou Dianzi University
No.2 Street, Xiasha Higher Education Zone, Hangzhou, 310018, China
yyaoprivate@gmail.com

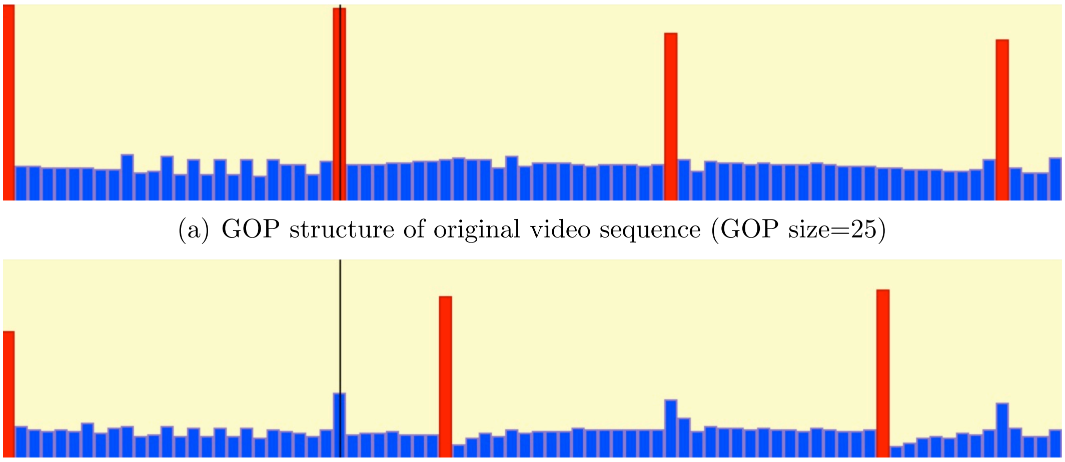
Received February 2018; revised XXX 2018
(Communicated by xxxx)

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. A computer-aided annotation method for video tampering dataset is presented in this paper. This annotation method can be used 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 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 temporal domain to generate the annotation information of forged areas in other video frames which 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 modify the location of the bounding box during the preview period. This method combines the manually 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 by manually, and meets 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 widely 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, the 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 for 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 that required for deep learning.

The computer-aided annotation [5] [6] for video tampering dataset cannot simply 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 object of the original video frames and the forged video frames, we can not get the location information of forged areas on the temporal domain and spatial domain accurately. Secondly, after the moving object in the video sequence has been forged, the video frames need 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 needs 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 the 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 and forged

A computer-aided annotation method for video tampering dataset is presented in this paper. This annotation method can be used 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 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 temporal domain to generate the annotation information of forged areas in other video frames which 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 modify the location of the bounding box during the preview period. This method combines the manually 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 by manually. It can meet the needs of annotation for an enormous amount of forged videos in the research of video passive forensics.

2. **Related Work.**

2.1. **Datasets.** There are several video tampering dataset for video forensics research. As far as we know, the first video tampering dataset is SULFA(Surrey University Library for Forensic Analysis) [7], which was designed and built by the forensics research group at Surrey University. SULFA contains several original and several forged video sequences, which is freely available through the website of research group[8]. There are 5 pairs of video sequences with copy-move forgery. Each forged video includes information of the forged area namely as annotation in this paper.

REWIND tampering dataset[9] 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 for each frame of the original sequences and the forged sequences, which serves as the annotation information.

Al-Sanjary *et al.* developed a video tampering dataset for forensic research[10]. 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 are provided in the ground truth, but there is no information about the tampering area in the temporal domain.

Although SULFA and REWIND include annotation information, the quantity of forged video sequences in these datasets is too small. It can not meet the requirement of deep learning based forensic method. SYSU-OBJFORG is the largest object-based forged video dataset according to the report in [11]. It consists of 100 pristine video sequences and 100 forged video sequences. All video sequences are of 11 seconds, 1280×720 H.264/MPEG-4 encoded sequences with a bitrate of 3 Mbit/s and a frame rate of 25. But there is no annotation information provided with the SYSU-OBJFORG dataset. In order to extract positive samples 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 or remove video objects to or from video sequences. However, These softwares cannot export coordinate information of the forged area in each video frame.

Ardizzone *et al.* present a tool for forensics researcher to create forged videos[12]. The author also created a forged video dataset from 6 video sequences. This dataset is available at[13] with the information of bounding boxes for the modified objects. This tool has the capability to clone video objects from a video frame to another video frame and generate a forged video, but this tool can not be used to calculate the difference between an original video sequence and a forged video sequence. That is to say, we can not use this tool to create an annotation for a existed tampering dataset such as SYSU-OBJFORG.

As a result, we present a computer-aided annotation method for video tampering dataset in this paper. This annotation method can be used 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.**

3.1. **Select the key point.**

The first step in our method is to select the key point in the trajectory of the forged object. This method requires the operator to select the key points manually. We need 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 difference between the original video frame and the forged video frame with our naked eyes, we can find out the forged area in each forged video frame. If we take each forged

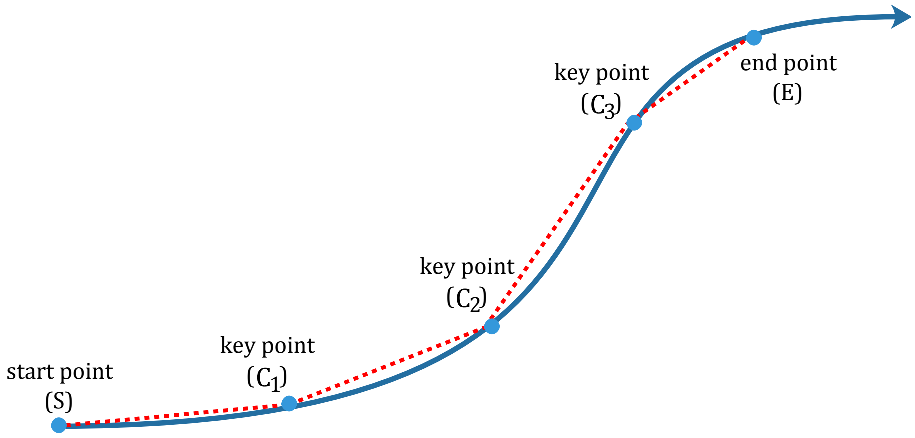


Figure 2. The top view for the actual trajectory of the forged areas (Solid blue line) and the manually annotation point (key point).

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 on the trajectory, we can select some points as the 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 let the prediction results more closer 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 we have to be a trade-off between the prediction accuracy and the manpower cost. We have given an example of the selection of the key point in Fig. 2.

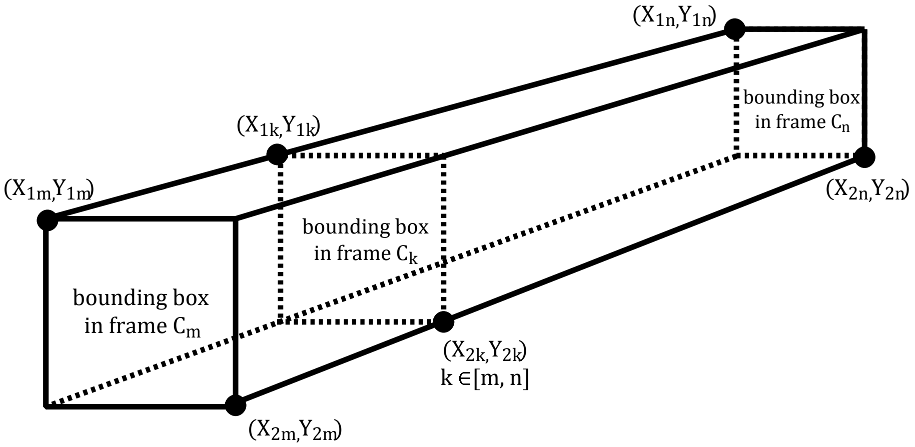


Figure 3. Predict bounding-box in frame k based on bounding boxes in frame m and frame n.

3.2. **Predict other points.** After we finished the selection of the key point, 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 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 point m and n, we use Formula (1) to get the coordinate of bounding box in the forged frame C_k. Note that according to Fig.3, the frame C_k is between the

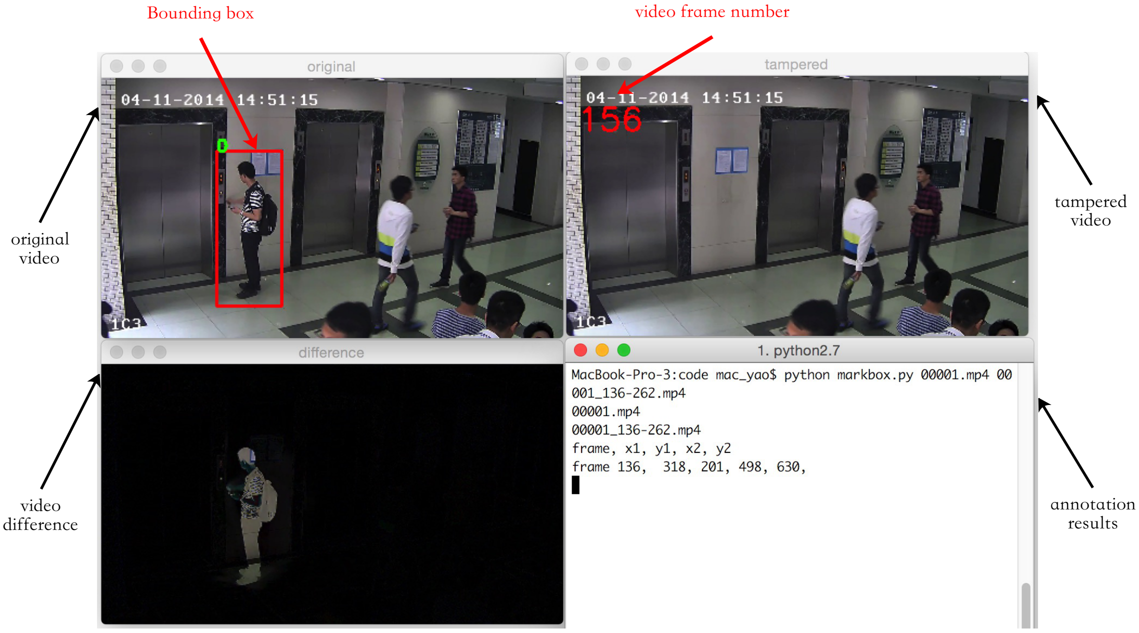
Algorithm 1 Predict other points between start point S and end point E

```
Input: start point $S$, key points $\{C_1, C_2, ..., C_n\}$, end point $E$
Output: prediction result $C_k, k \in [S, E]$
  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 point $C_k$ between $[m, n]$
    start key point $m \leftarrow idx$
  end for
```

frame C_m and the frame C_n. Therefore, the prediction resulting (X_{1k}, Y_{1k}) and (X_{2k}, Y_{2k}) can be regarded as the coordinate of the bounding box in frame C_k, which represents the forged area in the forged video frame C_k.

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

where m, n, k are the frame number 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, (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.



```
MacBook-Pro-3:code mac_yao$ python markbox.py 00001.mp4 00
001_136-262.mp4
00001.mp4
00001_136-262.mp4
frame, x1, y1, x2, y2
frame 136,  318, 201, 498, 630,
```

FIGURE 4. Graphical user interface.

4. Results.

4.1. **Tool.** The source code corresponding to the annotation method has been freely available through github [14]. 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 frame difference between the original and the forged respectively. The last one shows the information of the bounding box outputting by the annotation tool.

In order to let users more easily to find out the forged area, we show the frame difference between the original and the forged 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 area in the forged video sequence with a mouse.

We also add some codes to support the modification of bounding box. The user can delete one bounding box by means of clicking on the bounding box through the right button of the mouse. After deleting 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 on the first window, namely, are processed through the window of the original video sequence.

There are some 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. The 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 result

TOBEADD		TXT TODO	TXT TODO
p_1	p_2	quality (in dB)	(in %)
0	0	32.53	100
0.05	0.05	30.69	97.56
0.1	0.1	28.90	95.14
0.25	0.25	24.59	88.18
0.3	0.3	23.14	86.04
0.4	0.4	21.52	81.46
0.5	0.5	19.98	76.25
0	1	26.18	73.34
1	0	26.09	80.55

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 [14] to researchers who want to train their tampering detection models. The statistics of the annotation on the SYSU-OBJFORG dataset is shown in Table 1.

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 area in the forged video sequence. With the aid of computer algorithms, we can improve 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 deep learning based algorithms.

Acknowledgment. This work is partially supported by the Public Technology Application Research Project of Zhejiang Province under Grant 2017C33146, partially supported by the Humanities and Social Sciences Foundation of Ministry of Education of China under Grant 17YJC870021. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

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, May 2016.

[4] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication*, vol. 39, pp. 46–74, 2015.

[5] 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)*, vol. 00, July 2017, pp. 7464–7473. [Online]. Available: doi.ieeecomputersociety.org/10.1109/CVPR.2017.789

[6] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: Lasiesta," *Computer Vision and Image Understanding*, vol. 152, pp. 103 – 117, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314216301138

[7] 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)*, July 2012, pp. 1–6.

[8] (2012) Sulfa dataset. [Online]. Available: http://sulfa.cs.surrey.ac.uk/

[9] (2013) Rewind dataset. [Online]. Available: http://sulfa.cs.surrey.ac.uk/

[10] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0379073816303176

[11] 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.

[12] E. Ardizzone and G. Mazzola, "A tool to support the creation of datasets of tampered videos," in *Image Analysis and Processing — ICIAP 2015*, V. Murino and E. Puppo, Eds. Cham: Springer International Publishing, 2015, pp. 665–675.

[13] (2015). [Online]. Available: http://www.diid.unipa.it/cvip

[14] (2018). [Online]. Available: http://github.com/yeffreyyao/MarkTamperedRect