VIDEO PICTURE-IN-PICTURE DETECTION USING SPATIO-TEMPORAL SLICING

Mengren Qian, Luntian Mou, Jia Li, Yonghong Tian[+]

National Engineering Lab for Video Technology, School of EE & CS, Peking University

ABSTRACT

For video copy detection and near-duplicate retrieval applications, picture-in-picture (PiP) is one of widely-used but especially difficult transformations to be detected. Traditionally, PiPs in a video are detected by extracting edges within key frames sampled from the video. However, without taking the temporal continuity between frames into account, the performance of these frame-based methods is not that promising. In this paper, we propose a new video PiP detection method by introducing spatio-temporal slicing (STS) to establish the corresponding edge surface probability measurement. An optimization algorithm is then designed to refine vertical and horizontal edge lines by filtering noisy edges. This PiP detection method can be used to improve the performance of video copy detection particularly in the case of the most challenging PiP transformation. The experimental results on the TRECVID-CCD 2010 dataset demonstrate the effectiveness and efficiency of the proposed method.

Index Terms—Video, Picture-in-picture, Edge detection, Spatio-temporal slicing (STS), Content based video copy detection

1. INTRODUCTION

Picture-in-picture (PiP) is a kind of visual transformation which allows one or more pictures to be resized and inserted as foreground pictures into a background picture. Usually, PiPs in a video enable people to watch two or more videos at the same time. Due to simplicity in implementation, video PiP is configured in most TV sets to benefit people, although it is also exploited by the advertisers to insert advertisements or to insert logos on unauthorized copies so as to elude copyright lawsuits as well.

Since two or more videos probably with completely different content are combined into one PiP video, it is challenging to analyze or retrieve such a video in a traditional way using global or local features [8], [10], [12], [16]. On the one hand, as foreground videos account for only a small portion of the whole video frame, it is difficult to use the global features which take the whole video frame into account to retrieve or analyze these foreground videos. On the other hand, the retrieving or analyzing of the background video could be interfered by the change of global or local features caused by the foreground videos. This problem may cast the influence on most of the video analyzing or retrieving systems such as content based video copy detection, content based video retrieval, and video classification. Fig. 1 shows how content based video copy detection (CCD) systems can be misleading using the traditional Bag-of-Words (BoW) approaches [13]. While the query video frame has 11 matching pairs with the actual copied video frame (Fig. 1(b)), it has 30 matching pairs with an irrelevant video frame caused by the interference of background video (Fig. 1(a)).

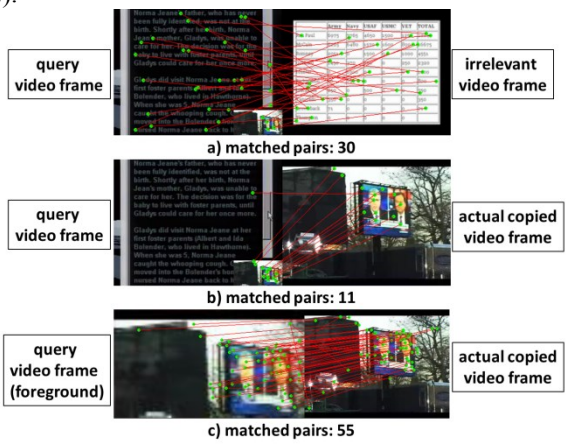

Fig. 1. Sample of misleading CCD systems using BoW

In existing work, two different kinds of solutions are practiced to handle the PiP issue. In [8], spatial verification techniques are used as post-processing to efficiently refine the basic BoW approaches. Despite quite efficient, the accuracy of this refinement is still not satisfactory especially when retrieved from large datasets. The other solution, seemly the most effective, is to first separate the foreground videos and the background video and then carry out the retrieving or analyzing to the foreground videos and background video respectively [1]-[4], [6], [11]. For example, with copy detection being performed on the foreground video alone, 55 matching pairs can be found between the query video frame and the actual copied video frame (Fig. 1(c)). This overtakes the irrelevant video frame, consequently reducing the mismatching.

[+]: This work was supported in part by grants from the Chinese National Natural Science Foundation under contract No. 61370113 and No. 61035001. Contact the authors via yhtian@pku.edu.cn.

Unfortunately, while it is easy to produce a PiP video, it is very difficult to detect PiPs from a video, namely, to automatically split the video into foreground videos and a background video. The traditional methods to detect PiP regions are mainly done by detecting the edges between the foreground videos and the background video. For the sake of efficiency, the PiP region detection is often simplified into detecting the edges within key frames sampled from the video. However, the sampling process incurs loss of temporal continuity of the video to some extent, consequently decreasing the detection performance.

To address this issue, we propose a new video PiP detection method by utilizing spatio-temporal slicing (STS) to establish the corresponding edge surface probability measurement. This novel method takes the temporal continuity into account, thus ensuring more effective edge detection between foreground videos and the background video. To reduce the time complexity, a refinement algorithm for vertical and horizontal edge lines is designed to filter noisy edges in the detection process. This PiP detection method can be used to improve the performance of video copy detection particularly in the case of the most challenging PiP transformation. The experimental results on the TRECVID-CCD 2010 dataset demonstrate the effectiveness and efficiency of the proposed method.

The remainder of this paper is organized as follows. Sec. 2 reviews related works. Sec. 3 details the proposed method. Experiments and performance evaluation are described in Sec. 4. And the work is concluded in Sec. 5.

2. RELATED WORKS

Related research works on video PiP detection are mainly focused on detecting frame level PiPs [1]-[4], [6], [11]. And image PiP detection can be modeled as detection in an edge image for likely rectangles, which might be the edges between foreground and background images. Fig. 2(b) is the edge image of Fig. 2(a), as the rectangle with the most probability being marked red.

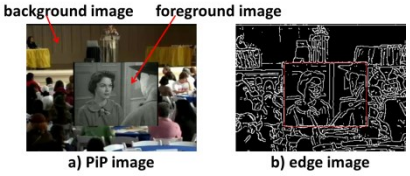

Fig. 2. Sample of image PiP detection

As edge line detection is concerned, Hough transform [1], Sobel [2], Laplacian [3], and Canny [4] edge detectors are used to detect either horizontal or vertical edge lines within the image. In [1], only horizontal lines are detected. If one horizontal line is consistent over the whole video, it is taken as a persistent horizontal line. The PiP region can be located with a pair of persistent horizontal lines which may constitute two parallel lines of a rectangle. In [2]-[4], [6], [11], both vertical and horizontal edge lines are detected, and the most likely rectangle with two vertical lines and two horizontal lines is considered as a PiP region. In [3], [4], the edges are detected in every key frame, and thus a mean-edge frame is obtained for PiP localization.

Fusion of frame level PiP results is applied to form video PiP results more precisely. In [6], after the voting and summation of rectangles detected from every key frame, the best rectangle will be returned as the video PiP region.

However, there are some weaknesses with these existing methods. For one thing, if every frame in the video is processed, the time consumption is unacceptable. For another, if some key frames are sampled to simplify the detection, time continuity information will be partially lost and the PiP region cannot be detected and located precisely. The proposed method in the following section is to eliminate both folds of weaknesses.

3. THE PROPOSED METHOD

The overview of our proposed method is shown in Fig. 3. Sampled key frames and spatio-temporal slices are first extracted from the video clip. Vertical and horizontal edge lines are then detected and refined from the key frames and spatio-temporal slices. With the probabilities of four surfaces being measured by the proposed spatio-temporal slicing (STS) based measurement, a PiP region can be measured and determined.

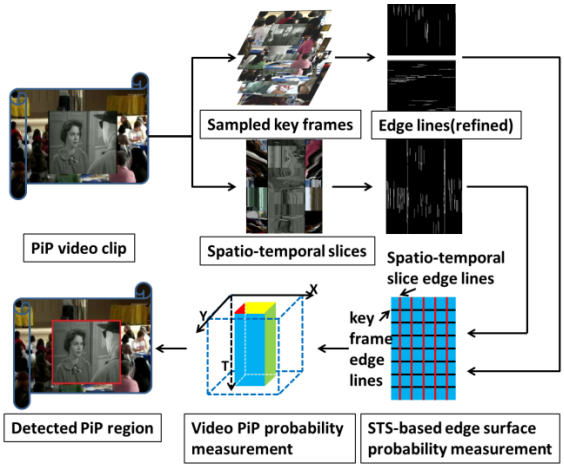

Fig. 3. Overview of the proposed method

3.1. Model for image PiP detection

The probability of a rectangle region being an image PiP region can be established by fusing the probabilities of four lines which are supposed to constitute four edges of a rectangle in the edge image:

$$P_{\text{imagePiP}} = F(P_{\text{left}}, P_{\text{right}}, P_{\text{top}}, P_{\text{bottom}}), \quad (1)$$

where P_{left}, P_{right}, P_{top} and P_{bottom} denote the probabilities of left, right, top and bottom edge lines of the rectangle respectively. Function F fuses the four probabilities of edge lines and generates a final probability of the rectangle. The return value of function $F(a,b,c,d)$ will be $\frac{(a+b+c+d)}{4}$ if either of the following conditions is satisfied:

1. If all of a, b, c and d are larger than T_{mid};
2. If three of a, b, c and d are larger than T_{high} and the other one is larger than T_{low}.

Otherwise, the return value would be 0. In our work, the values of the three thresholds T_{high}, T_{mid} and T_{low} are experimentally set to 0.95, 0.85 and 0.75 respectively.

Assuming that the left, right, top, and bottom positions of the rectangle are $x1$, $x2$, $y1$ and $y2$ respectively, with constraints of $x1 < x2$ and $y1 < y2$, the probability of left edge line is defined as follows:

$$P_{\text{left}} = \left(\sum_{i=y1}^{y2} E(x1,i)\right)/(y2 - y1 + 1), \quad (2)$$

where $E(x,y)$ represents whether point (x,y) is an edge point or not (set to 1 if TRUE, and 0 otherwise). Naturally, P_{top}, P_{right} and P_{bottom} can be similarly defined.

3.2. Model for video PiP detection

Since a video can be viewed as a series of images, and the PiP region through a video clip is almost persistent, a time dimension of T can be introduced to extend the model of image PiP into the model of video PiP, as shown in Fig. 3. The fundamental goal of video PiP detection, therefore, extends from detecting a rectangle represented by four edge lines between foreground and background images to detecting cubes consisting of four edge surfaces between foreground and background videos. The four edge surfaces are filled with blue, green, yellow, and red respectively in Fig. 4.

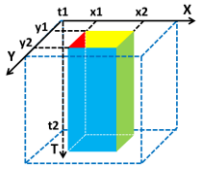

Fig. 4. Model for video PiP detection

Similarly, the probability of the PiP position in the video can then be formulated as follows:

$$P_{\text{videoPiP}} = F'(P'_{\text{left}}, P'_{\text{right}}, P'_{\text{top}}, P'_{\text{bottom}}), \quad (3)$$

where P'_{left} stands for the probability of the left edge surface in the video cube (marked red in Fig. 4.) and P'_{right} (green), P'_{top} (yellow) and P'_{bottom} (blue) have similar meanings. Similar to F, F' is used to fuse the probabilities of the four edge surfaces to form the probability of video PiP.

Let $(x1, x2, y1, y2, t1, t2)$ denote such edge surfaces (as Fig. 3 shows), the following formulations can be deduced:

$$P'_{\text{left}} = \frac{\sum_{j=t1}^{t2}\sum_{i=y1}^{y2} E'(x1,i,j)}{(y2-y1+1)\times(t2-t1+1)}, \quad (4)$$

where $E'(x,y,t)$ stands for whether point (x, y) is an edge point or not of the frame at time t (set to 1 if TRUE, and 0 otherwise). Also, P'_{right}, P'_{top} and P'_{bottom} can be defined in a similar way.

3.3. STS-based edge surface probability measurement

Although the probability evaluation of four surfaces in (4) is precise, it is time consuming to extract edges from all frames of the video. Therefore, certain techniques are used to accelerate video PiP detection. In [3], [4], edges of sampled key frames are detected, and each key frame accounts for one horizontal line in an edge surface (shown in Fig. 5). By measuring the probabilities of these lines, the probability of the edge surface can be obtained. However, temporal continuity of the video cannot be preserved by these separated horizontal lines alone, thus leading to inaccurate measurement of the probability of the edge surface especially in videos with frequent scene switching.

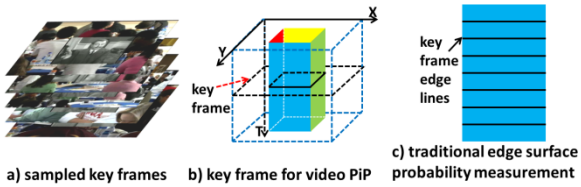

Fig. 5. Key frame used for video PiP detection

To address the above issue, we introduce spatio-temporal slicing (STS) [5] in edge surface probability measurement, and thus both horizontal and vertical lines are taken into consideration. While the horizontal lines are taken from the edge image of key frames, the vertical lines are taken from the edge image of spatio-temporal slices.

As a video can be viewed as a series of images with spatial dimension (X,Y) and temporal dimensionT, the spatio-temporal slices can be viewed as a set of images with dimension (X, T) or (Y, T). For example, the same line of a same y (marked red in Fig. 6(a)) is chosen from all the video frames, and thus a spatio-temporal slice is correspondingly produced with dimension (X, T) of the whole video (Fig. 6(b)).

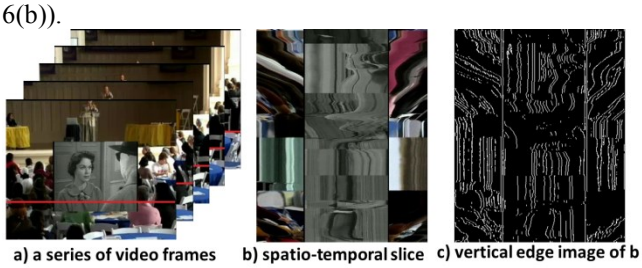

Fig. 6. Sample of spatio-temporal slice

Fig. 7 shows how vertical lines can be obtained from the spatio-temporal slices. These kinds of vertical lines, located in the edge surfaces between foreground and background videos, are then used together with the horizontal lines from key frames to measure the probability of the edge surface.

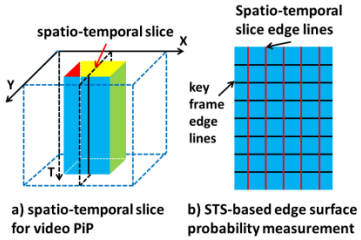

Fig. 7. Spatio-temporal slice used for video PiP detection

Given both the vertical and horizontal lines, the probability of the edge surface can be measured in a new formula:

$$P'_{\text{surf}} = G(AVG(P_{\text{vLine}}), AVG(P_{\text{hLine}})), \quad (5)$$

where $AVG(P_{\text{vLine}})$ denotes the average probability of vertical lines and $AVG(P_{\text{hLine}})$ for horizontal lines. The probability of a single vertical line can be calculated as follows:

$$P_{\text{vLine}} = (\sum_{i=t1}^{t2} E'(x, y, i))/(t2 - t1 + 1), \quad (6)$$

where $E'(x, y, t)$ has a same definition here as (4). The probability of a horizontal line can be similarly calculated.

Function G is used here to fuse the probability of vertical and horizontal lines where T_{line} is used to avoid the situation that any of the two probability values is too small:

$$G(a, b) = \begin{cases} 0 & \text{, if } a < T_{\text{line}} \text{ or } b < T_{\text{line}} \\ \frac{(a+b)}{2} & \text{, otherwise} \end{cases}. \quad (7)$$

3.4. Edge line refinement

Canny edge detector [7], which proves to be the best among traditional edge detectors, is modified to meet our specific needs here: the vertical and horizontal edge images are extracted respectively, which relieves the computation of gradient directions by reducing the original eight directions to the current two directions. The vertical and horizontal Canny edges are shown in Fig. 8(c) and Fig. 8(d) respectively, with the original Canny edges shown in Fig. 8(b). Here a rather low threshold for Canny edge selection is chosen for the sake of recall rate, raising the probability of actual edges and noise edges as well.

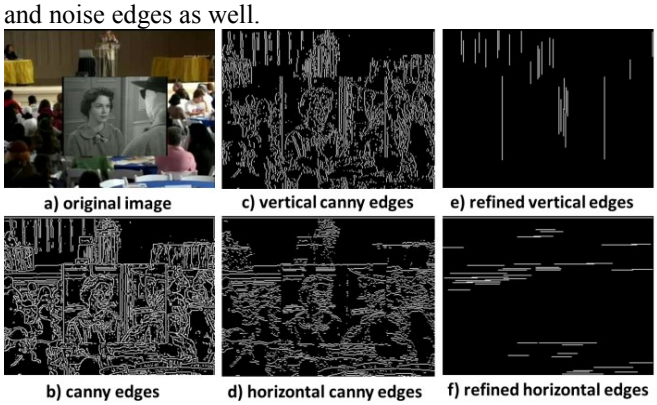

Fig. 8. Refined edge lines extraction

Then, an optimization algorithm in edge image extraction is designed to combine short neighboring edge segments while eliminating short isolating ones, which reduces the noisy edges so that less edge lines will be chosen for further procedures. The following is the refinement algorithm for vertical edges:

Input: original edge image with width w and height h
Output: refined edge image with width w and height h
for $x = 0$ to w do
1.**for** $0 \le y1 < y2 < y3 < y4 \le h$
if $(P_{Line}(x, y1, x, y2) > p \; and P_{Line}(x, y3, x, y4) > p)$
if $(P_{Line}(x, y1, x, y4) > p)$
Merge line$(x, y1, x, y2)$ with $(x, y3, x, y4)$
2.Remove line segments shorter than l after merging.

$P_{Line}(x1, y1, x2, y2)$ here stands for the probability of an edge line from $(x1, y1)$ to $(x2, y2)$, which should be evaluated only in the original edge image. As for horizontal edges, the refinement algorithm is similar. And the parameters l and p denote the minimum length of the edge line and the minimum probability of a line to be kept respectively. Experimentally, we set $p = 0.7$ and $l = 20$ according to the dataset. The refined horizontal and vertical edges are shown in Fig. 8(e) and Fig. 8(f).

Our refinement algorithm has something in common with [11], but also has some differences. Firstly, the edge points are not ranked, therefore all possible edges are taken into consideration. Secondly, we combine short edge segments by the probability of long edge segment if combined, rather than a fixed way to combine adjacent edge segments, which may face some problems in the context like dotted lines.

3.5. Content based video PiP copy detection system

Fig. 9 shows the diagram of our video PiP copy detection system, which is extended from [13], [15] and [16]. PiP is detected and localized (if detected) in every query video. After that, SIFT [14] BoWs are extracted from foreground frames and original frames respectively. Inverted indexes are used to speed up searching the most similar reference frames compared with query frames. These similar reference frames will eventually lead to the decision of whether the query video is a copied video and which reference video it copied. The performance of our video PiP copy detection system will be evaluated in Sec. 4.2.

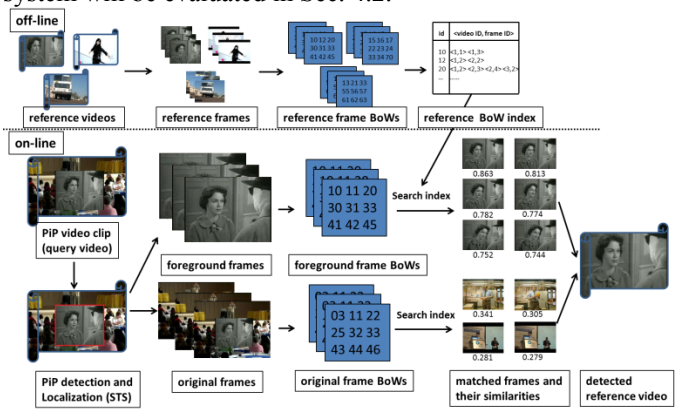

Fig. 9. Diagram of our video PiP copy detection system

4. EXPERIMENT

In this section, we present our experiments on two different aspects: video PiP localization, so as to compare our proposed PiP localization method with others', and content based video PiP detection, so as to validate the performance of our proposed video PiP detection system.

4.1. Experiment on PiP localization using STS

4.1.1. Dataset and evaluation

The query videos of TRECVID 2010 content based copy detection (CCD) task [9] is used as the dataset for PiP localization performance evaluation. Totally, 392 videos are chosen, which contain 213 PiP regions in 196 videos with possible one video containing two or more PiP videos. Correspondingly, a ground truth of this dataset is acquired by manually marking those PiP positions of all the videos.

Once a PiP position is detected from a video, it is checked against the ground truth. If one PiP detected matches anyone in the ground truth, it is a true positive; otherwise it is a false positive. The detected PiP region $(x1', y1', x2', y2')$ is determined as a match with the ground truth PiP region $(x1, y1, x2, y2)$ if and only if the following inequalities are satisfied:

$$\begin{cases} x1 - d < x1' < x1 + d \\ x2 - d < x2' < x2 + d \\ y1 - d < y1' < y1 + d \\ y2 - d < y2' < y2 + d \end{cases}, \quad (8)$$

where d is the maximum deviation allowed for the detected PiP region (d = 4 in our work). If one ground truth item is not matched with any PiP detected, it is a miss.

We have implemented the methods in [4] and [6] for comparison. In [6], PiPs are detected in each frame, and the PiP region which appears most times through the whole video is returned as the video PiP region. In contrast, the method in [4] obtains a mean-edge frame first from the sampled key frames and then calculates the video PiP region from this mean-edge frame (practically same as Fig. 5).

4.1.2. Result and analysis

The experiment is carried out under different settings shown in Table. 1 and under a Windows® server equipped with 12 core 2.0GHz CPU (E5-2620) and 32GB memory.

Table 1. Experiment settings of PiP localization

Method Name	[6]	[4]	10 frames*	20 frames*	All frames*	Proposed
Key frames	All	10	10	20	All	10
Spatio-temporal slices	/	/	/	/	/	20(X,T) 20(Y,T)
Line refinement	No	No	Yes	Yes	Yes	Yes
Proc time(s)	15	5.5	0.28	0.39	2.61	0.38

*These methods are based on [4] but with edge line refinement and different number of sampled key frames.

The recall-precision curves of different methods are shown in Fig. 10, while the processing time is shown in Table. 1. It can be seen that compared with [6], which detect PiP regions on frame level, PiP detection over the whole video ([4]) has higher precision and lower time consumption. And with edge line refinement, even higher precision and lower time consumption can be achieved (see [4] and 10frames) since less candidate positions are needed to be verified for video PiP.

The proposed method has a compatible detection accuracy performance with the ideal method of using all frames, which has the best performance but with much higher time consumption (see Proposed and Allframes). Furthermore, the proposed method outperforms the method which does not use spatio-temporal slices and simply takes more key frames into account (see Proposed and 20frames).

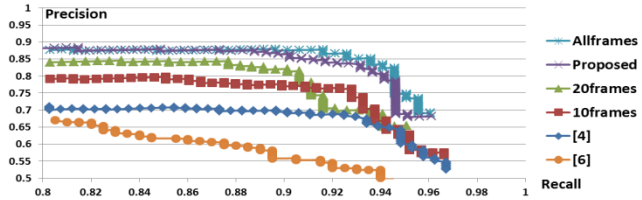

Fig. 10. Recall-precision curves of video PiP localization

Examples of video PiPs detected by the proposed method are displayed in Fig. 11, with right localization marked in light blue and wrong localization marked in red. It can be found that the proposed method cannot handle well those rectangular objects with strong and continuous edges, such as books, subtitles, and even fences. The reason is that these rectangle shaped objects are not easy to be distinguished from PiPs in our model.

Fig. 11. Examples of detected video PiPs

4.2. Experiment on PiP detection using STS

4.2.1. Dataset and evaluation

We applied our solution to the content based video PiP detection task to check whether the proposed STS PiP detection method works or not. The procedures of the copy detection system are explained in Sec. 3.5. The experiment settings are similar with [8], but with some difference. We conducted experiments on both small and large datasets chosen from TRECVID 2010 CCD task dataset, with the detailed setting shown below:

1. Small Dataset: consists of 130 query videos and 130 reference videos. All of these query videos are PiP videos copied from the 130 reference videos.
2. Large Dataset: consists of 196 query videos and 11524 reference videos, of which the small dataset is actually a subset. The additional 66 query videos are PiP videos but neither the foreground videos nor the background video is copied from the reference videos. The additional 11394 reference videos are used to interfere with the process of copy detection.

Precision-recall curve is used to show the performance of PiP detection, ordering by the similarity between query and reference videos using BoW approaches [13]. Besides, in order to make a comparison with [8], the average retrieval accuracy is also used for measurement, which is defined as dividing the number of correctly retrieved results with that of the total retrieved results.

4.2.2. Result and analysis

The experiment on video PiP copy detection is carried out under the settings shown in Table. 2.

Table 2. Experiment settings of video PiP copy detection

Method	[8]		STS (Proposed)	
Dataset	small	large	small	large
Query videos	120	120	130	196
Reference videos	120	12620	130	11524
Proc Time(s) (Feature extract)	3.0*	3.0*	5.1	5.1
Proc Time(s) (Feature match)	0.8*	5.5*	1.4	12.6

* The processing time is based on our realization of LP-MCP method in [8], which may have some difference with its original version.

As can be seen from Fig. 12, the basic BoW approach (labeled BoW) cannot handle well the problem of video PiP copy detection. In [8], spatial verification techniques are used as post-processing to refine the basic BoW approaches, which are efficient but still not promising. In our system, features from foreground and original frames are extracted and matched respectively, which is slower than [8], but with significant performance improvement (labeled STS), especially with large dataset (Fig. 12(b)).

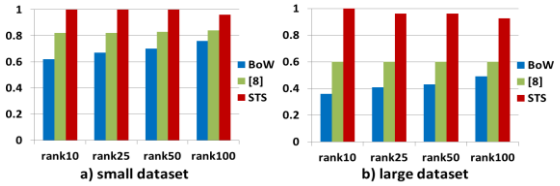

Fig. 12. Average retrieval accuracy of video PiP copy detection

Moreover, a further experiment is carried out to test the influence of PiP localization to the video PiP copy detection system, where we use the ground truth query video PiP localizations (with localization recall 1.0 and precision 1.0, labeled GT) to make a comparison with our STS-based video PiP localizations (with localization recall 0.96 and precision 0.7). The result is shown in Fig. 13. Although wrong video PiP may be localized in our STS-based approaches, they have limited influence to the whole copy detection system since no matching reference videos could be found with them. However, with some actual PiP localizations missed in our STS-based video PiP detection approaches, the recall rate is a bit lower than using ground truth PiP localizations.

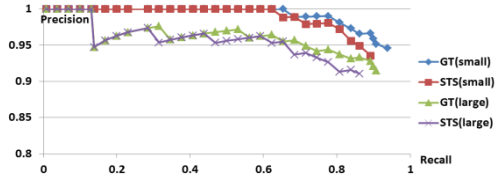

Fig. 13. Recall-precision curves of video PiP copy detection

5. CONCLUSIONS

To facilitate the effective video PiP detection, a new video PiP method is proposed with the corresponding edge surface probability measurement based on spatio-temporal slicing (STS). Besides, an algorithm is designed to refine vertical and horizontal edge lines by filtering noisy edges. Experimental results show that our method is effective and efficient in video PiP localization. When our method is applied in the video PiP copy detection system, the average retrieval accuracy could be significantly improved.

In the future work, we will further improve the proposed method to reduce false positive rate by eliminating the disturbing effects imposed by fence-like objects or subtitles.

6. REFERENCES

[1] Q. B. Orhan, J. Liu, J. Hochreiter, J. Poock, Q. Chen, Ajay Chabra, M. Shah, "University of Central Florida at TRECVID 2008 Content Based Copy Detection and Surveillance Event Detection", *TRECVID Workshop*, Gaithersburg, Nov. 2008

[2] J. Chen and J. Jiang. "University of Bradford at TRECVID 2008 Content Based Copy Detection Task", *TRECVID Workshop*, Gaithersburg, Nov. 2008

[3] J. M. Barrios and B. Bustos, "Content-Based Video Copy Detection: PRISMA at TRECVID 2010", *TRECVID Workshop*, Gaithersburg, Nov. 2010

[4] C. Sun, J. Li, B. Zhang and Q. Zhang, "THU-IMG at TRECVID 2010", *TRECVID Workshop*, Gaithersburg, Nov. 2010

[5] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion", *J. Opt. Soc*. Am. A, pp 284-299, Vol. 2, No. 2, Feb. 1985

[6] Z. Liu, E. Zavesky, N. Zhou and B. Shahraray, "AT&T Research at TRECVID 2011", *TRECVID Workshop*, Gaithersburg, Nov. 2011

[7] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 679-698, Nov. 1986

[8] S. Purushotham, Q. Tian and C. C. J. Kuo, "Picture-in-Picture Copy Detection Using Spatial Coding Techniques". *ACM international workshop on Automated media analysis and production for novel TV services*, Scottsdale, Dec. 2011

[9] NIST, "Guidelines for TRECVID 2010", *http://www-nlpir.nist.gov/projects/tv2010/tv2010.html#ccd*, available in Apr. 2014

[10] T. T. Do, L. Amsaleg, E. Kijak and T. Furon, "Security-oriented picture-in-picture visual modifications", In ACM International Conference on Multimedia Retrieval, Hong Kong, Jun. 2012

[11] Z. Liu, T. Liu and B. Shahraray, "AT&T research at TRECVID 2009 content-based copy detection", *TRECVID Workshop*, Gaithersburg, Nov. 2009

[12] H. Liu, H. Lu, X. Xue, "A Segmentation and Graph-Based Video Sequence Matching Method for Video Copy Detection", *IEEE Transactions on Knowledge and Data Engineering*, pp. 679-698, Aug. 2013

[13] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos", *IEEE International Conference on Computer Vision*, vol. 2, pp. 1470, Oct. 2003

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, pp. 91–110, Jan. 2004

[15] H. Liu, H. Lu and X. Y. Xue, "A Segmentation and Graph-Based Video Sequence Matching Method for Video Copy Detection", *IEEE Transactions on knowledge and data engineering*, vol. 25, pp. 1706-1718, Aug. 2013

[16] S. Wei, Y. Zhao, C. Zhu, C. Xu and Z. Zhu, "Frame Fusion for Video Copy Detection", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 15-28, Jan. 2011