A VIDEO COPY DETECTION ALGORITHM COMBINING LOCAL FEATURE'S ROBUSTNESS AND GLOBAL FEATURE'S SPEED

Xiaoguang Gu[1,2,3], *Dongming Zhang*[1,3], *Yongdong Zhang*[1,3], *Jintao Li*[1,3], *Lei Zhang*[1,2,3]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Beijing Key Laboratory of Mobile Computing and Pervasive Device
(Institute of Computing Technology, Chinese Academy of Sciences)
xggu@ict.ac.cn

ABSTRACT

This paper presents a novel algorithm for fast and robust video copy detection. The idea is to use local features to estimate the copy transformation parameters first and then use the estimated parameters to guide the global-feature-based matching at a later stage. It is based on the fact that the copy transformations generally remain unchanged in a continuous video clip even in the whole video. Local-feature-based matching can find the candidates which are difficult to be detected only using global features. Furthermore, the matched local feature points can provide enough information to estimate the copy transformations. After the copy transformations are estimated, the subsequent detection can be accelerated by doing global-feature-based matching. The experimental results show that the proposed algorithm can get the same good robustness as the local-feature-based method but the faster detection speed.

Index Terms— Content-based copy detection, local feature, global feature, approximate nearest neighbour search

1. INTRODUCTION

Content-based video copy detection (CBCD) plays an important role in many practical applications, such as digital copyright protection, video tracking, large-scale video databases, and so on. The goal of video copy detection is to identify the original and modified copies of a video from a large amount of videos. To date, many algorithms have been developed.

Because of the success of local features in the area of image retrieval, they have also been adopted to many CBCD systems[1][2][3]. Local features are inherently resistant to the transformations caused by some post-production operations, for example cropping, since a part of original content always remains in the copy. However, local feature extracting and matching are all very compute expensive[4].Compared with local features, global features are more compact and efficient. There have been many CBCD systems based on global features[5][6]. Global features are normally based on the statistics of the entire frame or the whole clip. Therefore, global-feature-based approaches are sensitive to geometric transformations[4].

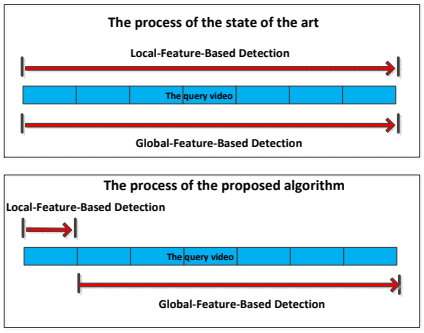

Fig. 1. The difference between the state of the art and the proposed algorithm

Some approaches which integrate local and global features are brought forward [7][8][9][10].The detection results based on global and local features are integrated in a post-processing stage to make the results reliable and accurate. The process is shown in the top of **Figure 1**. Although these approaches can get a better detection result, the compute complexity of their algorithms becomes more higher.

In this paper, we provide a novel integration algorithm to do a robust and fast CBCD. It is every different from the existing integration algorithms. The proposed algorithm use local features to estimate the copy transformation parameters first and then use the estimated parameters to guide the global-feature-based matching at a later stage. The process is shown in the bottom of **Figure 1**. It is based on the fact that the copy transformations generally remain unchanged in a continuous

Thanks to National Nature Science Foundation of China(61273247,61271428); National key technology support program(2012BAH39B02).

video clip even in the whole query video. Local-feature-based matching can find the candidates which are difficult to be detected only using global features. Furthermore, the matched local feature points can provide enough information to estimate the copy transformations. After the copy transformations are confirmed, the subsequent detection can be accelerated by doing global-feature-based matching. Therefore, the proposed algorithm can get the same good robustness as the local-feature-based method but the faster detection speed.

2. OVERVIEW OF THE PROPOSED ALGORITHM

Figure 2 shows the processing procedure of the proposed algorithm. We use SIFT feature[11] to do the local-feature-based detection. Product Quantization provided in [12] is used to index the SIFT features. The similar frames are identified by voting the matched local feature points and validating the spacial relationship. The matched frames are further filtered by sequential relationship in the video. These processes are common. The main differences are detailed in Section 3 and 4.

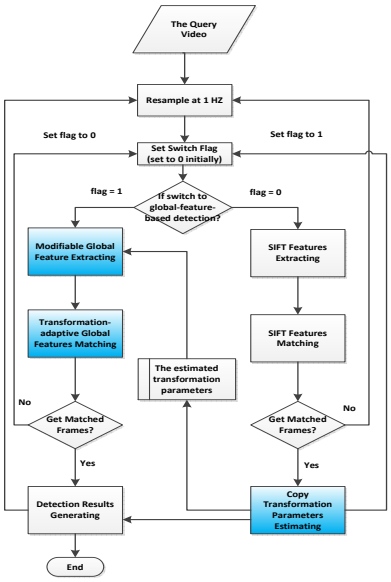

Fig. 2. The proposed CBCD algorithm

3. COPY TRANSFORMATION PARAMETERS ESTIMATION

Firstly, we employ RANSAC algorithm to estimate the affine transformation that maps the points in the query frame to those in its matched reference frame. The affine transformation can model the geometric changes introduced by the transformations such as picture in picture, shift, zoom, ratio, etc. It has been exploited to remove the mismatched local feature points in many reported CBCD systems.

For a local feature point at pixel (x_q, y_q) in the query frame, it is mapped to the pixel (x_r, y_r) in the reference frame by the following formula:

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_q \\ y_q \\ 1 \end{bmatrix} \quad (1)$$

There are six affine transformation parameters to be determined in the formula (1). Therefore, three pairs of matched points are required. These points must not be on the same line. The parameters estimated by randomly selected three pairs of matched points are not reliable. The algorithm using RANSAC to precisely estimate the parameters is detailed in **Algorithm 1**.

Algorithm 1 The Affine Transformation Parameters Estimating Algorithm Based on RANSAC

Require:

The set of the matched local feature point pairs;

Ensure:

Six affine transformation parameters a, b, c, d, t_x, t_y;

1: Calculate the orientation difference of every pair of matched points. Generate a 36-dimensional histogram of the orientation difference;
2: Find the peak bin of the histogram. The point pairs in the peak bin and its left and right two neighbour bins are used in the next step. The other point pairs are filtered out.
3: Randomly select 3 pairs of matched local feature points;
4: Calculate the affine transformation parameters using formula (1) and record them. Transform all points in the query frame using the calculated parameters;
5: Calculate the distance from the transformed coordinate of point in the query frame to the coordinate of its matched point in the reference frame. If the distance is lower than the threshold, this point is inlier point. Count the number of the inlier points in the query frame. The counted number is recorded as a score. Update the best score and the recorded parameters;
6: Randomly select 3 pairs of matched local feature points from the inlier points set. Do the same process as 4 to 5;
7: Repeat 3 to 6 to get stable parameters;
8: **return** The estimated parameters which are corresponding to the best score;

Figure 3 presents the result of **Algorithm 1** in the case of Picture in Picture. The proposed algorithm can precisely estimate the transformation parameters.

After the affine transformation parameters are estimated precisely, we can align the query frame with the reference frame in space and scale by doing the estimated transformation on the query frame. Through comparing the transformed query frame with the original reference frame, we can capture the local difference introduced by some copy transformations

between them. We partition the frame as presented in **Figure 4(b)**. The whole frame is partitioned to 25 blocks. Every block is assigned an order number. We calculate the similarity, s_i, between the block in the query frame and its corresponding block in the reference frame using gray histogram intersection. There are 25 similarity values to be calculated. Suppose the average is u. The blocks with the obvious low similarity must contain some distortions. We construct a 25-dimensional integer vector, W, to represent which block is unchanged and which block has local distortion. The first dimension is corresponding to the first block, and so on. Every dimension is set to 0 or 1 according to the similarity between the corresponding blocks. If the similarity is less than the average, the corresponding dimension is set to 0, and vice versa. The definition is shown in formula (2). **Figure4 (c) and (d)** show an example.

$$W =< w_1, w_2, w_i, ..., w_{25} >, w_i = \begin{cases} 1, if s_i >= u \\ 0, if s_i < u \end{cases} \quad (2)$$

Finally, we get six affine transformation parameters and a 25-dimensional vector. They can model all kinds of copy transformations to which the global features have no invariability.

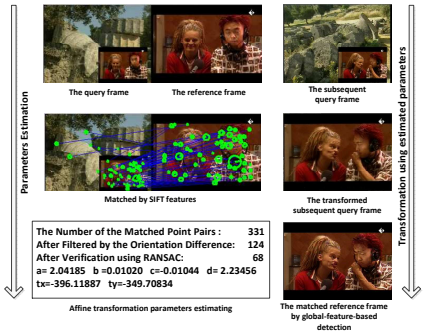

Fig. 3. Algorithm 1 can precisely estimate the affine transformation parameters. The query video and the reference video are all from the TRECVID 2009 dataset.

4. GLOBAL FEATURE EXTRACTING AND MATCHING

To extract the global features used in our work, the frames are first partitioned into 25 blocks as **Figure 4(b)** shows. Secondly, an OM feature[5] is extracted from 1-9 blocks. The OM feature can be represented by a 32-bits integer. Thirdly, the average and the variance of gray are calculated for each left block. We concatenate all these features into a 33-dimensional feature vector according to the ordinal number of the blocks. The definition of the proposed global feature is shown in **Figure 5**.

For the reference frames, the global features are directly extracted from the original frame. For the query frames,the

Fig. 4. (a)The query frame in TRECVID 2009 dataset. (b)Partition the frame. (c)The transformed query frame using the estimated parameters. (d)The matched reference frame. By comparing the corresponding blocks in (c) and (d), we get a 25-dimensional vector to represent which block is unchanged and which block has local distortion.

global features are extracted from the transformed query frames. The extracted global features can be matched adaptively to tolerate all kinds of local changes. We use the OM feature of the first nine blocks to construct the inverse index. When the global feature is extracted from a query frame, we use its first dimension to query in the inverse index. As a result, the reference frames which have the same OM feature as the query frame can be returned. Next, the left 32 dimensions are used to refine the results. The final similarity is calculated as follows:

$$Sim_{refine}(p,q) = \sqrt{\sum_{i=2}^{33}((p_i - q_i) \cdot w_{\lfloor i/2 \rfloor + 9})^2} \quad (3)$$

W, defined in (2), is used to do a transformation-adaptive matching. Using formula (3) to calculate the similarity, the blocks which contain the distortions are excluded. Therefore, the matching result is robust to the distortions.

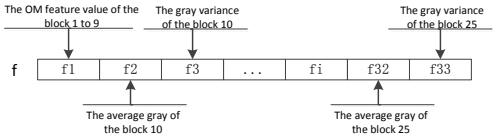

Fig. 5. The proposed global feature.

5. EXPERIMENTS

We evaluate our CBCD algorithm using the TRECVID 2009 dataset [13] which includes seven kinds of copy transformations. TRECVID defines three key performance measures.

They are normalized detection cost rate (NDCR), copy location accuracy, and copy detection processing time.

NDCR is defined by a weighted mean of the two errors: false negatives and false positives. In this paper, we show the results only for the NOFA profile because NDCR values become almost the same for BALANCED and NOFA profiles on the TRECVID 2009 settings.

Copy location accuracy is measured by the F1 score, which is the harmonic mean of the precision and recall of the detected copy location relative to the true video segment. It is calculated only for the correctly detected copies.

Copy detection processing time is the mean processing time per query. It includes all processing time from reading in the query video to the output of results.

The experiments are performed on a workstation with 24G memory, 2.13GHz Intel CPU and 64-bit Operating System. The top-500 similar local features are used to identify similar frames. The top-50 matched frames are used to identify the copy video clip. For global-feature matching, the similarity threshold is set to 10.

5.1. Experiment results

Figure 6 shows the copy detection performance of the proposed algorithm. As the comparison, we also present the performance of the individual global-feature-based(using our global feature) and local-feature-based (using SIFT feature) methods, as well as the best TRECVID2009 submission[13]. The proposed algorithm has the same good performance as the local-feature-based method. It gets a comparable performance to the best TRECVID2009 submission. The individual global-feature-based method has the worst performance. This is because the global feature is not robust to the transformations such as PIP, pattern insertion and post-product.

Furthermore, in **Figure 7**, the proposed algorithm gets a query processing time which is obviously faster than the individual local-feature-based method. Because we use the original SIFT feature, the processing time is only the median of the TRECVID2009 submissions which is about 80 seconds. If we use some faster local feature extracting algorithm, the processing time can be further shortened. In the TRECVID2009 dataset, there are third query videos whose whole content is copy . The proposed algorithm gets a very fast processing speed for these query videos. The query processing time is very close to the global-feature-based algorithm. This is because our algorithm does local-feature-based detection only on the several start frames. The left part of the query video is all detected using our global feature.

Figure 8 shows the localization performance of the proposed algorithm. The proposed algorithm gets a better result than the individual local-feature-based method and global-feature-based method.

Synthesizing the experiment results shown in **Figure 6 to 8**, we can conclude that the proposed algorithm has the same good copy detection performance as the local-feature-based method and the significantly faster detection speed than the local-feature-based method. In some cases, the proposed algorithm can get a very similar efficiency as the global-feature-based method.

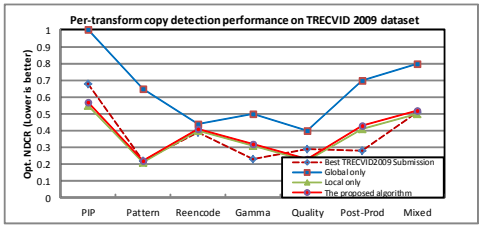

Fig. 6. Detection performance on TRECVID 2009 dataset.

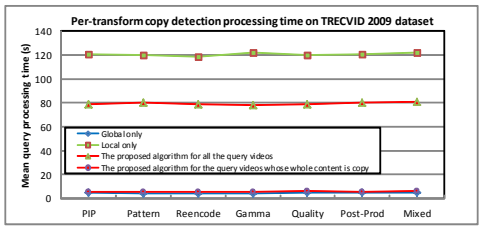

Fig. 7. The query processing time on TRECVID 2009 dataset.

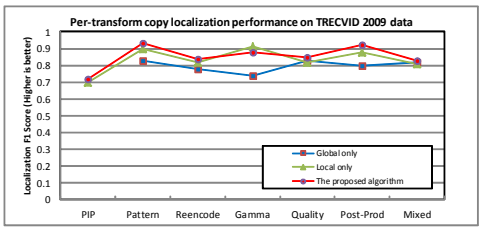

Fig. 8. Localization performance on TRECVID 2009 dataset.

6. CONCLUSIONS

Based on the fact that the copy transformations generally remain unchanged in a continuous video clip even in the whole video, this paper presents a novel algorithm which combines the local-feature-based and global-feature-based detection methods. The proposed algorithm uses local features to estimate the copy transformation parameters at the beginning and then use these parameters to do a transformation-adaptive global-feature-based matching at a later stage. The experimental results show that the proposed algorithm can get the same robustness as the local-feature-based method but the faster detection speed.

7. REFERENCES

[1] M. Douze, H. Jegou, and C. Schmid., "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. on Multimedia*, vol. 12(4), pp. 257–C266, 2010.

[2] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust voting algorithm based on labels of behavior for video copy detection," in *in Proc. ACM Int. Conf. Multimedia*, 2006, pp. 835–844.

[3] S Liu, P Cui, H Luan, W Zhu, S Yang, and Q Tian, "Social visual image ranking for web image search," *Advances in Multimedia Modeling*, pp. 239–249, 2013.

[4] J. Law-To, L. Chen, A. Joly, and I. Laptev., "Video copy detection: a comparative study," in *In Proc. of CIVR*, 2007, pp. 371–378.

[5] X. Hua, X. Chen, and H. Zhang, "Robust video signature based on ordinal measure," in *In Proc. of ICIP*, 2004, pp. 685–688.

[6] Lifeng Shang, Linjun Yang, Fei Wang, Kwok-Ping Chan, and Xian-Sheng Hu, "Real-time large scale near-duplicate web video retrieval," in *in Proc. ACM Int. Conf. Multimedia*, 2010, pp. 531–540.

[7] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo, "Practical elimination of near-duplicates from web video search," in *in Proc. ACM Int. Conf. Multimedia*, 2007, pp. 218–227.

[8] Jingkuan Song, Yi Yang, Zi Huang, Hengtao Shen, and Richang Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *ACM Multimedia*, 2011, pp. 423–432.

[9] Yusuke Uchida, Motilal Agrawal, and Shigeyuki Sakazawa, "Accurate content-based video copy detection with efficient feature indexing," in *In Proc. of ICMR*, 2011.

[10] Yue Wang, ZuJun Hou, Karianto Leman, Nam Trung Pham, TeckWee Chua, and Richard Chang, "Combination of local and global features for near-duplicate detection," in *In Proc. of MMM*, 2011, pp. 328–338.

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

[12] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. on PAMI*, vol. 33(1), pp. 117–128, 2011.

[13] W. Kraaij, G. Awad, and P. Over, "Trecvid-2009 content-based copy detection task overview," in *in TRECVID-2009 Workshop*.