International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

SIFT Based Video Copy Detection

Swapnil S.Kharat[1], Mayur Ingale[2], Lalit Shevkani[3], Rakeshkumar Yadav[4]

[1]*Department of Electronics & Telecom. Engineering, SSJCET,Asangaon, swapnilskharat@hotmail.com*
[2]*Department of Electronics & Telecom. Engineering, SSJCET,Asangaon, mayu_ingale@yahoo.co.in*
[3]*Department of Electronics & Telecom.Engineering, SSJCET,Asangaon, lalitshevkani@gmail.com*
[4]*Department of Electronics & Telecom.Engineering, SSJCET,Asangaon, rakeshkumaryadav1098@gmail.com*

Abstract- Due to advances in multimedia compression technology and growth of internet, digital videos are widely used. These videos can be found on public video web servers, TV channels and video blogs. Hence there is need for automated procedures to protect the owner of video against unauthorized use of their video content, preventing copyright violations. The video can be altered by various transformations. This paper presents video copy detection method using SIFT (Scale Invariant Feature Transform) descriptors. To reduce the computational complexity of SIFT; first we used the dual-threshold method to segment the videos into segments and extract key-frames from each segment. SIFT features are extracted from the key-frames of the segments. Then a nearest neighbor method is used to match two video frames with SIFT point set descriptors. We have also compared the performance of SIFT descriptor with Trajkovic detector for various types of transformations. The distinguishing feature of our method is that it is capable of detecting the most difficult transformation: Picture-in-Picture. We have also tested our system for four other types of transformations as well

Keywords- SIFT, Auto dual thresholding, key frame, descriptors, TRECVID 2008

I. INTRODUCTION

With the rapid development of multimedia hardware and software technologies, the cost of image and video data collection, creation and storage is becoming increasingly low. As a consequence, an effective and efficient method for video copy detection has become more and more important. Content-based video copy detection addresses the issue that automatically determines whether a query video contains a copy from a given database of reference videos and if so from where the copy comes. Here the term "copy" means a video segment derived from another video usually by visual and/or audio transformations.

In order to facilitate the discussion of "video copy" we use the definition of video copy in TRECVID 2008 tasks in which 10 transformations [5] are defined. These 10 transformations are as below [6].

T1. Cam Cording, T2. Picture in picture, [6], T3. Insertions of pattern: Different patterns are inserted randomly captions, subtitles, logo, sliding captions, T4. Strong re-encoding, T5. Change of gamma, T6, T7. Decrease in quality: Blur, change of gamma (T5), frame dropping, contrast, compression (T4), ratio, white noise; T8, T9. Post production: Crop, Shift, Contrast, caption (text insertion), flip (vertical mirroring), Insertion of pattern (T3), Picture in Picture (the original video is in the background), T10.

The organization of our paper is as follows. Related research work is explained in section II. The proposed block diagram for video copy detection system and auto dual-threshold method for eliminating redundant frames is presented in Section III and IV respectively. Trajkovic operator and SIFT for extracting the key points are explained in section V and VI respectively. Based on the extracted SIFT features for two key frames, the feature matching on Euclidean distance is explained in Section VII. Experimental results are presented in Section VIII. And we conclude our work in Section X.

II. RELATED RESEARCH

The methods on copy and near duplicate detection can be grouped into two types. One type of copy detection method uses global descriptor. Specifically, Hampapur et al. compares distance measures and video sequence matching methods for video copy detection [4]. They employed convolution for motion direction feature, L1 distance for Ordinal Intensity Signature (OIS), and histogram intersection for color histogram feature. The results show that the method using OIS performs better. Yuan et al. combined OIS with color histogram feature as a tool for describing video sequence [3]. Another type of methods are based on local descriptors. The local descriptors on points, lines, and shape play an important role in image and video copy detection.

Methods based on global descriptor are carried out primarily by using spatiotemporal low level features of the whole image. The features used include color histogram, color layout descriptor, ordinal intensity signature, etc. The methods based on local descriptor must first detect local spatiotemporal feature points on the video sequence, i.e. interest points or key points, and then use the content around the feature points to them. Mikolajczyk and Schmid made a comparative study on many local descriptors. The study showed that the SIFT descriptor performs better in identifying the objects.

III. BLOCK DIAGRAM OF VIDEO COPY DETECTION SYSTEM

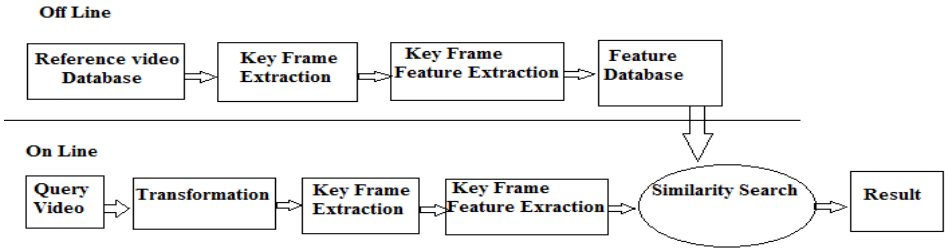

Figure1. A Block Diagram of Video Copy Detection System

Fig. 1 shows the block diagram of video copy detection system [1]. It consists of two parts

1) **An off-line step**. Key frames are extracted from the reference video database and features are extracted from these key frames. The features can be stored in an indexing structure in order to make similarity comparison efficacious.

2) **An on-line step**. Query videos are analyzed. Features are extracted from these videos and compared to those stored in the reference database. The matching results are then analyzed and the detection results are rendered.

IV. AUTO DUAL-THRESHOLD METHOD

Normally visual information of video frames is temporally redundant. So, video sequence matching is not necessarily to be carried out using all the video frames. An effective way of

reducing non-necessary matching is to extract certain key frames to represent the video content. And the matching of two video sequences can be first performed by matching the key frames. Specifically, Guil et al. [9] proposed to cluster video frames by computing the similarity between neighboring frames and choose a key frame from each cluster to represent it.

This proposed method uses auto dual-threshold method to get rid off redundant video frames. This method cuts consecutive video frames into video segments by eliminating temporal redundancy of the visual information of consecutive video frames. This method has the two characteristics. Out of two thresholds first is used for detecting abrupt changes of visual information of frames and second for gradual changes. Specifically, $Th=\mu + \alpha\ \sigma$ where μ and σ are the mean and standard deviation of difference values between consecutive frames and α is suggested to be between 5 and 6 according to empirical study[5] . And the low threshold Tl is set to b x Th, where b is selected from the range 0.1-0.5. The auto dual threshold method to get rid off redundant frames is shown in Fig. 2.

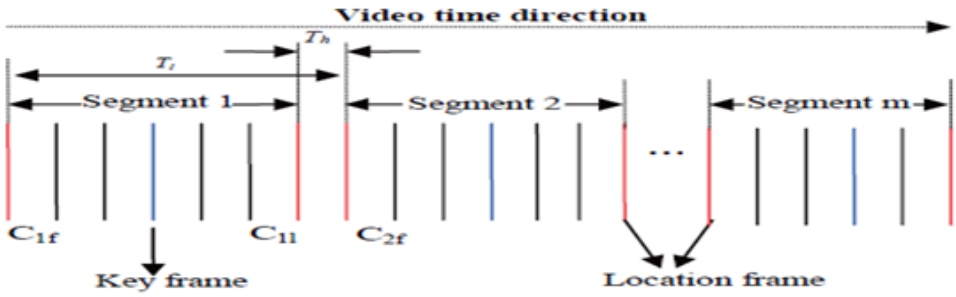

Figure2. Auto dual-threshold method to get rid off redundant video frames, select key frames.C1f denotesthe first frame of Segment 1, C1l the last frame of the Segment 1; C2f denotes the first frame of the Segment 2.

Three frames are extracted from each video segment, which are the first frame, the key frame and the last frame of this segment. The key frame is determined by the frame that is the most similar to the average frame. The key frame is used for video sequence matching, while the first and the last frames for accurately determining the segment location for copy detection and assisting matching.

V. TRAJKOVIC OPERATOR

This operator was developed by Miroslav Trajkovic and Mark Hedley in 1998 with the intent of obtaining comparable repeatability rates and localization performance as the most popular corner detectors, while requiring a minimum of computation. The Trajkovic corner detector is stated formally below:

Input: greyscale image, scale for low resolution version of image, threshold T1, threshold T2

Output: map indicating position of each detected corner

5.1 Apply Corner Operator

This step takes as input the image and typically a few parameters required by the corner operator. For each pixel in the input image, the corner operator is applied to obtain a cornerness measure for this pixel. The cornerness measure is simply a number indicating the degree to which the corner operator believes this pixel is a corner as below.

$$C_{simple}(x,y) = \min (r_A, r_B)$$

Where $r_A = (I_A-I_C)^2 + (I_A'-I_C)^2$) is horizontal intensity variation and $r_B = (I_B - I_C)^2 + (I_B' - I_C)^2$ is vertical intensity variation. Flag any pixel (x, y) with a cornerness measure $C_{SIMPLE}(x, y) \geq T1$ as a potential corner. Where I_A , I_B , I_C are intensities of pixels at points A, B, C respectively.

5.2 Threshold Cornerness Map

Initialize a cornerness map M, with dimensions of the input image, to be all zeros. Interest point corner detectors define corners as local maximum in the cornerness map. However, at this point the cornerness map will contain many local maximum that have a relatively small cornerness measure and are not true corners. To avoid reporting these points as corners, the cornerness map is typically thresholded.

5.3 Interpixel approximation cornerness measure

Compute the interpixel approximation cornerness measure as below:

$$C_{INTERPIXEL}(X,Y)= \begin{cases} c-\frac{B^2}{A} & ; if\ B<0\ and\ (A+B)>0 \\ C_{simple}(x,y) & ; otherwise \end{cases}$$

Where, B1 = $(I_B - I_A) (I_A - I_C) + (I_{B'} - I_{A'}) (I_{A'} - I_C)$

B2 = $(I_B - I_A') (I_B' - I_C) + (I_{B'} - I_A) (I_A - I_C)$

C = r_A

B = min (B1, B2)

A = $r_B - r_A - 2C$

If $C_{INTERPIXEL}(x, y) < T2$ leave the cornerness measure in M as zero, else set M(x, y) to $C_{INTERPIXEL}(x, y)$.

5.4 Non-maximal Suppression

The thresholded cornerness map contains only nonzero values around the local maximums that need to be marked as corner points. To locate the local maxima, non-maximal suppression is applied. For each point in the thresholded cornerness map, non-maximal suppression sets the cornerness measure for this point to zero if its cornerness measure is not larger than the cornerness measure of all points within a certain distance. After nonmaximal suppression is applied, the corners are simply the non-zero points remaining in the cornerness map.

VI. Scale Invariant Feature Transform (SIFT)

Following are the major stages of computation used to generate the set of SIFT features [2].

6.1 Scale-space extrema detection

Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. It has been observed that under a variety of assumptions the only possible scale-space kernel is the Gaussian function. Therefore, the scale space of an image is a function, L(x, y,σ) that is derived from the convolution of a variable-scale Gaussian, G(x, y, σ), with an input image, I(x, y)

$$L(x,y, \sigma) = G(x,y, \sigma) * I(x,y)$$

Where * is the convolution operation in x and y, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

To expeditiously detect stable key point locations we have proposed scale-space extrema in the difference-of-Gaussian function convolved with the image D(x, y, σ), which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k.

For Laplacian of Gaussian (LoG) operation .Take an image, blur it a little and then calculate second order derivatives on it which will locates edges and corners on the image. These edges and corners are good for finding keypoints. But the second order derivative is noise sensitive. The blur smoothes out the noise and also stabilizing the second order derivative uses the scale space to generate Laplacian of Guassian images quickly. Then calculate the difference between two consecutive scales. Or, the Difference of Gaussians shown in figure 3.

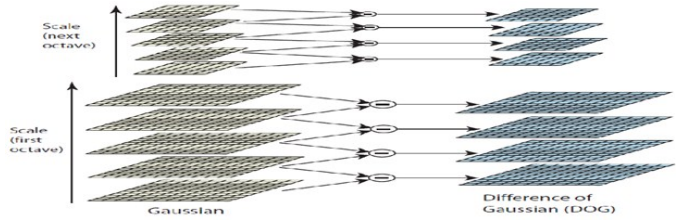

Figure3. Difference of Gaussian

For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. The difference-of-Gaussian images are produced by subtracting adjacent Gaussian images. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

6.2 Maxima/minima location in DoG images

The first step is to locate the maxima and minima. Iterate through each pixel and check all its adjacent. The check is done within the current image, and also its neighbors. Something like shown in figure 4. A total of 26 checks are made by considering X as current pixel and green circles as its neighbors. X is marked as a key point if it is the greatest or least of all 26 neighbors. Usually, a non-maxima or non-minima position won't have to go through all 26 checks. A few initial checks will usually sufficient to discard it. Key points are not detected in the lowermost and topmost scales. Once this is done, the approximate maxima and minima points are marked. They are "approximate" because the maxima/minima almost never lies exactly on a pixel. It lies somewhere between the pixel. So mathematically locate the sub pixel location.

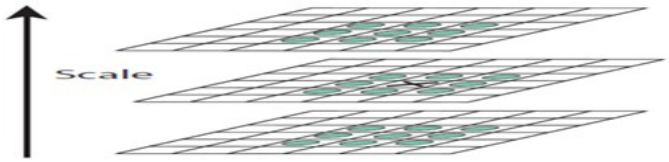

Figure4. Detection of Maxima and Minima

6.3 Detection of local extrema

Using the valid pixel data, sub pixel values are generated which is done by the Taylor expansion of the image around the approximate key point. Mathematically,

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

Using above equation it is easy find the extreme points by differentiating and equating to zero. After which sub pixel key point locations can be found. These sub pixel values increase chances of matching and stability of the algorithm.

6.4 Edge responses elimination

The difference of Gaussian function will have a strong response along edges and hence it is unstable to small amounts of noise. If the magnitude of the intensity at the current pixel in the DoG image is less than a certain value, it is discarded due to sub pixel key points so again need to use the Taylor's expansion to get the intensity value at sub pixel locations. If its magnitude is less than a certain value, the key point is rejected. The principal curvatures can be computed from a 2x2 Hessian matrix, H, computed at the scale and location of the key point.

$$[H] = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The derivatives are calculated by taking differences of neighboring sample points .Calculate two gradients at the key point to remove edges .Based on the image around the key point, three possibilities exist. The image around the key point can be:

A flat region
In this case, both gradients will be small.

An edge

In this case one gradient will be big (perpendicular to the edge) and the other will be small (along the edge)

A corner
Both gradients will be big.

Corners are great key point [2]. If both gradients are big enough so let it pass as a key point. Otherwise, it is rejected.

6.5 Assigning orientation

By assigning a continuous orientation to each key point based on local image properties, the key point descriptor can be represented relative to this orientation and thus achieve invariance to image rotation. For each image sample, L(x, y), at this scale, the gradient magnitude m(x, y), and orientation, Φ(x, y) are computed using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Within a region around the key point an orientation histogram is formed from the gradient orientations of sample points. The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with σ that is 1.5 times that of the scale of the key point .In this histogram; the 360 degrees of orientation are broken into 36 bins (each 10 degrees). Once done this for all pixels around the key point, the histogram will have a peak at some point .Also the highest peak are converted into a new key point which are above 80%.

6.6 SIFT Features Generation

Take a 16×16 window of "in-between" pixels around the key point, split that window into sixteen 4×4 windows. From each 4×4 window generate [2] a histogram of 8 bins. Each bin corresponding to 0-44 degrees, 45-89 degrees, etc. Gradient orientations from the 4×4 are put into these bins. This is done for all 4×4 blocks. Finally normalize the 128 values.

II. MATCHING SIFT FEATURE POINTS FOR COPY DETECTION

The best candidate match for each key point is found by identifying its nearest neighbor in the database of key points from training images. The nearest neighbor is defined as the key point with minimum Euclidean distance for the invariant descriptor vector. However, many features from an image will not have any correct match in the training database because they arise from background clutter or were not detected in the training images. Therefore, it would be useful to have a way to discard features that do not have any good match to the database. A global threshold on distance to the closest feature does not perform well, as some descriptors are much more discriminative than others. A more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. If there are multiple training images of the same object, then we define the second-closest neighbor as being the closest neighbor that is known to come from a different object than the first, such as by only using images known to contain different objects.

III. EXPERIMENTAL RESULTS

Figure 5 shows the comparison between SIFT and Trajkovic operator for video copy detection in GUI of Matlab for PiP transformation

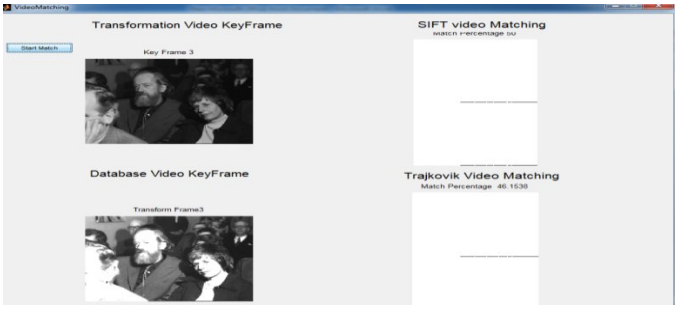

Figure5. Video matching for TRECVID 2008

We can summarize the video copy detection for all kinds of database in tabulated form as shown below in table no.1 and 2.

Table2. Comparison of SIFT and Trajkovic for different transformation

DATASET	TRANSFORMATIONS					
	CAMCORDING		CHANGE OF GAMMA		LOGO INSERTION	
	SIFT	TRAJOVIC	SIFT	TRAJOVIC	SIFT	TRAJOVIC
TRECVID	50	15.38	50	46.67	50	20
BBC NEWS	50	3.57	62	5.67	50	21.42
NGC	50	2.67	53	5.67	50	20
AERIAL	60	21	50	16.67	50	14.23

II. CONCLUSION

This paper first analyzes different video copy types and the features used for copy detection. To describe video frames it uses local feature of SIFT. The video copy detection using SIFT features has high computational cost since the number of SIFT points extracted from a video is

large. Then a dual-threshold method is used to discard redundant video frames. Furthermore, for video sequence matching, we propose a nearest member matching based on Euclidean distance for video sequence matching method. It skillfully detects the video sequence matching result. In real-world applications, flip or flip-like transformations are commonly observed in images due to artificial flipping, opposite capturing viewpoint, or symmetric patterns of objects. In order to detect flip like transformations we plan to use flip-invariant SIFT (or F-SIFT) in our future work. However, our proposed method is very effective and efficient for detecting the video copies.

REFERENCES

[1] Hong Liu, Hong Lu, and Xiangyang Xue, "A segmentation and Graph-based Video Sequence Matching Method for Video Copy Detection", ieee transactions on knowledge and data engineering.
[2] David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 2004
[3] J.Yuan, L.-Y. Duan, Q. Tian, S. Ranganath, and C. Xu, "Fast and robust short video clip search for copy Detection," in Pacific Rim Conf. on Multimedia (PCM), 2004..
[4] A. Hampapur and R. Bolle, "Comparison of distance measures for video copy detection," in Proc. IEEE Int. Conf. Multimedia and Expo (ICME), 2001, pp. 188–192.
[5] TRECVID 2008 Final List of Transformations, 2008, http://www-nlpir.nist.gov/projects/tv2008/active/copy detection/final.cbcd.video.transformations.
[6]Fina lCBCD EvaluationPlanTRECVID2008(v1.3),2008,http://www-nlpir.nist.gov/projects/tv2008/Evaluation cbcd-v1.3.htm.
[7] Z. Huang, H. T. Shen, J. Shao, B. Cui, and X. Zhou, "Practical online near-duplicate subsequence detection For continuous video streams," IEEE Transactions on Multimedia, vol. 12, no. 5, pp. 386–397, August 2010.
[8] N. Guil, J. M. Gonzlez-Linares, J. R. Czar, "A clustering technique for video copy detection," in Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Girona, Spain, June 2007, pp. 452–458.

