Event Video Segmentation using Trajectory Discontinuity

Charlie Maere
Postgraduate Researcher, College of Information and Communication Engineering
Harbin Engineering University, China
Email: cmaere [AT] kcn.unima.mw

Zhang Lei
Professor, College of Information and Communication Engineering
Harbin Engineering University, China

Abstract—**The ability of accurately segmenting a video into events or scenes without any human intervention has been a challenge in computer vision. In this paper, we present a more accurate way of segmenting a video into events using trajectory discontinuities, which consist of two stages. In the first stage, we estimate possible regions where the scene changed using comparison of motion features we extracted by edge detection segmentation and the second stage, we use the estimated regions to estimate trajectory discontinuity using Large Displacement Optical flow (LDOF) which calculates the actual frame where the scene has changed. Our experiments have shown that this method have high accuracy rate and it is faster that the conventional way of event video segmentation.**

Keywords— ***Large Displacement Optical flow, Video event segmentation, Edge detection, spatial information***

I. INTRODUCTION

There has been a lot of research on video segmentation, where the goal is to segment a video sequence into moving objects and the world scene, researchers defined this process as assigning labels to pixels in a video sequence. But this paper focuses on another kind of video segmentation, called video event segmentation in which the main goal is to separate individual events in a video sequence. The concept is derived from how a human has the ability to identify or perceive an event when observing a video sequence. In this paper we will use the word event and scene interchangeably, which is defined as a sequence of continuous action in a video.

[1] Define video events as those high-level semantic concepts that humans perceive when observing a video sequence. A human brain is an interesting organ of the human, it resides a lot of secrets to science. One of it is the ability to identify or perceive an event from a video sequence. The brain has this amazing ability to differentiate between a wedding ceremony and a football match. The brain does this by interpreting a video we watch into meaningful concepts based on its memory and human perception. In this paper we attempt to give a computer this similar ability of this human-like perception of an event. Video event detection is one of a high-level task in computer vision and video/image processing to be precise. It depends on output data from many lower level tasks such as feature extraction that involves image/video segmentation, edge detection and optical flow estimation, feature analysis that involves object recognition, object classification and tracking. Each part of these lower level tasks has spawned major research interests.

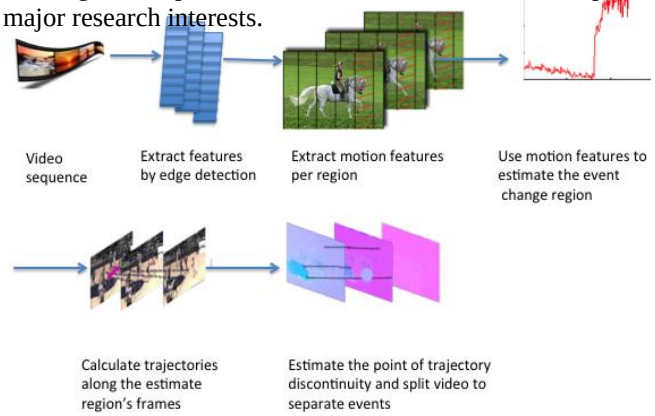

Fig. 1. Whole framework for video event segmentation by trajectory discontinuity

Video event segmentation is a fundamental operation used in many multimedia applications such as digital libraries, video on demand (VOD), interested event retrieval or video event classification. With the overwhelmingly rapid progressing in multimedia, video event segmentation will be urgent required.

As for video segmentation, most works focus on finding the boundary based on shot or different scene. A typical framework is to match the difference between two consecutive frames, where the feature for frames could be low-level global features such as the luminance pixel-wise difference, luminance or color histogram difference [1]. Since luminance or color is sensitive to small change, other mid-level features on STIP (spatial-temporal interested points) are proposed [2]. There still are some researches to fuse audio and video features to analysis or recognizing video event.

However, for video event segmentation, most attention should be paid on interactions between object and people, that is, the moving part in the video sequence. Since for movement detection, optical flow [3] outperforms other features [4], in

this paper, we consider applying it in video event segmentation. Compared with traditional optical flow, the large displacement optical flow (LDOF) [5], a recent variation optical flow method, can deal with faster motion than previous optical flow techniques. In [3], LDOF is applied for video segmentation to separate moving objects from background, here; we firstly apply it in video event segmentation.

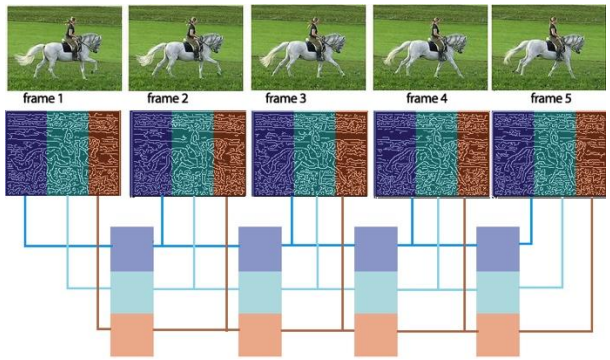

Fig. 2. Stripe partition and motion vector generation scheme

Our main contributions in this paper are as follows:

- Firstly applied LDOF to obtain trajectory discontinuity in video event segmentation.
- In order to lessen the computing complexity, we added a pre-processing stage before LDOF.
- In pre-processing stage, we combined edge detection with spatial information to detect all shot boundaries and treat them as the candidates for trajectory discontinuity detection.

The whole framework of our approach is given in Fig. 1. By edge detection and spatial pyramid, we extract motion feature in pre-processing stage, and then calculate LDOF and find its discontinuity part along video sequence.

II. PRE-PROCESSING BY MOTION FEATURE

Although LDOF can be parallel implemented and is about 78 times faster than the conventional one, it has been proven that method is time consuming for long videos of more than 1000 frames. Since for event lasting from tens second to several minutes, it is impossible to apply LDOF directly here.The pre-processing stage aims to shorten the possible candidate length for later applying LDOF.

The motion feature extraction is considered from two aspects: the information of each frame content where canny edge detection is adopt here and motion information extraction, where we combine canny edge information with spatial information and match them between the consecutive frames.

A. Edge detection by canny detector

- Canny edge detector have proven to be fast and efficient in capturing edge features [6], which has 5 steps as follows:
- Apply Gaussian filter to smooth the frame to remove the noise.
- Find the intensity gradients of the frame.
- Apply non-maximum suppression to get rid of spurious response to edge detection.
- Apply double threshold to determine potential edges.
- Track edge by hysteresis and finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

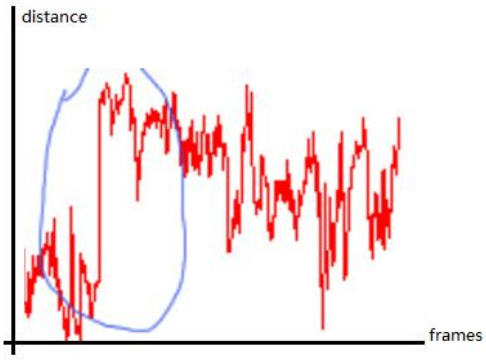

Fig. 3. Interested region by motion information

After canny detector, one frame can be expressed as a zero or a one in a pixel matrix.

B. Combined with spatial information

[7] shows that spatial pyramid is an effective way to combine spatial information into histogram or other features with no location details.

Let $u_i(x, y)$ is the pixel matrix obtained by canny detector as above for frame i, and by equally dividing each frame into 50 stripes, matrix $u_i$can be represented by:

$$u_i(x, y) = [u_i(x_{r_1}, y_{r_1}) \quad u_i(x_{r_2}, y_{r_2}) \cdots u_i(x_{r_{50}}, y_{r_{50}})] \quad (1)$$

Then the intersection of each stripe between successive frames can be calculated as

$$u_i(j) = u_i(x_{r_j}, y_{r_j}) \bigcap u_{i+1}(x_{r_j}, y_{r_j}) \quad (2)$$

Where *j* notes the stripe number

For each stripe *j*, we can calculate a number of the share edge between successive frames as Eq. (2). If we concatenate $u_i(j)$ for different *j*together, we can use this vector to express the motion information of this stripe to some extent, which is shown in Fig. 2.

After computing the Euclidean distance among vectors in Eq. (3), we can obtain the results as Fig. 3.

$$d_i = \sqrt{\sum_j [u_i(j) - u_{i+1}(j)]^2} \quad (3)$$

III. TRAJECTORY DISCONTINUITY

In above stage we can obtain the candidates of possible segmentation. In this stage, we use the trajectory information to verify the real segmentation of event. Since for long video sequence, LDOF is not the same effectiveness as the short one, in this part, we just compute the LDOF in the region of 20 frames centered by the segmentation candidates in former stage.

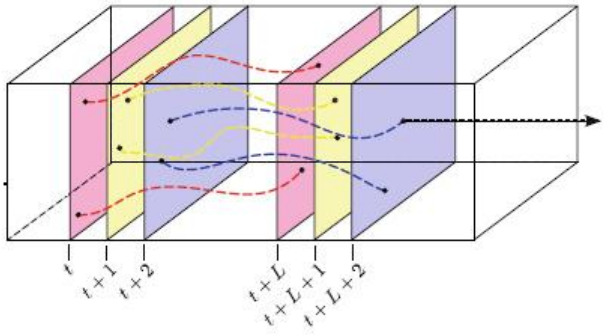

Fig. 4. Illustration of our approach to extract and characterize dense trajectories

A. LDOF

Motion is an intrinsic property of the world and an integral part of our visual experience. Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. We are interested in the extraction of motion features from the static/background part in video sequence, namely, LDOF to represent motion from background.

The main idea of LDOF is to estimate large displacements by minimizing the energy objective function, as $E(w)$ and $w := (u, v)^T$ is the optical flow we want. Suppose $X := (x, y)^T$ denotes a point in the image, this objective function is defined as:

$$E(\mathbf{w}) = E_{color}(\mathbf{w}) + \gamma E_{grad}(\mathbf{w}) + \alpha E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}) \quad (4)$$

where $\Psi(S^2) = \sqrt{S^2 + \epsilon^2}$.

Due to illumination effects, matching the color or gray value is not always reliable; in Eq. (4) it adds the constraints as smooth information, gradient information and match information, where each element is shown as follows.

$$E_{color}(\mathbf{w}) = \int \Psi(|I_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - I_1(\mathbf{x})|^2) d\mathbf{x} \quad (5)$$

$$E_{grad}(\mathbf{w}) = \int \Psi(|\Delta I_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - \Delta I_1(\mathbf{x})|^2) d\mathbf{x} \quad (6)$$

$$E_{smooth}(\mathbf{w}) = \int \Psi(|\Delta u(\mathbf{x})|^2 + |\Delta v(\mathbf{x})|^2) d\mathbf{x} \quad (7)$$

$$E_{match}(\mathbf{w}) = \int \rho(\mathbf{x}) \Psi(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2) d\mathbf{x} \quad (8)$$

In Eq. (8), $w_1(X)$ denotes the correspondence vectors obtained by descriptor matching at some points x.

B. Tracking Trajectory Discontinuity by LDOF

From LDOF in above subsection, we can obtain w matrix based on pixels. Then from the start frame, we can trace the point's location w matrix on frame by frame to obtain the corresponding trajectory. To handle the problem that trajectories tend to drift from their initial locations during the tracking process, we limit the length in 20 frames centered by the candidates obtained in the first stage.

Then trajectory tr_i is defined as

$$tr_n = \left\{ \left({x_n}^k, {y_n}^k, {t_n}^k \right) \quad k = 1 \ldots T_n \right\}, n = 1 \ldots N \quad (9)$$

Where T_n is the length of tr_n and N is the number of trajectories. The whole procedure is shown in Fig. 4. If no tracked point is found in a $W \times W$ neighborhood, the trajectory is ended. We just count the number of trajectories for each frame, and if there is a sudden drop for this number, we treat it as the trajectory discontinuity

IV. EXPERIMENT

A. Dataset

When conducting our research we found out that there is no open dataset with multi-event video sequences. Since there is no open dataset for video event segmentation, we composed it based on CCV videos database [8]. CCV has 4659 training videos and 4658 test videos from YouTube over 20 semantic categories. For videos in CCV, each one just represents one event. We select 1000 videos with less than 30 seconds each and randomly combine two or three videos together. We conducted our experiment on 500 combined videos. As for evaluation, we adopt recall rate and precision rate as follows to determine the system performances.

$$precisionrate(PR) = \frac{correctly\ detected\ points}{all\ points\ detected}$$
$$recallrate(RR) = \frac{correctly\ detected\ points}{actually\ real\ points} \quad (10)$$

B. Performance of motion feature in first stage

As seen in Fig.5, since the motion feature just compare the edge change in stripes between consecutive frames, there was a high false alarm in event segmentation. For instance, in Fig.5, the first seven frames (in fact, much more than seven and in order to show the result, we just list the frames nearby detected points) belong to the same event, in which the animals are playing in meadowland. While for only motion feature detection, it has several segmentation points when the background or foreground has some changing. From Table 1, it can be seen that although motion feature can give 100% recall rate for event boundary, the precision rate is too low.

Fig. 5. The different results of event detection by only motion feature (with black box) and motion feature combined with trajectory discontinuity (without black box)

TABLE I.

Different Approach	**PR**	**RR**
Only motion feature	50.5%	100%
Combined with trajectory discontinuity	98.3%	100%

Comparison of performance between trajectory discontinuity and without trajectory discontinuity

C. Performance of trajectory discontinuity in second stage

In this paper we used a fast GPU implementation of large displacement optical flow (LDOF), which is a parallel implementation version.In Fig. 5, when motion feature combined with trajectory discontinuity information, the false alarm problem is solved. Since for the animal in the meadowland we can find the tracing trajectory to show that it is a moving object, and we could not treat it as segmentation just because of the background changing or a new animal is adding. While when the event is totally varied, then the trajectory will be ended, and thus the number of trajectories will drop down dramatically. By tracing the number of trajectories in each frame, it is not hard to find the real segmentation based on event. The improvement of precision rate from 50.5% to 98.3% in Table 1 also proves the effectiveness of trajectory discontinuity information in event segmentation on videos. It shows the helpful information provided by moving object detection by trajectory tracing.

V. CONCLUSION

We have presented a method of separating events from a video sequence, which uses edge detection motion features to estimate the possible region where the event have changed in a video sequence, and then analyze this region using tracking trajectory discontinuity. Our experimental results demonstrate that this method is more efficient in terms of time and accuracy in segmenting events in a video sequence. For future works we would like to explorer further in classifying

ACKNOWLEDGMENT

I would like to thank my supervisor Professor Zhang Lei her generous support, and critic my research work in order to perfect the work. I would also like to thank the Harbin Engineering University Management for providing me with resources for the success of the research work. Last but not least I would like to thank my Wife, and Family for the emotional support.

REFERENCES

[1] C. F. Shu D. Swanberg and R. Jain, “Knowledge guided parsing in video database,” Storage and Retrieval for Image and video Databases, vol. 1908, San Jose, CA, 1993.

[2] A. Yoshitaka and M. Miyake, “Scene detection by audio-visual features,” IEEE International Conference on Mulatimedia and Expo (ICME01), 2001.

[3] Geng; Shi Jianbo Fragkiadaki, Katerina ; Zhang, “Video segmentation by tracing discontinuities in a trajectory embedding,” IEEE, 2012.

[4] D; Song M; Bu J; Chen C Liu, X; Tao, “Nearest neighbor-based label transfer for weakly supervised multiclass video segmentation,” CVPR 2014, 2014.

[5] T. Brox N. Sundaram and K. Keutzer, “Dense point trajectories by gpu-accelerated large displacement optical flow,” ECCV, 2010.

[6] Rohini Saxena Rashmi, Mukesh Kumar, “Algorithm and technique on various edge detection a survey,” Signal and Image Processing An International Journal, 2013.

[7] C.; Ponce Lazebnik, S.; Schmid, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR’06), 2006.

[8] Shih-Fu Chang Daniel Ellis Alexander C. Loui Yu-Gang Jiang, Guangnan Ye, “Consumer video understanding: A benchmark database and an evaluation of human and machine performance,” ACM International Conference on Multimedia Retrieval (ICMR), 2011.

[9] DavidMWPowers, “Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation,” Journal of Machine Learning Technologies, 2011.

[10] quasi-dense correspondes, “Comparison between large displacement optical flow algorithms,” 2014.