Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 3983 – 3993
© TÜBİTAK
doi:10.3906/elk-1608-308

A novel optical flow-based representation for temporal video segmentation

Samet AKPINAR*, Ferdanur ALPASLAN
Department of Computer Engineering, Faculty of Engineering, Middle East Technical University, Ankara, Turkey

Received: 28.08.2016 • **Accepted/Published Online:** 18.05.2017 • **Final Version:** 05.10.2017

Abstract: Temporal video segmentation is a field of multimedia research enabling us to temporally split video data into semantically coherent scenes. In order to develop methods challenging temporal video segmentation, detecting scene boundaries is one of the more widely used approaches. As a result, representation of temporal information becomes important. We propose a new temporal video segment representation to formalize video scenes as a sequence of temporal motion change information. The idea here is that some sort of change in the optical flow character determines motion change and cuts between consecutive scenes. The problem is eventually reduced to an optical flow-based cut detection problem from which the average motion vector concept is put forward. This concept is used for proposing a pixel-based representation enriched with a novel motion-based approach. Temporal video segment points are classified as cuts and noncuts according to the proposed video segment representation. Consequently, the proposed method and representation is applied to benchmark data sets and the results are compared to other state-of-the art methods.

Key words: Temporal video segmentation, optical flow, temporal video segment representation, average motion vector, cut detection

1. Introduction

Temporal video segmentation is a field of multimedia research enabling us to temporally split video data into semantically coherent scenes from a number of observations. The necessity of temporal video segmentation especially arises from the needs of video action detection, which is an issue of video information retrieval. In order to recognize video actions or events, extracting action candidate scenes for action classification plays an important role. Although there are methods of video action detection that use both temporal segmentation and action classification completely in one shot, temporal video segmentation is still a hot topic because of its contribution to action detection performances.

As a first step, we need to formally represent temporal video information for detecting scene boundaries in temporal video segmentation. Visual features, such as corners and visual interest points, of video frames are the basics for constructing our model. They are used for building a more sophisticated motion feature, namely optical flow. In this study, we propose a new temporal video segment representation that formalizes the video scenes as a sequence of temporal motion change information for detecting scene boundaries. The representation is fundamentally based on the optical flow vectors calculated for the frequently selected frames of a video scene. The idea is that some sort of change in the optical flow character determines the motion change and cuts between consecutive scenes.

The main contribution of this work is a proposed temporal video segment representation method that aims to be a generic model. The representation is based on an optical flow concept that spatially segments the

*Correspondence: samet@ceng.metu.edu.tr

video frames. Optical flow vectors are grouped according to this segmentation. We propose the novel concept of an average motion vector to specify the direction vector of each segmented part. In this way, the motion characteristic of the spatial frame segments is more specifically represented to describe scene cut behavior.

The outline of the paper is as follows. In Section 2, the related work is given. The optical flow concept is described in Section 3 and the proposed method is discussed in Section 4. In Section 5, the implementation of the method, experiments, and their results are presented. Finally, the conclusion is given in Section 6.

2. Related work

Temporal video representation and segmentation is an important issue of content-based video information retrieval methods and is defined as the extracting of scenes from a video instance. This can be done by using different approaches.

The representation methods of video information are widely elaborated on in the literature. The studies in [1,2] focus on the perception of the visual world and present facts about how to more philosophically detect visual features. Key-frame, bag-of-words, and motion-based approaches are the groups of methods reflecting the way of representation. Key-frame-based representation approaches focus on detecting the key frames in video segments in order to use them in classification. This kind of representation is used in [3,4] for video scene detection and video summarization. There are two important drawbacks of key-frame-based approaches. First, these approaches lack important motion information in videos. Second, ignoring all of the frames except the key-frames makes these approaches mostly inapplicable for detecting boundaries in temporal video segmentation problems. Another approach for representing temporal video information uses the bag-of-words concept. Histogram-based bag-of-words approaches represent the frames of video segments over a vocabulary of visual features. Examples of such approaches are found in [5,6]. While [5] conceptualizes temporal video information as bag-of-word vectors composed of histograms of visual features, [6] proposes a new motion feature, an expanded relative motion histogram of bag-of-visual-words. The most important disadvantage of the bag-of-words approach is the restricted nature of code words. Representing a visually rich frame with a label results in the loss of an important amount of information. The motion-based approach leverages motion features that are important in terms of their strong information content and robustness to spatiotemporal visual changes. Studies using motion features can be found in [7–10]. While [7] represents video motion by using image points with their trajectories, [8,9] present methods for representing video segments with a more complicated feature optical flow. The space-time interest point concept is proposed by [10]. Interest points are spatially defined in 2D and extended over time. With this extension, interest points change and have a 3D nature representing motion. Motion-based approaches are good candidates in terms of representing temporal video information as they can associate time with visual information in a descriptive and integrated way.

In our study, a motion-based representation is proposed to deal with the temporal video segmentation problem. Optical flow is the motion feature, integrating time with visual features, utilized for constituting the model.

It is also important to mention the temporal video segmentation methods in the literature. Temporal video segmentation is the problem of temporally splitting the video into coherent scenes. It generally originated from the needs of video segment classification. In order to semantically classify the video scenes as segments, they need to be extracted considering all video information. Temporal video segmentation methods tackle the problem from different points of view. Because we are dealing with visual feature-based segmentation, we analyze and group the methods accordingly. The methods in question include pixel difference-based, histogram comparison-based, edge-oriented, and motion-based methods [11].

Pixel difference-based methods use pixel intensities or color differences between the frames in order to characterize cuts between video scenes. A threshold-based automatic cut detection method is introduced in [12]. The method uses visual features for representing cut candidates and, according to these features, threshold values are estimated. Despite its simplicity and time efficiency, the most important drawback of the pixel difference-based approach is its sensitivity to motion. The histogram comparison-based method uses color histogram differences between frames. It is especially successful in cases that are independent of motion. The most important drawback of this approach results from the meaning of the histogram itself. Histogram similarity, in many cases, does not mean real similarity in the context. In some studies, including [13], this problem is tackled by extending the color histogram. The edge-oriented approach uses edge-related metrics, such as the edge change ratio and edge histogram, in video frames. This method deals with edge magnitudes and changes related with frames and their neighbors. In [14], a histogram-based edge tracking method is proposed for cut detection. Detected edges are followed and the cut property is estimated using changes in edge magnitudes. The drawback of the edge-oriented approach is that it is sensitive to high-speed motion. The motion-based approach uses motion features in video frames. This kind of approach is based on the fact that motion breaks define the cuts. Therefore, it focuses on motion detection and tracking. Methods based on scene change detection are used in [11,15,16]. Descriptive examples of motion-based temporal video segmentation approaches can be found in [15,16]. Camera motion analysis and optical flow estimations are used in these studies for cut detection by using scene changes. The author of [11] proposes novel methods for classifying cuts according to local visual features of video without using any threshold information. In [16,17] cut detection and action recognition methods based on optical flow features are used. An optical flow-based model using linear prediction is presented in [17], while [16] proposes a novel method for temporal segmentation of a video into scenes based on motion features including optical flow. The approach is based on the fact that optical flow changes differ at shot boundary points. In [18], a cut detection method using fuzzy rules is proposed. These rules are applied to the video frames having spatiotemporal features to detect cut boundaries. The most important drawback of these approaches is their computational cost and dependency on the extraction and tracking of motion features.

In our study, we propose a representation that allows us to enrich a pixel-based approach with a novel motion-based one. Optical flow is the motion feature used for describing a motion concept integrating the motion information to a pixel difference-based metric.

3. Optical flow

Optical flow is commonly defined as the apparent motion of brightness patterns in images in a video domain. An optical flow vector is defined along with a point (pixel) of a video frame. Selection of descriptive points is important in optical flow estimation. This is accomplished by using visual features. The calculation of optical flow vectors of the extracted features can be reduced to the following problem: "Given a set of points in a video frame, find the same points in another frame". Many algorithms according to different approaches have been proposed for optical flow estimation. According to [19], optical flow estimation algorithms can be grouped according to the theoretical approach used for interpreting optical flow. These are differential techniques, region-based matching, energy-based methods, and phase-based techniques.

Differential techniques utilize some sort of velocity estimation from spatial and temporal derivatives of image intensity [19]. They are based on the theoretical approach proposed by [20]. Well-known methods using differential techniques are found in [20,21]. Region-based matching is an alternative to differential techniques when differentiation is not suitable due to noise or a small number of frames [19]. In region-based matching,

concepts such as velocity and similarity are defined between image regions [22]. Energy-based methods are based on the output energy of filters tuned by velocity [19–23]. In phase-based techniques, which are different from energy-based methods, velocity is defined as the filter output having phase behavior. Examples of phase-based techniques using spatiotemporal filters can be found in [24,25].

4. Proposed method

In this study, an optical flow-based temporal video information representation is proposed for temporal video segmentation. Optical flow vectors need to be calculated for the selected sequential frames. Detection of features and estimation of optical flow according to these features are the main steps of optical flow estimation.

4.1. Optical flow estimation

In our approach, the Shi–Tomasi algorithm proposed in [26] is used for feature detection. The Shi–Tomasi algorithm is based on the Harris corner detector [27] and finds corners as interest points. The algorithm is especially robust for tracking. The Lucas–Kanade algorithm is selected [21] for estimation of the optical flow. The Lucas–Kanade algorithm is especially successful for videos with sufficient information and no noise. It works with the corners obtained from the Shi–Tomasi algorithm in our case. The following function should be minimized for each detected corner point, as seen in differential approaches:

$$\in (\delta x, \delta y) = I\,(x,\, y) - I\,(x + \delta x,\, y + \delta y) \tag{1}$$

According to our optical flow implementation, video frames are selected according to a frequency of 6 frames/s for 30 fps videos. The implementation is first applied to the "Hollywood Human Actions" data set [28]. Figure 1 shows the optical flow vectors estimated for the detected points in the video frame sequence.

Figure 1. Frame with optical flow vectors.

Optical flow vectors are calculated for every detected point in all frequently selected frames. The set of optical flow vectors is the temporal information source for our representation. The model below forms the backbone of our representation method. The optical flow vector set with an operator constructs the representation.

$$R = [S\,(V)\,,\, \Phi] \tag{2}$$

$S\,(V)$ is the set of optical flow vectors, while Φ is the descriptor operator. The operator defines the relation of the elements of the optical flow vector set of the frames.

4.2. Proposed representation

Temporal video segmentation is a direct field of interest in this study as the video scenes need to be extracted from the whole video. Therefore, segments are obtained by detecting the cuts between the scenes and our problem is reduced to a cut detection problem.

Cut detection is a commonly studied field in video information retrieval. Optical flow is an essential motion feature used in cut detection, as well as in temporal segment representation. The key study in terms of our research, [16], presents a cut detection method using optical flow. Temporal segmentation of a video into scenes is carried out based on motion features. The approach is based on the fact that optical flow changes differ at shot boundary points. The aim is to detect the outliers using optical flow magnitudes.

Optical flow is a key concept and behaves as an operator in the pixel difference calculations in our method. The fundamental idea here is that some sort of change in the optical flow character determines the cuts. More specifically, the hypothesis is constructed as the difference of intensity values between the pixels, mapped with optical flow vectors, changing in the consecutive frames at the cut points. Calculated optical flow vectors are used here as building block features. This was inspired by the idea in [29], which focused on operating on pixel difference calculations to represent scene changes.

The proposed method partially resides in the group of pixel difference-based approaches. However, as they have the drawback of being sensitive to motion, we solve this problem by strengthening the pixel-based cut detection methods with the motion-based ones. In our method, average motion vectors are calculated according to optical flow vectors. In addition, pixel matching based on these average vectors is carried out between consecutive frames. Pixel matching takes place for each frame transition, which forms the basis of our representation formalism. In this way, a binary cut classification method can be applied on this constructed set. In order to have a more descriptive feature vector, the frames are spatially divided into blocks and the mentioned calculations are carried out for blocks instead of frames. The method is also strengthened by some additional improvements, as explained in Section 5.

Eq. (2), which gives the optical flow-based generic representation, is adapted to cut detection. From this point of view, Φ is the operator that defines the relations between the optical flow vectors $S(V)$ and gives their meaning for representing cuts. The other parameters used in the model are explained in Table 1.

Table 1. Segment representation parameters.

Parameter	Definition
$P_{m,i}$	Spatial block m in ith frame
$S(V_{P_{m,i}})$	Set of optical flow vectors in block $P_{m,i}$
$V'_{P_{m,i}}$	Average motion vector in block $P_{m,i}$
$D_{P_{m,i}}$	Block distance of block $P_{m,i}$ with $P_{m,i+1}$
$px_j^{P_{m,i}}$	jth pixel in block $P_{m,i}$
$O(px_j^{P_{m,i}})$	Location of optical flow pixel $px_j^{P_{m,i}}$ in $(i+1)$th frame
$\mid S \mid$	Cardinality of set S
$I(px)$	Intensity value of pixel px

The first step in our method is to partition the frames into spatial locations. These locations hold visual features. They keep color information along with optical flow vectors when the pixels holding the optical flow vectors are grouped according to the spatial locations. The parameters defined in Table 1 are the basic building

blocks for constructing the model.

$$V'_{P_{m,i}} = \frac{\sum\limits_{v \in S(V_{P_{m,i}})} V\left(r, \angle\varphi\right)}{\left| S\left(V_{P_{m,i}}\right) \right|} C \tag{3}$$

The above representation, describing the average motion vector concept, calculates the sum of optical flow vectors in a block of the related frame. The sum is averaged by the total number of optical flow vectors in that partition. C is a threshold between 0 and 1 that is used for avoiding noise. It depends on the ratio shown in Eq. (4). If the ratio produces a result smaller than 1%, C is set to 0. Otherwise, the ratio is scaled between 90% and 100%.

$$\frac{\left| S\left(V_{P_{m,i}}\right) \right|}{\left| S\left(V\right) \right|} \tag{4}$$

The average motion vector is used in the pixel mapping formulation. A metric is needed for calculating the differences between the blocks of sequential frames. The Euclidean distance between frame blocks is used in our method. Assuming that each block of a video frame includes N pixels, the distance is calculated as follows:

$$D_{P_{m,i}} = \sqrt{\sum_{j=0}^{N-1} \left| I\left(px_j^{P_{m,i}}\right) - I\left(px_{j'}^{P_{m,i+1}}\right) \right|^2} \tag{5}$$

This calculation is based on the Euclidean distance between the intensity values of the mapping blocks of consecutive frames. The mapping block in the latter frame is constructed from the optical flow vectors of the related block in the former frame. The mapped pixel originating from pixel j of the former frame block is represented as $px_{j'}^{P_{m,i+1}}$ in the above representation. The mapping function for frame blocks is as follows:

$$px_{j'}^{P_{m,i+1}} = \begin{cases} O\left(px_j^{P_{m,i}}\right), & \text{if} px_j^{P_{m,i}} \text{ is a part of an optical flow vector} \\ px_j^{P_{m,i}} + V'_{P_{m,i}}, & \text{otherwise} \end{cases} \tag{6}$$

According to the above function and distance formula, distance values are expected to differ in cut points. The feature vector can then be represented by using the D values. First, distance is calculated for each block in the frame and combined for the vector representation. Assuming that the frames are partitioned into M spatial locations, the frame transition can be represented as:

$$F_{i,i+1} = \left[D_{P_{1,i}}, D_{P_{2,i}}, D_{P_{3,i}}, \ldots, D_{P_{M,i}}\right] \tag{7}$$

The vector set of the video is shown below, assuming that the video has W frames:

$$S = \{F_{1,2}, F_{2,3}, \ldots, F_{w-1,w}\} \tag{8}$$

Now the operator Φ in the generic optical flow-based representation model $R = \left[S\left(V\right), \Phi\right]$ is defined in Eq. (9). The model is updated with the intensity values as $R = \left[I, S\left(V\right), \Phi\right]$. *The* Φ operator maps pixel intensities I by the optical flow vector set $S\left(V\right)$ to the feature vector R in Eq. (9):

$$\Phi : \{I, S\left(V\right)\} \rightarrow R \tag{9}$$

4.3. Classification and experimental results

Temporal video segmentation leverages the vector representation proposed in Section 4. Support vector machines (SVMs) with a Gaussian radial basis function are used for nonlinear binary (cut/noncut) classification.

The Video Segmentation Project in the Carleton University data set [12] is used for evaluation. The set contains 10 different sets of video data. Color videos with motion and actions are selected rather than black/white and static videos such as news. The sets are named E, G, H, and J. The curves in the figures represent the values obtained from the collection of videos in these data sets.

First, the size of spatial partitioning for the frames is estimated. Here the contribution of the average motion vector concept is also shown by making a slight modification to the method. The modification is performed by removing the average motion vector concept from Eq. (6). The results of the experiments for determining the best value for partitioning and showing the effect of the average motion vector are shown in Figure 2.

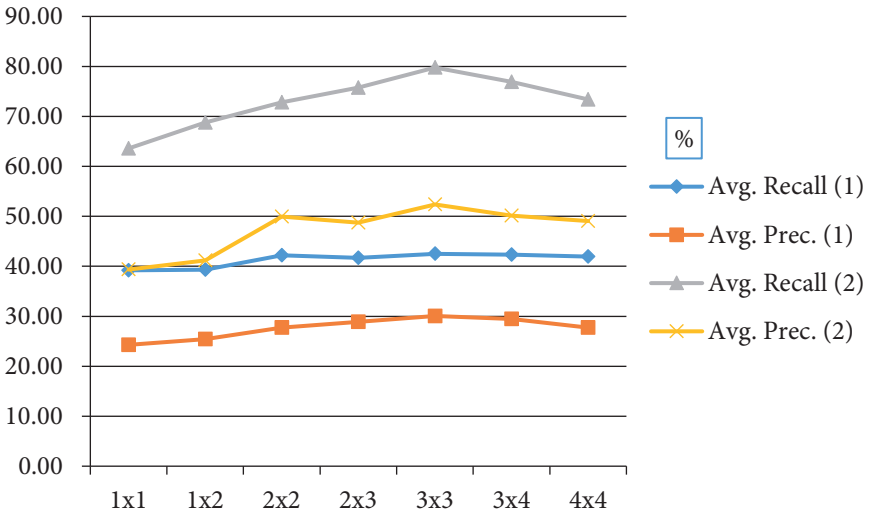

Figure 2. Partition size estimation in temporal video segmentation.

According to the figure, the optimal partition size is estimated as 3×3 regarding the cut detection recall/precision results. It can also be observed that the results of the proposed method (Eq. (2)) are far better than those of the modified one (Eq. (1)). Therefore, the contribution of the average motion vector is obvious. After selecting the partition size, the detailed experimental results of the proposed method with this size are found and compared to the ones obtained from the study in [12], as shown in Table 2.

Table 2. Comparison of the results of temporal video segmentation.

Set	Recall		Precision	
	Whitehead et al. [12]	Ours	Whitehead et al. [12]	Ours
E	100%	83.3%	93.8%	50.0%
G	94.4%	77.7%	81.0%	51.8%
H	89.5%	78.9%	89.5%	48.4%
J	89.7 %	79.3%	49.7%	57.5%

As seen in Table 2, both precision and recall rates are far different from the ones in [12]. According to our analysis of the vectors, the separation could not be implemented under these conditions. In order to make

the vector more descriptive, the sliding windows method is used. As the cut frame is a peak point compared to its neighboring frame transitions, neighboring frames should be used in the representation of the cut point. Sliding windows methods are widely used in many areas. Textual information extraction algorithms specifically utilize this approach. The main idea is that extracting the meaning of a word cannot be done only by using its meaning. Its context is also needed. The meaning of a word in a sentence needs to be extracted. While the word itself does not give the desired meaning, chunks, i.e. neighboring words, help extract the desired meaning by using sliding windows. Likewise, in cut frames, neighboring frames contribute to the meaning of the cut frame. The frame representation introduced by Eq. (7) in Section 4 is modified by using a window of size s, as follows:

$$F_i^{'} = [\ldots, F_{i-2,i-1}, F_{i-1,i}, F_{i,i+1}, F_{i+1,i+2}, \ldots] \tag{10}$$

The size of the new vector and the new frame set are as follows:

$$\left|F_i^{'}\right| = s \times |F_{i,i+1}| \tag{11}$$

$$S = \left\{F_1^{'}, F_2^{'}, \ldots, F_w^{'}\right\} \tag{12}$$

The experiments are carried out again with this new improvement of the method.

First, the size of the sliding window is estimated using the frame partition size of 3 × 3 in Figure 3. According to the figure, the optimal sliding window size is estimated as 4. The detailed results are summarized in Table 3.

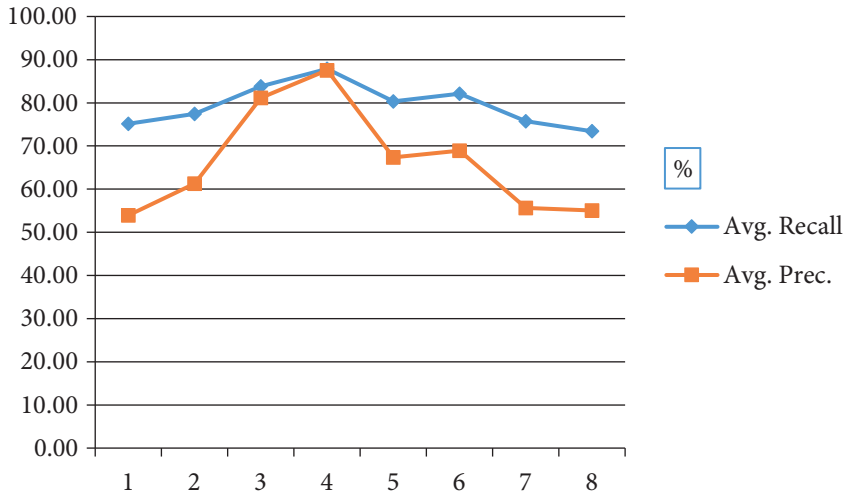

Figure 3. Estimation of sliding window size.

Satisfactory results were obtained with these new improvements. In [12], a feature-based cut detection method with automatic threshold selection is proposed. This comparison shows that our approach based on optical flow representation is more successful in terms of recall and precision in most cases.

We compared our method with other state-of-the-art methods using the same data set. The video with computer animations (J) is excluded to have a common context with the other methods. The study in [14] uses histogram-based edge tracking methods in cut detection, while [18] proposes a fuzzy rule-based approach that

Table 3. Comparison 1 of temporal video segmentation using sliding window.

Set	Recall		Precision	
	Whitehead et al. [12]	Ours	Whitehead et al. [12]	Ours
E	100%	100%	93.8%	96.6%
G	94.4%	94.4%	81.0%	94.4%
H	89.5%	92.0%	89.5%	90.2%
J	89.7 %	80.0%	49.7%	62.5%

tackles the cut detection problem. The authors in [16] introduce a dynamic threshold-based method using optical flow for cut detection. The results of the comparisons are shown in Table 4. It can be seen that our method produces better results than threshold oriented visual feature-based and edge histogram-based methods [12–14], respectively. This results from the fact that the motion effects are better handled by our method. The simple features used by thresholds and edge histograms cannot describe the cut nature as well as our motion-based representation, especially in the videos with motion. Moreover, the results are close to the ones found in [18]. It can be seen that the rules defining the domain knowledge in the fuzzy rule-based method positively affect the precision values. However, rule-based approaches make the methods more domain specific. Lastly, we compare our method with another optical flow-based method, i.e. that of [16], and generally obtain better results. This proves the success of the average motion vector concept with our descriptive representation formalism against dynamic thresholding with average optical flow vector magnitude. The complexities are similar as both of the methods include optical flow calculations and window-based analyses of optical flow vectors as the dominant cost operations. Our method is computationally more complex than the other methods, with the exception of [16], because of the optical flow vector calculations.

Table 4. Comparison 2 of temporal video segmentation using sliding window.

Set	Recall				
	Whitehead et al. [12]	MOCA [14]	Ours	Rogayeh et al. [18]	Kowdle et al. [16]
E	100%	81.0%	100%	92.3%	93.3%
H	89.5%		92.0%	92.5%	94.7%
G	94.4%	61.1%	94.4%	88.9%	88.9%
Set	Precision				
	Whitehead et al. [12]	MOCA [14]	Ours	Rogayeh et al. [18]	Kowdle et al. [16]
E	93.8%	95.3%	96.6%	85.7%	94.2%
H	89.5%		90.2%	100%	88.8%
G	81.0%	91.7%	94.4%	100%	90.4%

5. Conclusion

An optical flow-based approach is proposed in this study to represent temporal video information. This generic approach is applied to temporal video segmentation. The temporal video segmentation problem is converted to a cut detection problem in our work as the first phase of content-based video information retrieval. First, optical flow vectors are calculated. This representation is then used for cut detection.

In [8–24], cut detection methods based on the use of optical flow features are proposed. These are the key studies that influenced our work. The research in [24] presents an optical flow-based model using linear prediction. The authors in [8] propose a novel method for the temporal segmentation of a video into scenes based on motion features, including optical flow. The approach is based on the fact that optical flow changes

differ at shot boundary points. From the fact that optical flow changes differ at shot boundary points, an optical flow-based cut detection model is proposed. The average motion vector concept is put forward for video frames. Using this concept and optical flow vectors, consecutive frame differences are modeled. The model is then tested using a SVM-based binary classifier (cut/noncut). The results are compared to reference studies and the contribution of the model is shown.

The main feature of our method is the way it solves the temporal video representation problem in a video information retrieval domain. In addition to the successful results, this novel representation formalism also offers a solution for the dimensionality problem for video data. The classification is carried out using low-dimensional but descriptive vectors. The results show that our method surpasses the threshold-oriented visual feature-based and edge histogram-based methods while also competing with rule-based methods. While rule-based methods provide high success rates with explicit domain knowledge, they tend to be more domain-specific.

Future work will involve enhancing the optical flow features with audio and other visual features. We intend to make the method more robust by using these features. We also see that it is important to integrate the method with a scene classifier as temporal video segmentation is more meaningful with a video scene/event classifier. With further improvements, they can both behave as an event recognizer.

References

[1] Gibson JJ. The Perception of the Visual World. Boston, MA, USA: Houghton Mifflin, 1950.

[2] Royden CS, Moore KD. Use of speed cues in the detection of moving objects by moving observers. Vision Res 2012; 59: 17-24.

[3] Vasileios TC, Aristidis CL, Nikolaos PG. Scene detection in videos using shot clustering and sequence alignment. IEEE T Multimedia 2009; 11: 89-100.

[4] Gianluigi C, Raimondo S. An innovative algorithm for key frame extraction in video summarization. Journal of Real-Time Image Processing 2006; 1: 69-88.

[5] Ballan L, Bertini M, Bimbo AD, Serra G. Video event classification using string kernels. Multimed Tools Appl 2010; 48: 69-87.

[6] Wang F, Jiang YG, Ngo CW. Video event detection using motion relativity and visual relatedness. In: ACM Multimedia; 27–31 October 2008; Vancouver, BC, Canada. New York, NY, USA: IEEE. pp. 239-248.

[7] Sand P, Teller S. Particle video: long-range motion estimation using point trajectories. Int J Comput Vision 2008; 80: 72-91.

[8] Chaudry R, Ravichandran A, Hager G, Vidal R. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: Conference on Computer Vision and Pattern Recognition; 22–24 June 2009; Miami, FL, USA. New York, NY, USA: IEEE. pp. 1932-1939.

[9] Lertniphonphan K, Aramvith S. Human action recognition using direction histograms of optical flow. In: International Symposium on Communications and Information Technologies; 12–14 October 2011; Hangzhou, China. New York, NY, USA: IEEE. pp. 574-579.

[10] Laptev I, Lindeberg T. Space-time interest points. In: International Conference on Computer Vision; 13–16 October 2003; France. New York, NY, USA: IEEE. pp. 432-439.

[11] Barbu T. Novel automatic video cut detection technique using Gabor filtering. Comput Electr Eng 2009; 35: 712–721.

[12] Whitehead A, Bose J, Laganière R. Feature based cut detection with automatic threshold selection. In: International Conference on Image and Video Retrieval; 21–23 July 2004; Dublin, Ireland. New York, NY, USA: IEEE. pp. 410-418.

[13] Pass G, Zabih R. Comparing images using joint histograms. Multimedia Syst 1999; 7: 234-240.

[14] Effelsberg W. The MoCA Project–Movie Content Analysis Research at the University of Mannheim. Mannheim, Germany: Jahrestagunug University of Mannheim, 1998.

[15] Xiong W, Chung J, Lee M. Efficient scene change detection and camera motion annotation for video classification. Comput Vis Image Und 1998; 71: 166-181.

[16] Kowdle A, Chen T. Learning to segment a video to clips based on scene and camera motion. In: European Conference on Computer Vision; 7–13 October 2012; Firenze, Italy. New York, NY, USA: IEEE. pp. 272-286.

[17] Fatemi O, Zhang S, Panchanathan S. Optical flow based model for scene cut detection. In: Canadian Conference on Electrical and Computer Engineering; 26–28 May 1986; Calgary, AB, Canada. New York, NY, USA: IEEE. pp. 470-473.

[18] Roghayeh D, Hamidreza R, Rashidy K. AVCD-FRA: A novel solution to automatic video cut detection using fuzzy-rule-based approach. Computer Vis Image Und 2013; 117: 807-817.

[19] Barron J, Fleet D, Beauchemin S. Performance of optical ?ow techniques. Int J Comput Vision 1994; 12: 43-77.

[20] Horn KP, Schunck BG. Determining optical ?ow. Artif Intell 1981; 17: 185-203.

[21] Lucas B, Kanade T. An iterative image registration technique with an application to stereo vision. In: Proceedings of the Imaging Understanding Workshop; 1981. New York, NY, USA: IEEE. pp. 121-130.

[22] Anandan P. A computational framework and algorithm for the measurement of visual motion. Int J Comput Vision 1989; 2: 283-310.

[23] Heeger DJ. Optical flow using spatiotemporal filters. Int J Comput Vision 1988; 1: 279-302.

[24] Buxton B, Buxton H. Computation of optical flow from the motion of edge features in image sequences. Image Vision Comput 1984; 2: 59-74.

[25] Fleet DJ, Jepson AD. Computation of component image velocity from local phase information. Int J Comput Vision 1990; 5: 77-104.

[26] Shi J, Tomasi C. Good features to track. In: Conference on Computer Vision and Pattern Recognition; 21–23 June 1994; Seattle, WA, USA. New York, NY, USA: IEEE. pp. 593-600.

[27] Harris C, Stephens M. A combined corner and edge detector. In: 4th Alvey Vision Conference; 31 August–2 September 1988; Manchester, UK. New York, NY, USA: IEEE. pp. 147-151.

[28] Laptev I, Marszalek M, Schmid C, Rozenfeld B. Learning realistic human actions from movies. In: Conference on Computer Vision and Pattern Recognition; 23–28 June 2008; Anchorage, AK, USA. New York, NY, USA: IEEE. pp. 1-8.

[29] Akpınar S, Alpaslan FN. Optical flow-based representation for video action detection. In: Deligiannidis L, Arabnia H, editors. Emerging Trends in Image Processing, Computer Vision and Pattern Recognition. Boston, MA, USA: Morgan Kaufmann, 2014. pp. 331-351.