MULTI-VIEW SEMANTIC TEMPORAL VIDEO SEGMENTATION

Thomas Theodoridis*, *Anastasios Tefas** *and Ioannis Pitas**†

*Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
†Department of Electrical and Electronic Engineering, University of Bristol, UK

ABSTRACT

In this work, we propose a multi-view temporal video segmentation approach that employs a Gaussian scoring process for determining the best segmentation positions. By exploiting the semantic action information that the dense trajectories video description offers, this method can detect intra-shot actions as well, unlike shot boundary detection approaches. We compare the temporal segmentation results of the proposed method to both single-view and multi-view methods, and also compare the action recognition results obtained on ground truth video segments to the ones obtained on the proposed multi-view segments, on the IMPART multi-view action data set.

Index Terms— temporal video segmentation, action recognition, IMPART multi-view action data set

1. INTRODUCTION

Human action recognition has recently been a very active research area [1], spanning across many applications, such as human-computer interaction [2], daily action recognition for improving the quality of life of patients [3] and elderly people [4], content-based video retrieval [5], etc. However, until recently, the majority of research efforts were focused on the analysis of single-view video sequences. In part due to the decreased cost of video shooting equipment and in part due to computer hardware advances, a number of multi-view data sets and methods have appeared in the literature in the last few years [6, 7], thus allowing multi-view human action recognition.

Human action recognition in video sequences presents several challenges [8]. Variations in the action execution style among individuals can be substantial. Each person has a unique style of execution, which may, in part, be due to physiological reasons (e.g., the pace of a short person walking will be different to that of a tall person). The speed of action execution can also be another source of variation, as the same action may be perceived differently when executed fast or slowly. Furthermore, the video recording set-up and the scene content during filming can influence action recognition performance, as actions viewed from various perspectives can appear quite different; this also applies to lightning conditions. View occlusion and background video content can negatively impact the action recognition algorithm as well. Multi-view methods have the potential to overcome some of these problems by creating more robust representations of the performed actions.

An important component of the action recognition process is detecting a set of desired actions within videos that contain multiple action segments. Various techniques have been proposed in the literature to this end [1]. Each video either has to be temporally segmented by an unsupervised algorithm into sub-sequences that depict single actions, or the action recognition algorithm must be applied repeatedly to many consecutive sub-sequences of the video, in order to detect the actions. In this work, we propose a multi-view temporal segmentation method, denoted by mv-GS, that is based on the single-view algorithm presented in [9], which uses a Gaussian scoring system for determining the best segmentation positions. To showcase the effectiveness of the proposed method, we compare it against the mean label multi-view method [9], denoted by mv-ML, and the single-view algorithm that is based upon. We also present multi-view action recognition results, both on ground truth video segments and on segments produced by the proposed method, on the IMPART multi-view action data set [10].

The rest of our work is structured as follows. Section 2 describes the single-view temporal segmentation and action recognition algorithms, while the multi-view approaches are discussed in Section 3. Section 4 presents the IMPART data set, the experimental set-up and results. Finally, conclusions are discussed in Section 5.

2. SINGLE-VIEW METHODS

2.1. Temporal Segmentation

The temporal video segmentation method employs the dense trajectories video description [11], which achieved state-of-the-art results in action recognition on various data sets. After calculating the description for the entire video in question,

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). The European Union is not liable for any use that may be made of the information contained therein.

each descriptor is assigned to a visual word using k-means clustering [12]. Subsequently, bag-of-visual-words representations of successive, overlapping frame sequences are generated. An iterative algorithm, taking into account these representations, searches for the best segmentation position of the input video by minimizing the Fisher ratio [13]. Each time the optimal cut position is found, the same process continues for the two resulting video segments. The stopping criterion of the algorithm is based on the video segment length. If the video segment to be further segmented becomes less than a certain number of m_0 frames, the process terminates.

2.2. Action Recognition

The recognition process is based on the dense trajectories video descriptors as well. During the training phase, dense trajectories descriptors are calculated for video segments that depict the desired actions. Each video segment contains only a single action (e.g., *hand-waving*). A codebook of visual-words is generated, from a random subset of the calculated descriptors, through k-means clustering. Using this codebook, a bag-of-visual-words representation is obtained for each video segment. The classifier is trained using the training video segment action labels and a kernel matrix, which contains the distances among the bag-of-words representations of all the actions. During testing, the input video is segmented by the temporal segmentation method described previously. Dense trajectories descriptors are calculated for each video segment. Using the codebook generated during the training phase, corresponding representations are computed for each video segment. The distance between the representation of each segment and the representations calculated in training are given as input to the classifier, which produces a label for the recognized action. In the course of our experiments, we have used a feed-forward neural network as our classifier. However, special configuration and training algorithm were employed, which were shown to improve classification performance [14].

3. MULTI-VIEW METHODS

3.1. Temporal Segmentation

By applying the single-view temporal segmentation algorithm to each individual video of a n-camera set-up, we end up with n distinct single-view segmentations $S_1, \ldots, S_n$. If nc_i is the number of cut positions for video i, each segmentation $S_i = \{s_i^j,\ j = 1, \ldots, nc_i + 1\} = \{c_1^i, \ldots, c_{nc_i}^i, m\}$ is a list of frame numbers indicating the cut positions, plus an additional element m, which is the index of the last video frame.

The individual segmentations $S_i,\ i = 1, \ldots, n$, that are given as input to the multi-view methods, need not have the same value of nc_i in order to be combined, as both multi-view methods can handle segmentations with different number of cut positions. However, the mv-ML algorithm is influenced by nc_i regarding the number of cut positions that it produces, while the mv-GS can be adjusted to produce the desired amount.

3.1.1. *Temporal Segmentation Method mv-ML*

This method translates the segmentation sets $S_i,\ i = 1, \ldots, n$ into sets of frame labels $L_i = \{l_i^k,\ k = 1, \ldots, m\}$. Each video frame receives a label which is determined by the formula:

$$l_i^k = \operatorname*{argmin}_p (k - 1 \leq s_i^p), \quad p \in [1,\ nc_i + 1] \tag{1}$$

In other words, label k is the segment index that the corresponding frame of segmentation S_i belongs to. The multi-view labels $L_{MV} = \{l_{MV}^k,\ k = 1, \ldots, m\}$ are produced by calculating the mean frame label, across all n label sets, for each position k:

$$L_{MV} = \left\{ \left\lfloor \frac{1}{n} \sum_{i=1}^{n} l_i^k \right\rfloor,\ k = 1, \ldots, m \right\}, \tag{2}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Figure 1 illustrates this method more clearly.

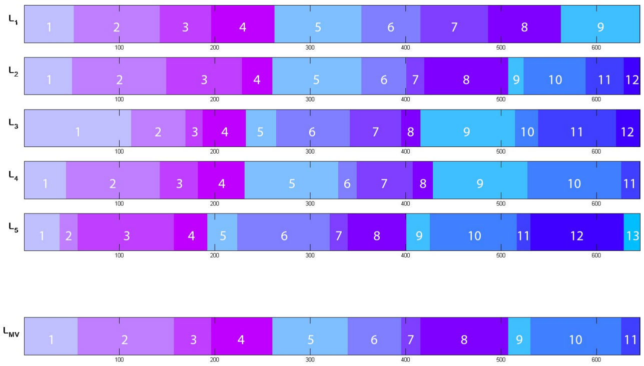

Fig. 1. An illustration of five label sets $L_1, \ldots, L_5$ and the multi-view labels L_{MV} as determined by the mv-ML method. The horizontal axis represents frame numbers.

3.1.2. *Temporal Segmentation Method mv-GS*

The proposed multi-view segmentation method employs a scoring system in order to determine the best positions for segmenting the captured footage. A set of score values $SV_i = \{sv_i^k,\ k = 1, \ldots, m\}$ is derived from each segmentation S_i, by scoring the positions around the cut frames according to a Gaussian probability density function. For each cut position c within segmentation $S_i^{'} = S_i \setminus \{m\}$, a Gaussian centered at c, with zero mean and variance σ^2, determines the scores of the surrounding positions. Without loss of generality, we assume that only positions within an area of

α frames away from c receive a score, whereas the rest have zero values. As will become more clear when we describe the subsequent steps of the algorithm, this does not affect the overall result as low scores are discarded anyway. More formally, given an integer $\alpha \geq 1$ that controls the scoring span and a sigma value $\sigma > 0$, the scores are derived from the formula:

$$sv_i^k = \begin{cases} e^{-\frac{(k-c)^2}{2\sigma^2}}/\sigma\sqrt{2\pi}, & \exists\, c \in S_i^{'} : |k-c| \leq \alpha \\ \quad 0 \quad\quad , & otherwise \end{cases} \tag{3}$$

If a certain position k is close to more than one cut frame c, the different scores are combined.

The next step of the algorithm is summing up the individual score values: $SV_{MV}^{+} = \{\sum_{i=1}^{n} sv_i^k,\ k = 1, \ldots, m\}$. Then, using a thresholding procedure, we zero out all values smaller than a chosen threshold sv_0. The remaining significant scores are stored into a new set SV_{MV}^{S}. Finally, in order to determine the best cut positions from the remaining values, we employ a sliding window approach over the values of SV_{MV}^{S}. Inside this window of length $2\alpha + 1$, all values below the maximum therein are zeroed out. If the maximum appears at more than one position, their median position is chosen as the cut point. By sliding the window one position to the right at a time and repeating this procedure, we end up with SV_{MV} that contains the multi-view cut positions. Figure 2 provides an illustration of this process.

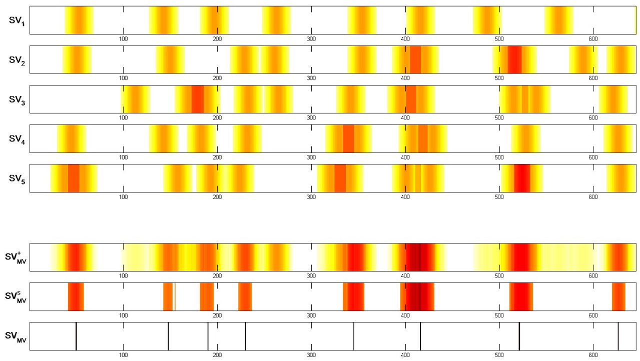

Fig. 2. A depiction of the score values corresponding to five cameras $SV_1, \ldots, SV_5$, the combined scores SV_{MV}^{+}, the remaining significant scores after thresholding SV_{MV}^{S} and the final cut positions of the algorithm SV_{MV}. Darker areas indicate higher scores. The horizontal axis represents frame numbers.

3.2. Action Recognition

In correspondence to the temporal segmentation case, a set of action recognition labels $R_i = \{r_i^k,\ k = 1, \ldots, m\}$ is obtained for each camera, by applying the single-view algorithm discussed in Section II. Each label r_i^k indicates the recognized action at frame k from camera i. Our multi-view fusion approach consists of calculating the label with the highest frequency at each position k. However, since the label $other$ represents a group of actions we are not interested in recognizing, the next most frequently appearing label is chosen in such cases, provided that its frequency is at least f_0. Let us denote by $Q_k = \{r_1^k, \ldots, r_n^k\}$ the set of recognition labels for frame k, by $r_k = mode(Q_k \setminus \{other\})$ the mode label for each position when excluding the label $other$ and by f_{r_k} the frequency of label r_k within $Q_k \setminus \{other\}$. The multi-view recognition labels $R_{MV} = \{r_{MV}^k,\ k = 1, \ldots, m\}$ are given by the formula:

$$r_{MV}^k = \begin{cases} \quad r_k \quad , & f_{r_k} \geq f_0 \\ other, & otherwise \end{cases} \tag{4}$$

4. EXPERIMENTAL RESULTS

4.1. IMPART data set

The IMPART multi-view action data set [10] was filmed using a high definition multi-camera set-up in two different locations: one indoors and one outdoors. The indoor filming set-up consists of 12 cameras placed around and at the ceiling of a room with every-day objects, capturing three non-professional actors that perform the actions $walk$, $handwave$, run and $other$, where the category $other$ contains distraction actions, such as $jump\ in\ place$, $jump\ forward$ and $open/close\ door$. The outdoor set-up consists of 10 cameras, placed in a 180° arc configuration, capturing four actors that perform the same actions, plus another distraction action, $bend\ forward$. Also, it has a dynamic background of moving people and objects. One script was drafted for each shooting location, which was executed successively by all actors. Furthermore, three different sessions were filmed for each script, which contain slight variations among them. In total, 30 videos with an average length of 5,492 frames were recorded for the outdoor set-up and 36 videos of 3,592 frames on average for the indoor set-up. Figure 3 showcases some of the performed actions.

4.2. Parameter Values

Starting off with the single-view methods, the dense trajectories for the segmentation algorithm were calculated with the same parameters proposed in [11]. The codebook cardinality was set to 100 words and the length criterion for terminating the process was set to $m_0 = 125$ frames. In the action recognition case, dense trajectories were calculated as before, with the difference that the trajectory length was set to 7 frames. The codebook cardinality during training was set to 2,000 words. The algorithm was trained on video segments from the Hollywood2 [15], Hollywood3D [16], i3DPost [17], IXMAS [18] and previous IMPART [19] databases; no videos

Fig. 3. Sample frames from the IMPART outdoor and indoor capture sessions.

of the new data set were used in training. Footage from one indoor camera, which was located at the ceiling, was not used for recognition, as the algorithm was not trained on this viewing angle and produced low quality results.

Regarding the multi-view temporal segmentation algorithm mv-GS, the timespan area α was set to 23 frames, σ was set to 6 and the threshold sv_0 for keeping the significant values was set to the 75th percentile of the non-zero score values. The threshold frequency for multi-view recognition was set to $f_0 = 3$.

A generalization of the Temporal Segmentation Accuracy (TSA) metric [20] was used for measuring the temporal segmentation performance. Given the ground truth segmentation G and the produced segmentation S, the TSA metric is given by:

$$TSA = \frac{2}{|G| + |S|} \sum_{i=1}^{|G|} \sum_{j=1}^{|S|} \frac{|G(i) \cap S(j)|}{|G(i) \cup S(j)|} \qquad (5)$$

For measuring the action recognition performance, we computed the F-measure for each of the three main actions and took the average value over the 6 sessions (3 sessions in each set-up), in order to produce the final score. The video segments in the rec-GT case were manually produced by annotators, whereas in the rec-MV case were generated by the proposed mv-GS method.

4.3. Video Segmentation and Recognition Results

The video segmentation results of the single-view method, as well as the two multi-view ones, are presented for each of the six sessions in Table 1. The column for the single-view approach contains the average (best) score across all cameras.

We can see that the mv-ML method performs better than the average single-view result in 5 out of 6 cases, with a performance difference between -0.68% and 3.74%, and scores higher than the best in 2 out of 6 cases, having a difference of -3.89% in the worst case and 1.3% in the best. The proposed multi-view method mv-GS scores higher than the average as well as the best single-view result in all sessions. The biggest gain with respect to the average single-view result was 9.93% and the lowest was 4.25%, while the performance gain compared to the best result varied between 0.07% and 7.21%. These results indicate that there are substantial gains to be realized by combining the information of multiple cameras. Finally, the gain of the proposed method with respect to the mv-ML method was between 0.78% and 10.15%, which shows the effectiveness of the proposed scoring approach.

The multi-view action recognition results on ground truth video segments (rec-GT) and on segments produced by the proposed mv-GS multi-view method (rec-MV) are presented in Table 2. The action *run* was the easiest to recognize in both cases, with scores 98.33% and 97.16% respectively, while *hand-wave* was the most challenging to recognize with the proposed segmentation. Overall, both approaches produced a high average F-measure score, with rec-GT achieving 96.37% and rec-MV 92.86%.

Table 1. The temporal segmentation accuracy of the single-view, mv-ML and mv-GS methods on the IMPART data set.

		Single-view (%)	mv-ML (%)	mv-GS (%)
IMPART Indoor	Session 1	66.14 (68.40)	65.46	75.61
	Session 2	70.22 (74.69)	70.80	77.96
	Session 3	69.89 (72.89)	70.85	79.82
IMPART Outdoor	Session 1	72.92 (75.01)	75.86	79.54
	Session 2	73.83 (78.01)	74.43	78.08
	Session 3	70.75 (73.19)	74.49	75.27
Average		70.63 (73.70)	71.98	77.71

Table 2. The recognition results per action on the IMPART data set.

	rec-GT (%)	rec-MV (%)
run	98.33	97.16
walk	95.08	91.61
hand-wave	95.69	89.80
Average	96.37	92.86

5. CONCLUSION

In this work, we have proposed a multi-view video segmentation approach which managed to outperform the single-view as well as the mv-ML multi-view approach in all cases when tested on the IMPART data set. Using ground truth video segments and segments produced by the proposed method, multi-view action recognition results were computed on the IMPART data set. The small performance difference in action recognition between the two cases indicates the effectiveness of the proposed multi-view segmentation approach.

6. REFERENCES

[1] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[2] V. Pavlovic, R. Sharma, and T.S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997.

[3] A. Ghali, A.S. Cunningham, and T.P. Pridmore, "Object and event recognition for stroke rehabilitation," *Visual Communications and Image Processing*, pp. 980–989, 2003.

[4] H. Foroughi, B.S. Aski, and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," *11th International Conference on Computer and Information Technology (ICCIT-2008)*, pp. 219–224, 2008.

[5] H.J. Zhang, J. Wu, D. Zhong, and S.W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern recognition*, vol. 30, no. 4, pp. 643–658, 1997.

[6] M.B. Holte, C. Tran, M.M. Trivedi, and T.B. Moeslund, "Human action recognition using multiple views: a comparative perspective on recent developments," *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pp. 47–52, 2011.

[7] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view human action recognition: A survey," *Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 522–525, 2013.

[8] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[9] N. Kourous, A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Video characterization based on activity clustering," *International Conference on Electrical and Computer Engineering (ICECE-2014)*, pp. 266–269, 2014.

[10] "IMPART multi-view action data set," `http://cvssp.org/impart/`.

[11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2011)*, pp. 3169–3176, 2011.

[12] O.R. Duda, E.P. Hart, and G.D. Stork, *Pattern classification. 2nd Edition.*, John Wiley & Sons, 2001.

[13] G. McLachlan, *Discriminant analysis and statistical pattern recognition*, vol. 544, John Wiley & Sons, 2004.

[14] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, 2015.

[15] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2009)*, pp. 2929–2936, 2009.

[16] S. Hadfield and R. Bowden, "Hollywood 3d: Recognizing actions in 3d natural scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2013)*, pp. 3398–3405, 2013.

[17] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," *Conference for Visual Media Production (CVMP-2009)*, pp. 159–168, 2009.

[18] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.

[19] H. Kim and A. Hilton, "Influence of colour and feature geometry on multi-modal 3d point clouds data registration," *2nd International Conference on 3D Vision (3DV-2014)*, vol. 1, pp. 202–209, 2014.

[20] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.