Skeleton-based temporal segmentation of human activities from video sequences

Sébastien Quirion Alexandra Branzan-Albu Robert Bergevin
Computer Vision and Systems Laboratory
Dept of Electrical and Computer Engineering
Laval Univ., Ste-Foy, Qc, Canada, G1K 7P4
{squirion,branzan,bergevin}@gel.ulaval.ca

ABSTRACT

This paper presents a new multi-step, skeleton-based approach for the temporal segmentation of human activities from video sequences. Several signals are first extracted from a skeleton sequence. These signals are then segmented individually to localize their cyclic segments. Finally, all individual segmentations are merged with respect to the global set of signals. Our approach requires no prior knowledge on human activities and can use any generic stick-model. Two different techniques for signal segmentation and for the fusion of the individual segmentations are proposed and tested on a database of fifteen video sequences of variable level of complexity.

Keywords

Video Understanding, Motion Analysis, Periodicity Analysis, Signal Segmentation, Data Fusion.

1. INTRODUCTION

Human activity description and recognition is currently an active research area in computer vision. As in [Pol94a], cyclic activities are defined as regularly repeating sequences of motion events. Among others, [Pol94a] and [Bob01a] have proposed new methods for the description and recognition of such activities. These methods are based on the assumption that each video sequence contains exactly one activity. However, activity recognition systems should be able to handle sequences containing several cyclic and non-cyclic activities.

Our research aims at extending the applicability of such algorithms by extracting activities from video sequences through temporal segmentation. The proposed approach uses skeleton sequences to represent the human motion and divides the segmentation task into three sequential steps. Our algorithms are based on periodicity analysis.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

The use of a skeleton[1] model enables us to accurately follow the periodic motion of a human subject despite the periodic motion present in the background. Another advantage of this high-level motion description is the ability to detect an activity by using anatomical information.

Related Work

Few relevant papers have been published on the temporal segmentation of human activities. Among the few, [Yaz04a] describes a temporal segmentation method of symmetric activities based on a 2D inter-frame similarity plot and requires no prior knowledge on human activities.

2. PROPOSED APPROACH

The proposed approach is divided in three steps. In the first step, signals are *extracted* from a skeleton sequence in order to obtain simple but significant data to work with. In the second step, these signals are *segmented individually* in order to localize their cyclic segments. In the final step, all individual segmentations are coherently *merged* with respect to the entire set of signals. This approach is highly modular and allows for different implementations depending on the priorities of the task at hand.

[1] A skeleton is a generic graph where each node has spatial coordinates. These coordinates are changing over time, thus forming a skeleton sequence.

Signal Extraction

Using signals to represent significant attributes of an activity (e.g. movement of a skeleton node, variation of a joint angle, etc.) is an interesting choice as one can expect such attributes to exhibit cyclic behavior during activities. As a general rule, every node or edge of the stick-model subject to independent and significant motion (e.g. hands, feet, etc.) is to be represented by at least one signal. These signals are needed for the analysis of relative cyclic movements (e.g. waving an arm). Also, at least one global attribute, like the instantaneous speed of the centroid of all nodes, is to belong to the signal-set. This global feature is needed for the analysis of global cyclic movements (e.g. jumping up and down). The actual set of signals depends on the stick-model used.

Individual Signal Segmentation

A straightforward approach to implementing a signal segmentation algorithm is based on periodicity analysis. This approach computes a periodicity score to rate different possible segments in the signal and uses these ratings to select relevant segments. Other approaches are also possible. For instance, an effective way to reduce the complexity of the problem is to first form a rough segmentation by removing the 'silences' (i.e. portions of very low amplitude) in the signal. A more informed approach based on periodicity analysis can then be applied to the resulting segments to refine this rough segmentation.

2.1.1 Periodicity Score

The periodicity score rates the periodicity of a signal S in the interval [0,1]. The periodicity scoring algorithm used in this paper is a 1-D adaptation of the lattice matching technique presented in [Cut00a].

Considering A_S, the normalized mean-removed autocorrelation of the signal S, M_S, the ordered set of maxima found in A_S, and c_S, the estimated cycle-length (i.e. the mean length between consecutive autocorrelation maxima), the expression of the score is:

$$\Psi_S = \frac{1}{|M_S|-1}\sum_{i=2}^{|M_S|}\left(1-\frac{|c_S\cdot(i-1)-M_S(i)|}{c_S}\right)\cdot A_S(M_S(i)) \qquad (1)$$

This score is a measure of proximity in lag and value between actual maxima of A_S and expected maxima for a perfectly periodic signal of period c_S. The score of a periodic signal is equal to one and decreases as the signal becomes less and less cyclic. The score may be negative for degenerate cases.

For increased robustness, an adjusted score Ψ_S' can be computed in the same fashion using only the first 90% of the mean-filtered values of A_S. Moreover, as long cyclic segments are preferred over smaller ones, the length of the segment (i,j) can be normalized by the length l_S of the whole given signal S. To favor length only in cyclic segments, a threshold η is needed to define what is considered cyclic and what is not. The length-normalized score is computed as follows:

$$\Upsilon_{S(i,j)} = \eta^{1-\left(\frac{(j-i+1)}{l_S}\right)}\cdot\Psi_{S(i,j)}{}'^{\frac{(j-i+1)}{l_S}} \qquad (2)$$

The length-normalized score properties can be summarized in a few key statements:

- As the length of the segment (i,j) approaches l_S, $\Upsilon_{S(i,j)}$ approaches $\Psi_{S(i,j)}'$;
- As the length of the segment (i,j) approaches 0, $\Upsilon_{S(i,j)}$ approaches η ;
- $(\Psi_{S(i,j)}' < \eta) \Leftrightarrow (\Upsilon_{S(i,j)} < \eta)$;
- $(\Psi_{S(i,j)}' > \eta) \Leftrightarrow (\Upsilon_{S(i,j)} > \eta)$.

These statements imply that length improves the score of a cyclic segment (i.e. a segment (i,j) with $(\Psi_{S(i,j)}' > \eta)$) but decreases the score of a non-cyclic segment.

2.1.2 Segmentation

The proposed algorithm first detects the 'most cyclic' segment in a given signal using a ***bestSegment*** algorithm. It then validates if the segment is cyclic enough. If its periodicity score is above a threshold τ_L, it is included in the segmentation set. The algorithm then proceeds in the same fashion on the remaining portions of the signal until the analysis of the entire signal is completed. With the exception of ***bestSegment***, the remainder of the implementation is rather simple.

A first implementation of the ***bestSegment*** algorithm simply computes the length-normalized score for every possible segment (i,j) and returns the segment with the maximum score. The segmentation algorithm is called ***MaxS*** when it uses this implementation of ***bestSegment***.

A second implementation seeks the segment with the best length-normalized score through numerical optimization[2]. This approach is based on the fact that the length-normalized score increases as a segment grows within a cyclic portion of a signal, and it decreases as this segment grows out of the cyclic portion. The segmentation algorithm is called ***OptS*** when it uses this implementation of ***bestSegment***.

[2] Our implementation uses the DHC algorithm [Yur94a].

Segmentations Fusion

In this last step, the idea is to use the segments detected on each individual signal as candidates for the global segmentation. The score threshold τ_L for the signal segmentation should be high in order to minimize the number of false detections. The set of candidates therefore contains temporal segments during which there is good reason to believe an activity occurs. To create a robust segmentation, the general idea is to compute a global score for each candidate and keep the highest scoring non-overlapping subset of candidates.

In the presented implementations, the global score of a segment is the sum of the scores greater than a threshold τ_G obtained on each signal in the signal-set.

SimF is a straightforward fusion algorithm. It iteratively removes the highest scoring candidate from the initial set, adds it to the final segmentation and removes from the set of candidates any remaining candidate that overlaps the chosen candidate.

A second fusion algorithm, ***GenF***, is an alternative of the one presented above. Instead of discarding overlapping candidates entirely, ***GenF*** discards the overlapping portion of the candidates. The remainders are being considered as candidates if their periodicity score is greater than τ_L on at least one signal.

3. EXPERIMENTAL RESULTS

Fifteen test sequences were captured in front of a static background using a monocular camera at 30fps. A skeleton was then adjusted at each frame using the single-frame extraction algorithm presented in [Vig03a]. The resulting unfiltered skeleton sequences are typically noisy and represent input data for the approach. The duration of these test sequences ranges from 300 to 1200 frames. The activities in the sequences are cyclic, articulated motions such as arm waving, side-stepping, etc. Nine signals were extracted from each skeleton sequence to form the corresponding signal-set: four angle signals (shoulders and hips), four vertical position signals (hands and feet) and the instantaneous speed of the centroid of all nodes.

The test sequences are partitioned according to their complexity. The sequences of type A are sequences where all the activities are temporally bounded by pauses. The sequences of type B are sequences where at least one activity is temporally adjacent to another activity or to non-cyclic movements. Finally, the sequences of type C are sequences where at least one activity fuses with another, like waving an arm immediately followed by waving the two arms.

The two fusers (***SimF*** and ***GenF***) were first confronted using ***MaxS*** in both cases. The most promising fuser was then used to compare the two segmentation algorithms.

In all performed tests, $\tau_L = 0.8$ was used as the lower limit on the periodicity score for signal segmentation, $\tau_G = 0.5$ as the lower limit on the periodicity score for global score computing and $\eta = 0.5$ as the lower limit on the periodicity score for length-normalized score computation.

The result tables provide two different validation measures. The first measure indicates the number of activities in the reference segmentation that were correctly detected[3] by the program and the number of false detections made (i.e. the total number of activities detected minus the number of correct detections). These numbers can be compared to the number of activities present in the reference segmentation, noted on the right of the sequence identifier. The second measure is a similarity score between the reference segmentation (**R**) and the segmentation obtained by the program (**F**). In short, it represents how well the proposed approach detects all the activities in their *totality*. This score is in the interval [0,1], 1 being a perfect match, and is computed in the following fashion:

1. Considering segments as sets of frames, a matching score for each pair (**r**,**f**) in **R**×**F** is computed as the cardinal of their intersection divided by the cardinal of their union;
2. For each **r**, the (**r**,**f**) pair with the maximum matching score is noted. The value of segment **f** is set to the score of (**r**,**f**) divided by |**R**|;
3. The value of all other segments in **F** is set to a penalty value of $-1/(2\cdot|\mathbf{R}|)$ for false detection;
4. The values computed in step 2 and 3 are summed. A negative score is set to 0.

The reference segmentation in these two measures is the mean of manual segmentations performed by ten volunteers. Considering the small standard deviation in these different segmentations, which was always smaller than 6 frames, the mean of segmentations represents a robust estimated ground truth.

Table 1 presents the results of ***MaxS-SimF*** and ***MaxS-GenF*** on all of the test sequences. It can be observed that both combinations make very few false detections. Also, in both cases, in sequences of type A and B, activities are almost always detected in their totality as their high scores show. The

[3] If over 90% of the length of a segment overlaps a reference segment, the activity is correctly detected.

advantage of using ***GenF*** rather than ***SimF*** is observed with the scores on sequences of type C.

Sequence (number of activities)	***MaxS - SimF***		***MaxS - GenF***	
	Detections (good:bad)	Score	Detections (good:bad)	Score
A_1 (2)	(2 : 0)	0.98	(2 : 0)	0.98
A_2 (4)	(4 : 0)	0.92	(4 : 0)	0.92
A_3 (3)	(3 : 0)	0.92	(3 : 0)	0.92
A_4 (3)	(3 : 0)	0.92	(3 : 0)	0.92
A_5 (2)	(1 : 1)	0.73	(1 : 1)	0.73
B_1 (5)	(5 : 0)	0.74	(5 : 0)	0.74
B_2 (2)	(2 : 0)	0.93	(2 : 0)	0.93
B_3 (4)	(3 : 1)	0.81	(3 : 1)	0.81
B_4 (3)	(3 : 0)	0.86	(3 : 0)	0.86
B_5 (4)	(3 : 0)	0.69	(4 : 0)	0.87
C_1 (2)	(1 : 0)	0.41	(1 : 1)	0.85
C_2 (5)	(2 : 0)	0.32	(2 : 0)	0.32
C_3 (5)	(1 : 1)	0.36	(2 : 2)	0.70
C_4 (3)	(3 : 0)	0.86	(3 : 0)	0.92
C_5 (5)	(1 : 1)	0.30	(1 : 1)	0.30

Table 1. *SimF / GenF* comparison on all test sequences

Table 2 presents the results of ***OptS-GenF*** on all of the test sequences. Since ***OptS*** is non-deterministic, these results are presented as means and standard deviations based on 500 runs of the program on each sequence. It can be observed that ***OptS*** generally makes little false detection but generally makes fewer good detections than ***MaxS***. Nonetheless, in the most simple cases, ***OptS*** performs well and represents an interesting alternative to ***MaxS*** in terms of efficiency: while the running time of ***MaxS*** on the tests sequences ranges from 4 to 240 seconds, the running time of ***OptS*** ranges from 0.4 to 2 seconds[4].

Sequence (number of activities)	***OptS – GenF***			
	Detections (good:bad)		Score	
	mean	st.dev.	mean	st.dev.
A_1 (2)	(1.53 : 0.00)	±(0.71 : 0.00)	0.63	±0.30
A_2 (4)	(2.61 : 0.00)	±(0.67 : 0.00)	0.47	±0.14
A_3 (3)	(1.23 : 0.03)	±(0.66 : 0.16)	0.31	±0.19
A_4 (3)	(1.71 : 0.02)	±(0.78 : 0.13)	0.45	±0.22
A_5 (2)	(1.76 : 0.24)	±(0.43 : 0.44)	0.68	±0.16
B_1 (5)	(3.57 : 0.04)	±(0.61 : 0.21)	0.51	±0.09
B_2 (2)	(1.00 : 0.78)	±(0.00 : 0.51)	0.16	±0.17
B_3 (4)	(3.01 : 0.76)	±(0.78 : 0.70)	0.80	±0.11
B_4 (3)	(2.10 : 0.13)	±(0.39 : 0.34)	0.61	±0.14
B_5 (4)	(2.86 : 0.05)	±(0.72 : 0.22)	0.62	±0.15
C_1 (2)	(1.80 : 0.20)	±(0.40 : 0.40)	0.79	±0.07
C_2 (5)	(1.05 : 0.08)	±(0.22 : 0.22)	0.14	±0.06
C_3 (5)	(1.07 : 0.97)	±(0.78 : 0.78)	0.36	±0.13
C_4 (3)	(1.73 : 0.34)	±(0.49 : 0.47)	0.54	±0.11
C_5 (5)	(0.84 : 0.21)	±(0.37 : 0.43)	0.17	±0.03

Table 2. *OptS - GenF* results on all test sequences (500 runs)

[4] Running times are based on a C++ implementation running on a P4 at 3GHz.

4. CONCLUSIONS & FUTURE WORK

This paper presents a new approach for the temporal segmentation of human activities in video sequences. This approach requires no prior knowledge on the activities to be detected and does not impose severe constraints on the type of the activity. We have proposed two different versions of the individual signal segmentation techniques and of the segmentation fusion techniques respectively.

MaxS-GenF proved to be effective on simple and moderately complex sequences. Moreover, all presented implementations have a low rate of false detections, making their detections reliable. It is also worth mentioning the noise robustness of the algorithms, which faced noisy skeletons in all test sequences. It was also demonstrated that in simple cases ***OptS*** can be effective and efficient. Finally, while a few thresholds were involved in the presented algorithms, experimental results showed that reliable results were obtained on all test sequences using fixed thresholds.

5. ACKNOWLEDGMENTS

We would like to thank NATEQ for the financial support of this research through a postgraduate scholarship. We would also like to thank Annette Schwerdtfeger and the ten volunteers for their help.

6. REFERENCES

[Bob01a] Bobick, A.F. and Davis, J.W., The Recognition of Human Movement Using Temporal Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.23, no.3, 2001, pp. 257-267

[Cut00a] Cutler, R. and Davis, L.S., Robust Real-Time Periodic Motion Detection, Analysis, and Applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no.8, 2000, pp. 781-796

[Pol94a] Polana, R. and Nelson, R., Low Level Recognition of Human Motion, *IEEE Computer Science Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, 1994, pp.77-82

[Vig03a] Vignola, J., Lalonde, J.F. and Bergevin, R., Progressive Human Skeleton Fitting, *Proceedings of the 16th Conference on Vision Interface*, 2003, pp. 35-42

[Yaz04a] Yazdi, M., Branzan-Albu, A. and Bergevin, R., Morphological Analysis of Spatio-Temporal Patterns for the Segmentation of Cyclic Human Activities, *17th International Conference on Pattern Recognition*, 2004, pp. 240-243

[Yur94a] Yuret, D., *From Genetic Algorithms to efficient optimization,* Technical Report 1569, MIT AI Laboratory, 1994