Robust motion segmentation for on-line application

Lukáš Klicnar, Vítězslav Beran

Brno University of Technology
Faculty of Information Technology
Department of Computer Graphics and Multimedia
Božetěchova 2, 612 66 Brno, CZ

xklicn00@stud.fit.vutbr.cz, beranv@fit.vutbr.cz

ABSTRACT

This paper presents an approach for on-line video motion segmentation. Common methods were designed for off-line processing, where time to process one frame is not so important and varies from minutes to hours. The motivation of our work was an application in robotic perception, where a high computational speed is required. The main contribution of this work is an adaptation of existing methods to a higher computational speed and on-line processing. The proposed approach is based on sparse features, we utilized the KLT tracker to obtain their trajectories. A RANSAC-based method is used for initial motion segmentation, resulting motion groups are partitioned by a spatial-proximity constraints. The correspondence of motion groups across frames is solved by one-frame label propagation in forward and backward directions. Finally, an approximation of dense image segmentation is obtained by using the Voronoi tessellation.

Keywords

Motion segmentation, moving objects detection, KLT tracker, Voronoi tessellation.

1. INTRODUCTION

The automatic pre-processing of digital content is getting high importance over the last decade. The growing importance is accelerated by the amount of video recordings, visual surveillance data or multimedia content that are easily acquired and shared. Such amount of digital data is of very limited use when only pixel-level knowledge is available.

All mentioned scenarios are convenient applications for off-line processing. Present applications of digital content analysis are usually efficient data storage, indexing, image or video retrieval, content-based copy detection, semantic indexing, etc. Such solutions pre-process the data in off-line stage, when the computational performance has low priority. Developed techniques are then designed to work off-line in general. The promising results of developed off-line techniques lead our research to design a method that processes the video stream in on-line manner. This is motivated by needs in applications like TV-broadcast monitoring, assistive systems utilizing a machine vision or robotic perception where operating in real-time is a crucial demand.

Presented work is focused to develop and evaluate a pre-processing method for video segmentation. Based on state-of-the-art methods that were developed for off-line processing, we design an approach for on-line video-content segmentation by common motion constraints. The video content is usually described by set of key-frames and any high-level processing is than applied to key-frames separately. The temporal attribute naturally included in video sequence is then suppressed or back-projected by high-level methods for wide base-line matching. Our method is designed to describe the video content in on-line manner by i) spatial segments in each frame using common motion constraints and ii) correspondences of spatial segments in temporal domain.

The rest of the paper is organized as follows: Section 2 review of existing methods extracting the information from video sequences in spatio-temporal domain. The segmentation method is introduced in Section 3. We illustrate the performance of the method and achieved results

in Section 4. Finally, in Section 5 the proposed method and its possible extensions are discussed.

2. RELATED WORK

The spatio-temporal segmentation and clustering of region trajectories is related to the research topics about interesting region detection, segmentation and tracking. Numerous existing methods dealing with video motion segmentation are designed to work off-line. The motion segmentation methods based on frame-two-frame optical flow [Shi98] [Cre05] [Won02] evaluate a motion difference between objects of two adjacent frames. The methods' locality in temporal domain tends to over-segment cluttered scenes or to lose the tracks when fast motion appears. The approach based on tracking of local patches has appeared in structure-from-motion tasks [Fit00] [Rot07] where the focus is more on 3D object model building and only rigid objects are considered. The motion segmentation was applied for matching, recognition and retrieval tasks [Siv06] where not only rigid objects are considered. The existing approaches also widely differ in density of trajectories from quite sparse [Siv06] [Bas08] to highly dense [Fra09] [Che09] [Bro10] sampling. Some previous works are closely related to extract spatio-temporal segments of particular objects where the training stage takes place to adopt the method to particular object type and temporal behavior, e.g. pedestrians [Lei05] [Rod07]. The problem of object segmentation in changing environments and moving backgrounds has been addressed also in robotic field [Bea11]. Based on probabilistic models, the knowledge of the robot's motion is used to determine the shape and location of objects. In contrary to our method, the knowledge of the robot's motion constrains the usage of this method for robotic application only.

The all previous related work are quite similar in the computational cost where most of the approaches work off-line and the time needed to process the frame range from minutes to hours. One of the promising works solving the problem of unsupervised on-line video segmentation [Vaz10] can effectively handle long sequences, create and terminate labels as the video is processed, and still preserve the photometric consistency of the segmentation across several frames.

3. SEGMENTATION

The proposed approach can be divided into these four consecutive blocks: i) feature detection and tracking, ii) initial motion segmentation, iii) object extraction and tracking, and iv) dense segmentation approximation. The method is based on a sparse feature tracking, because all image points cannot be processed, if a high computational speed is demanded. Trajectories of these features are input for the motion segmentation. At first, initial motion segmentation is estimated by a RANSAC-based robust algorithm, which results to groups of tracks with a similar motion. Groups are then partitioned to satisfy spatial proximity constraints of associated tracks. These steps provide a local label for each track, which is valid for a particular frame only, so their frame-to-frame correspondence has to be solved. This is called object extraction and tracking. Finally, the Voronoi tessellation is used to obtain a dense image segmentation to overcome the low density of sparse features. A block diagram of the whole system is in Figure 1.

Feature detection and tracking

One of the most important parts of this approach is a robust tracking method, which provides "good" trajectories. This means that they are as long as possible and they don't contain outliers. The length of trajectories has an influence on correct object tracking, while the low occurrence of outliers is essential for good motion segmentation. An outlier in this case means point of measured trajectory, which significantly differs from the actual position.

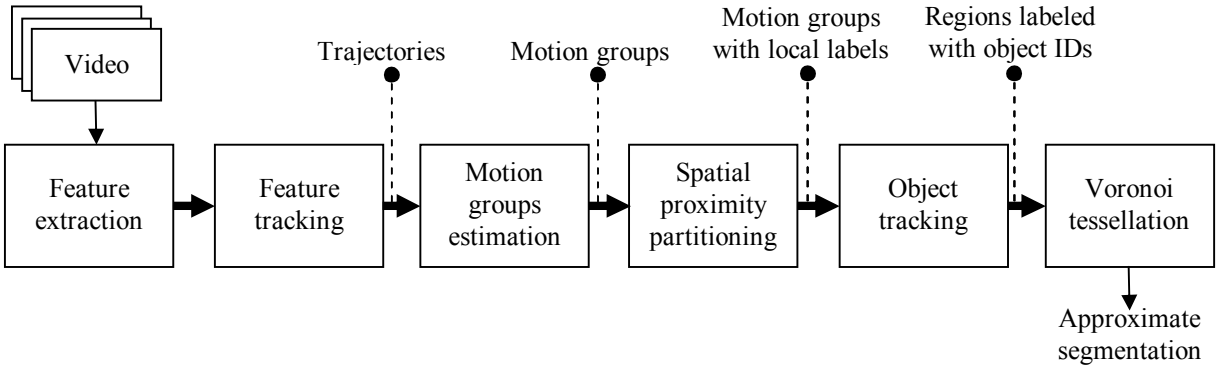

Figure 1. A block diagram of proposed system for motion segmentation.

We found tracking with the KLT tracker [Tom91] based on optical flow very suitable for this task. It doesn't match features from two frames, but it rather searches for a best occurrence in the image directly. This builds an invariance to the stability of used feature detector. Features are still detected in every frame, but they are used only for a continuous adding of new tracks. These aspects imply that any feature detector should be suitable. In the proposed approach, points provided by the Harris corner detector [Har88] are used. Features are detected on image scales of 1x, 1/2x, 1/4x and they are all merged and tracked together. This improves the ability to segment some moving objects, which are too fast for current shutter speed and appear blurred in the image – in undersampled image, the blurred structure becomes sharper and more features are detected.

The disadvantage is that the accuracy decreases with the scale, so a point detected e.g. at scale 1/4x is projected on 1x with an error of 4px. This didn't show as a problem, because the KLT tracker working on image pyramids handles these points correctly. In addition, matched features are checked by normalized cross correlation and if it's below threshold (values of 0.8-0.9 are used), the particular track is terminated instead of being extended. That results to trajectories with no visible outliers. In opposite to [Siv06] [Bas08], there is no need for short-range track repair, we found KLT tracking sufficient.

Initial motion segmentation

The initial motion segmentation is performed by a RANSAC based algorithm [Siv06] [Bas08], you can see a block diagram in Figure 2. It progressively extracts groups of features with a similar motion, a one group is extracted in every iteration. Four-tuples of tracks are randomly chosen to estimate a homography matrix that represents the motion between the current and the previous frame. A total reprojection error is then computed, that forms the criterion for suitability of found transformation – the goal is to find the homography corresponding with the lowest reprojection error. This is done iteratively until the maximal number of iterations is reached. In opposite to [Bas08], iterating can be stopped also when a desired reprojection error is satisfied (we use condition that average reprojection error is lower than 1.0px). This founds a sufficient transformation on average after 11 iterations. Compared to 100-200 iterations (which is the maximal number), this significantly increases the computational speed.

Inlying tracks (with reprojection error below 3.0px) are then removed as a single motion group and the rest of them goes through the same process in a next iteration. When a sufficient transformation is found, the corresponding homography is recomputed using inliers only. Every motion group is then divided to meet the spatial proximity constraint, that the distance between any point from this group and its nearest neighbour from the same group has to be lower than a threshold, so the group becomes spatially-compact. This step is not necessary, but it

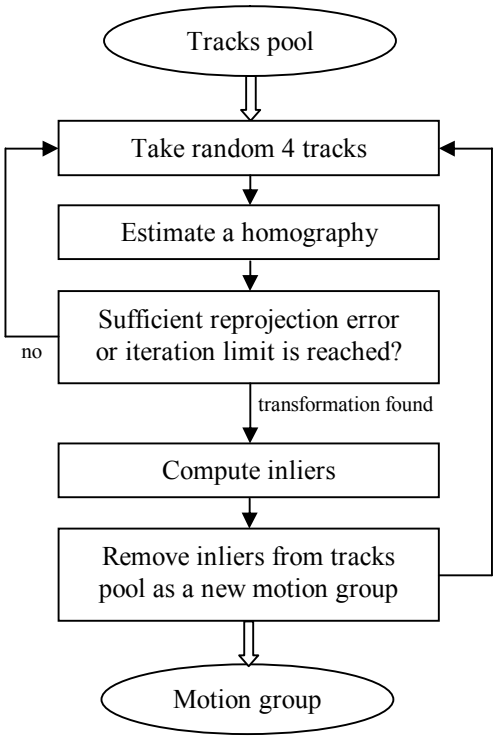

Figure 2. RANSAC motion segmentation.

allows recognizing distant objects moving in the similar way. The spatial proximity constraint may cause an oversegmentation, but this is handled by the next stage, object extraction and tracking.

Object extraction and tracking

From the previous step, every track in a motion group has a local label, which corresponds with a motion segment in a particular frame. For moving objects tracking, the frame-to-frame correspondence of these labels has to be solved. This has to be considered to be 1:N, because several motion groups from the current frame can match a one group from the previous frame (this may be caused e.g. by oversegmentation). A motion group correspondence across frames is solved with label propagation. In [Bas08], authors developed a method of propagation in forward and backward direction, which can handle situations, when motion groups split and merge, and it labels them correctly.

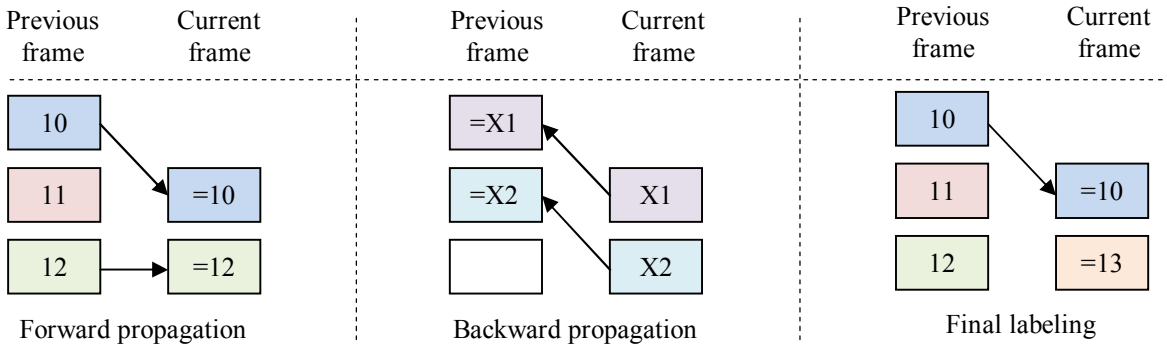

Figure 3. Label propagation in both directions and final labeling. Rectangles represent motion groups with their labels. New label (13) is generated, because groups assigned from both directions differ.

Unfortunately, this violates the condition of on-line processing, so only a limited version of this approach can be used.

The limitation consists in restricting the label propagation to a one frame, i.e. between the current and the previous frame, but in both directions. First, labels from the previous frame are propagated forwards, into the current frame. For every local label from the current frame, a corresponding label from the previous frame is found – that is the label of the group, which has the highest number of common tracks with the currently processed group. Subsequently, a new, temporary label for each group in the current frame is generated and propagated in the same manner, but in the opposite direction (backwards). Labels are then merged for each pair of groups from the current and the previous frames: The same label (from the forward labeling) is assigned only if both groups correspond together in both directions, otherwise a new one is generated (see Figure 3). New labels aren't assigned immediately to motion groups, but only if the same new label is stably suggested for at least 2-3 frames. This helps to overcome the problem that a motion group can occasionally disappear (tracks are incorrectly assigned to another group with different label), which tends to be stable for max. 1-2 frames.

Dense segmentation approximation

The disadvantage of a sparse approach is the low density of points, for which an object label is provided. That information is known only for points, where one of the tracked features occurs. To overcome this problem, the Voronoi tessellation is used to obtain an approximate dense segmentation of the whole image. The positions of tracks in the current frame are used as a cell generators, all pixels in each cell are then labeled with the same label as the track that generated the cell. This divides the image into segments, where points are labeled according to their closest track, where the motion information is known.

4. RESULTS

The evaluation of motion segmentation is a relatively difficult task, because it needs videos with masks of all moving objects, which is very time-consuming to obtain. There are very few dataset available that can be used for exact comparison of motion segmentation methods.

Dataset and evaluation description

To evaluate the proposed method, we used The Berkeley Motion Segmentation Dataset [Bro10]. It consists of a total of 26 videos: (i) 10 sequences of cars, which are rigid objects with predictable movement (ii) 13 sequences of Marple detective stories, which contains mostly people. These are non-rigid, relatively slowly moving objects, which may event stop for a while (iii) 2 sequences of people and (iv) 1 sequence from tennis match. Most of the objects in this dataset are people, typical sizes are at least 20 % of frame area. Lighting condition doesn't change significantly, but most of the videos are obtained by a moving camera. Sequences don't contain cuts.

This dataset consists of a 4243 frames, 189 frames are annotated with a pixel-precise mask. The annotation is dense in space but sparse in time, approximately every tenth frame is annotated. The resolution of frames varies from 350x288 (detective stories) to 640x480 (cars, people, tennis). Examples are in Figure 5.

Authors of this dataset also include methodology and tools for evaluation, which are even obligatory to use when evaluating on this dataset. This means that results are directly comparable to other algorithms. The following list is a free quotation from [Bro10] – for detailed information, please refer to it. The provided evaluation tool measures these parameters:

Density – the number of points for which a cluster label is provided over the total number of image

points. Higher density means that more information from the image is extracted.

Overall clustering error – the number of bad labels over the total number of labels on a per-pixel basis. The tool automatically assigns clusters to regions from annotation, all points covering their assigned region are counted as good labels and all others count as bad labels.

Average clustering error – similar to the overall error but averages across regions after computing the error for each region separately. It is usually much higher than the overall error, which is caused by incorrect detection of small regions.

Over-segmentation error – corresponds with the number of clusters merged to fit the regions from the annotation.

Number of objects extracted with less than 10 % error. A background region is excluded.

Results

Experiments with a different length of videos were made (only the first 10, 50, 200 or all frames were processed). In Table 1, the results are shown. The low density is caused by setting of the used feature detector. It can be set to detect more features, but this significantly slows the computation and it is not necessary, dense image segmentation by the Voronoi tessellation extends the density to 100% coverage. Although the results are presented without the tessellation, to show the performance of a motion segmentation itself. The high segmentation error is caused by the characteristics of the dataset. More than a half of objects include people, slowly moving non-rigid objects, which are relatively difficult to segment and track with this method. Slow movement causes that the object falls below the RANSAC threshold ant it gets merged with the background group. The on-line manner prevents to label the tracks according to knowledge if they will move faster and be part of different motion group in the future. By visual evaluation of videos containing moving cars only (they usually move significantly faster), the results would be better.

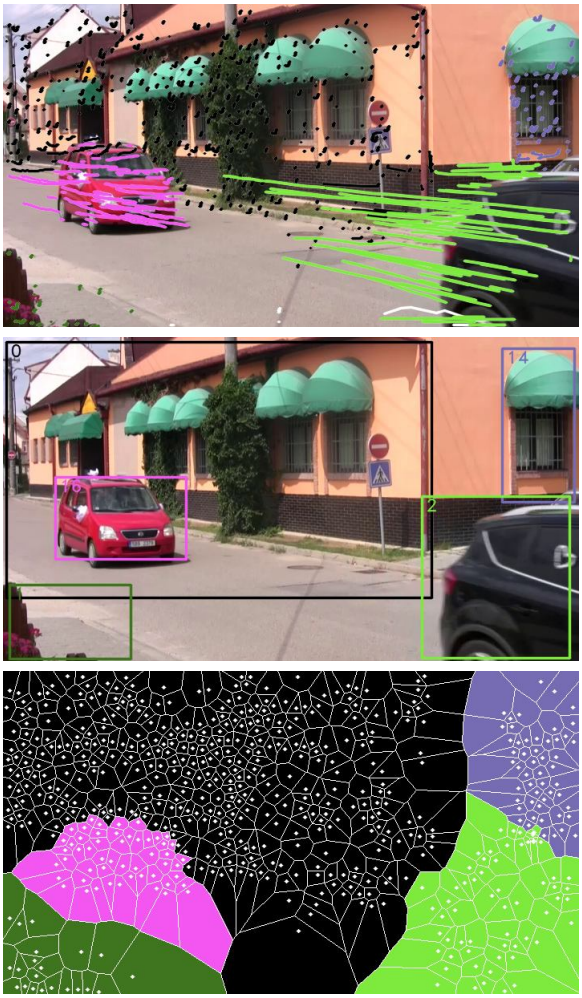

Figure 4. Motion segmentation of scene with two moving cars. From top: tracks, bounding boxes, Voronoi tessellation of the whole image.

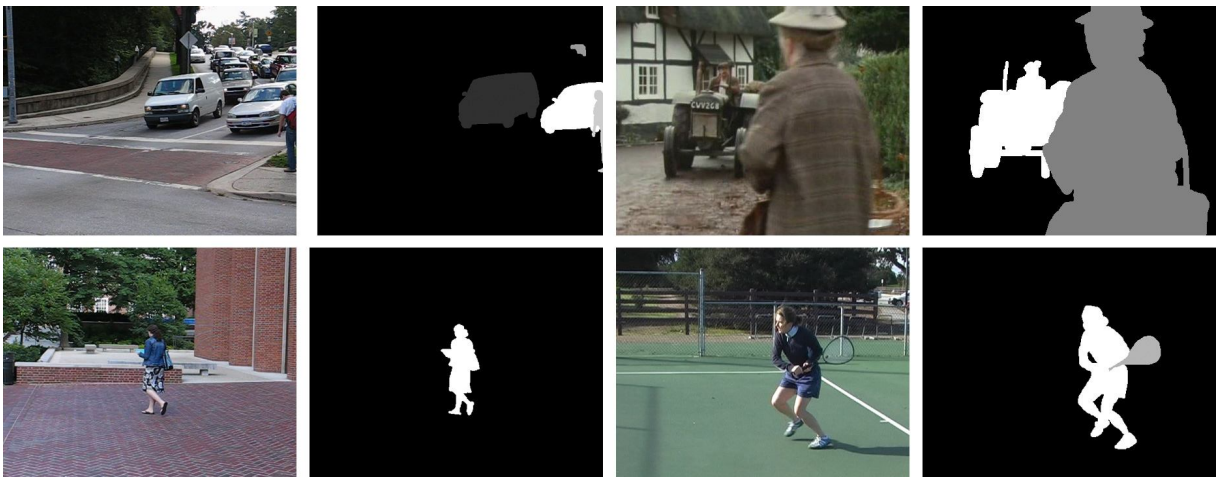

Figure 5. Examples of frames and their annotations from the Berkeley Motion Segmentation Dataset.

	Density	Overall error	Average error	Over-segment.	Extracted objects
First 10 frames (26 sequences)					
This method	**0.14%**	**19.68%**	**40.37%**	**3.87**	**14**
Brox and Malik	3.34%	7.75%	25.01%	0.54	24
GPCA	2.98%	14.28%	29.44%	0.65	12
LSA	2.98%	19.69%	39.76%	0.92	6
RANSAC	2.98%	13.39%	26.11%	0.50	15
ALC corrupted	2.98%	7.88%	24.05%	0.15	26
ALC incomplete	3.34%	11.20%	26.73%	0.54	19
First 50 frames (15 sequences)					
This method	**0.13%**	**20.22%**	**52.39%**	**2.40**	**3**
Brox and Malik	3.27%	7.13%	34.76%	0.53	9
ALC corrupted	1.53%	7.91%	42.13%	0.36	8
ALC incomplete	3.27%	16.42%	49.05%	6.07	2
First 200 frames (7 sequences)					
This method	**0.17%**	**19.39%**	**46.89%**	**4.29**	**3**
Brox and Malik	3.43%	7.64%	31.14%	3.14	7
ALC corrupted	0.20%	0.00%	74.52%	0.40	1
ALC incomplete	3.43%	19.33%	50.98%	54.57	0
All available frames (26 sequences)					
This method	**0.13%**	**19.41%**	**41.53%**	**3.54**	**13**
Brox and Malik	3.31%	6.82%	27.34%	1.77	27
ALC corrupted	0.99%	5.32%	52.76%	0.10	15
ALC incomplete	3.29%	14.93%	43.14%	18.80	5

Table 1. Results of the motion segmentaion on the Berkeley Motion Segmentation Dataset. The performance of other methods is taken from [Bro10], for their description, please refer to the original paper.

The results in Table 1 show that the segmentation error is higher than other methods. The overall performance is comparable to the ALC, but the main advantage is the computational speed, which will be discussed further. The proposed method is more sufficient for detection of rigid and not too slowly moving (frame-to-frame motion larger than threshold for inliers estimation in motion segmentation part) objects. The Berkeley dataset mostly contains moving people, which are objects that violate these conditions and this causes the overall increase of the segmentation error. The high average error means that the method doesn't work correctly on large amount of objects (people in this dataset), but it works reasonable on some types (cars). This method can better detect and larger objects, e.g. cars in the foreground, rather than smaller objects. Sometimes it can detect objects parts (wheels of the car, legs or hands of people), but it is capable to hold them only for a several frames.

The dense segmentation obtained by Voronoi tessellation is only very approximate. Results in Table 2 show that the values are not dependent on the density of tracks. The most of approximation errors is caused by image areas with low density, where the Voronoi cells are large and inaccurately approximate image segments.

Density	Precision	Recall	F-measure
0.06%	0.52	0.71	0.60
0.13%	0.56	0.73	0.63
0.46%	0.61	0.69	0.65

Table 2. Precision, recall and F-measure of dense segmentation for different densities of tracks.

In [Bro10], authors state the computation time for the first 10 frames of the people1 sequence. Unfortunately, they don't mention the parameters of used computer and 10 frames appear to be too little for reasonable comparison. The results are shown in Table 3. We measured the computation time on all sequences of the Berkeley dataset with several densities of tracks. With the low density of 0.13 %, the method could process about 420 tracks/s, but this increases up to approx. 823 tracks/s at a density of 2,0 %. The used computer has the Intel Core2 DuoT7100 CPU at 1.80 Ghz, 4 GB RAM, Windows 7 Professional x64, no GPU acceleration. We don't know the computer used in [Bro10], so these numbers are not directly comparable, but they differ so much, that the speed difference should be obvious.

	Tracks	Time	Tracks/s
Brox and Malik	15486	497s	31.16
ALC	957	22837s	0.042
This method	**Up to approx. 823 tracks/s**		

Table 3. Computation times for the first 10 frames of the people1 sequence from the Berkeley dataset. Results of other methods are taken from [Bro10]. The number of tracks and time for this method is not given, because of the different lengths of used sequences. This method was evaluated on all frames of all sequences of the Berkeley dataset.

On the Berkeley dataset, the motion segmentation part takes average 64.2% of algorithm running time, feature detection takes 10.5% and feature tracking 15.5%, which means that used features and KLT tracking are suitable for high performance motion segmentation. The spatial proximity partitioning takes 6.6% of total computational time and the Voronoi tessellation takes only a 2.9%.

5. Conclusion

The objective of the presented work was to design and evaluate a method for video on-line motion segmentation with demands for low computational cost. The solution is based on sparse feature tracking and RANSAC motion segmentation. The optical flow tracker proved to be very suitable for this task, the most time-demanding part is the motion segmentation, which makes the best candidate for further improvements. The results of evaluation show that the speed-up of this approach is a tradeoff for overall segmentation error. The Voronoi tessellation provides only a very approximate dense segmentation, it will be the one of the subjects of further improvements. But the number of extracted objects and oversegmentation appears to be usable. This method can process video of a standard TV resolution at approximately 1-1.5 fps, which is not yet sufficient for application in real-tim, such is the intended robotic perception. In the current state, this method could be used for example in the tasks involving moving objects detection, instance search, etc., when it is crucial to process to process the data quickly, but not in real-time.

6. Acknowledgments

This work was supported by the Ministry of Education project No. MSM0021630528 and the European Community's 7th Framework Artemis JU grant agreement n. 100233 (R3-COP).

7. References

[Bas08] Basharat, a, Zhai, Y., Shah, M. (2008). Content based video matching using spatiotemporal volumes. *Computer Vision and Image Understanding*, *110*(3), 360-377. doi:10.1016/j.cviu.2007.09.016.

[Bea11] Beale, D., Iravani, P. (2011). Probabilistic models for robot-based object segmentation. *Robotics and Autonomous Systems*, *59*(12), 1080-1089.

[Bro10] Brox, T., Malik, J. (2010). Object segmentation by long term analysis of point trajectories. *European Conference on Computer Vision*, 6315, 282-295. doi:10.1007/978-3-642-15555-0_21.

[Che09] Cheriyadat, A. M., Radke, R. J. (2009).Non-Negative Matrix Factorization of Partial Track Data for Motion Segmentation. *International Conference on Computer Vision*, 865 - 872.

[Cre05] Cremers, D., Soatto, S. (2005). Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, *62*(3), 249-265.

[Fit00] Fitzgibbon, A., Zisserman, A. (2000). Multibody structure and motion: 3-D reconstruction of independently moving objects. *European Conference on Computer Vision*, 891-906.

[Fra09] Fradet, M., Robert, P., Pérez, P. (2009).Clustering point trajectories with various life-spans. *Visual Media Production*.

[Har88] Harris, C., Stephnes, M. (1988). A combined corner and edge detector, *Proceedings of the 4th Alvey Vision Conference*, 147-151.

[Lei05] Leibe, B., Seemann, E., Schiele, B. (2005). Pedestrian Detection in Crowded Scenes. *Computer Vision and Pattern Recognition*, *1*, 878-885. Ieee. doi:10.1109/CVPR.2005.272.

[Rod07] Rodriguez, M. D., Mubarak, S. (2007). Detecting and Segmenting Humans in Crowded Scenes. *International Conference on Multimedia*. doi:10.1145/1291233.1291310.

[Rot07] Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J. (2007). Segmenting, modeling, and matching video clips containing multiple moving objects. *Pattern Analysis and Machine Intelligence*, *29*(3), 477-91. doi:10.1109/TPAMI.2007.57.

[Shi98] Shi, J., Malik, J. (1998). Motion segmentation and tracking using normalized cuts. *International Conference on Computer Vision*, 1154-1160.

[Siv06] Sivic, J., Schaffalitzky, F., Zisserman, A. (2006). Object Level Grouping for Video Shots. *International Journal of Computer Vision*, 67(2), 189-210. doi:10.1007/s11263-005-4264-y.

[Tom91] Tomasi, C., Kanade T. (1991). Detection and Tracking of Point Features. T*echnical Report CMU-CS-91-132*, Carnegie Mellon University.

[Vaz10] Vazquez-Reina, A., Avidan, S., Pfister, H. (2010). Multiple hypothesis video segmentation from superpixel flows. *European conference on Computer vision*, *6315*, 268-281.doi:10.1007/978-3-642-15555-0_20.

[Won02] Wong, Y., Spetsakis, M. (2002). Motion Segmentation and Tracking . International Conference on Vision Interface, , 42(4), 80-87. doi: 10.1016/j.jbiomech.2008.11.038.