Robust Realtime Motion-Split-And-Merge for Motion Segmentation

Ralf Dragon[1], Jörn Ostermann[2], and Luc Van Gool[1]

[1] Computer Vision Lab (CVL), ETH Zurich,
{dragon,vangool}@vision.ee.ethz.ch
[2] Institut für Informationsverarbeitung (TNT), LUH Hannover,
ostermann@tnt.uni-hannover.de

Abstract. In this paper, we analyze and modify the Motion-Split-and-Merge (MSAM) algorithm [3] for the motion segmentation of correspondences between two frames. Our goal is to make the algorithm suitable for practical use which means realtime processing speed at very low error rates. We compare our (robust realtime) RMSAM with J-Linkage [16] and Graph-Based Segmentation [5] and show that it is superior to both. Applying RMSAM in a multi-frame motion segmentation context to the Hopkins 155 benchmark, we show that compared to the original formulation, the error decreases from 2.05% to only 0.65% at a runtime reduced by 72%. The error is still higher than the best results reported so far, but RMSAM is dramatically faster and can handle outliers and missing data.

1 Introduction

In the past years, the motion segmentation of tracked features has been receiving increasing attention since it can be used as a strong prior in dense object segmentation [9], for the unsupervised learning of object detectors [11], or for tracking [14]. In order to track objects under occlusions, [3] proposed to apply motion segmentation on the basis of correspondences between independently detected SIFT keypoints instead of tracked features. In order to handle outliers and missing data, they carry out motion segmentation on the basis of multiple frame-to-frame motion segmentations, called multi-scale motion clustering (MSMC).

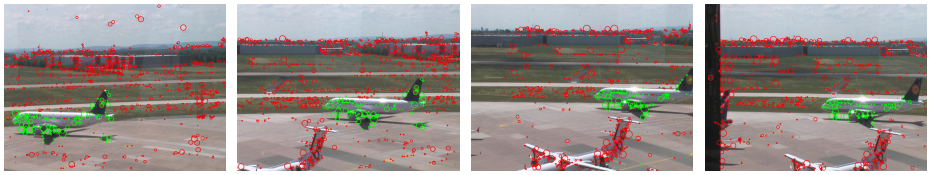

Fig. 1. Sequence *panning* from the Airport Dataset, segmented with our RMSAM & MSMC. The segmentation allows learning object keypoints from object motion.

For the frame-to-frame core, they propose an adapted version of the classical split-and-merge for images, called motion-split-and-merge (MSAM). On one side MSMC allows the segmentation of trajectories with large amounts of unknown or erroneous data. On the other side, the MSAM core contains many parameters and is not stable and fast enough for real time applications.

In this paper, we thoroughly analyze the components of the MSAM algorithm and derive a robust realtime variant, coined RMSAM. Our contributions are the formulation of RMSAM, a detailed analysis of its parameters, and an extensive evaluation on the Hopkins 155 [18] benchmark and our new Airport Dataset[3]. This paper is organized as follows: In Sec. 2, we give an overview of related work. In Sec. 3, the original approach MSAM is explained, analyzed and modified towards realtime RMSAM. In Sec. 4, we provide experimental evaluations, and in Sec. 5, a summary and a short conclusion is given.

2 Related Work

2.1 Multi-Frame Motion Segmentation

Existing approaches to multi-frame motion segmentation can be classified into *subspace-based* and *affinity-based*. In the first class, a measurement matrix $\mathbf{W}$ is constructed consisting of all points of all trajectories. Trajectories from different motions lie in different subspaces of $\mathbf{W}$, so the motion segmentation can be found by subspace assignment. This is algebraically elegant and allows very good results, as in agglomerative lossy clustering (ALC) [12] or sparse subspace clustering (SSC) [4]. However, these approaches require that tracked points only go missing (e.g. due to occlusions) in a certain way and up to a limited degree. For instance, ALC and SSC need at least one trajectory with complete data.

Affinity-based approaches do not come with such constraints since only the pairwise relationship between trajectories (affinity) is analyzed. Cheriyadat and Radke [2] decompose the trajectory features speed and direction using non-negative matrix factorization (NNMF). The resulting weights are used for an affinity measure. Fradet et al. [7] propose affine motion similarity as basis for affinities which are clustered by J-Linkage [16]. Brox and Malik [1] use spatial distance and similarity of translational motion to compute affinities which are then clustered with Spectral Clustering [8]. In order to render the motion similarity more precise, they propose to analyze triplets of trajectories [10] which allows them to add scale and rotation to the motion model. The MSMC approach [3] allows for arbitrary frame-to-frame motion models. First, correspondences between pairs of frames are motion-segmented at different time scales using MSAM. In a second step, ambiguities are resolved by observing common frame-to-frame motion during a longer time span.

[3] Available at `http://www.vision.ee.ethz.ch/~dragonr/airport`.

2.2 Frame-to-Frame Motion Segmentation

In the MSMC-approach, the parametrization of motion plays an important role to resolve ambiguities. RANSAC [6] first tackles the parameter estimation of data from one underlying parametrization mixed with random data. Fischler and Bolles derived the number r of trials to find an inlier-only minimal sampling set (MSS) – the smallest set of inlier data points to estimate parameters:

$$r \geq \frac{\log(1-p)}{\log(1-w^L)}\,, \tag{1}$$

where L is the cardinality of an MSS, w is the inlier ratio and p is the probability of finding an inlier-only MSS with r trials. In this paper we use $p = 0.95$. As it can be easily verified, L has a huge impact on r, especially if w is small.

Torr [17] extended RANSAC to the multi-model case by applying it sequentially on the remaining outliers. However, w is very small for the first motion since all other segments count as outliers. Additionally, sequential RANSAC is biased towards rendering small segments too small and big segments too big since it is too greedy [15]. A further problem pointed out by [19] is that sequential RANSAC detects *phantom motions* originating from the interaction of different moving objects. In the field of model selection, the problem is known as overfitting (too many degrees of freedom) vs. oversegmentation (too few degrees).

In order to tackle the multi-model problem, Schindler and Suter [13] proposed a sample-and-cluster paradigm. In the first step, N motion models are randomly sampled. In a model selection step, multiple models are selected such that a precise modeling is achieved at a low complexity. Similarly, in J-Linkage [16], an inlier matrix $\mathbf{I}$ is constructed. It specifies the inlier- and outlier relationship between all data points and N different motion parameters which are estimated from random local MSSs. The data is clustered bottom-up with agglomerative clustering. The affinity between segments is derived from common occurrences of inliers using the Jaccard distance.

A parameterless alternative is the graph-based image segmentation (GBS) [5]. Stalder et al. [14] adapted it to motion segmentation. In this approach, a graph between neighboring correspondences is established with difference in translational motion as weights. GBS finds segments such that intra-segment weights are low and inter-segment weights are high. In summary, in the MSMC context, the parametrization is essential to resolve ambiguities. We can use GBS as a strong parameterless baseline in frame-to-frame motion segmentation.

3 Robust Realtime Motion-Split-And-Merge (RMSAM)

Next, we derive the RMSAM algorithm. Its parameters are determined using ground-truth-labeled correspondences $\mathcal{Y}$ from our Airport Dataset. Given the ground truth object segments $\mathcal{V}_k$ as well as the best-matching permutation of the estimated segments $\mathcal{S}_k$, we compute the average object specific

$$\text{precision} = \frac{|\mathcal{S}_k \cap \mathcal{V}_k|}{|\mathcal{S}_k \cap \mathcal{V}_k| + |\mathcal{S}_k \cap (\mathcal{Y} \backslash \mathcal{V}_k)|}\,,\ \text{recall} = \frac{|\mathcal{S}_k \cap \mathcal{V}_k|}{|\mathcal{S}_k \cap \mathcal{V}_k| + |(\mathcal{Y} \backslash \mathcal{S}_k) \cap \mathcal{V}_k|} \tag{2}$$

over all objects k and ground truth frames in a sequence. The results over all sequences are averaged. To show the variation over the sequences we additionally give 20% and 80% quantiles.

3.1 Enforcing Convergence

The original MSAM formulation consists of the following 4 steps which are carried out every iteration until all segments remain unchanged:

1. Split segments: Motion parameters $\boldsymbol{p}_k$ of each segment $\mathcal{S}_k$ are estimated using RANSAC, assuming an inlier ratio of $\tau(\mathcal{S}_k \mid \boldsymbol{p}_k) \geq \theta_s$ to determine the number of trials r according to (1). If this assumption holds, correspondences from $\mathcal{S}_k$ which are not inlier according to $\boldsymbol{p}_k$ are removed from $\mathcal{S}_k$ and added to the outlier segment $\mathcal{S}_O$. Otherwise, the segment is split into two parts using an adapted version of J-Linkage, as further explained in Sec. 3.2.
2. Merge segments: Each pair of segments ($\mathcal{S}_k$,$\mathcal{S}_l$) is merged if the inlier ratio $\tau(\mathcal{S}_l \mid \boldsymbol{p}_k)$ of the smaller segment $\mathcal{S}_l$ according to parameters $\boldsymbol{p}_k$ is larger than a threshold θ_m.
3. Split outliers: Using the adapted J-Linkage, the outlier segment $\mathcal{S}_O$ is split into two like a regular segment in step 1. Resulting parts which are large enough are added as regular segments.
4. Merge outliers: Each correspondence from the outlier segment $\mathcal{S}_O$ is assigned to a segment $\mathcal{S}_k$ if it is inlier according to its parameters $\boldsymbol{p}_k$.

Even if enforcing that merged segments cannot immediately be split again ($1 + \theta_m \geq 2\theta_s$), it is easy to find an example in which a segment becomes cyclically split and merged. For instance, RANSAC might not find good parameters and the segment is accidentally split which is corrected by a merge in the following iteration, leading to the original state.

By discarding step 3 (Sec. 3.3) and imposing the following constraint, we enforce convergence: Once a valid parameter vector $\boldsymbol{p}_k$ with an inlier ratio $\tau(\mathcal{S}_k \mid \boldsymbol{p}_k) > \theta_s$ has been found in step 1, it is kept fixed for the segment. $\boldsymbol{p}_k$ is not recomputed if we merge a smaller segment (step 2) or outliers (step 4) to it. We call such a segment *consistent* since after the removal of outliers in step 1 of the following iteration, the inlier ratio is $\tau(\mathcal{S}_k \mid \boldsymbol{p}_k) = 1$. Besides enforced convergence, this measure allows speeding up step 2 by remembering a decision for further iterations. In RMSAM, we choose to split as early as possible, i.e. $\theta_s = 1/2 + \theta_m/2$. As Fig. 2a shows, the performance is not very dependent on the choice of θ_m. For highest precision, we use $\theta_m = 0.9$, thus $\theta_s = 0.95$.

Proof of Convergence: The approach finishes if no step is carried out. Equivalently, all segments $\mathcal{S}_k$ have an inlier ratio $\tau(\mathcal{S}_k \mid \boldsymbol{p}_k) = 1$, they cannot be merged and all correspondences from $\mathcal{S}_O$ cannot be merged into any segment. Since all steps necessarily lead to an increasing[4] $\tau(\mathcal{S}_k \mid \boldsymbol{p}_k)$, the algorithm has to converge.

[4] In the merge step 2 (including the successive outlier removal), the bigger segment has increasing τ and the smaller is dissolved. This ordering by size prevents cyclic growing and dissolving of a segment.

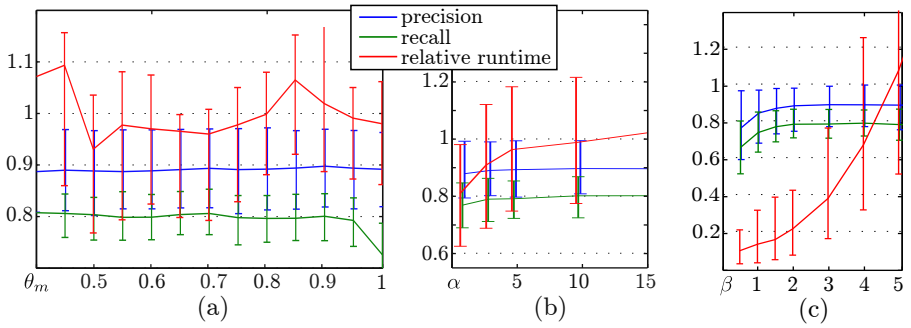

Fig. 2. (a) Analysis of the merging threshold θ_m, (b) the inlier sampling factor α and (c) the smallest detectable segment parameter β.

3.2 Adaptive Fast Split into 2 Segments

In the original MSAM formulation, an adapted J-Linkage is proposed to split a segment $\mathcal{S}$ into two parts. It differs by the sampling strategy to construct the inlier matrix $\mathbf{I}$, and by the clustering method:

1. To raise the inlier ratio, a best-of-n strategy is applied which makes no assumption on the spatial distribution of the segments.
2. The segmentation is carried out with K-Means instead of agglomerative clustering since the number of clusters ($K = 2$) is known a-priori. The cluster vectors are the columns from an affinity matrix incorporating motion similarity through the Jaccard distance and spatial similarity through the Mahalanobis distance.

In order to raise the probability of sampling an inlier-only MSS, we take into account the spatial distribution. However, in contrast to the J-Linkage approach, we choose the neighborhood size adaptive to the clustering problem. We propose adaptive localized sampling: The first point $\boldsymbol{x}_0$ is drawn with uniform probability. All remaining points are drawn with

$$p(\boldsymbol{x}) \propto \exp\left(-\frac{M^2(\boldsymbol{x}, \boldsymbol{x}_0)}{2\sigma_s^2}\right), \tag{3}$$

where M is the Mahalanobis distance inside $\mathcal{S}$, and σ_s, empirically set to 1, determines the neighborhood size. By this, enough inliers are found such that the best-of-n-sampling is not necessary. This leads to a significant runtime reduction to obtain the same amount of inliers, e.g., in the Airport Dataset by 84%.

An open question is how many samples N should be used to construct $\mathbf{I}$. Too few samples lead to high uncertainty during the clustering, while too many lead to a long runtime. In MSAM, $N = 10$ was proposed while it is much higher in J-Linkage (usually 1000 or more). However, different split problems require a different amount of investigation. A heuristic that prevents using either too few

or too many samples is to sample until more than $\alpha|\mathcal{S}|$ inliers are found, where a reasonable choice is $\alpha = 10$ (Fig. 2b). With this, simple clustering problems are solved with a low N and complex ones with a high N to ensure a sufficiently detailed analysis of the data.

The sampling goal of finding $\alpha|\mathcal{S}|$ inliers might never be reached, e.g. when the segment completely consists of outliers. In order to limit the number of samples to an upper bound $N_{\max}$, we follow the idea of having a smallest detectable segment $\mathcal{S}_m$ in $\mathcal{S}$. Let $\beta = (|\mathcal{S}| - |\mathcal{S}_m|)/|\mathcal{S}_m|$ be its ratio of outliers to inliers. If $\mathcal{S}$ contains exactly one segment $\mathcal{S}'$ with $|\mathcal{S}'| \geq |\mathcal{S}_m|$, an inlier-only MSS is found with a probability p according to (1) in $r = N_{\max}$ trials:

$$N_{\max} = \frac{\log(1-p)}{\log\left(1 - \left(\frac{1}{\beta+1}\right)^L\right)}. \tag{4}$$

As shown in Fig. 2c, $\beta = 2$ is a reasonable trade-off between runtime and precision. This means that segments which are smaller that one third of the number of outliers are unlikely to be detected. In order to detect such segments, the best choice is to reduce the number of outliers by measures like stricter matching thresholds.

3.3 Re-Distilling

Step 3 from MSAM is necessary to find small segments which were accidentally added to the outlier segment. Apart from wanting to avoid the convergence problem (Sec. 3.1), it does not seem reasonable to carry it out during every iteration since $\mathcal{S}_O$ may remain unchanged. Especially if $\mathcal{S}_O$ is large or if many iterations are carried out, this has a huge impact on the runtime.

RMSAM handles this differently: Initially, all correspondences are put into one segment, which is iteratively processed with steps 1, 2 and 4. Upon convergence, the outlier segment $\mathcal{S}_O$ is treated as initial segment and iteratively processed with steps 1, 2 and 4 again – it is "distilled" a second time. All other consistent segments $\mathcal{S}_k$ stay the same during the re-distilling since all correspondences from $\mathcal{S}_O$ are not inliers to the parameters $\boldsymbol{p}_k$ and thus will never be merged into $\mathcal{S}_k$ in steps 2 or 4. Since the outlier segment may contain only true outliers, the re-distilling is only carried out a certain number of times ν_r. We choose $\nu_r = 3$ as this is the largest value that still yields a significant increase in precision (Fig. 3a).

4 Experiments

4.1 Model for Frame-to-Frame Motion

In this section, we compare the performance of the translational, affine, homographical and epipolar motion models[5]. If the model is too general, segments are

[5] Since epipolar motion only allows analyzing the model error in the direction orthogonal to the epipolar line, its inlier threshold ϵ is not comparable to the thresholds of

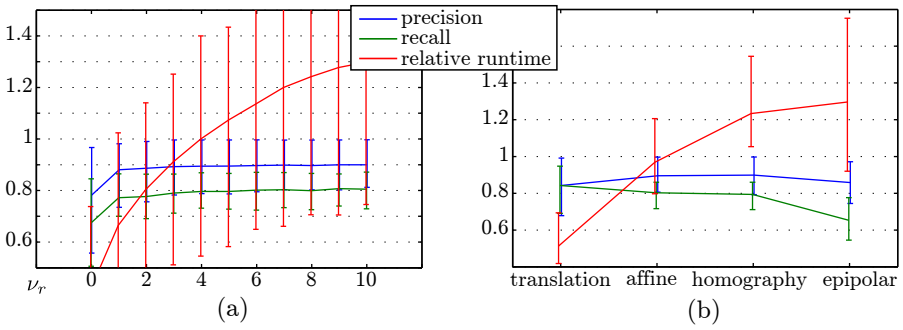

Fig. 3. (a) Analysis of the number ν_r of redistillation steps. (b) Comparison of translational, affine, perspective and epipolar motion models.

accidentally combined and precision and recall decrease (2). On the other hand if a motion model is too specific, the precision decreases and the recall increases due to oversegmentation.

The results are displayed in Fig. 3b. The ratios of runtimes are $0.53 : 1 : 1.27 : 1.3$, so the simple translational model is more than a factor of 2.5 faster than the epipolar model. Lower runtimes are not only related to lower computational complexities but also to smaller minimal sampling sets (MSS). The latter raises the probability of sampling an MSS from one underlying motion, thus allowing for a lower N during an adaptive split. The precision is the highest for the affine and homographical motion model with only very small differences between them. We use the first since it is significantly faster. The epipolar model performs worst in recall and runtime, so it can be discarded.

To conclude, the affine motion model is best suited for our motion segmentation. For fast segmentation, a translational motion model may be used at the cost of lower precision. We have to keep in mind that the choice of motion model is dependent on the observed scene and its motion, here the Airport Dataset. However, as we will show in Sec. 4.3, the affine motion model is also applicable in the Hopkins benchmark, so we are not overfitting to the training data.

4.2 RMSAM vs. GBS vs. JL

We now compare our RMSAM algorithm with J-Linkage (JL) and the graph-based segmentation (GBS) from [14]. For fair comparison, we treat segments with less than 8 correspondences as outlier segments. In GBS, the parameter k which describes the homogeneity of segments is empirically optimized as $k = 50$. JL is run with parameters which combine good performance and reasonable runtime: $N = 1000$ samples with a neighborhood of $\sigma = 1000$ pixel. The outlier threshold of RMSAM and JL is set to $\epsilon = 3$ pixel.

the translational, affine and homographical motions. We empirically found the best results for an ϵ-threshold which is half that of the others.

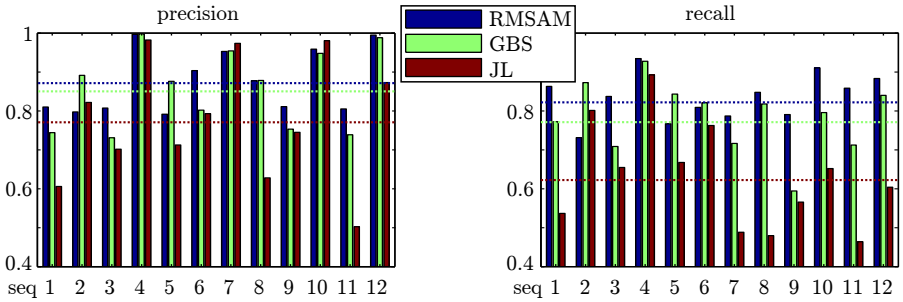

Fig. 4. Object precision and recall over the different sequences (bars) and their averages (dotted lines), using RMSAM, GBS and JL. The average runtimes per frame are 0.21 s, 0.20 s, and 4.90 s, respectively.

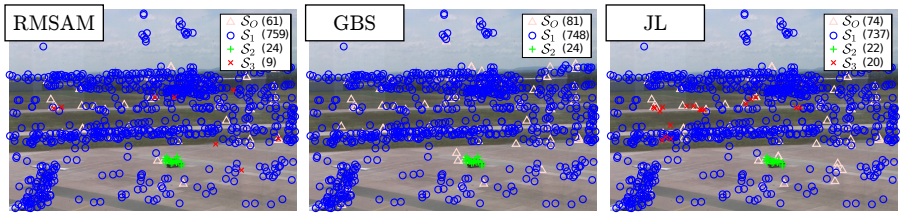

Fig. 5. Qualitative comparison of the segmentation of RMSAM, GBS and JL in the sequence 4 of the Airport Dataset. The numbers in parenthesis denote the number of correspondences inside a segment. JL tends towards oversegmentation, RMSAM towards combining outliers to ghost segments and GBS towards integrating outliers into regular clusters or vice-versa since it does not enforce a consistent parametrization along one segment.

The results in terms of object precision, recall and runtime are given in Fig. 4. In Fig. 5, a qualitative comparison is given which allows to assess the different characteristics. Our RMSAM achieves the best average object precision and recall at low processing times. Compared with JL, it performs better in almost all sequences at lower runtime. In sequences with slow motions, GBS, which does not use an outlier threshold, performs better than RMSAM. However, its results are much worse for sequences with large or non-translational motions since GBS' translational motion assumption is violated.

4.3 Hopkins 155 Benchmark

In order to evaluate the performance of RMSAM in a multi-frame motion segmentation context, we use it in combination with MSMC in the Hopkins 155 benchmark [18] consisting of the 3 categories Checkerboard, Traffic and Articulated. Since the Checkerboard sequences are mainly designed to verify analytic properties of subspace-based approaches and are not intuitive to cluster for humans, we skip these sequences, as [2,3,7] also do.

Approach	ALC [12]	SSC [4]	NNMF [2]	MSAM [3]	VLS [7]	**RMSAM**
Missing Data	constrained	constrained	yes	yes	yes	yes
Articulated, 2 motions, 11 sequences						
Error	10.70 %	0.62 %	10.00 %	6.03 %	5.38 %	2.38 %
Articulated, 3 motions, 2 sequences						
Error	21.08 %	1.42 %	15.00 %	8.27 %	20.41 %	1.91 %
Traffic, 2 motions, 31 sequences						
Error	1.75 %	0.02 %	0.10 %	0.66 %	1.92 %	0.06 %
Traffic, 3 motions, 7 sequences						
Error	8.86 %	0.58 %	0.10 %	0.17 %	4.89 %	0.16 %
All 51 sequences						
Runtime	261.3s	111.8s	3.0s	13.8s	5.7s	3.9s
Recall	1	1	1	0.977	1	0.978
Error	**5.41 %**	**0.28 %**	**2.82 %**	**2.05 %**	**3.80 %**	**0.65 %**

Table 1. Segmentation error rates and the recall in the Hopkins benchmark without missing data. The average CPU runtimes per sequence are collected from different papers with comparable systems (our system: 3.0 GHz quadcore standard desktop PC).

We compare our RMSAM & MSMC with results reported for subspace-based approaches (ALC [12] and SSC [4]) and for affinity-based approaches (NNMF [2], MSAM & MSMC [3], and VLS [7]). As MSMC may classify trajectories as outliers, we also report the recall according to [3, Eq. (15)]. To suppress randomness in RMSAM and K-Means, we average over 10 repetitions.

The results in terms of average error rate and recall are displayed in Table 1. Among all unconstrained approaches, we receive by far the best error rate at a recall very close to 1 and at a reasonable runtime. Compared to the very best result of SSC, we loose some accuracy, but 1) the speed goes up dramatically, and 2) the applicability is raised further by the fact that far fewer restrictions apply (like being able to handle any kind of missing data). It is also important to note that all other methods are optimized for the Hopkins benchmark, while we trained on the Airport Dataset, yet tested on the Hopkins benchmark. As has been shown in other comparisons like object class recognition, benchmarks tend to be biased and, as a result, training and then testing on different benchmarks may lead to rather serious losses in performance.

5 Summary and Conclusion

In this paper, we presented the robust realtime Motion-Split-and-Merge approach (RMSAM) for motion segmentation based on correspondences between two frames. We enforced convergence and introduced the adaptive fast split and re-distilling. Our experimental evaluation showed that RMSAM is superior to J-Linkage and has slightly better results than Graph-based Segmentation. However, because we parametrize motion, our approach can resolve ambiguities on a multi-frame level. Analyzing the performance of RMSAM on the Hopkins 155

benchmark, we showed that compared to the original formulation, the runtime is reduced by 72%, and the error from 2.05% to only 0.65%. The error is still higher than the best results reported so far, but RMSAM is dramatically faster, and it can handle outliers and unconstrained missing data.

Acknowledgment This work was partially funded by ERC project VarCity, SNF project AerialCrowd and BMBF project ASEV.

References

1. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. pp. 282–295 (Sep 2010)
2. Cheriyadat, A.M., Radke, R.J.: Non-negative matrix factorization of partial track data for motion segmentation. In: ICCV. pp. 865–872 (Oct 2009)
3. Dragon, R., Rosenhahn, B., Ostermann, J.: Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In: ECCV (Oct 2012)
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR. pp. 2790–2797 (2009)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Intl. Journal of Computer Vision 59(2) (September 2004)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus. Commun. ACM 24, 381–395 (Jun 1981)
7. Fradet, M., Robert, P., Perez, P.: Clustering point trajectories with various life-spans. In: CVMP. pp. 7–14 (2009)
8. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS. vol. 14, pp. 849–856 (2002)
9. Ochs, P., Brox, T.: Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In: ICCV (2011)
10. Ochs, P., Brox, T.: Higher order models and spectral clustering. In: CVPR (2012)
11. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR (Jun 2012)
12. Rao, S., Tron, R., Vidal, R., Ma, Y.: Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 32, 1832–1845 (2010)
13. Schindler, K., Suter, D.: Two-view multibody structure-and-motion with outliers. In: CVPR (2005)
14. Stalder, S., Grabner, H., Van Gool, L.: Dynamic objectness for adaptive tracking. In: ACCV (2012)
15. Stewart, C.V.: Bias in robust estimation caused by discontinuities and multiple structures. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 19, 818–833 (1997)
16. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-Linkage. In: ECCV. pp. 537–547 (2008)
17. Torr, P.H.S.: Geometric motion segmentation and model selection. Phil. Trans Mathematical, Physical and Engineering Sciences 356(1740), 1321–1340 (May 1998)
18. Tron, R., Vidal, R.: A benchmark for the comparison of 3D motion segmentation algorithms. In: CVPR (2007)
19. Wills, J., Agarwal, S., Belongie, S.: A feature-based approach for dense segmentation and estimation of large disparity motion. Intl. Journal of Computer Vision 68(2), 125–143 (2006)