Supplementary: Motion Segmentation by Exploiting Complementary Geometric Models

Anonymous CVPR submission

Paper ID 4172

1. KITTI 3D Motion Segmentation Dataset

The introduction of the Hopkins155 dataset [3] has done much to stimulate 3D motion segmentation research in the last decade. However, performance levels on this dataset by recent algorithms have hit a plateau and it is clear that further results on this dataset can no longer reveal critical insights into what aspects of the problem we need to improve on, especially for real-world scenarios where large field of views and significant camera translations are not uncommon. For this reason, we proposed the KITTI 3D Motion Segmentation Benchmark (KT3DMoSeg) to spur further research into this problem by the community. This benchmark is constructed by selecting short clips from the KITTI benchmark [1], a dedicated dataset for autonomous driving research. We select clips according to three principles: (1) We wish to study sequences with significant camera translation. Therefore, we choose clips where the camera is mounted on a moving car. (2) Complex background structure occurring in the real world is also not well-represented in existing datasets; therefore we choose clips with large depth relief and rich clutter. (3) Current motion segmentation is limited to have at most 3 motions; real world scenes can have a larger number of motions, and often face an attendant greater likelihood of subspace overlap. We therefore select clips with up to 5 motions. Overall, we chose 22 clips from 15 sequences of the KITTI benchmark. Each clip is between 10-20 frames. Thumbnails of the chosen clips are given in Fig. 1. We tag each clip with the unique sequence identifier from the KITTI benchmark, e.g. Seq005 indicates the raw sequence '2011_09_26_drive_0005' of the KITTI benchmark, and the order in each sequence say, clip01 indicates the first clip selected from the sequence.

2. Preprocessing and Labelling

In this section, we introduce the details of how we preprocess and label all clips.

Tracking We first of all extract dense trajectories from each video clip using the code provided by [2]. Key points are densely sampled on the whole frame with a gap of 8 pixels. Occlusion is detected by checking the consistency of the forward and backward flow [2]. Trajectories which fail the check are considered to be occluded or the underlying flow is incorrect. These trajectories are stopped. In the next frame, new trajectories are densely sampled in those areas not occupied by existing trajectories. Finally, we further filter out short trajectories less than 5 frames for robustness. The resultant feature trajectories are very dense, on the order of 1500-5000 points per sequence, and more importantly, the background points account for an overwhelming majority of all feature points. Examples of raw dense feature points are shown in Fig. 2. This huge imbalance of background and foreground point sets renders hypothesis-driven methods liable to miss small foreground objects, and also impose high computational load on most algorithms. To relieve these problems, we sub-sample 10% of the background points so that the average number of points is between 200-1000 for all sequences. The distribution of the number of points for all rigid motion groups is given in Fig. 3.

Manual Labelling Due to the large number of tracked points we need to come up with a method that can substantially reduce the effort in labelling. Our method is to only manually label the foreground moving objects, with the remaining unlabelled points all treated as stationary background points. Clearly, both the foreground and background obtained in this simple manner have many feature trajectories that do not belong well, either due to tracking errors or non-rigidities in the foreground motions. We next propose an efficient way to remove these outliers.

Outlier Removal We witness many erroneous trajectories generated by the dense tracking. Typical errors include point drifting, in particular background points adhering themselves to moving foreground objects. In this dataset, we identify outliers in a semi-autonomous manner with human-in-the-loop. In particular, we estimate via

Figure 1. Thumbnails for KT3DMoSeg benchmark.

RANSAC a single fundamental matrix $\mathbf{F}$ over two frames using all points in each rigid motion group defined above. This is repeatedly done for all consecutive pairs of frames. The goodness of a trajectory is based on the sum of Sampson errors w.r.t the respective $\mathbf{F}$s along the trajectory and normalized by the number of frames the point has appeared in. Points with accumulated residuals greater than $Q3 + 7IQR$ are considered as outliers and removed; in the above, $Q3$ is the third quartile and IQR is the inter-quartile range of all the residuals for points within a single motion. We do not claim that all bad features have been removed as a result, just as some small amount of bad feature trajectories still exist in the Hopkins 155 dataset. A completely automatic and reliable outlier detection module remains elusive, but addressing this problem is beyond the scope of our paper. To encourage further research on outlier detection for motion segmentation in-the-wild (i.e. without any manual intervention), we also publish the untrimmed feature trajectories. An illustration of the individual sequences with their various rigid motion groups and outliers is presented in Fig. 4, in which red dots indicate detected outliers.

3. Additional Motion Segmentation Results

We have earlier explored the merits of having the affine model as one of the views. Even one could argue that the affine model is just a special case of the homography model and is therefore redundant, its inclusion could still make sense numerically as the simpler affine model lends itself to more stable estimation. This has been corroborated in the results obtained for the largely small field-of-view Hopkins 155 sequences, presented in the main paper. The relative merits of including the affine view are far from clear for a strong-perspective dataset like our KT3DMoSeg. We now

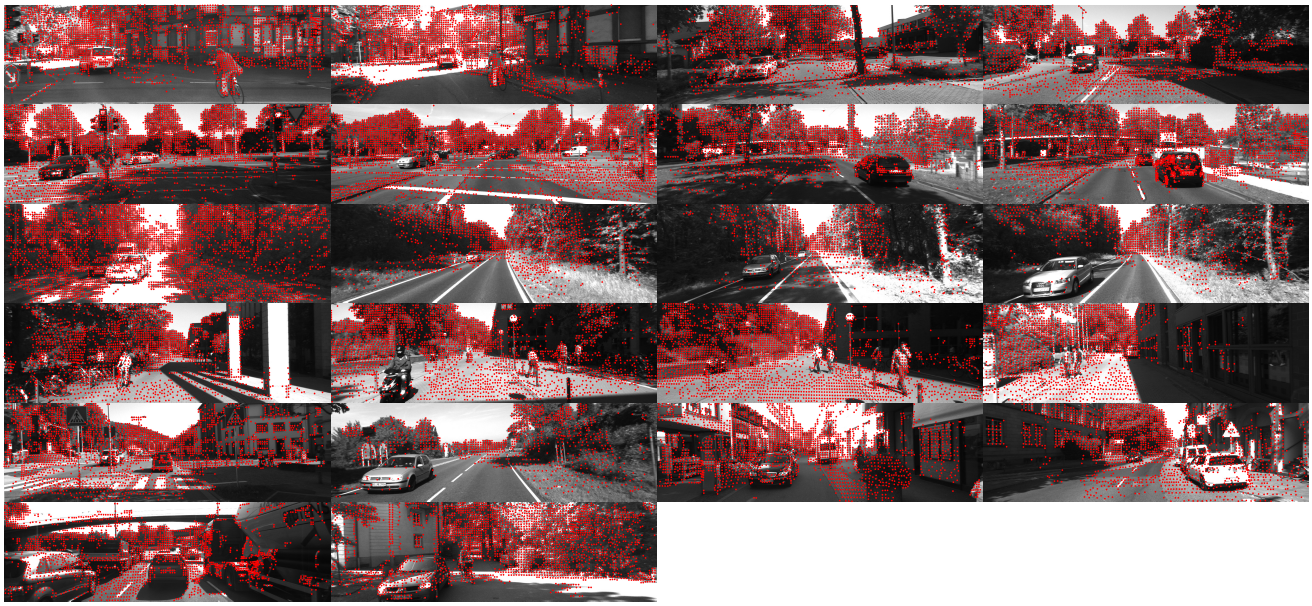


Figure 2. Raw feature trajectories for all sequences in KT3DMoSeg.

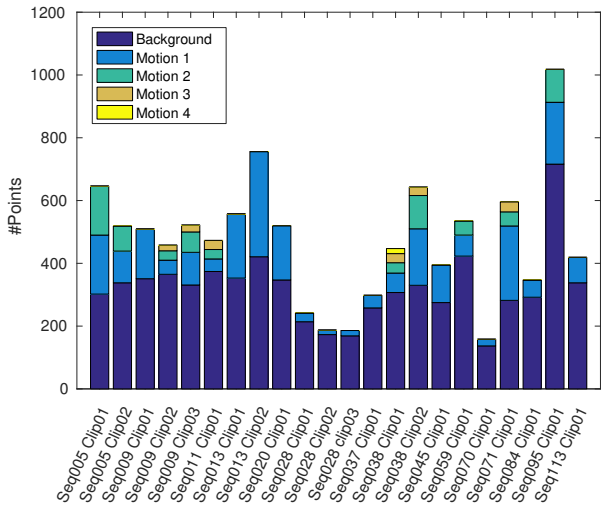


Figure 3. Number of points per motion for all sequences in KT3DMoSeg.

explore this important issue[1]. With a two-view scheme (**H** and **F** only), we obtained better performances for both the Subset Constrained and Co-Regularization fusion scheme, improving their performance levels from the three-view results 8.08% and 7.92% to 7.62% and 7.27% respectively.

The detailed quantitative results for individual sequences obtained from the two-view fusion are depicted in Fig. 5. The segmentation results for each sequence are presented pictorially in Fig. 4, where it is clear that our model takes advantage of different views successfully and produces results very close to the ground-truth on most sequences. On the few occasions our fusion model failed to generate meaningful results, in particular sequence (c), (k), (l) and (f), the reasons are similar as before. For the first 3 cases, neither **F** or **H** view is able to produce reasonable segmentation. This is probably due to the freedom in translating along epipolar line rendering **F** view ineffective, while **H** failed to link the background when confronted with non-compact objects and large depth relief. Sequence (f) failed by splitting the background and merging the two vehicles (green dots in "Subset"). The affinity between these two vehicles are inherited from the **F** view (due to its greater susceptibility to subspace overlap, and that the two cars have rather similar motions over much of the duration under observation). Unfortunately, this erroneous linkage failed to be corrected by the **H** view. The competing models, GPCA, SSC, LRR and LSA, failed on most sequences with few successes. One possible cause of failure could be due to the difficulty in filling in the missing data, especially when confronted with high missing rate (as high as 50%) and violation of the low-rank assumption (breakdown of affine model, higher number of motions). All four approaches are inclined to over-segment the background; this is clearly due to the limitation of the affine camera assumption which is suitable only for scenes with weak depth relief.

Finally, we present video demos for the motion segmentation results in the supplementary material as a separate file named "SuppID4172.mp4".

[1]We are not able to do this in time for the main paper submission, but our results obtained here do not invalidate the conclusions established in the main paper.

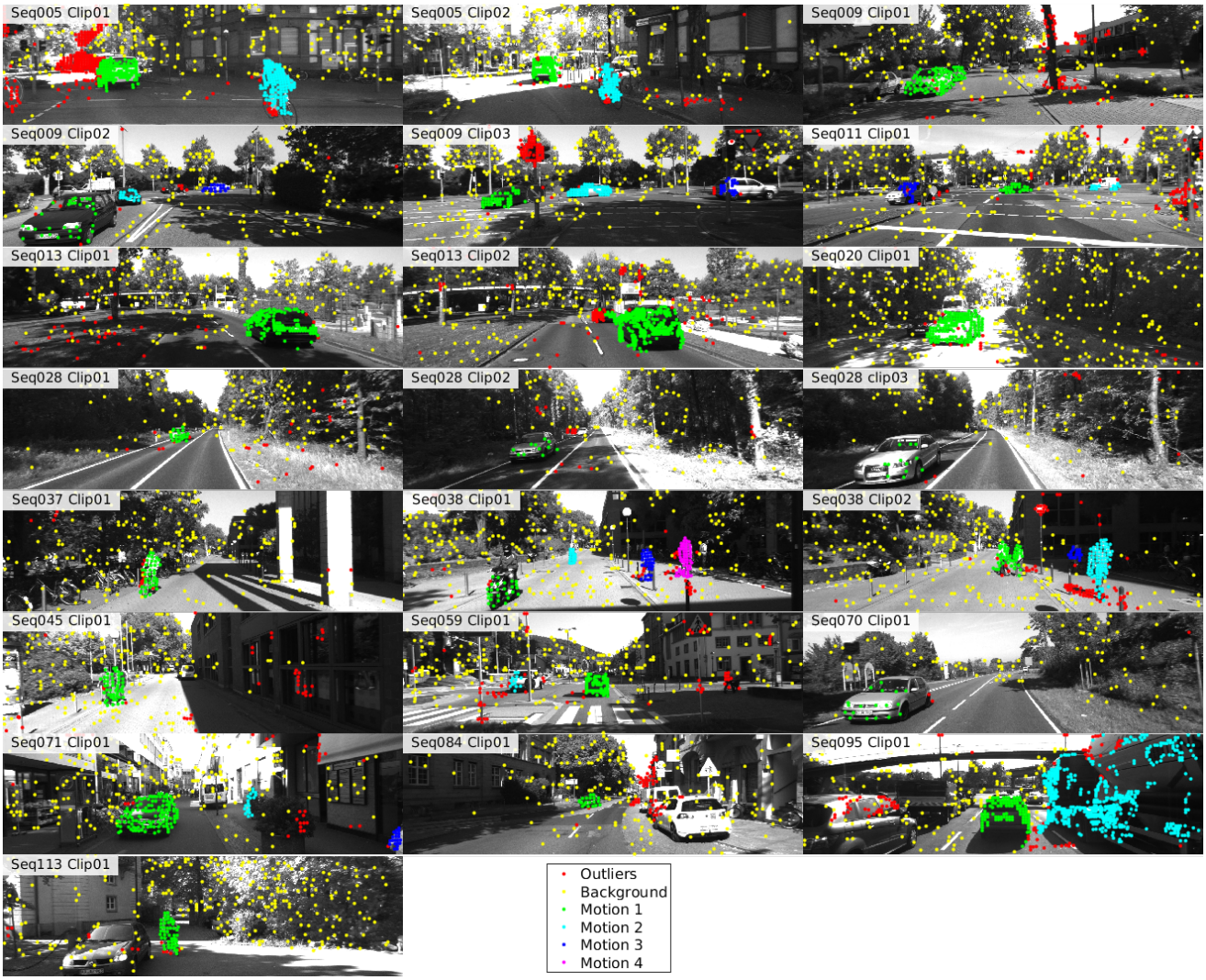


Figure 4. Ground-Truth trajectories for each sequence of KT3DMoSeg.

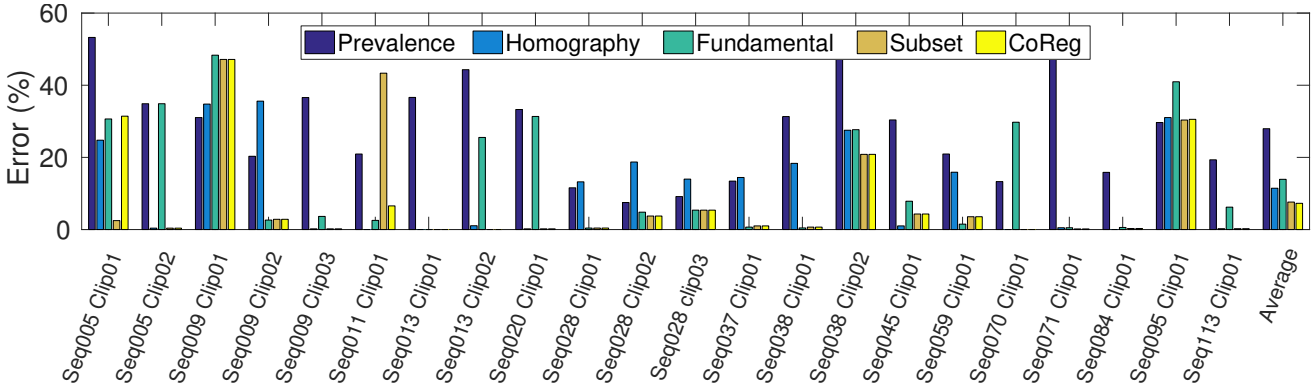
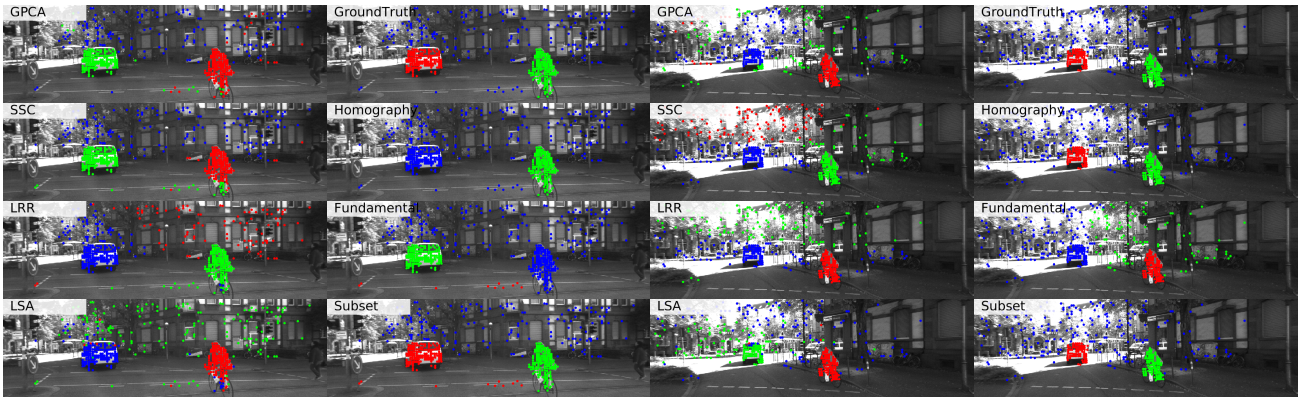


Figure 5. Motion segmentation errors of individual sequences with two-view fusion scheme.

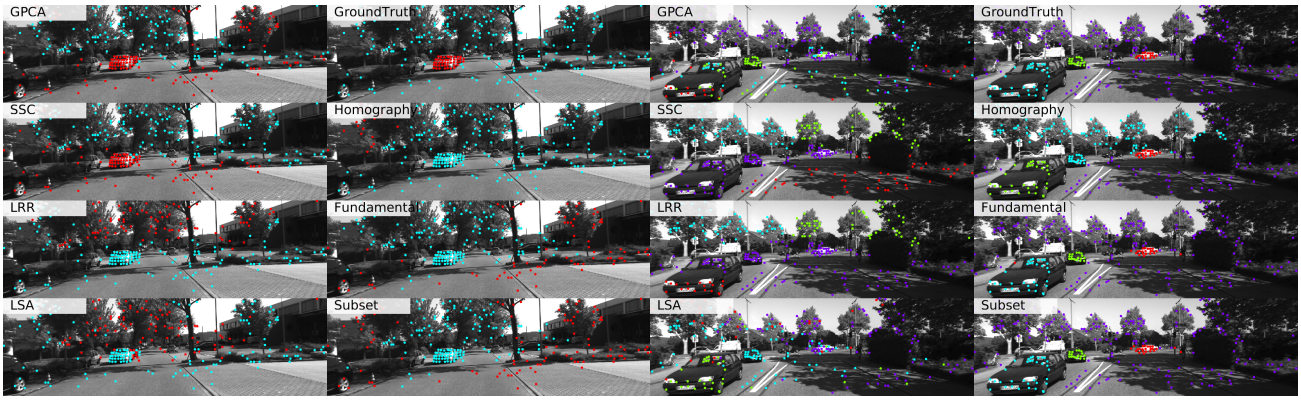
References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013. 1

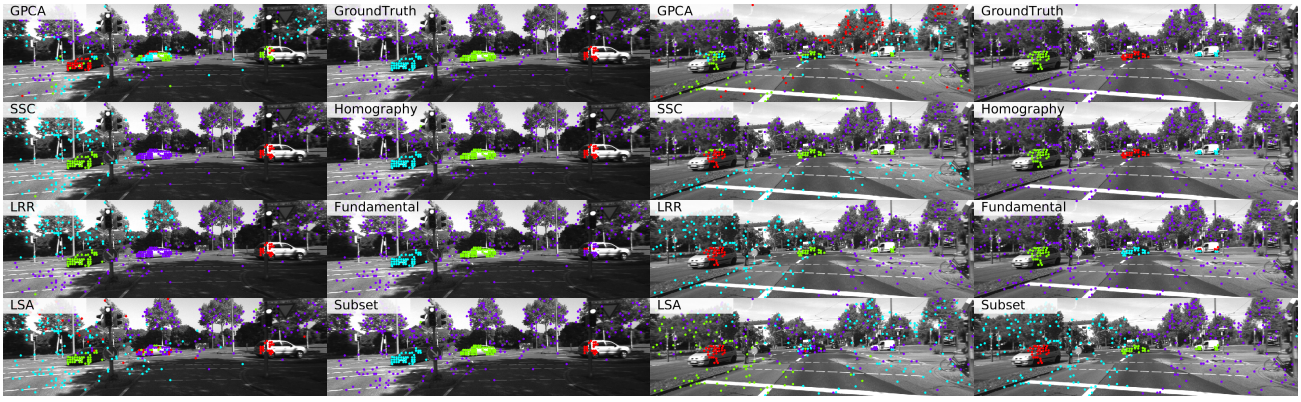
[2] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In



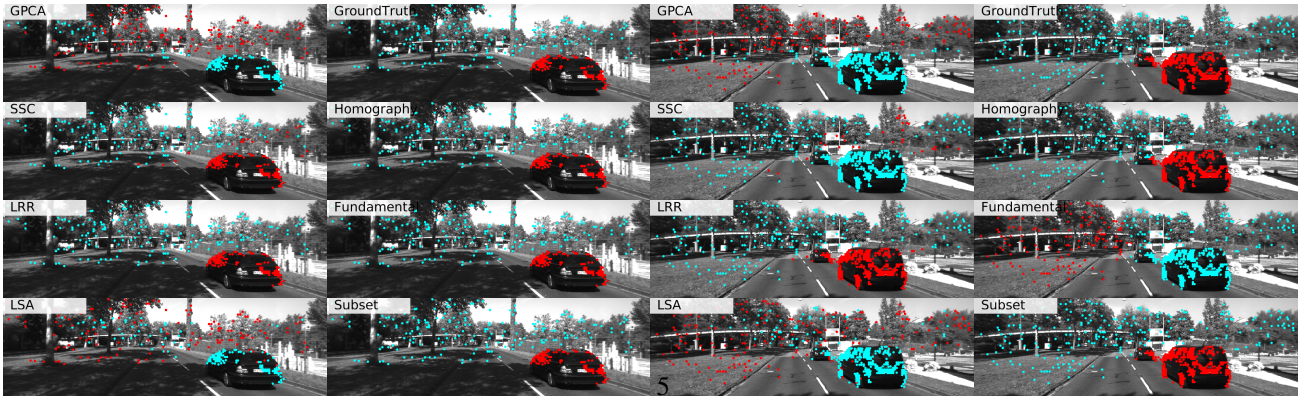
(a) Seq005 Clip01 (b) Seq005 Clip02



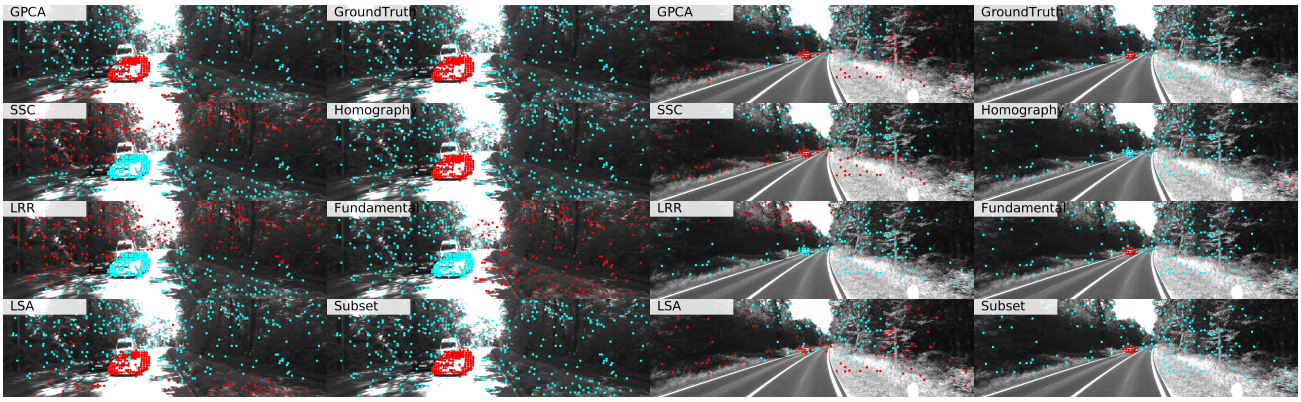
(c) Seq009 Clip01 (d) Seq009 Clip02



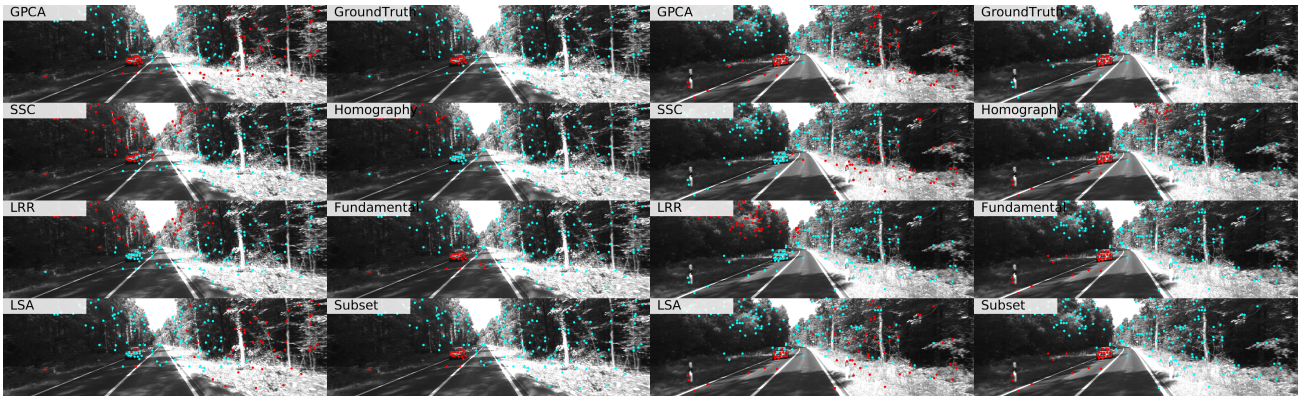
(e) Seq009 Clip03 (f) Seq011 Clip01



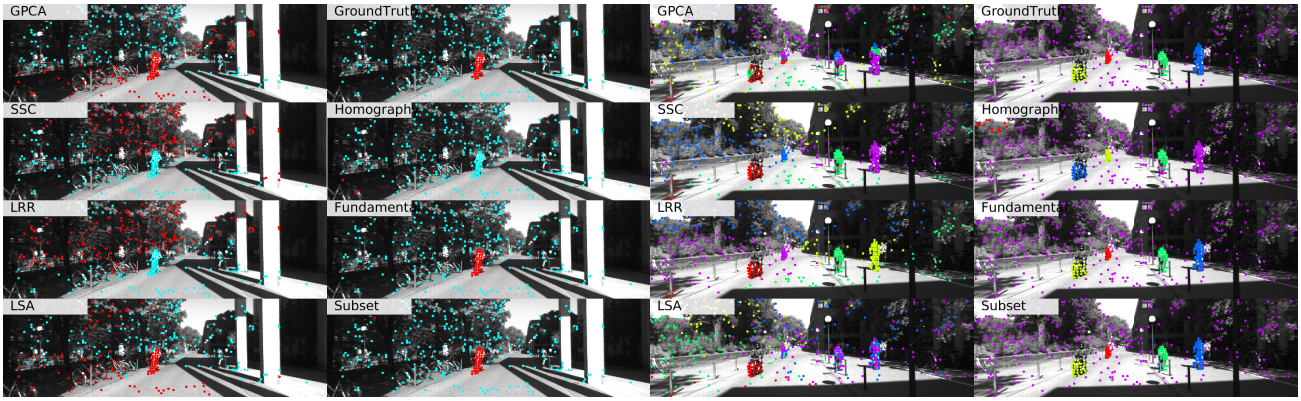
(g) Seq013 Clip01 (h) Seq013 Clip02



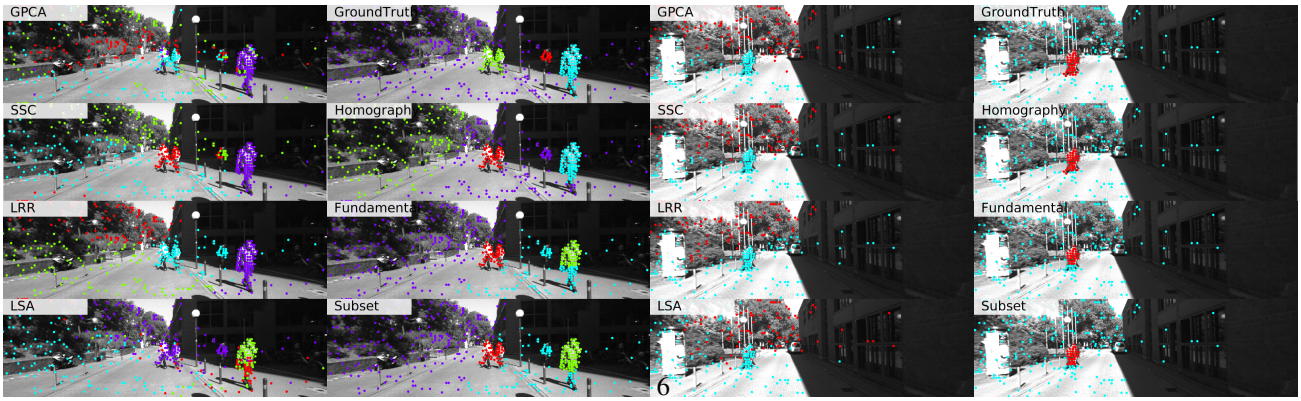
(i) Seq020 Clip01 (j) Seq028 Clip01



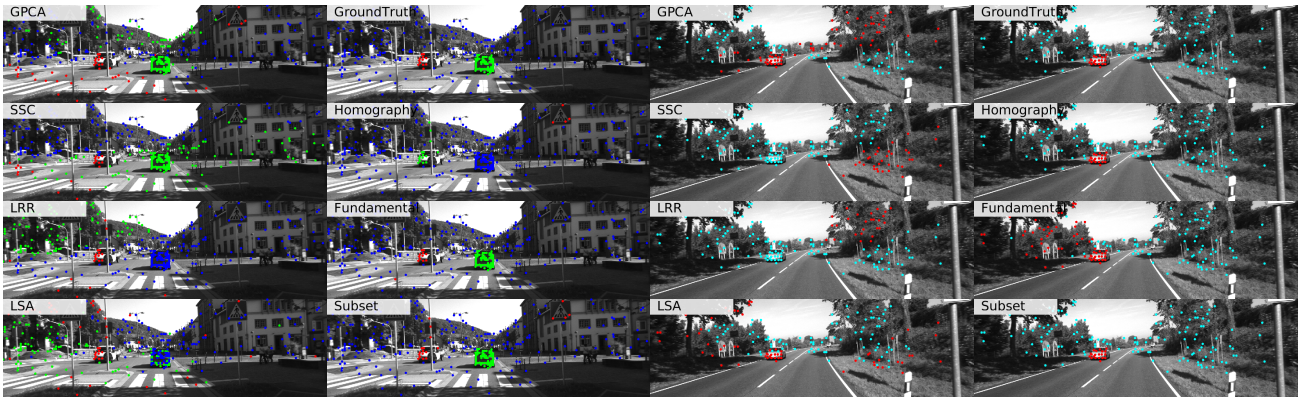
(k) Seq028 Clip02 (l) Seq028 Clip03



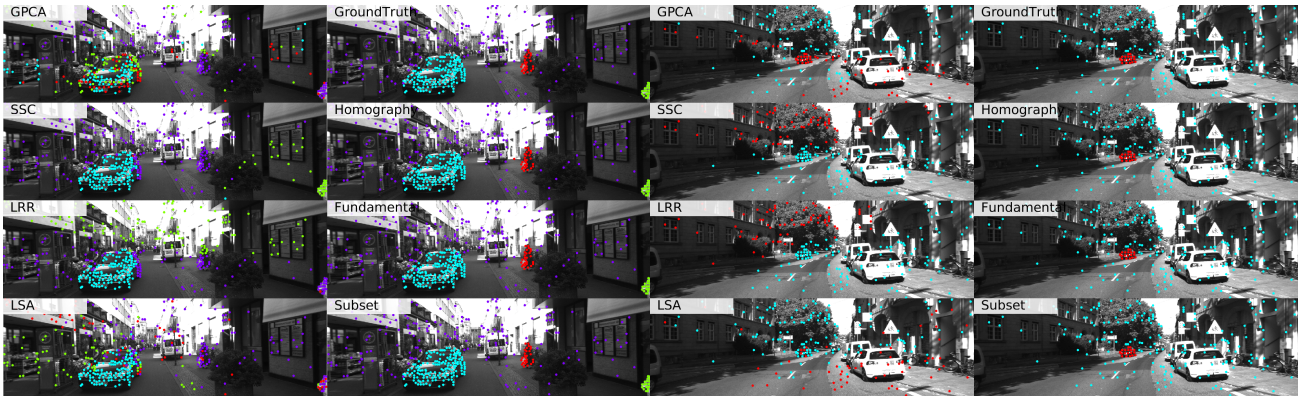
(m) Seq037 Clip01 (n) Seq038 Clip01



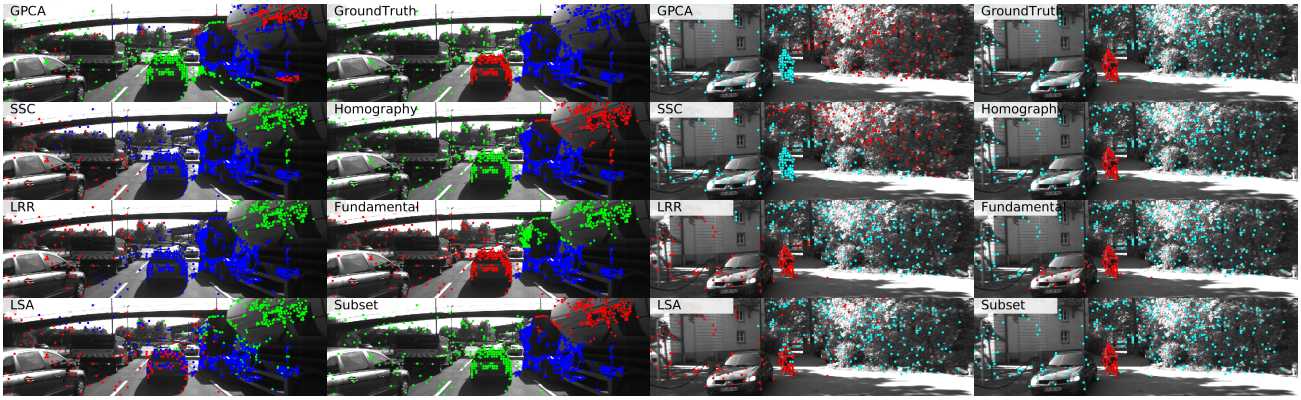
(o) Seq038 Clip02 (p) Seq045 Clip01



(q) Seq059 Clip01 (r) Seq070 Clip01



(s) Seq071 Clip01 (t) Seq084 Clip01



(u) Seq095 Clip01 (v) Seq113 Clip01

Figure 4. (a-v) Example frames of motion segmentation results for all sequences of KT3DMoSeg.

ECCV, 2010. 1

[3] R. Tron and R. Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *CVPR*, 2007. 1