# Motion-coherent Affinities for Hypergraph based Motion Segmentation

Kai Cordes, Christopherus Ray'onaldo, and Hellward Broszio

VISCODA GmbH, www.viscoda.com,
{cordes,rayonaldo,broszio}@viscoda.com

**Abstract.** Motion segmentation is the task of classifying the feature trajectories in an image sequence to different motions. Hypergraph based approaches use a specific graph to incorporate higher order similarities for the estimation of motion clusters. They follow the concept of hypothesis generation and validation. For the sampling of hypotheses, a high probability of selecting clean samples, i. e. samples consisting of points from the same cluster, is desired. Many approaches use spatial proximity to build an auxiliary graph for the sampling. But, spatial proximity is often not sufficient to capture the main affinities for motion segmentation. Thus, we introduces a simple but effective model for incorporating motion-coherent affinities into the auxiliary graph. The hypotheses generated from the resulting hypergraph lead to a significant decrease of the segmentation error evaluated on two state of the art benchmarks. Additionally, less computation time is required due to a reduced hypergraph complexity.

## 1 Introduction

Motion segmentation algorithms classify feature trajectories in an image sequence to a number of motions. Most approaches are multi-frame methods [1,4,5,20]. They take trajectories from many frames (e.g. 30) as input. On the other hand, there are two-view methods [7,12,14,18,20] which use correspondences of only two frames as input. In practice, it is often impossible to establish a sufficient number of trajectories on moving objects with large trajectory length. Often, a short response time is desired. Thus, our aim is to solve the motion segmentation task using small trajectory lengths, such as $5 - 10$ frames.
Recently, hypergraph based methods have been proposed for clustering tasks, such as motion segmentation [8,15,20], geometric model fitting [19,20] or face clustering [15]. A hypergraph contains higher order similarities instead of pairwise similarities which results in more accurate results with the same number of samples. In contrast to previous approaches [2,9] where only small degrees for hyperedges are used, recent approaches employ large hyperedge degrees [15,20]. Using large degrees of hyperedges yields better clustering accuracy because more information of the relationship between vertices is included. To incorporate large hyperedges, the approach [15] makes use of a special sampling technique [13,16]. Therefor, a neighborhood structure is required which

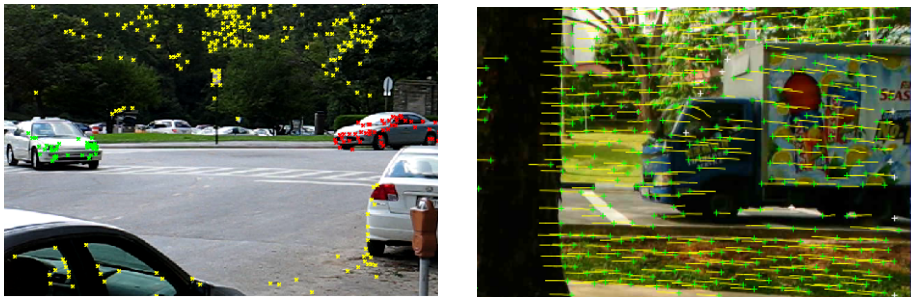

**Fig. 1.** Feature distribution for the example *cars5* from the Hopkins155 benchmark [17] (left) and for the example *van* from the MPTV benchmark [7] (right). For *cars5*, feature points on different objects mostly have large spatial distances while for *van*, features all features show small spatial distances, but different motion vectors.

guides the hypothesis sampling. The neighborhood structure is encoded in the auxiliary graph. In [15], the auxiliary graph is based on spatial proximity. The idea is that adjacent trajectories are likely to belong to the same motion segment. Thus, it is advantageous that they share the same hyperedge. Sampling from these hyperedges guarantee a relatively high probability of selecting candidates from the same motion segment, so-called pure hyperedges. Experiments [15] show superior performance of large hyperedges for the applications face clustering (Yale face database [6]) and motion segmentation (Hopkins155 data set [17]).
For the evaluation of motion segmentation approaches the well known Hopkins155 benchmark [17] is widely used. It consists of image sequences and trajectory data together with ground truth information providing the motion segmentation result (cf. Figure 1, left: red, green, and yellow colors indicate three motion segments). All trajectories in the Hopkins benchmark consist of points which are visible in every frame of the sequence. Trajectories which discontinue within the sequence were deleted from the data set. Thus, only large trajectory lengths are provided in the benchmark. Since its publication in 2007, many approaches result in low segmentation errors on the Hopkins data set.
The MPTV benchmark [7] includes eight challenging sequences. Like in Hopkins155, a ground truth labeling of the trajectories is provided. In contrast to Hopkins155, feature tracks may start and end in any frame. Thus, trajectories of arbitrary length are included in MPTV. For applications such as motion estimation from a camera mounted on a car, short trajectories are of special importance since long trajectories are rarely available on moving objects.
We propose a method for building the auxiliary graph which incorporates the motion of a trajectory instead of using spatial proximity only [15]. The new auxiliary graph leads to improved hypotheses which lead to a decrease of the segmentation error on both benchmarks. Furthermore, less computation time is required since the complexity of the graph is reduced. The extension is evaluated on Hopkins155 and MTPV. For the evaluation, the sequences are divided into subsequences of length $f_s$ to

provide results for limited trajectory length. For the subsequences, all trajectories which are visible in all images are included in the evaluation. This leads to a large number of experiments for each sequence. For MPTV, this step is necessary since all trajectories have arbitrary length. To summarize, the contributions of this paper are as follows:

- an improved method for the generation of the auxiliary using motion cues from the trajectory data
- the evaluation on two reference benchmarks [7,17] based on limited trajectory lengths
- the evaluation of the applicability of the benchmarks [7,17] since Hopkins155 only includes long trajectories while MTPV provides arbitrary trajectory lengths

In the following Section 2, the hypergraph based approach of motion segmentation is introduced. In Section 3, the new auxiliary graph is presented. Section 4 shows experimental results. In Section 5, the paper is concluded.

## 2 Motion Segmentation based on Hypergraphs

We briefly review the reference method proposed in [15]. An overview diagram is shown in Figure 2. Based on the input data matrix $\mathtt{X}$, the *auxiliary graph* is generated. This graph guides the generation of hyperedge of the hypergraph $G^t$ at iteration $t$. The hypotheses are generated using the RCM (Random Cluster Model) [13] and Swendsen-Wang sampling [16]. The sampling procedure is designed for larger than minimal subsets and a large hyperedge degree [13]. After several iterations, the motion segmentation result is achieved based on a minimal subspace approximation error resulting from the subspace estimation.

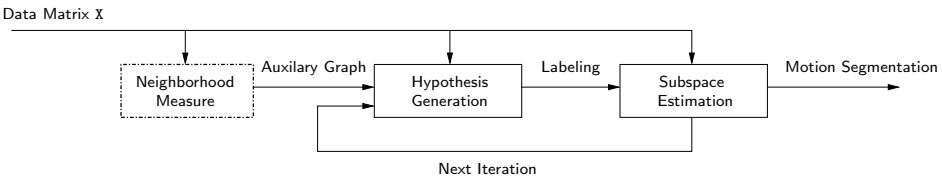


**Fig. 2.** Motion segmentation using hypergraphs (cf. Section 2). The hypotheses are generated using the auxiliary graph as neighborhood structure. The proposed neighborhood measure based on motion affinities exchanges the spatial proximity proposed in [15] in the box with the dotted border. It leads to improved hypotheses.

*Hypergraphs* A hypergraph $H = (V, E)$ consists of a set of vertices $V$ and a set of hyperedges $E$. In a weighted hypergraph $H$, a weight $w(e)$ is associated with each hyperedge $e$. The degree of a hyperedge $\delta(e)$ is defined as the number of vertices in edge $e$. The degree of a vertex $d(v)$

is the sum of all weights of the hyperedges incident with vertex $v$. If all hyperedges have the same degree $r$, the hypergraph is a $r$-uniform. A two-uniform hypergraph describes a common graph in which an edge connects two vertices.

### 2.1 Auxiliary Graph

For the RCM sampling, a spatial neighborhood structure is required. This structure is represented as a graph $G$. In [15], the spatial proximity, i.e. the Euclidean distance of trajectory vectors, is encoded in $G$.
Initially, mean subtracted coordinates of the trajectory vectors are computed from the data matrix `X`. In the case of motion segmentation, `X` consists of all point coordinates of a trajectory in the images. The graph is built by firstly using a PCA on the trajectory vectors to reduce the data dimension to $dim$ (default: $dim = 5$) followed by a k-nearest-neighbor algorithm (default: $k = 3$). The k-nearest-neighbor employs the Euclidean distance for each vector in the reduced data matrix $\mathtt{Y} = \{y_i\}_{i=i}^{N}$. Two vertices $v_i, v_j$ are connected if $v_j$ is a k-nearest neighbor of $v_i$ and vice versa. The distance between two data points determines the weight $p_e$ of the edge $e$ connecting these points as follows [15]:

$$p_e = exp\left(-\frac{||y_i - y_j||^2}{2\sigma_e^2}\right) \tag{1}$$

The weight $p_e$ indicates the probability of regarding two data points of the same structure. $\sigma_e$ is computed as the standard deviation of the nearest neighbor distances in `Y`. The auxiliary graph shares the same vertices as the hypergraph $G^0 = (V, E^0)$.

### 2.2 Hypothesis Generation using RCM

Hypothese are generated using the Random Cluster Model (RCM) [13] as well as Swendsen-Wang sampling (SWS) [16]. The RCM provides the partitioning of the auxiliary graph while SWS is an enhanced Monte Carlo sampling procedure [11]. The Swendsen-Wang approach [16] introduces a binary bond variable $d \in \{0, 1\}$ for each edge $e$ which can be turned on ($d = 1$) or off ($d = 0$). The vector $f = \{f_i\}_{i=1}^{N}$ represents labels of the vertices, $f \in \{1, \ldots, K\}$. A realization of $(f, d)$ effectively partitions the vertices into a set of connected components. Each connected component is a subset of $V$ such that all the bond variables between vertices in the component are turned on. Additionally, vertices in a connected component must have the same label. This leads to the graph $G^{(t)} = (V, E^{(t)})$ with $E^{(t)} = \{e =< i, j > | f_i^{(t)} = f_j^{(t)}\}$ at iteration $t$.
In [15], the vector $f$ and the samples $d$ are updated alternatingly. Given $f$, a number of samples $d$ provides a set of hyperedges. Given the hyperedges, $f$ is updated using NCut [21]. Each subcluster is a set of vertices connected by bond variables that have been turned on. Subclusters with a size less than a given hyperedge size $D$ ($D = 10$ by default) are removed. The newly generated subset is added to the set of all generated hyperedges. The iterative process is terminated when either the labels $f$ do

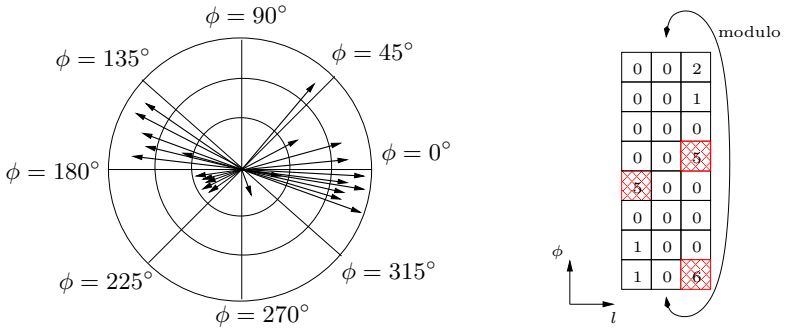


**Fig. 3.** Calculation of the dominant motions from a set of feature correspondences. The three dominant motions in this example are marked with red color.

not change significantly, or a maximum number of iterations is reached. Initially, the bond variables $d \in \{0, 1\}$ are determined probabilistically by comparing a random number $r \in [0; 1[$ with the edge weight $p_e$ (cf. equation (1)). If $r \leq$ the edge probability $p_e$, the bond variable is turned on.

### 2.3 Subspace Estimation

In the last step of each iteration, the subspace error of the current labeling $f$ is determined. The weight $w(e)$ of a hyperedge $e$ is calculated as:

$$w(e) = \begin{cases} \exp(\frac{-r^2(v,\phi_s)}{2\sigma^2}) & \text{if } v \notin s \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here, $s$ is a sampled $D-1$ tuple, and $v$ is an arbitrary vertex in $V$. $\phi_s$ is the model fitted in a least squares manner on $s$, and $r(v, \phi_s)$ is the residual of $v$ with respect to $\phi_s$. The parameter $\sigma$ is problem dependent and needs to be tuned [15]. It is determined as the sigma which gives the lowest subspace approximation error [3].

## 3 Auxiliary Graph based on Dominant Motions

The input data for the auxiliary graph is given by the correspondences of all feature points in the current image $\mathtt{I}_k$ to the corresponding feature in image $\mathtt{I}_{k-s}$. The auxiliary graph serves as initialization for the hypergraph $G^0$. Hypergraph edges between differently moving objects should be avoided. The reference algorithm [15] employs spatial proximity to generate the auxiliary graph. We argue that this is not an appropriate measure as shown in the example input sequence *van* in Figure 1, left. Many data point pairs have a low spatial distance from each other but belong to different motions, e.g. points on the trees in the background are near to points on the upper region of the van. But, their motion vectors are very different from each other (one motion pointing to the

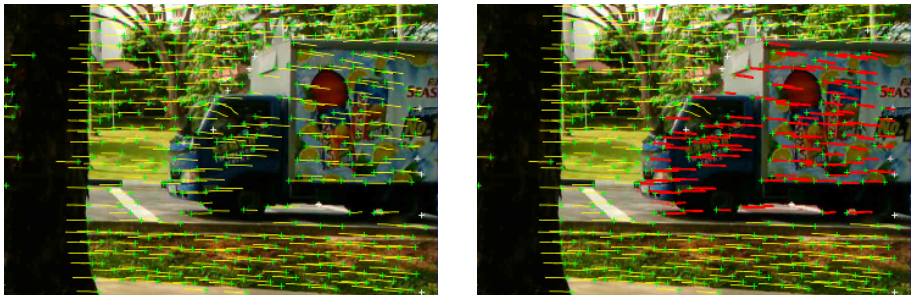

**Fig. 4.** Visualization of two dominant motions. The trajectories classified as in the histogram bins as dominant motions are marked with yellow and red color, respectively. The image subsequences is taken from the *van* sequence [7], cf. Figure 4, right.

left, the other pointing to the right). Thus, the motion of the trajectories should be incorporated in the computation of the auxiliary graph.

The proposed approach determines the dominant motions in the scene. The main idea is borrowed from the task of computing the main orientation for scale invariant features, such as SIFT [10]. In SIFT, the main orientation of a texture patch surrounding a feature is estimated using orientation histograms build from the angles of gradients. The angles are grouped into bins and the maximum is determined. The maximum gives the dominant orientation of a feature.

For our task of computing the dominant motions in the scene, we extend this idea to two dimensions, angle $\phi$ and length $l$ of a trajectory. The two dimensional histogram array of size $d_\phi \times d_l$ is shown in Figure 3. The angle $\phi$ is determined from corresponding points $\mathbf{x}_{i-s} = (x_{i-s}, y_{i-s})^T$, $\mathbf{x}_i = (x_i, y_i)^T$ as $\phi_i = \tan^{-1}(\frac{y_i - y_{i-s}}{x_i - x_{i-s}})$ while the length $l_i$ is $l_i = |\mathbf{x}_i - \mathbf{x}_{i-s}|$. The lengths $l_i$ are normalized using the maximum length of all trajectories. All correspondences are grouped into a two dimensional histogram array as shown in Figure 3. Then, local maxima are determined (marked with red color in Figure 3). The feature correspondences included in the maximum bin determine the dominant motion of the object. A threshold value $M_{thres}$ ensures that a dominant motion has at least $M_{thres}$ entries. Otherwise, it is discarded. The dimensions of the 2D histogram are given by parameters $d_\phi$ and $d_l$. In Figure 3, $d_\phi = 8$ and $d_l = 3$ are illustrated while in all experiments in this paper $d_\phi = 10$ and $d_l = 4$ are used. The number of minimum bin entries for the dominant motion is set to $M_{thres} = 8$.

The dimensions of the 2D histogram are given by parameters $d_\phi$ and $d_l$. In Figure 3, the configuration $d_\phi = 8$, $d_l = 3$ is shown while in all experiments in this paper $d_\phi = 16$ and $d_l = 8$ are used. The threshold $M_{thres}$ for the minimal number of trajectories to build a dominant motion cluster is set to $M_{thres} = 5$. An example for the determination of a dominant motion is shown in Figure 4. The trajectories of the dominant motion induced by the moving van is marked with red color while the dominant motion of the background is shown with yellow trajectories.

The information of trajectories located on the extracted dominant motions are incorporated in the generation of the auxiliary graph such that vertices in the auxiliary graph which belong to different dominant motions are not allowed to share an edge in the hypergraph (cf. Section 2). Thus, no hypotheses are generated from points on different dominant motions. This leads to a decreased number of samples required for a accurate segmentation result. Furthermore, the hypergraph has reduced complexity which leads to smaller computation times.

## 4 Experimental Results

The performance of the proposed approach is evaluated on the *Hopkins155* [17] database and the MTPV benchmark [7]. Both benchmarks provide image sequences together with annotated feature tracks as ground truth information. The benchmarks do not contain outliers. While the trajectories in the *Hopkins155* benchmark consist of points visible in every frame of the sequence (up to 53 frames), feature tracks may start and end in arbitrary frames in MTPV [7].
For the evaluation, the matlab code provided by the authors of [15] is used. The proposed approach using motion-coherent affinities is implemented as an extension of this code. We focus on the performance with limited trajectory lengths. For all evaluations, subsequences with a fixed number $f_s, f_s = 3, \ldots, 10$ of consecutive images are generated. Then, all trajectories of length $f_s$ in this subsequence are used as input data. To get a sufficient number of experiments per sequence, the next subsequence starts with a frame distance of $f_d$ ($f_d = 2$ used for all experiments). The segmentation error is defined as [17]:

$$\text{segmentation error} = \frac{\#\text{ of misclassified points}}{\text{total }\#\text{ of points}} \tag{3}$$

Additionally to the segmentation error mean, we report the median values. The segmentation results for Hopkins155 and MTPV are subsumed in Table 1 and Table 2, respectively. Generally, the mean segmentation

**Fig. 5.** Visualization of the auxiliary graphs for Pulak et al. [15] (left) and the proposed method (right). Edges between the bus (blue) and the background (red) are not established by the proposed method since their motions differ. Even spatially long edges are possible for small subgraphs. The image example is taken from the *bus* sequence [7].

**Table 1. Segmentation error** [%] on benchmark **Hopkins155** [17] for Pulak et al. [15] and the proposed method. Bold numbers depict the lower error.

| | Mean | | Median | |
|---|---|---|---|---|
| $f_s$ | reference | proposed | reference | proposed |
| 3 | 7.96 | **7.03** | 0.84 | **0.67** |
| 4 | 5.76 | **5.32** | **0.0** | **0.0** |
| 5 | 4.94 | **4.71** | **0.0** | **0.0** |
| 6 | 4.64 | **4.36** | **0.0** | **0.0** |
| 7 | 3.79 | **3.61** | **0.0** | **0.0** |
| 8 | 4.03 | **3.58** | **0.0** | **0.0** |
| 9 | 3.76 | **3.75** | **0.0** | **0.0** |
| 10 | 3.87 | **3.83** | **0.0** | **0.0** |

error decreases with increasing $f_s$. While the improvement of the proposed method is small for Hopkins155, its performance is significantly better for the MPTV benchmark. The decrease in the mean segmentation error is larger than 30% for $f_s \in \{4, \ldots, 10\}$. For $f_s = 7$ the largest gain of 41 % is achieved.

Obviously, the reference auxiliary graph leads to a much lower probability of sampling pure hyperedges from the input data MTPV. Since only trajectories with a large lengths are included in Hopkins155, the spatial proximity appears to provide a suitable metric for the auxiliary graph. This is not the case for MTPV since trajectories with short lengths are are not excluded from this data set. Thus, trajectories on differently moving objects may have a small spatial distance in the images (cf. Figure 5). This is often the case in applications using trajectory data extracted from natural sequences since trajectories with small length are required to capture moving objects. Thus, we can conclude that: (1) our approach provides significantly improved results for the MTPV dataset (2) Hopkins155 may not be a suitable benchmark for the evaluation of motion segmentation approaches if short trajectory lengths are of interest.

**Table 2. Segmentation error** [%] on benchmark **MTPV** [7] for Pulak et al. [15] and the proposed method. Bold numbers depict the lower error.

| | Mean | | Median | |
|---|---|---|---|---|
| $f_s$ | reference | proposed | reference | proposed |
| 3 | 26.86 | **21.19** | 32.01 | **22.13** |
| 4 | 21.35 | **13.89** | 22.63 | **0.46** |
| 5 | 19.80 | **12.77** | 17.89 | **0.30** |
| 6 | 18.27 | **12.60** | 9.74 | **0.30** |
| 7 | 19.01 | **11.21** | 12.30 | **0.19** |
| 8 | 18.66 | **11.84** | 10.75 | **0.24** |
| 9 | 17.46 | **11.04** | 1.21 | **0.25** |
| 10 | 16.83 | **10.97** | 1.87 | **0.21** |

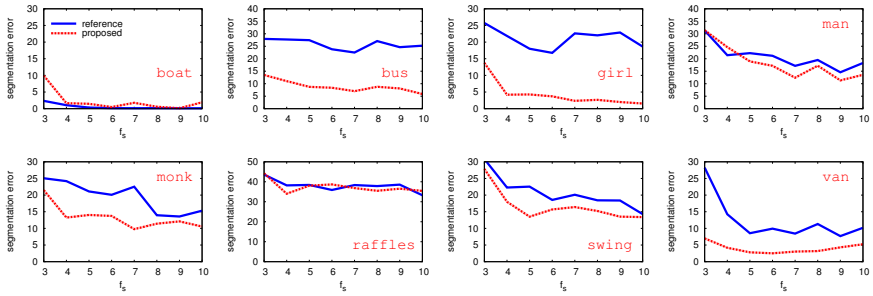


**Fig. 6.** Mean segmentation error [%] on the **MTPV** benchmark for each individual sequence with varying length of input trajectories $f_s$. In blue (solid line) the reference method [15] is shown, while in red (dashed line) the proposed auxiliary graph is employed. Note that the $y$ axes have different scalings.

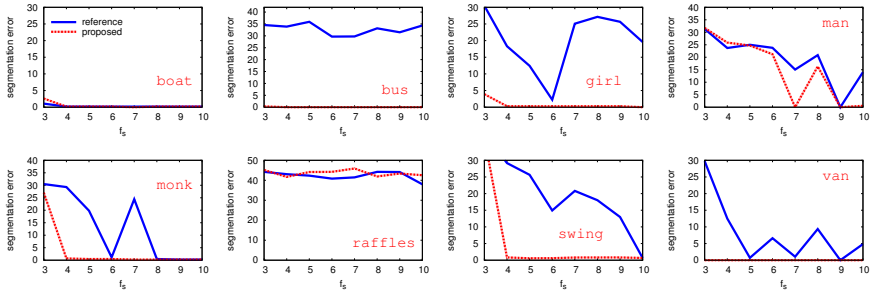


**Fig. 7.** Median of segmentation error [%] on the **MTPV** benchmark for each individual sequence with varying length of input trajectories $f_s$. In blue (solid line) the reference method [15] is shown, while in red (dashed line) the proposed auxiliary graph is employed. Note that the $y$ axes have different scalings.

The MTPV results for different trajectory lengths $f_s$ are shown in Figure 6 and Figure 7 in detail. For 6 of the 8 sequences, the proposed method leads to a significant decrease in the segmentation error. For the other 2 sequences (boat and raffles), the results are comparable. The sequence boat leads to small segmentation errors for both approaches. The sequence raffles fails for both methods. The proposed approach can not recover the motion because too few trajectories are located on the object and no dominant motion can be established.

The computation times are shown in Table 3. They are measured on a 3.20 GHz AMD Processor with Matlab2016. On the one hand, the computation time is heavily dependent on the number of trajectories. On the other, it is dependent on the complexity of the hypergraphs built at each iteration. As the hypergraphs resulting from the proposed auxiliary are less complex, the computation times for our method are much smaller than for the reference.

**Table 3. Computation time** in seconds for the a segmentation processing for Pulak et al. [15] and the proposed method. Bold numbers depict better values. The influence of $f_s$, $f_s \in \{3, \ldots, 10\}$, on the computation time is negligible.

| | Mean | | Median | |
|---|---|---|---|---|
| Benchmark | reference | proposed | reference | proposed |
| Hopkins | 8.83s | **2.49s** | 5.53s | **2.12s** |
| MPTV | 7.98s | **4.30s** | 4.26s | **2.75s** |

## 5 Conclusions

Hypergraph based approaches for motion segmentation reduce the number of required samples by exploiting higher order similarities for the hypothesis generation. The reference method proposed by Pulak et al. [15] uses an auxiliary graph which guides the sampling process. The auxiliary graph is based on spatial proximity of the trajectories.
The proposed approach uses a simple but effective scheme to combine motion direction and length of the trajectories for the determination of an improved auxiliary graph. Thus, less samples are required for accurate segmentation results.
The evaluation of the approaches employs two benchmarks, the well-known *Hopkins155* and the *MTPV* benchmark. While for Hopkins155, the proposed approach provides similar results regarding classification accuracy, it decreases the segmentation error significantly for MTPV. Since only long trajectories are included in *Hopkins155*, spatial proximity is an adequate measure to discriminate differently moving objects. It is shown that this measure is suboptimal for the *MTPV* benchmark. The proposed auxiliary provides a significantly lower segmentation error of up to 41%. Additionally, the computation time decreases since the resulting hypergraphs have reduced complexity.

## References


1. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI) 33(3), 500 –513 (2011)
2. Brox, T.: Higher order motion models and spectral clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 614–621. IEEE Computer Society (2012)
3. Chen, G., Lerman, G.: Spectral curvature clustering (SCC). International Journal of Computer Vision (IJCV) 81(3), 317–330 (2008)
4. Chin, T.J., Yu, J., Suter, D.: Accelerated hypothesis generation for multistructure data via preference analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI) 34(4), 625–638 (2012)
5. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. International Journal of Computer Vision (IJCV) 96(1), 1–27 (2012)

6. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI) 23(6), 643–660 (2001)
7. Li, Z., Guo, J., Cheong, L.F., Zhou, S.: Perspective motion segmentation via collaborative clustering. In: IEEE International Conference on Computer Vision (ICCV). pp. 1369–1376 (2013)
8. Li, Z., anf Loong-Fah Cheong, S.Y., Toh, K.C.: Simultaneous clustering and model selection for tensor affinities. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Liu, H., Yan, S.: Efficient structure detection via random consensus graph. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 574–581 (2012)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) 60(2), 91–110 (2004)
11. MacKay, D.J.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, chapter: The Swendson-Wang Method
12. Ozden, K., Schindler, K., Van Gool, L.: Simultaneous segmentation and 3d reconstruction of monocular image sequences. In: IEEE International Conference on Computer Vision (ICCV). pp. 1–8 (2007)
13. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI) 36(8), 1658–1671 (2014)
14. Poling, B., Lerman, G.: A new approach to two-view motion segmentation using global dimension minimization. International Journal of Computer Vision (IJCV) 108(3), 165–185 (2014)
15. Purkait, P., Chin, T.J., Ackermann, H., Suter, D.: Clustering with hypergraphs: The case for large hyperedges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) European Conference on Computer Vision (ECCV), LNCS, vol. 8692, pp. 672–687. Springer (2014)
16. Swendsen, R.H., Wang, J.S.: Nonuniversal critical dynamics in monte carlo simulations. Phys. Rev. Lett. 58, 86–88 (1987)
17. Tron, R., Vidal, R.: A benchmark for the comparison of 3d motion segmentation algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
18. Vidal, R., Ma, Y., Soatto, S., Sastry, S.: Two-view multibody structure from motion. International Journal of Computer Vision (IJCV) 68(1), 7–25 (2006)
19. Wang, H., Xiao, G., Yan, Y., Suter, D.: Mode-seeking on hypergraphs for robust geometric model fitting. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2902–2910 (2015)
20. Xiao, G., Wang, H., Lai, T., Suter, D.: Hypergraph modelling for geometric model fitting. Pattern Recognition 60, 748 – 760 (2016)
21. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: Advances in Neural Information Processing Systems (NIPS) 19. MIT Press (2006)