Segmentation of moving objects by long term video analysis

Peter Ochs, Jitendra Malik,Thomas Brox

Abstract—Motion is a strong cue for unsupervised object-level grouping. In this paper, we demonstrate that motion will be exploited most effectively, if it is regarded over larger time windows. Opposed to classical two-frame optical flow, point trajectories that span hundreds of frames are less susceptible to short term variations that hinder separating different objects. As a positive side effect, the resulting groupings are temporally consistent over a whole video shot, a property that requires tedious post-processing in the vast majority of existing approaches. We suggest working with a paradigm that starts with semi-dense motion cues first and that fills up textureless areas afterwards based on color. This paper also contributes the Freiburg-Berkeley motion segmentation (FBMS) dataset, a large, heterogeneous benchmark with 59 sequences and pixel-accurate ground truth annotation of moving objects.

Index Terms—motion segmentation, point trajectories, variational methods

1 Introduction

It has been shown that bottom-up segmentation based on color can successfully provide so-called superpixels – small, homogenous regions, which are actively used in many vision applications [4]. But what about the segmentation of whole objects or meaningful parts of objects? A person could wear clothes of very different color; see Fig. 1. How can a bottom-up approach decide which of these regions must be grouped together? Top-down object priors can resolve such ambiguities, but based on which data can these priors be learned in the first place?

In this paper, we reemphasize the value of motion and the Gestalt principle of “common fate” [34]. Motion vectors are typically more homogeneous within an object region than color and texture. Consequently, ambiguities in color based segmentation disappear as soon as objects move. Studies with formerly blind people indeed show that learning from moving objects is easier than learning from static ones [49].

However, most objects do not move permanently. There can be long periods during which an animal is as static as a pillar. Moreover, articulated objects do not move homogeneously. Arms and legs of a walking person move in opposite directions. All this causes severe problems in typical motion segmentation approaches based on two-frame optical flow. Tracking the interplay of the articulated parts over longer periods yields the missing information about the overall motion. Hence, in this paper, we argue that motion should be analyzed over longer periods. Such long term analysis decreases the motion’s intra-object variance relative to the inter-object variance. Moreover, motion information can be propagated to frames in which the object is mainly static.

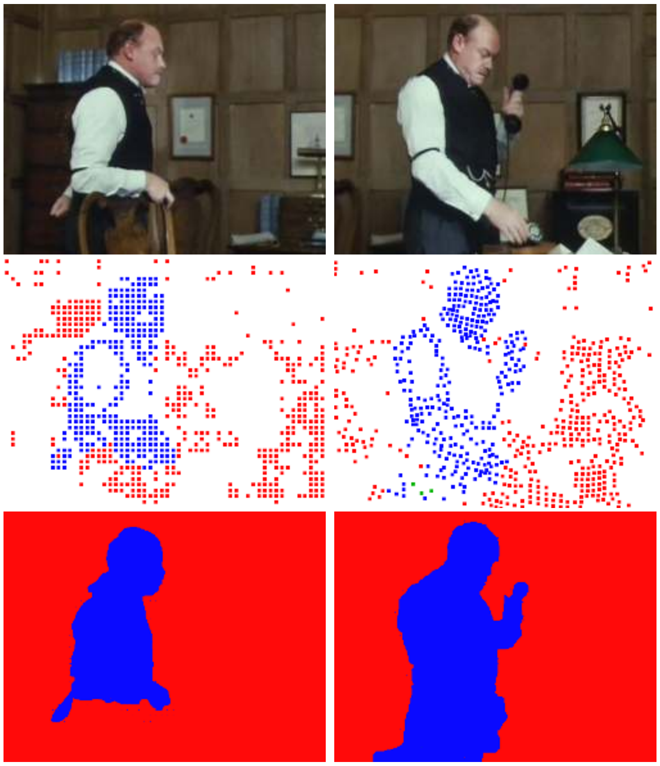


Fig. 1. **Top:** Two images from a video shot. Color based segmentation would not provide object regions. **Center:** Clustering of point trajectories indicates regions with similar motion. **Bottom:** Segmentation based on these clusters provides object regions.

Clearly, some kind of tracking is necessary for such long term analysis. We found that pursuing a “sparse to dense” strategy works best. Point tracking is more reliable than attempts to track superpixels, as stable features are located on

- *P. Ochs and T. Brox are with the Department of Computer Science and with the BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany. J. Malik is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley.*
E-mail: {ochs,brox}@cs.uni-freiburg.de, malik@eecs.berkeley.edu

This work was supported by a fellowship within the postdoc program of the German Academic Exchange Service (DAAD), by ONR MURI N00014-06-1-0734, and by the German Research Foundation (DFG) grant BR3815/5-1.

edges and corners rather than in flat areas. Between frames, superpixels can split or merge and significantly vary in shape. This is especially true near occlusion and disocclusion areas. We propose the use of a semi-dense point tracker based on optical flow that yields reliable trajectories for hundreds of frames with only little drift and that ensures a wide coverage of the video shot.

Trajectories are usually asynchronous, i.e., they start and end in different frames. This effect is mainly due to occlusion and disocclusion. Most existing trajectory clustering methods cannot deal with this problem. We define distances between all pairs of trajectories that share some common frames, which allows to deal with general scenes independent of how much occlusion is to be expected. The sparsity of the point trajectories is advantageous for the computational efficiency of the approach. It enables running spectral clustering [57] globally on all trajectories extracted from a video shot. Only after this powerful long term motion information has been exploited, we make use of the complementary color information to propagate object labels to the remaining homogenous areas in the video. This is achieved with a variational method.

The field of video segmentation, and motion segmentation in particular, has been lacking a dataset that is large enough to allow for credible quantitative evaluation. In our earlier conference publication a public benchmark dataset with pixel-accurate ground truth segmentation of moving objects was presented [17]. Here we extend this dataset with several new videos. These are a combination of short and long sequences at various resolutions. The dataset covers many typical challenges in motion segmentation, such as multiple objects, various types of motion, occlusion, and changing lighting conditions. The benchmark dataset comes with a tool for standardized evaluation in the motion segmentation scenario. We modified the original evaluation metric in [17] such that results with different granularities of over- and under-segmentation can be compared based on a single criterion, the F-measure.

2 Related work

The classical approach to motion segmentation is based on two-frame optical flow. While early approaches estimate the optical flow and the segmentation independently [69], [56], optical flow estimation and segmentation were later considered as a joint optimization problem [25], [3], [15], [63]. Obviously, object segmentation is only possible with such an approach, if the object motion is distinct and different from the background motion in *all* frames. Moreover, as pairs of frames are considered independently, the resulting segmentations are not consistent over time. [71], [50] approach these problems by combining motion analysis with a learned appearance model.

Above problems also disappear when the motion is analyzed over longer periods. There are many works that produce over-segmentations and connect the emerging superpixels over time using optical flow and/or clustering methods [12], [33], [67], [39]. This yields dense, temporally consistent segmentations, but usually they remain over-segmentations. It is not trivial to retrieve object regions from these results. Interactive video segmentation methods can avoid over-segmentation, but they require significant user input [6], [52].

While most of the above methods include some sort of region tracking, we rely on point tracking. The most popular point tracker is the Kanade-Lucas-Tomasi (KLT) tracker [58]. In the meantime, this tracker has been improved [73] and fast GPU versions are available [59], [73]. Other trackers are proposed in [9], [54]. We present a tracker that makes use of the dense motion field obtained with one of today's high quality variational optical flow methods [64]. We use large displacement optical flow [18], but it can be replaced by any other optical flow method. Compared to other trackers, the video can be covered densely, and fast motion of, e.g., body limbs does not lead to immediate tracking failure.

A dominant paradigm for clustering point trajectories has emerged from the technique of multi-body factorization. It decomposes the data matrix of the tracked point coordinates into a 3D rigid body motion and a structure matrix based on an affine camera model [23], [31], [10]. More recent methods model the (linear) dependency of the data samples [72], [26], i.e., motion segmentation is cast as the problem of segmenting samples drawn from a union of linear (or affine) subspaces. This allows definition of affinities between trajectories and the use of spectral clustering, as in the present work. For instance in [26] the dependency is modeled as an optimization program where data points express themselves as linear combinations with a sparsity prior on the representatives. In [37] the dimensionality of the ambient space is explored and affinities are defined using angular information. A few works also explored the projective dependency among the data samples [40], [55]. While initially all these techniques were very sensitive to noise, more recent models have solved this problem [26], [68], [41], [53], [42], [74]. However, the main limitation remains the requirement of a dominant subset of complete trajectories. Consequently, the methodology cannot deal with strong occlusion and disocclusion, which hampers sincere long term motion analysis.

There are few works which analyze point trajectories without the need to have a dominant subset of trajectories covering the full time line [61], [13], [22], [27]. Apart from [13], which analyzes trajectories but runs the clustering on a single frame basis, these methods provide temporally consistent clusters. The general idea of defining affinities between trajectories has been used in traffic scenarios already in 1997 [8]. Technically, however, all these methods are very different from our approach with regard to the density of trajectories, how the distance between trajectories is defined, and the algorithms used for clustering.

In some works, motion segmentation is not the primary goal but a way to achieve a higher level goal. Ommer et al. [48] couple motion segmentation with a recognition task, in [70] tracking and detection is combined with geometric information for 3D scene modeling, [14] focus on road scene understanding, and [62] reconstruct 3D point clouds for semantic segmentation and object recognition.

The part that converts the sparse trajectory clusters into dense spatio-temporal regions is related to interactive segmentation, where the user draws a few scribbles into the image and the approach propagates these labels to the non-marked areas. Several techniques based on graph cuts [11], random walks [32], and intermediate methods [60] have been proposed. The latest techniques are built upon variational convex relaxation methods [66], [19], [38], [45], which avoid the typical discretization artifacts of graph based formulations. The variational technique we propose here is in line with these methods.

The present paper builds upon three earlier conference papers. In [64] we proposed a dense point tracker based on variational optical flow. The clustering of such point trajectories was introduced in [17]. In [46] we presented a variational method that fills the empty gaps between the trajectories based on color and texture. These papers have led to several follow-up works by us [47] and other groups [39], [28], [29], [30], [75]. For the present paper we have consolidated the overall technique, particularly the MRF optimization on top of the eigenvectors of the affinity matrix in Section 5 and the variational model in Section 6. Moreover, we extend the benchmark dataset that came with [17]. Apart from adding more diverse sequences and providing a split into training and test set, we have improved the evaluation metric.

3 Point tracking with variational optical flow

The most important part of the presented object segmentation strategy is the long term aspect: we consider motion not independently for each frame but regard the whole motion *history* of a point to make a grouping decision. This requires point trajectories rather than just motion vectors. At the same time, we must avoid suffering from typical drawbacks of classical point tracking methods, such as the widely used KLT tracker [58]. Usually these trackers cover the image only very sparsely, have limited accuracy, and cannot deal with large motion of small, independently moving parts, such as arms and legs.

We obtain high quality point trajectories by using a very simple, but also very successful idea: we track points based on a current state-of-the-art optical flow method; here we use large displacement optical flow from [18]. This way, we benefit from all the progress made on optical flow estimation in the 30 years since the Lucas-Kanade method [43] was presented, which is the basis for the KLT tracker. In the following, we coarsely describe the most important aspects of the point tracker. For details we refer to [64]. The source code of the tracker is available at [1].

Initial points. Like in every tracker, a set of points is initialized in the first frame of a video. As we build on a dense optical flow method, in principle, we could initialize with every pixel. However, homogeneous areas can be problematic also for variational optical flow. To put more emphasis on points that can be tracked more reliably, we remove points that do not show any structure in their vicinity based on the smaller eigenvalue of the structure tensor.

As we will see in Section 5, the computational complexity of the motion segmentation method is quadratic in the number of point trajectories. For efficiency reasons, we spatially subsample the initial points. Fig. 2 shows a subsampling by factor 4 on the right and 16 on the left side. Factors larger than 12 lose details as there are not enough points to cover small object parts. On the other hand, factors smaller than 4 waste computation time, as smaller objects tend to be smoothed away by the optical flow anyway.

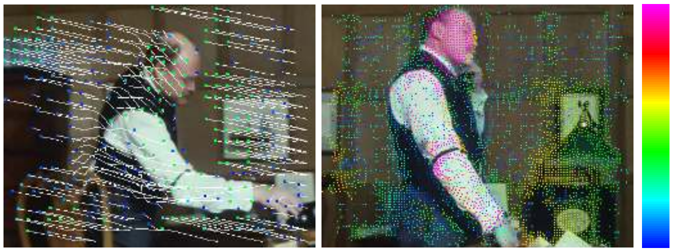


Fig. 2. Two visualizations of the trajectories. **Left:** Current position of the tracked points together with their trajectories. The subsampling factor was 16. **Right:** Only the current position of the tracked points is shown. The subsampling factor was 4. In both cases color shows for how long the points have been tracked as a percentage of the length of the shot (see color bar). Occlusion and disocclusion prohibit permanent tracking.

Tracking. Each of the points can be tracked to the next frame $t+1$ by using the optical flow field $\mathbf{w}_t$ at frame t. In principle, any optical flow method can be used here, yet many of the problems we find in motion segmentation are due to shortcomings of the optical flow, e.g., large displacements, sharp discontinuities, and accuracy for the occlusion detection. Hence, it is important to use a strong method. The approach from [18] combines the subpixel accuracy of variational approaches with combinatorial feature matching, which allows to capture large displacements. Moreover, an efficient GPU implementation [64] computes the optical flow between two 640×480 frames in less than 2 seconds. This enables tracking also in long, high resolution sequences in reasonable time.

Occlusion detection. Tracking has to be stopped as soon as a point gets occluded. This is very important, as otherwise the point trajectory will share the motion of two different objects. Occlusion detection is a common problem, considered especially in disparity estimation, but recently has appeared also more often in conjunction with optical flow. We refer to a recent work [5] and the references therein. In tracking, occlusion is usually detected by comparing the appearance of the local neighborhood of the tracked point over time. In contrast, we detect occlusions by verifying the consistency of the forward and the backward flow, as illustrated in Fig. 3. In a non-occlusion case, the backward flow vector points in the inverse direction of the forward flow vector. If this consistency requirement is not satisfied, the point is either getting occluded at $t+1$ or the flow was not correctly estimated. Both are good reasons to stop tracking this point at t. Since there are always some small estimation errors in the optical flow, we grant a tolerance interval.

Occlusion comes together with the opposite phenomenon: disocclusion or scaling. To fill these areas not covered by a trajectory yet, new trajectories are initialized in empty areas in

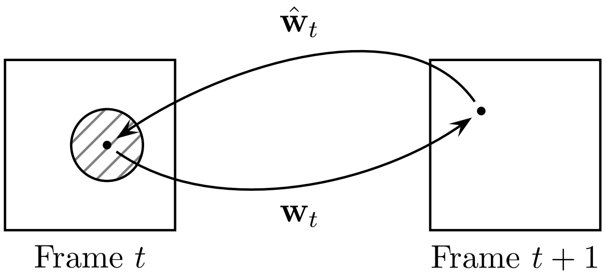


Fig. 3. Forward-backward matching criterion. Each pixel in frame t is mapped to frame $t+1$ via the optical flow vector $\mathbf{w}_t$. The backward map $\hat{\mathbf{w}}_t$ at the subpixel position is determined by bilinear interpolation. Concatenating the two mappings should result in approximately the original position.

each new frame using the same strategy as for the first frame.

A tracking example is shown in Fig. 2. Some points on the man are tracked across all 75 frames. However, most trajectories are newer due to disocclusion.

4 Affinities between trajectories

Clearly, trajectories of longer video shots are asynchronous, i.e., they cover different temporal windows in a shot. The set of points that are tracked across the whole scene is small or empty due to occlusion and disocclusion. A measurement matrix that takes the coordinates of the tracked points in all frames, as used by all multi-body factorization and linear subspace methods, will have many missing entries. We avoid a measurement matrix and rather set up pairwise affinities between trajectories. This only requires *some* trajectories to have *some* temporal overlap.

We define affinities between all pairs of trajectories that share at least one frame. They define the edge weights of a graph with trajectories as vertices. Trajectory pairs without overlap are assigned zero affinity. The emerging weighted graph is the basis for a grouping with spectral clustering. This way, even trajectories that do not share frames can get transitively connected via other trajectories.

According to the Gestalt principle of common fate [34], we should assign high affinities to pairs of points that move together. Clearly, there are many situations where this principle fails to segment objects. Two persons walking next to each other share the same motion although they are different objects. A person sitting in a chair shares the same motion as objects in the background. The Gestalt principle tells us that these situations should be treated conservatively and objects should not be separated. In motion segmentation based on two-frame optical flow, objects indeed cannot be separated in most frames as long as they do not show different motion permanently. At this point, the long term aspect of trajectories is important: as we are not forced any longer to make decisions for each frame independently, we can pick for each pair of trajectories the time instant where the motion is maximally different. According to the Gestalt principle, this instant provides maximum evidence that the two points do *not* belong to the same object. A man can sit in his chair for 1000 frames, but as he stands up, the motion difference provides the evidence that he is not part of the chair (Fig. 4).

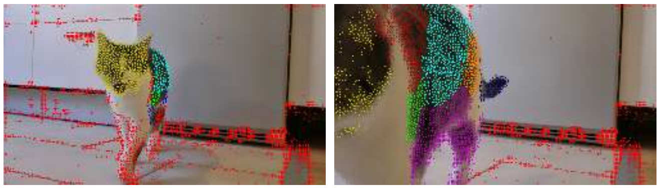


Fig. 5. Sequence with dominant scaling motion that cannot be captured well enough by a local translational model. These situations lead to an over-segmentation of the object.

Let A and B be two trajectories with coordinates (x_t^A, y_t^A) and (x_t^B, y_t^B), respectively, at frame t. According to the previous discussion, we define distances $d(A,B)$ for each pair of trajectories A and B exploiting their maximal dissimilarity, i.e., the maximal motion difference among all frames of common visibility

$$d^2(A,B) = \max_t d_t(A,B), \qquad (1)$$

and turn them into affinities via

$$w(A,B) = \exp\left(-\lambda d^2(A,B)\right). \qquad (2)$$

We fix the scale parameter $\lambda = 0.1$. This parameter brings us to another important issue, namely proper normalization. So far the affinity model is based on the two assumptions that the motion estimates are noise-free and all object motion is translational. Of course, both assumptions are not satisfied in real sequences, so we face the problematic question: when is a motion difference just due to noise and when is it significant enough to indicate different objects?

As this question is easier to answer if there is less noise, the first objective is to limit the noise that should be expected. On the side of the optical flow, we can add some more accuracy by averaging the motion over time. This is done by approximating the derivatives $\partial_t A$ and $\partial_t B$ of two continuous spatially temporal curves, defined by trajectories A and B, at time t with forward-differences over $T = 5$ frames:

$$\partial_t A = \frac{1}{T}(x_{t+T}^A - x_t^A, y_{t+T}^A - y_t^A)^\top \qquad (3)$$

The same for B. If less than T common frames are available between A and B, then T is set to the number of common frames for this pair. The exact choice of T is not critical. If T is chosen too large, we might lose relevant motion differences, e.g., due to a swinging arm. At frame rates of 30fps, $T = 5$ corresponds to just 160ms, and it is unlikely that this will smooth out some significant motion detail. We tested also values of $T = 10$ and $T = 15$ without any consistent positive or negative effect on the results[1].

[1] It is worth noting that temporal smoothing of the optical flow for the purpose of computing motion differences between trajectories is uncritical, whereas such smoothing can have very negative effects on the optical estimation process in case of camera jitter. The reason is that temporal smoothing during optical flow estimation hampers the correct matching of pixels. Such a problem does not exist when analyzing motion differences.

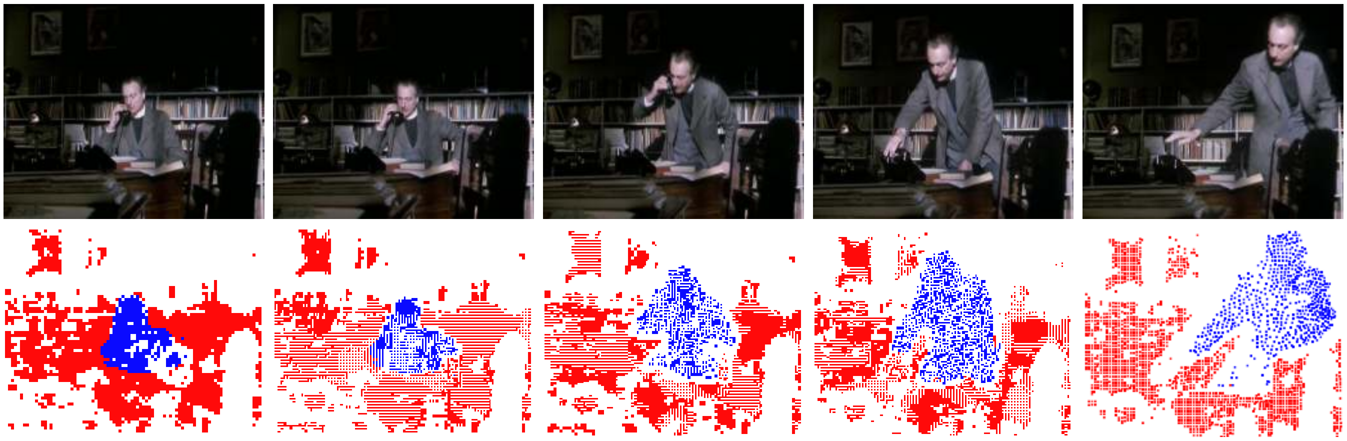


Fig. 4. Frames 0, 30, 50, 80, 93 of a shot from *Miss Marple: Murder at the vicarage*. Up to frame 30, there is hardly any motion as the person is sitting. Most information is provided when the person is standing up. This is exploited in the present approach. Due to long term tracking, the grouping information is also available at the first frames.

A second source of noise is model noise due to the assumption that all object motion is translational. Clearly, objects can undergo more complex motion. Fig. 5 shows a failure case, where motion is dominated by scaling. Pairwise distances only allow for verification of a translational model, whereas an affine motion model would require distances computed for at least 4 trajectories at a time to verify if they belong to the same group. This leads to a hypergraph and has been considered in [47]. Here we rather make use of the dense coverage of the image by trajectories and the fact that *locally* a higher order motion model can be approximated by a translational model. Consequently, we can limit the effect of model noise by damping the motion distance with the average spatial distance $d_{\text{sp}}(A, B)$ between trajectories A and B.

The distance at frame t between two trajectories A and B is defined as

$$d_t^2(A, B) = d_{\text{sp}}(A, B)\frac{|\partial_t A - \partial_t B|^2}{\sigma_t^2}, \tag{4}$$

where σ_t^2 is a locally adaptive normalization factor that deals with the fact that despite the above measures there is still some noise to be expected. The magnitude of the noise depends on the variation of the motion in the image. A larger variation indicates fast higher order motion and hence more model noise. Consequently, the distance should be normalized by the variance of the optical flow in the considered image. The intuition behind this normalization is that a motion difference of two pixels is a lot when there is hardly any motion in a scene, whereas the same motion difference is negligible in a scene with fast motion.

If there is just one object and the background, normalization by the global flow variance is sufficient, yet consider a scenario with one fast object and one slowly moving object. For the fast object, σ should be large, otherwise the object might be split into multiple regions. For the slow object, σ should be smaller to avoid that the object is merged with the background. This dilemma can be avoided by using a spatially adaptive variance estimate that is computed for each point individually in a local neighborhood. For an efficient computation of such local statistics we refer to [16].

5 Spectral clustering with spatial regularity

The pairwise affinities for n trajectories result in an $n \times n$ affinity matrix W. An (approximately) optimal partitioning of the underlying graph is obtained via spectral clustering [57], [44]. Let $D = \operatorname{diag}(d_A | A = 1, \ldots, n)$ be the $n \times n$ diagonal matrix with entries $d_A = \sum_B w(A, B)$. The eigendecomposition of the normalized graph Laplacian reads

$$V^\top \Lambda V = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}. \tag{5}$$

We keep the eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_m$ corresponding to the $m+1$ smallest eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_m$ according to the threshold $\max_i \lambda_i < 0.2$. The trivial solution $\lambda_0 = 0$ with the constant eigenvector $\mathbf{v}_0$ is omitted. Since the number of objects is expected to be significantly smaller than the number of trajectories, i.e., $m \ll n$, the eigenvectors and eigenvalues can be efficiently computed using the Lanczos method in $\mathcal{O}(n^2)$. For further computation we normalize the eigenvectors' range to $[0, 1]$.

We also determine the number of clusters automatically (model selection). In the ideal case, i.e., clearly distinguished translational motion and very few tracking errors, we obtain m piecewise constant eigenvectors and clusters are easily obtained with k-means clustering. There is a large number of model selection criteria in the literature, such as BiC or AiC, to automatically choose K in k-means clustering. As long as sufficiently many eigenvectors are computed, which is usually guaranteed with our conservative threshold on λ, such model selection will find a good number of clusters.

However, often the eigenvectors are not piecewise constant, as shown in Fig. 6. Standard k-means clustering is not suited for this setting as smooth transitions in the eigenvectors get approximated by multiple constant functions and, thus, leads to over-segmentation. This has a strong negative effect on the correct choice of the number of clusters K.

As a remedy, we suggest minimizing an energy function that comprises a spatial regularity term. This regularity term

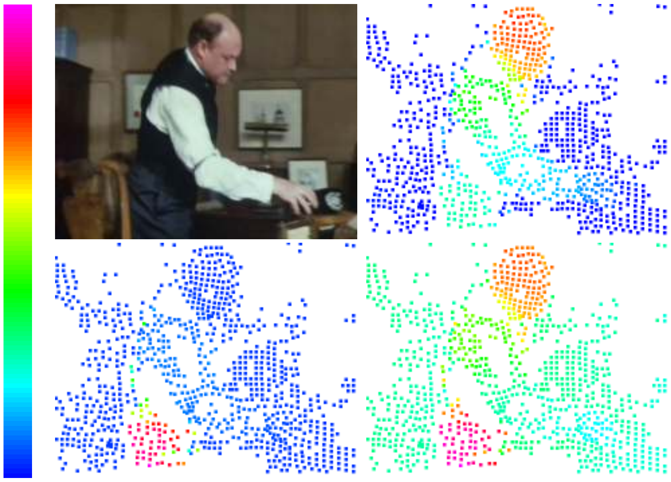


Fig. 6. **From left to right, top to bottom:** Input frame from a video shot and the first 3 eigenvectors with range of values represented by the colorbar. Clearly, the eigenvectors are not piecewise constant but show smooth transitions within the object regions. However, discontinuities in the eigenvectors correspond to object boundaries very well. This information needs to be exploited in the final clustering procedure.

not only prefers spatially compact clusters, it also acts as a criterion for model selection. Moreover, it takes edges in the eigenvectors into account. Let v_i^A denote the Ath component of the ith eigenvector and $\mathbf{v}^A$ the vector composed of the Ath components of all m eigenvectors. Index A corresponds to a distinct trajectory. Let $\mathcal{N}(A)$ be the symmetrized set of the 12 neighboring trajectories based on the average spatial distance of trajectories. We seek to choose the total number of clusters K and the assignments $\pi^A \in \{1, ..., K\}$ such that the following energy is minimized:

$$\begin{aligned} E(\pi, K) := & \sum_A \sum_{k=1}^{K} \delta_{\pi^A,k} |\mathbf{v}^A - \mu_k|_\lambda^2 \\ & + \nu \sum_A \sum_{B \in \mathcal{N}(A)} \frac{1 - \delta_{\pi^A,\pi^B}}{|\mathbf{v}^A - \mathbf{v}^B|}. \end{aligned} \tag{6}$$

The first term is the unary cost, where μ_k denotes the centroid of cluster k. The norm $|\cdot|_\lambda$ is defined as

$$|\mathbf{v}^A - \mu|_\lambda := \sum_i (v_i^A - \mu_i)^2 / \lambda_i, \tag{7}$$

i.e., each eigenvector is weighted by the inverse of the square root of its corresponding eigenvalue. This weighting is common in spectral clustering as eigenvectors that separate more distinct clusters correspond to smaller eigenvalues [7].

Clearly, if we do not add a penalty for additional clusters, each trajectory will be assigned its own cluster. The second term in (6) serves as a regularizer penalizing the spatial boundaries between clusters. The δ_{π^A,π^B} is the Kronecker delta, which is 1 if the trajectories A and B are assigned to the same cluster, and 0, else. The penalty is weighted by the inverse differences of the eigenvectors along these boundaries. Consequently, cutting a smooth transition in the eigenvectors will induce much higher cost than cutting along a strong discontinuity. This avoids splitting clusters at arbitrary locations due to smooth transitions in the eigenvectors. The parameter ν steers the tradeoff between the two terms. We obtain good results in various scenes by fixing $\nu = 60$.

Minimizing (6) is problematic due to many local minima. Fixing K, it becomes a multi-label MRF problem with unknown centroids. We start with an equidistant initial labeling, i.e., for $k \in \{1, \ldots, K\}$ all trajectories A_i with index $(k-1)\frac{n}{K} \leq i \leq k\frac{n}{K}$ are initially assigned $\pi^{A_i} = k$. In each iteration we update the centroids μ_k and optimize the label assignments using FastPD [35], [36]. We run at most 50 iterations and stop earlier as the energy decrease becomes less than 10^{-4}. We restrict the maximum number of objects per video shot to 20 and run this optimization for all $K \in \{1, ..., \min\{20, 2m\}\}$. Among all models we pick the one with the minimum energy.

Finally, we run a postprocessing step that merges clusters according to the mutual fit of their affine motion models estimated via least squares in each frame. We consider the average fit per frame and merge greedily until a threshold is reached for the average fitting error. This postprocessing step is not absolutely necessary, but corrects a few over-segmentation errors.

6 Dense Segmentation

The above clustering of point trajectories yields compact clusters of points that are by construction consistent over time. However, the approach so far does not yield a classical dense object segmentation, where each point is assigned to a region. In this section, we correct this shortcoming. While the trajectory clustering is focused on motion cues, the dense labeling brings in the complementary cues of color and texture and allows to decide on homogenous areas, where motion cues are not reliable. Our starting point is a labeled set of sparse trajectories, such as in Fig. 7b, where approximately 3% of the pixels are labeled. Depending on the trajectory sampling in the tracking even less pixels are labeled. We first present an approach that considers each frame independently. In Section 6.2 we introduce a model that enforces temporal regularity.

6.1 Variational label approximation

We cast the problem of making the sparse set of labels dense as optimization of a Potts model. The objective is to find a partitioning of the image domain $\Omega \subset \mathbb{R}^2$ into disjoint regions $E_1, \ldots, E_K \subset \Omega$, such that the region interface length Per and the cost for a weighting function $f = (f_1, \ldots, f_K)\colon \Omega \to \mathbb{R}^K$ becomes minimal. The generic Potts energy reads

$$\begin{aligned} & \min_{E_1,\ldots,E_K} \frac{1}{2} \sum_{k=1}^{K} Per(E_k;\Omega) + \sum_{k=1}^{K} \int_{E_k} f_k(x)\, dx \\ & \text{s.t.} \bigcup_{k=1}^{K} E_k = \Omega, \quad E_k \cap E_{k'} = \varnothing, \ \forall k \neq k'. \end{aligned} \tag{8}$$

In our case, the weighting function f in the data term is determined by the semi-sparse set of given labels. Defining

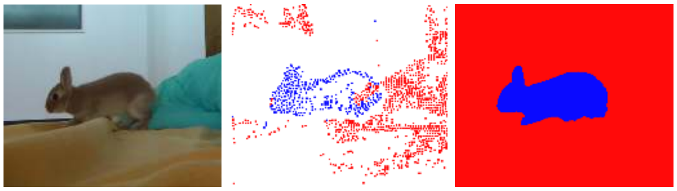


Fig. 7. Labels given by trajectories are penalized by a certain weight, and, thus, can be corrected in the variational optimization. **Left: (a)** Input image. **Center: (b)** Sparse point trajectory labels. **Right: (c)** Dense segmentation via minimization of a Potts model.

the set L_k of coordinates $x \in \Omega$ occupied by a trajectory with label k, let $f_k(x) = 0$ if $x \in L_k$ and $f_k(x) = \alpha$ otherwise. Thereby $\alpha \in \mathbb{R}^+$ is a positive weight, penalizing region $E_{k'}$ to contain $x \in L_k$, where $k \neq k'$. If $\alpha \to \infty$, regions are forced to enclose only points of a single label. As we can expect some erroneous trajectory labels smaller values for α are used and labels can be corrected, as demonstrated in Fig. 7. The α values may depend on the confidence in correctness of trajectory labels and on the cardinality of $\bigcup_{k=1}^{K} L_k$, i.e., the tracking subsampling factors 4, 8, or 16. Accordingly we set α to 200, 500, or 1000.

In order to practically minimize the Potts energy we rewrite it in terms of a convex total variation (TV) optimization problem [19]. The objective of finding a minimal partition is replaced by the minimization

$$\begin{aligned} &\min_u E(u) = \min_u TV(u) + \sum_{k=1}^{K} \int_\Omega u_k(x) f_k(x)\, dx \\ &\text{s.t. } u_k(x) \geq 0, \quad \forall k, \quad \sum_{k=1}^{K} u_k(x) = 1, \quad \forall x \in \Omega \end{aligned} \tag{9}$$

with respect to a label function $u = (u_1, \dots, u_K)\colon \Omega \to [0,1]^K$, where $TV(u)$ substitutes the measure for the region interface length. The value $u_k(x)$ can be though of as the probability of coordinate x taking label k.

Ideally, jumps in the label function should be located at image edges, i.e., where the image gradient $|\nabla I|$ is high. Therefore, we use image-driven weighted TV regularization

$$TV_g(u) = \frac{1}{2} \sum_{k=1}^{K} \int_\Omega g(x) |\nabla u_k(x)|\, dx, \tag{10}$$

where $g(x) = 1/\sqrt{|\nabla I|^2 + \varepsilon^2}$. The parameter $\varepsilon = 10^{-3}$ serves as a stabilizing parameter for homogeneous areas. This regularizer can be interpreted as ordinary total variation measure in the metric induced by the image as a Riemannian manifold.

In order to obtain the desired image partitioning, the minimizer u^* of the convex energy (9) has to be reprojected to the discrete label space $\{0,1\}^K$. We perform the projection by

$$u_k(x) = \begin{cases} 1, & \text{if } k = \text{argmax}_{k'} \{u^*_{k'}(x) | k' = 1, ..., K\} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

For the two-label case, the thresholding theorem [21] ensures global optimality of the solution with respect of the original Potts energy. In the general multi-label case, the integer solution usually is not a global optimum, but there is a computable tight upper bound [24].

The energy in (9) is minimized using Algorithm 1 from [20], a first order primal-dual algorithm. The algorithm is implemented on the GPU with CUDA. On a `GeForce GTX 580` it runs at about 1 frame per second for 680 × 480 images and 2 labels. The computation time scales linearly with the number of labels.

6.2 Temporal regularity

So far in this section, we treated each frame of the video independently. While the coarse result is consistent over time, thanks to the point trajectories, details near object boundaries may flicker. This can be avoided by extending the spatial total variation regularization to a spatio-temporal one

$$\begin{aligned} TV_{\text{temp}}(u) := \beta \int_\Omega c_{\mathbf{w}}^{(t)}(x) \sum_{k=1}^{K} \Big(&|u_k^{(t)}(x) - u_k^{(t+1)}(x + \mathbf{w}_t)|^2 \\ &+ |u_k^{(t)}(x + \hat{\mathbf{w}}_t) - u_k^{(t+1)}(x)|^2 \Big)\, dx. \end{aligned} \tag{12}$$

Function $u_k^{(t)}(x)$ indicates label k at time frame t at coordinate x, and $\mathbf{w}_t$ and $\hat{\mathbf{w}}_t$ refer to the forward and backward flow from frame t to $t+1$ and from $t+1$ to t, respectively. The binary function $c_{\mathbf{w}}^{(t)}(x)$ indicates whether the flow is reliable according to the consistency check of forward-backward matching as described in Section 3 (see Fig. 3). There is no need for a color based weighting, since optical flow links only pixels of similar color. The parameter β weights spatial against temporal regularization and can be chosen according to the desired amount of temporal smoothness.

7 Experimental Evaluation

7.1 Dataset

The field of motion segmentation lacks a sufficiently large and realistic benchmark dataset. There is the Hopkins 155 benchmark [65], but it focuses on short sequences with little occlusion and allows evaluation only of sparse, complete trajectories. The trajectories do not comprise outliers. In our previous work [17], we presented a dataset composed of 26 video sequences, among them shots from detective stories and 12 sequences from Hopkins 155. Several frames of each shot come with pixel-accurate ground truth segmentation of moving objects. The ground truth annotation is consistent over time.

We extended this dataset by adding 33 sequences. The new sequences show more variation in image resolution and comprise more non-translational motion than the previous sequences. Every 20th frame comes with ground truth, adding a total of 516 annotated frames to the benchmark. The full dataset with 59 sequences and 720 annotated frames is publicly available at [2].

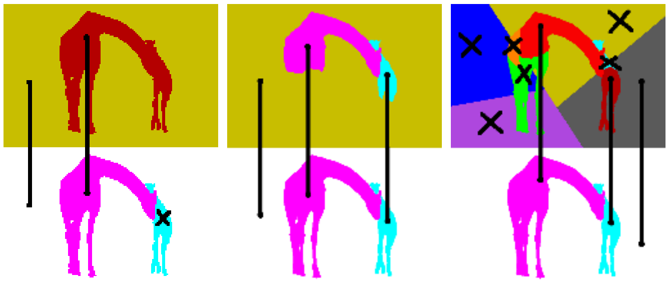


Fig. 8. Illustration of the evaluation metric for **(left to right)** an under-segmentation, an unprecise segmentation, and an over-segmentation. The ground truth is shown in the bottom row. The black lines show the cluster-region assignments and crosses indicates clusters or regions that have not been assigned. Average precision, recall, and F-measure are $(93.68\%, 66.67\%, 77.9\%)$, $(98.22\%, 80.31\%, 88.36\%)$, and $(100.0\%, 56.02\%, 71.81\%)$.

7.2 Evaluation method

Compared to [17] we also improved the evaluation methodology. The size of the new dataset allows to split it into a training set and a test set. We provide a fixed split into a roughly equal number of sequences in both sets. The split was chosen such that typical challenges appear in both sets.

We introduce an **average region density**, which is the average percentage coverage over all ground truth regions by labels. For a dense method as described in Section 6 the density is 100%; sparse trajectory clustering leads to lower densities depending on the spacing of the trajectories. By averaging over region densities rather than on a per-pixel basis, we penalize uneven spatial coverage.

To compare segmentations with different numbers of output regions, a metric must reflect the tradeoff between accuracy (usually maximized by increasing the number of regions) and a good coverage of the ground truth. In detection tasks, precision and recall have proven valuable to capture a similar tradeoff between false positives and misses. Here we provide a definition of precision and recall for segmentation. Let C be the set of pixels[2] labeled by the computer algorithm and $c_i \subset C$ the subset assigned to cluster i; $g_j \subset C$ be the corresponding subset of a ground truth region j, and let $|\cdot|$ denote the size of the set. The sets g_j only contain those pixels of a ground truth region that have been labeled by the evaluated computer algorithm. This allows the comparison of sparse and dense results on the basis of accuracy, whereas the density is measured by above density measure. **Precision** is defined as

$$P_{i,j} := \frac{|c_i \cap g_j|}{|c_i|}, \tag{13}$$

the ground truth fraction of a cluster, and **recall** as

$$R_{i,j} := \frac{|c_i \cap g_j|}{|g_j|}, \tag{14}$$

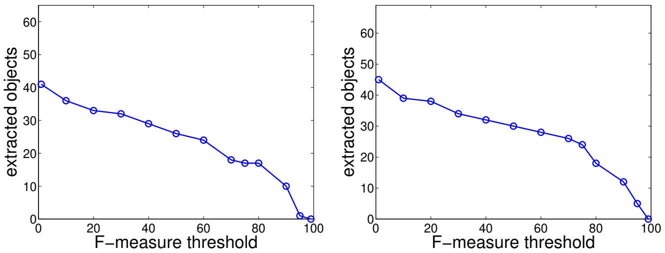


Fig. 9. Justification for the F-measure threshold of 75%. Results are obtained with the sparse method and trajectory sampling 8 for the **training set** (**left**) and the **test set** (**right**). With a larger threshold, only very few objects pass the criterion, with a smaller threshold, regions are allowed to be too inaccurate for being considered a meaningful object region; see also Fig. 10.

the fraction of a ground truth region covered by the cluster. They are defined for each pair of cluster i and ground truth region j. The best assignment of clusters to ground truth regions is found by the Hungarian method[3], a one-to-one matching algorithm, where we maximize the **F-measure**

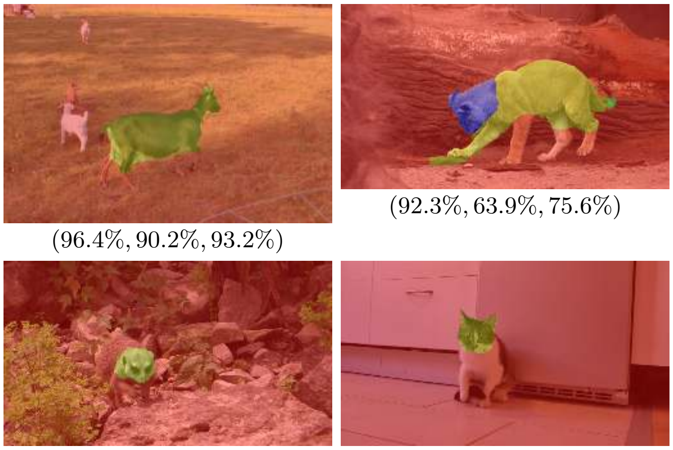
$$F_{i,j} := \frac{2P_{i,j}R_{i,j}}{P_{i,j} + R_{i,j}} \tag{15}$$

over all assignments. In case there are fewer clusters than ground truth regions, we introduce empty clusters. According to (14) their recall is $R = 0$, and we define $P = 1$. Unassigned clusters are ignored. Like in a typical detection setting, precision measures the percentage of correctly assigned pixels, and recall measures the covered fraction of the ground truth. However, since regions in a segmentation are disjoint, a lower recall usually does not increase precision, as in a typical detection setting. Fig. 8 illustrates the effect of certain error classes on the metrics. Both an under-segmentation (leftmost example) and an over-segmentation (rightmost example) lead to a reduction in recall. In the first case because one object is missed and obtains $R = 0$, in the second case because the assigned cluster covers only a small part of the ground truth region. Recall is mainly affected by a bad model selection. Precision is mostly affected by inaccurate clusters that overlap with multiple ground truth regions. The F-measure combines precision and recall and allows the comparison of approaches that yield different numbers of clusters, whereas with the evaluation metric in [17], it was unclear whether to prefer a result with lower pixel error or one with lower over-segmentation error. It is important to note that averages over precision and recall values are computed on a per region basis rather than on a per pixel basis. The latter would put too much weight on large background regions. The average F-measure always refers to the harmonic mean of the average precision and average recall rather than the average of single region F-measures.

Finally, we report the total number of **extracted objects**, which we define as clusters with F-measure $\geq 75\%$. This

[2] The set of pixels includes all frames with ground truth annotation.

[3] The complexity of the rectangular Hungarian method is $\mathcal{O}(mn^2)$. Since the number of ground truth regions is fixed and limited, we take n as the number of ground truth regions and m as the number of clusters.



$(96.4\%, 90.2\%, 93.2\%)$

$(92.3\%, 63.9\%, 75.6\%)$

$(99.3\%, 36.6\%, 53.5\%)$

$(99.2\%, 28.1\%, 43.8\%)$

Fig. 10. Qualitative justification for the F-measure threshold of 75%. Result for the first 10 frames and evaluation only on the first frame. **From left to right, top to bottom:** Dense result based on a trajectory sampling of 4 overlayed to the first frame of the sequence goats01, lion01, meerkats01, and cats02 with precision, recall, F-measure of the yellow region assignment. The F-measure for the lion is just above the threshold. Only the heads of the meerkat and cat are covered. Consequently, their F-measures are well below the threshold.

quantity is to indicate the number of objects that can be extracted with a certain accuracy from a dataset. One region is subtracted per sequence to account for the background, i.e., at least two regions must satisfy the above criterion to increase the counter. We refer to Fig. 9 and Fig. 10 to justify the threshold of 75%, which is in any case disputable and could be adapted to the quality requirements of an application.

7.3 Experimental setup

In order to demonstrate that there are challenges in the field of motion segmentation that cannot be properly handled by any previous methodology, we evaluated the proposed approach together with the factorization method in [53], which can deal with incomplete trajectories (ALC), the current state-of-the-art among subspace clustering methods: sparse subspace clustering [26] (SSC$_1$ and SSC), and a naive baseline method based on two-frame optical flow (Naive). SSC is the standard SSC and SSC$_1$ is an embedding of SSC into our motion segmentation framework where the only difference to our method is in the computation of the affinity matrix.

For ALC and SSC the correct number of labels is provided. Whereas SSC yields a segmentation with exactly this number of labels, ALC uses this number just as a prior. SSC$_1$ uses the model selection strategy from our framework. For all these methods, the same trajectories with an 8 sampling were used as input. In case of ALC, we randomly subsampled these trajectories by another factor 8 because the method is very slow. The trajectories for SSC$_1$ and SSC consist of the subset of complete trajectories only.

The baseline method was set up as follows: in the first frame, K reference flow vectors are chosen randomly, where K is set to the ground truth number of objects. All other flow vectors are assigned to the closest reference vector based on the Euclidean distance. In all further frames, the random flow vectors are replaced by the mean flow vector from the previous frame's segmentation with some inertia to account for noise.

For ALC we used the default parameters that came with the code. For all other methods we coarsely optimized the parameters on the training set by manual search before running the final version on the test set. Fig. 11 and Table 1 show the results on both datasets. The performance on the test and training set is quite similar. This shows that over-fitting is not an issue for the evaluated methods. A separate training and test set of reasonable size should avoid methods that over-fit also in the long run. By using the dataset, researchers must agree on running their method on the test set only once and on not using it for parameter optimization.

7.4 Quantitative results

The results in Table 1 and Fig. 11 show that the presented framework clearly outperforms all other methods. The main reason is that SSC$_1$, SSC, and ALC all cannot handle occlusions. SSC$_1$ and SSC work only on the subset of complete trajectories, which reduces the density but also the F-measure considerably. Many objects are missed completely, because they are not covered by *any* complete trajectories. While ALC can deal with incomplete trajectories, it fails when trajectories have little overlap. In contrast to SSC$_1$ and SSC, the density stays high, but the F-measure is not better. Also the baseline method based on two-frame optical flow performs poorly, especially on longer sequences. It cannot separate an object from the background if there is no clear motion difference in some frames or in case of articulated motion.

The presented dense segmentation inherits the temporal consistency from the long term analysis of sparse trajectories. A comparison to the sparse result shows that filling the gaps between the trajectories comes with a small loss in performance. Only with a very coarse subsampling of 16, performance drops significantly. Dense segmentation is clearly a harder task, as can be seen in Fig. 13. While the sparse segmentation just omits the difficult leg area of the cat, the dense segmentation is forced to decide for a label.

Since previous techniques have not been designed for very long sequences, we also show results for just the first 10 frames in Table 1. In this case, SSC$_1$ achieves approximately the same performance as the presented approach, while ALC and Naive still have problems.

We also evaluated the effect of model selection. On first glance it is surprising that the results with automatic model selection (SSC$_1$) are consistently better than those, where the correct number of objects is provided (SSC). We believe, this is due to imperfect trajectories that do not allow the detection of all ground truth objects. If affinities propose a bad segmentation, the constraint to yield a given number of clusters can be counterproductive, while automatic model selection adapts to such situations. We also ran a variant of our method (marked with *), where we replaced the optimization over K by the correct number of clusters. This number provides only

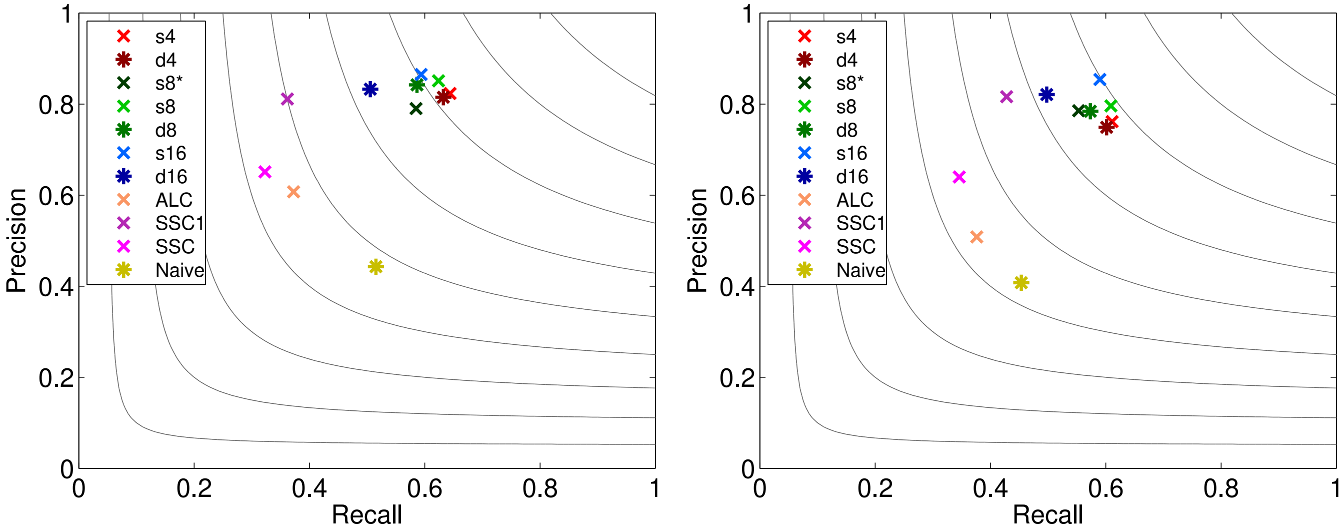


Fig. 11. Precision-recall graph for the **training set** (**left**) and the **test set** (**right**) for the evaluation on all sequences; see also Tab. 1. 's' denotes the sparse clustering result, 'd' the dense segmentation. The number indicates the subsampling rate of the trajectories. The proposed approach performs significantly better than previous approaches. Embedding affinities by SSC into the presented approach improves over the traditional SSC framework. This shows that the definition of affinities and the model selection framework both contribute to the performance.

an upper bound, since the regularization could still remove some of the K clusters. Also this version performed worse than the version with automatic model selection.

The density of trajectories has a similar effect on model selection as the regularization parameter ν: sparser trajectories lead to fewer clusters. This is because smaller object parts are no longer covered by sufficiently many trajectories to support a separate cluster. This reduces the over-segmentation of articulated objects like the bear in Fig. 12. On the other hand, smaller objects will be missed, if trajectories are too sparse. We found that decreasing the subsampling from 4 to 2 does not help capturing smaller objects anymore. A deeper analysis indicates that the optical flow is the limiting factor. Since for smaller objects the region area over the contour length gets smaller, misplaced discontinuities in the optical flow have a strong effect and hamper motion segmentation.

Trajectory subsampling has a positive effect on the computational cost. For subsampling rates of 4, 8, and 16, the average computation times of the clustering are 6s, 800ms, and 600ms per frame, respectively. Computation of the forward and backward flow takes on average 20s per frame on the CPU or 2s per frame on the GPU. The dense segmentation on average adds 1s per frame on the GPU. These computation times allow the application to large video datasets on commodity hardware.

7.5 Qualitative results

The qualitative results show that the method is applicable to a quite general set of sequences and can deal with many challenges. Fig. 12 highlights the possibility to deal with articulated motion. Strong articulation usually leads to an over-segmentation of the object, as the articulated parts are assigned to separate clusters. This can be a desired effect. The current parameter setting for the dense segmentation tends to smooth out these smaller parts, but these can be modified if necessary. A limitation of the method can be observed for feet that stay on the ground for a long time and then get occluded before they move. These limbs are assigned to the background because a true long term analysis of their motion is frustrated by the occlusion.

In contrast, partial occlusion of larger objects usually is not a problem, as shown in Fig. 13. Although the cat is occluded several times, the visible parts show sufficiently similar motion to keep the whole object in the same cluster. Only at the very end of the sequence, the overlap of trajectories is too small and the cluster gets split temporally. Also strong occlusion due to changing viewpoint can be handled, as shown in Fig. 14. The viewpoint changes by almost 180 degree, i.e., hardly any part of the horse in the first frame is still visible in the last one. Also the background changes completely. In contrast to many methods from literature, the proposed way to define affinities can handle this case easily.

The long term motion analysis can also deal with objects that are static for many frames, such as the sitting person in Fig. 4. Although the person is perfectly static at the beginning, it can be separated from the background by motion cues. The motion cues from the end of the sequence are successfully propagated to these first frames. For the method to work, it is only necessary that the object shows a different motion in at least one frame. We found that the method will *not* work, if there is a moving camera but the object is static in all frames. This is because at the footpoint, where the object is standing on the ground, the optical flow of foreground and background is identical, which leads to a leakage problem in spectral clustering.

We have also tested our model with temporal regularity from Section 6.2. The result videos look much better since there is

	Training set (29 sequences)					**Test set** (30 sequences)				
	D	P	R	F	F $\geq$ 75%	D	P	R	F	F $\geq$ 75%
	all frames					all frames				
MoSegSparse (4)	3.71%	82.33%	64.26%	72.27%	17/65	3.95%	76.15%	61.11%	67.81%	22/69
MoSegDense (4)	100.0%	81.50%	63.23%	71.21%	16/65	100.0%	74.91%	60.14%	66.72%	20/69
MoSegSparse (8)	0.87%	85.10%	62.40%	72.0%	17/65	0.92%	79.61%	60.91%	69.02%	24/69
MoSegSparse* (8)	0.87%	79.02%	58.49%	67.22%	14/65	0.92%	78.54%	55.29%	64.90%	16/69
MoSegDense (8)	100.0%	84.21%	58.67%	69.16%	15/65	100.0%	78.42%	57.32%	66.23%	17/69
MoSegSparse (16)	0.20%	86.51%	59.39%	70.43%	15/65	0.22%	85.41%	58.98%	69.77%	20/69
MoSegDense (16)	100.0%	83.27%	50.56%	62.91%	8/65	100.0%	82.08%	49.78%	61.97%	9/69
ALC	0.09%	60.73%	37.24%	46.18%	0/65	0.09%	50.83%	37.62%	43.24%	0/69
SSC_1	0.17%	81.11%	36.17%	50.03%	7/65	0.17%	81.62%	42.80%	56.16%	11/69
SSC	0.17%	65.12%	32.29%	43.17%	5/65	0.17%	63.98%	34.61%	44.92%	3/69
Naive	100.0%	44.29%	51.54%	47.64%	2/65	100.0%	40.77%	45.35%	42.94%	1/69
	first 10 frames					first 10 frames				
MoSegSparse (8)	0.95%	92.77%	65.44%	76.75%	16/53	0.97%	87.44%	60.77%	71.71%	19/55
MoSegDense (8)	100.0%	92.97%	63.18%	75.24%	13/53	100.0%	87.41%	58.73%	70.26%	14/55
ALC	0.12%	54.31%	54.80%	54.56%	8/53	0.12%	53.11%	56.40%	54.70%	5/55
SSC_1	0.89%	91.16%	63.34%	74.75%	14/53	0.88%	91.67%	50.57%	65.18%	10/55
SSC	0.89%	67.62%	73.04%	70.22%	13/53	0.88%	61.64%	60.63%	61.13%	11/55
Naive	100.0%	72.63%	51.63%	60.36%	2/53	100.0%	57.96%	53.41%	55.60%	2/55

TABLE 1

Results on training (left block) and test set (right block). Acronyms are **D**: average region density, **P**: average precision, **R**: average recall, **F**: F-measure and **F** $\geq 75\%$: extracted objects. Numbers are given for the sparse clustering and the dense segmentation with trajectory sampling rate given in parentheses. Details for ALC, SSC_1, SSC and Naive are discussed in the text.

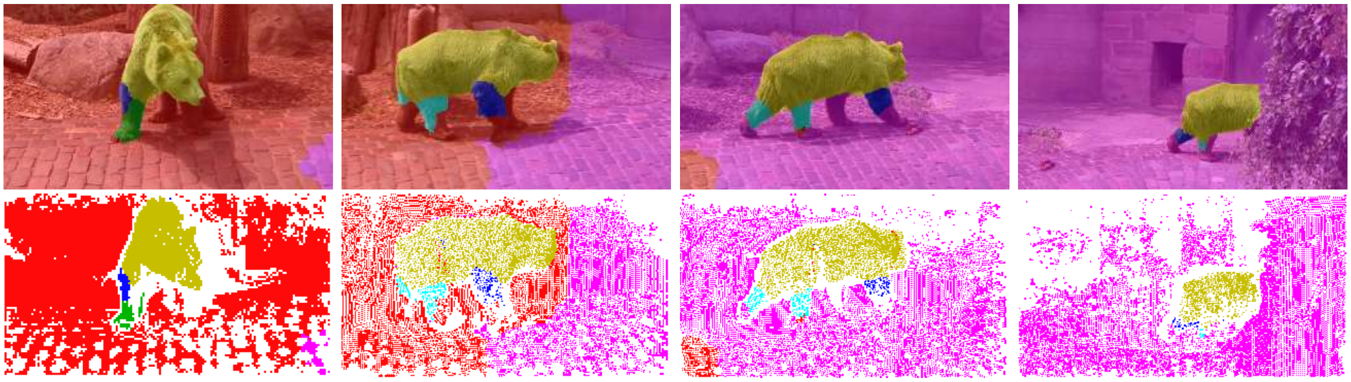


Fig. 12. Example of an articulated object from the benchmark dataset video shot bear02 with 458 frames and 24 annotated images. **Top row:** Dense segmentation obtained with the variational model from the point trajectories in the bottom row overlayed with the input images (precision: 98.1%, recall: 74.7%, F-measure: 84.8%). **Bottom row:** Clustered point trajectories (precision: 98.9%, recall: 78.4%, F-measure: 87.5%). Clearly, articulated motion leads to an over-segmentation of the object, yet the clusters could also indicate reasonable object parts.

no more flickering of labels, but there is not a significant effect on the quantitative numbers.

8 Summary

We have presented an approach for motion segmentation that exploits long term motion cues. Motion information is aggregated over the whole shot to assign labels also to objects that are static in a large part of the sequence. Occlusion and disocclusion is naturally handled by this approach, which allows to gather information about an object from multiple viewpoints. In contrast to video segmentation methods based on region tracking, we rely on point trajectories computed via optical flow. Focusing on areas in the image, where optical flow estimation works best, this makes tracking more reliable. A dense segmentation is obtained by a variational approach that fills the gaps between trajectories based on color. This strategy puts valuable long term motion information first and relies on color only in those areas, where motion is difficult to estimate. We provide a benchmark dataset for the general motion segmentation task with a variety of objects and resolutions. Results on the dataset look very promising and consistently outperform results with previous state-of-the-art methods. This is although only pairwise affinities have been considered, which restricts the motion model to be locally translational. In [47] the affinities have been extended to deal with higher-order motion models. This further improves results at the price of higher computational costs. In a recent work, the motion segmentation result of our method has already been

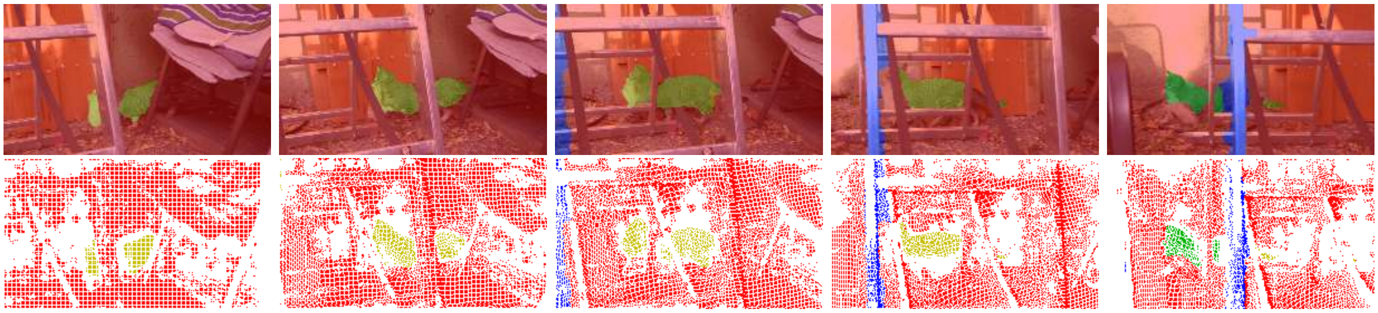


Fig. 13. Example of a partially occluded object from the video shot cats05 with 87 frames and 6 annotated images. **Top row:** Dense segmentation obtained with the variational model from the point trajectories in the bottom row overlayed with the input images (precision: 64.0%, recall: 52.8%, F-measure: 57.8%). **Bottom row:** Clustered point trajectories (precision: 65.0%, recall: 54.2%, F-measure: 59.1%). Although the trajectories on the cat are short due to occlusions, there is enough temporal overlap to assign trajectories on the cat to the same cluster across most of the obstacles.

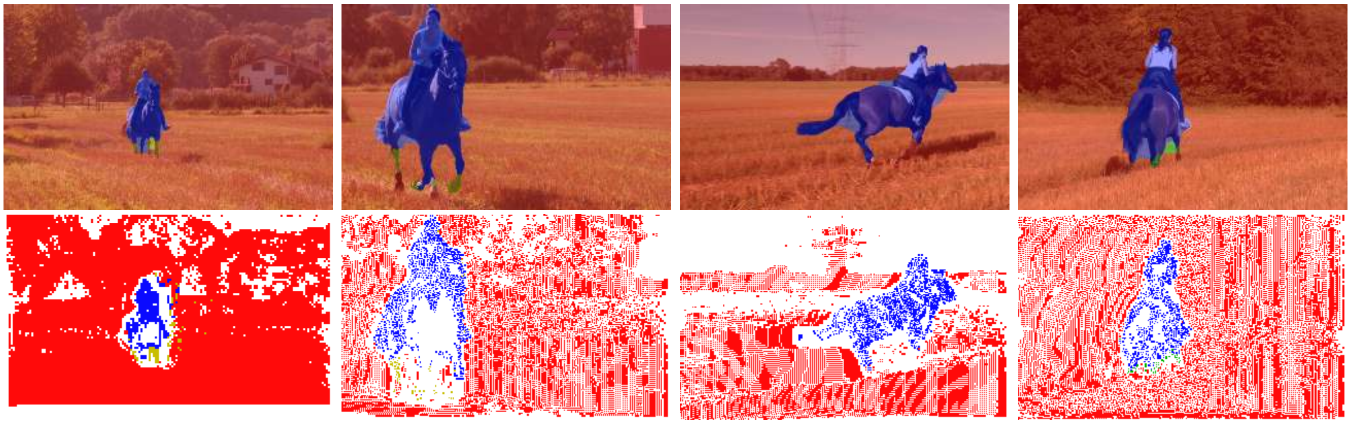


Fig. 14. Example of a sequence with large perspective background motion from the video shot horses01 with 500 frames and 26 annotated images. **Top row:** Dense segmentation obtained with the variational model from the point trajectories in the bottom row overlayed with the input images (precision: 94.8%, recall: 93.4%, F-measure: 94.1%). **Bottom row:** Clustered point trajectories (precision: 95.1%, recall: 95.3%, F-measure: 95.2%). The horse is seen from different viewpoints and the background changes completely. Nonetheless, there are two temporally consistent clusters, one for the horse and one for the background.

used to train object detectors from videos [51]. We hope that our work will continue fostering research in the field of motion segmentation.

Acknowledgements

We thank Thomas Pock for valuable comments on the variational model. We thank René Vidal for allowing us to use some of the sequences from Hopkins 155 in our benchmark dataset, and the reviewers for suggestions that improved the manuscript.

References

[1] Large displacement optical flow tracker. http://lmb.informatik.uni-freiburg.de/resources/binaries/eccv2010_trackingLinux64.zip.

[2] Motion segmentation dataset (FBMS). http://lmb.informatik.uni-freiburg.de/resources/datasets/FBMS_Trainingset.tar.gz and http://lmb.informatik.uni-freiburg.de/resources/datasets/FBMS_Testset.tar.gz.

[3] T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. *International Journal of Computer Vision*, 68(2):111–124, 2006.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011.

[5] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3):322–338, 2012.

[6] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *International Conference on Computer Vision (ICCV)*, 2007.

[7] S. Belongie and J. Malik. Finding boundaries in natural images: A new method using point descriptors and area completion. In *European Conference on Computer Vision (ECCV)*, volume 1406 of *Lecture Notes in Computer Science*, pages 751–766. Springer, 1998.

[8] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 495–501, 1997.

[9] S. Birchfield and S. Pundlik. Joint tracking of features and edges. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[10] T. Boult and L. Brown. Factorization-based segmentation of motions. In *In Proceedings of the IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[11] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[12] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *International Conference on Computer Vision (ICCV)*, 2009.

[13] G. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision (ECCV)*, pages 44–57, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conference on Computer Vision (ECCV)*, volume 3951 of *Lecture Notes in Computer Science*, pages 471–483. Springer, 2006.

[16] T. Brox and D. Cremers. On local region models and the statistical interpretation of the piecewise smooth Mumford-Shah functional. *International Journal of Computer Vision*, 84(2):184–193, 2009.

[17] T. Brox and J. Malik. Object segmentation by long-term analysis of point trajectories. In *European Conference on Computer Vision (ECCV)*, 2010.

[18] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:500–513, 2011.

[19] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Applied Mathematics*, 5(4):1113–1158, 2012.

[20] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[21] T. Chan, S. Esedoḡlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[22] A. Cheriyadat and R. Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *International Conference on Computer Vision (ICCV)*, 2009.

[23] J. Costeira and T. Kanande. A multi-body factorization method for motion analysis. In *International Conference on Computer Vision (ICCV)*, pages 1071–1076, 1995.

[24] D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[25] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[26] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, preprint, 2013.

[27] M. Fradet, P. Robert, and P. Pérez. Clustering point trajectories with various life-spans. In *Proc. European Conference on Visual Media Production*, 2009.

[28] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[29] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, Rhode Island, USA, 2012.

[30] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *European Conference on Computer Vision (ECCV)*, 2012.

[31] C. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.

[32] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[33] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[34] K. Koffka. *Principles of Gestalt Psychology*. Hartcourt Brace Jovanovich, New York, 1935.

[35] N. Komodakis and G. Tziritas. Approximate labeling via graph-cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1436–1453, 2007.

[36] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, Oct. 2008.

[37] F. Lauer and C. Schnörr. Spectral clustering of linear subspaces for motion segmentation. In *International Conference on Computer Vision (ICCV)*, 2009.

[38] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *International Conference on Computer Vision (ICCV)*, 2009.

[39] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[40] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

[41] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning (ICML)*, pages 663–670. Omnipress, 2010.

[42] R. Liu, Z. Lin, F. De la Torre, and Z. Su. Fixed-rank representation for unsupervised visual learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[43] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, Aug. 1981.

[44] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[45] C. Nieuwenhuis, B. Berkels, M. Rumpf, and D. Cremers. Interactive motion segmentation. In *Pattern Recognition - Proc. DAGM*, volume 6376 of *Lecture Notes in Computer Science*, pages 483–492. Springer, 2010.

[46] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *International Conference on Computer Vision (ICCV)*, 2011.

[47] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[48] B. Ommer, T. Mader, and J. Buhmann. Seeing the objects behind the dots: Recognition in videos from a moving camera. *International Journal of Computer Vision*, 83(1):57–71, 2009.

[49] Y. Ostrovsky, E. Meyers, S. Ganesh, U. Mathur, and P. Sinha. Visual parsing after recovery from blindness. *Psychological Science*, 20(12):1484–1491, 2009.

[50] M. Pawan Kumar, P. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76:301–319, 2008.

[51] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[52] B. L. Price, B. S. Morse, and S. Cohen. LIVEcut: learning-based interactive video segmentation by evaluation of multiple propagated cues. In *International Conference on Computer Vision (ICCV)*, 2009.

[53] S. R. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[54] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80:72–91, 2008.

[55] K. Schindler, D. Suter, and H. Wang. A model-selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision*, 79(2):159–177, 2008.

[56] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. 6th International Conference on Computer Vision*, pages 1154–1160, Bombay, India, Jan. 1998.

[57] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[58] J. Shi and C. Tomasi. Good features to track. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[59] S. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc. Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*, 22(1):207–217, 2011.

[60] A. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *International Conference on Computer Vision (ICCV)*, 2007.

[61] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *International Journal of Computer Vision*, 67(2):189–210, 2006.

[62] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference*. British Machine Vision Association, 2009.

[63] D. Sun, E. B. Sudderth, and M. Black. Layered segmentation and optical flow estimation over time. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[64] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. Springer, 2010.

[65] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[66] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. TVSeg - interactive total variation based image segmentation. In *British Machine Vision Conference*, 2008.

[67] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. Springer, 2010.

[68] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[69] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[70] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *Pattern Analysis and Machine Intelligence*, 35(4):882–897, 2013.

[71] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1644–1659, 2005.

[72] J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision (ECCV)*, volume 3954 of *Lecture Notes in Computer Science*, pages 94–106. Springer, 2006.

[73] C. Zach, D. Gallup, and J.-M. Frahm. Fast gain-adaptive KLT tracking on the GPU. *CVPR Workshop on Visual Computer Vision on GPUs*, pages 1–7, 2008.

[74] L. Zappella, X. Lladó, E. Provenzi, and J. Salvi. Enhanced local subspace affinity for feature-based motion segmentation. *Pattern Recognition*, 44(2):454–470, 2011.

[75] G. Zhang, Z. Yuan, D. Chen, Y. Liu, and N. Zheng. Video object segmentation by clustering region trajectories. In *International Conference on Pattern Recognition*, 2012.

Peter Ochs received the M.Sc. degree (honours degree) in Mathematics from Saarland University in Germany in 2010. Currently, he is working as a research assistant at the University of Freiburg in Germany. He spent three months as a visiting researcher at the TU-Graz in Austria. His research interests are in computer vision, where he focuses on mathematics, variational models, optimization, motion analysis, and segmentation in video data.

Jitendra Malik received the B.Tech degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, in 1980 and the PhD degree in Computer Science from Stanford University in 1985. In January 1986, he joined the University of California at Berkeley, where he is currently the Arthur J. Chick Professor in the Department of Electrical Engineering and Computer Sciences. He is also on the faculty of the Cognitive Science and Vision Science groups. He serves on the advisory board of Microsoft Research India, and on the Governing Body of IIIT Bangalore. His research interests are in computer vision, computational modeling of human vision, and analysis of biological images. His work has spanned a range of topics in vision including image segmentation, perceptual grouping, texture, stereopsis, and object recognition with applications to image based modeling and rendering in computer graphics, intelligent vehicle highway systems, and biological image analysis. He has authored or co-authored more than 150 research papers on these topics, and graduated 26 PhD students who occupy prominent places in academia and industry. According to Google Scholar, five of his papers have received more than a thousand citations each. He received the gold medal for the best graduating student in Electrical Engineering from IIT Kanpur in 1980 and a Presidential Young Investigator Award in 1989. At U.C. Berkeley, he was selected for the Diane S. McEntyre Award for Excellence in Teaching in 2000 and a Miller Research Professorship in 2001. He received the Distinguished Alumnus Award from IIT Kanpur in 2008 and was awarded the Longuet-Higgins Prize for a contribution that has stood the test of time twice, in 2007 and in 2008. He is a fellow of the IEEE and the ACM.

Thomas Brox received his Ph.D. degree in computer science from the Saarland University in Germany in 2005. He spent two years as a postdoctoral researcher at the University of Bonn and two years at the University of California at Berkeley. Since 2010, he is heading the Computer Vision Group at the University of Freiburg in Germany. His research interests are in computer vision, in particular video analysis and learning from videos. Prof. Brox is associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and the Image and Vision Computing journal. He was an area chair of ICCV 2011 and reviews for several funding organizations. In 2004, he received the Longuet-Higgins Best Paper Award at the European Conference on Computer Vision.