# Self-supervised Sparse to Dense Motion Segmentation

Amirhossein Kardoost[1][0000−0001−6477−6631], Kalun Ho[1,2,3], Peter Ochs[4][0000−0002−4880−7511], and Margret Keuper[1][0000−0002−8437−7993]

[1] Data and Web Science Group, University of Mannheim, Germany
[2] Fraunhofer Center Machine Learning, Germany
[3] Fraunhofer ITWM, Competence Center HPC, Kaiserslautern, Germany
[4] Mathematical Optimization Group, Saarland University, Germany

**Abstract.** Observable motion in videos can give rise to the definition of objects moving with respect to the scene. The task of segmenting such moving objects is referred to as motion segmentation and is usually tackled either by aggregating motion information in long, sparse point trajectories, or by directly producing per frame dense segmentations relying on large amounts of training data. In this paper, we propose a self supervised method to learn the densification of sparse motion segmentations from single video frames. While previous approaches towards motion segmentation build upon pre-training on large surrogate datasets and use dense motion information as an essential cue for the pixelwise segmentation, our model does not require pre-training and operates at test time on single frames. It can be trained in a sequence specific way to produce high quality dense segmentations from sparse and noisy input. We evaluate our method on the well-known motion segmentation datasets FBMS59 and DAVIS$_{16}$.

## 1 Introduction

The importance of motion for visual learning has been emphasized in recent years. Following the Gestalt principle of common fate [1], motion patterns within an object are often more homogeneous than its appearance, and therefore provide reliable cues for segmenting (moving) objects in a video. Motion segmentation is the task of segmenting motion patterns. This is in contrast to semantic segmentation, where one seeks to assign pixel-wise class labels in an image. Thus, for motion segmentation, we need motion information and at least two frames to be visible in order to distinguish between motion segments. Ideally, such motion patterns separate meaningful objects., e.g. an object moving w.r.t. the scene (refer to Fig.1). To illustrate the importance of motion segmentation, consider an autonomous driving scenario. A first step to classify potential danger caused by the presence of a possibly static object, e.g., a parking car, is the knowledge about its mobility. Unknowingly waiting for the observation that the door of the parking car suddenly opens may be too late to avoid an accident. The speed of the autonomously driving vehicle must be adjusted based on the mobility

Fig. 1: Motion segmentation example is provided for frames 1 and 160 of the "horses01" sequence (with their ground-truth motion segmentation) in the FBMS59 [2] dataset. Due to the reason that the person and the horse are moving together and have the same motion pattern, they are assigned the same motion label (blue color).

and danger of other objects. While models for the direct prediction of pixel-wise motion segmentations are highly accurate [3,4], they can only take very limited account of an objects motion history.

In this paper, we propose a model to produce high quality dense segmentations from sparse and noisy input (i.e. *densification*). It can be trained in a sequence specific way using sparse motion segmentations as training data, i.e. the densification model can be trained in a self-supervised way. Our approach is based on sparse (semi-dense) motion trajectories that are extracted from videos via optical flow (Fig. 2, center). Point trajectory based motion segmentation algorithms have proven to be robust and fast [2,5,6,7,8,9,10]. By a long term analysis of a whole video shot at once by the means of such trajectories, even objects that are static for most of the time and only move for few frames can be identified, i.e. the model would not "forget" that a car has been moving, even after it has been static for a while. The same argument allows articulated motion to be assigned to a common moving object.

In our approach we use object annotations that are generated using established, sparse motion segmentation techniques [5]. We also propose an alternative, a GRU-based multicut model, which allows to learn the similarity between the motion of two trajectories and potentially allows for a more flexible application. In both cases, the result is a sparse segmentation of video sequences, providing labels only for points lying on the original trajectories, e.g. every 8 pixels (compare Fig. 2). From such sparse segmentations, pixel-wise segmentations can be generated by variational methods [2]. In order to better leverage the consistency within the video sequences, we propose to train sequence specific densification models using only the sparse motion segmentations as labels.

Specifically, we train a U-Net like model [11] to predict dense segmentations from given images (Fig. 2), while we only have sparse and potentially noisy labels given by the trajectories. The training task can thus be interpreted as a label densification. Yet, the resulting model does not need any sparse labels at test time but can generalize to unseen frames.

In contrast to end-to-end motion segmentation methods such as e.g. [4], we are not restricted to binary labels but can distinguish between different motion

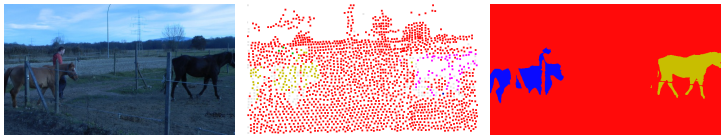

Fig. 2: Exemplary multi-label motion segmentation results showing *(left)* the image and its sparse *(middle)* and dense *(right)* segmentation. The sparse segmentation is produced by [5] and the dense segmentation is the result of the proposed model.

patterns belonging to different objects and instances per image and only require single images to be given at test time. Also, in contrast to such approaches, our model does not rely on the availability of large surrogate datasets such as ImageNet [12] or FlyingChairs [13] for pre-training but can be trained directly on the sequences from for example the FBMS59 [2] and $\text{DAVIS}_{16}$ [14] datasets.

To summarize, we make the following contributions:

- We provide an improved affinity graph for motion segmentation in the minimum cost multicut framework using a GRU-model. Our model generates a sparse segmentation of motion patterns.
- We utilize the sparse and potentially noisy motion segmentation labels to train a U-Net model to produce class agnostic and dense motion segmentation. Sparse motion labels are not required during prediction.
- We provide competitive video object segmentation and motion segmentation results on the FBMS59 [2] and $\text{DAVIS}_{16}$ [14] datasets.

## 2 Related Work

Our goal is to learn to segment moving objects based on their motion pattern. For efficiency and robustness, we focus on point trajectory based techniques as initial cues. Trained in a sequence specific way, our model can be used for label *densification*. We therefore consider in the following related work in the areas *motion segmentation* and *sparse to dense labeling*.

### 2.1 Motion Segmentation

Common end-to-end trained CNN based approaches to motion segmentation are based on single frame segmentations from optical flow [4,3,15,16,17]. Tokmakov et al. [4,3] make use of large amounts of synthetic training data [13] to learn the concept of object motion. Further, [4] combine these cues with an ImageNet [12] pre-trained appearance stream and achieve long-term temporal consistency by using a GRU optimized on top. They learn a binary video segmentation and distinguish between static and moving elements. Siam et al. [17]

use a single convolutional network to model motion and appearance cues jointly for autonomous driving. A frame-wise classification problem is formulated in [16] to detect motion saliency in videos. In [15] for each frame multiple figure-ground segmentations are produced based on motion boundaries. A moving objectness detector trained on image and motion fields is used to rank the segment candidates. Methods like [18,19] approach the problem in a probabilistic way. In [18] the camera motion is subtracted from each frame to improve the training. Variational formulations based on optical flow are used in [19,20], where [20] employ a strong geometric model. Addressing motion segmentation is different than object segmentation, as in motion segmentation, different motion patterns are segmented, which makes connected objects seem as one object if they move together with the same motion pattern. As an example, we refer to the example of a person riding a horse (Fig. 1). As long as they move together they will be considered as same object while in object segmentation we deal with two separate objects, this makes our method different than object segmentation approaches [21,22,23].

A different line of work relies on point trajectories for motion segmentation. Here, long-term motion information is first used to establish sparse but coherent motion segments over time. Dense segmentations are usually generated in a post-processing step such as [2]. However, in contrast to end-to-end CNN motion segmentation approaches, trajectory based methods have the desirable property to directly handle multiple motion pattern segmentations. While in [5,6] the partitioning of trajectories into motion segments uses the minimum cost multicut problem, other approaches employ sparse subspace clustering [24] or spectral clustering and normalized cuts [25,26,27]. In this setting, a model selection is needed to determine the final number of segments. In contrast, the minimum cost multicut formulation allows for a direct optimization of the number of components via repulsive cost.

Our GRU approach uses the same graph construction policy as [5] for motion segmentation while the edge costs are assigned using a Siamese (also known as twin) gated recurrent network. The Siamese networks [28] are metric learning approach used to provide a comparison between two different inputs. Similar trajectory embeddings have been used in [29] to predict a pedestrians future walking direction. For motion segmentation, we stick to the formulation as a minimum cost multicut problem [5].

### 2.2 Sparse to Dense Labeling

While the trajectory based motion segmentation methods can propagate object information through frames, they produce sparse results. Therefore, specific methods, e.g. [2,30] are needed to produce dense results. A commonly used densification approach is the variational framework of Ochs et al. [2]. In this approach the underlying color and boundary information of the images are used for the diffusion of the sparsely segmented trajectory points, which sometimes leaks the pixel labels to unwanted regions, e.g. loosely textured areas of the image. Furthermore, [31] address the densification problem using Gabriel graphs as per

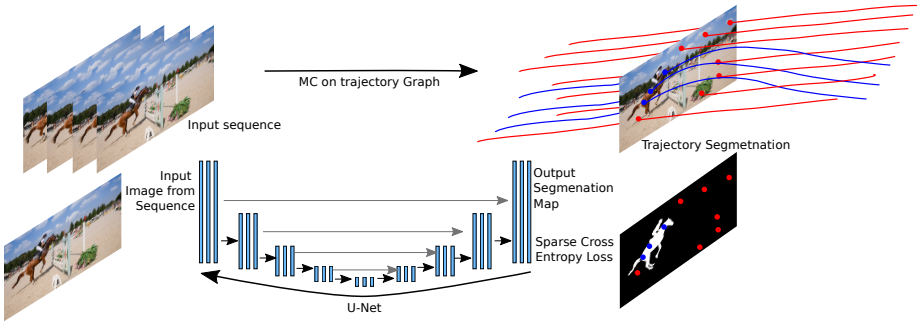


Fig. 3: Sparsely segmented trajectories are produced by minimum cost multicuts (MC) either with our Siamese-GRU model or simple motion cues as in [5] (top). The sparsely labeled points are used to train the U-Net model (bottom). At test time, U-Net model can produce dense segmentations without requiring any sparse labels as input.

frame superpixel maps. Gabriel edges bridge the gaps between contours using geometric reasoning. However, super-pixel maps are prone to neglect fine structure of the underlying image and leads to low segmentation quality.

Our method benefits from trajectory based methods for producing a sparse multi-label segmentation. A sparsely trained U-Net [11] produces dense results for each frame purely from appearance cues, potentially specific for a scene or sequence.

## 3 Proposed Self-Supervised Learning Framework

We propose a self-supervised learning framework for sparse-to-dense segmentation of the sparsely segmented point trajectories. In another words, a U-Net model is trained from sparse annotations to estimate dense segmentation maps (Section 3.2). The sparse annotations can be provided either with some potentially unsupervised state-of-the-art trajectory segmentation methods [5] or our proposed Siamese-GRU model.

### 3.1 Annotation Generation

Point trajectories are generated from optical flow [7]. Each point trajectory corresponds to the set of sub-pixel positions connected through consecutive frames using the optical flow information. Such point trajectories are clustered by the minimum cost multicut approach [32] (aka. correlation clustering) with respect to their underlying motion model estimated (i) from a translational motion model or (ii) from a proposed Siamese GRU network.

**Point Trajectories** are spatio-temporal curves represented by their frame-wise sub-pixel-accurate (x,y)-coordinates. They can be generated by tracking points using optical flow by the method of Brox et al. [7]. The resulting trajectory set aims for a minimal target density (e.g. one trajectory in every 8 pixels). Trajectories are initialized in some video frame and end when the point cannot be tracked reliably anymore, e.g. due to occlusion. In order to achieve the desired density, possibly new trajectories are initialized throughout the video. Using trajectories brings the benefit of accessing the object motion in prior frames.

**Translational Motion Affinities** can be assigned based on motion distances of each trajectory pair with some temporal overlap [5]. For trajectories $A$ and $B$, the frame-wise motion distance at time $t$ is computed by

$$d_t(A,B) = \frac{\|\partial_t A - \partial_t B\|}{\sigma_t}. \tag{1}$$

It solely measures in-plane translational motion differences, normalized by the variation of the optical flow $\sigma_t$ (refer to [2] for more information). The $\partial_t A$ and $\partial_t B$ represent the partial derivatives of $A$ and $B$ with respect to the time dimension.

The overall motion distance of a pair of trajectories $A$ and $B$ is computed by maximum pooling over their joint life-time,

$$d^{motion}(A,B) = \max_t d_t(A,B) \tag{2}$$

Color and spatial cues are added in [5] for robustness. Instead of using translational motion affinities we propose a Siamese Gated Recurrent Units (GRUs) based model to provide affinities between the trajectory pairs.

**Siamese Gated Recurrent Units (GRUs)** can be used to learn trajectory affinities by taking trajectory pairs as input. The proposed network consists of two legs with shared weights, where in our model each leg consists of a GRU model which takes a trajectory as input. Specifically, for two trajectories $A$ and $B$, the $\partial A$ and $\partial B$ (partial derivative of the trajectories with respect to the time dimension) on the joint life-time of the trajectories are given to each leg of the Siamese-GRU model. The partial derivatives represent their motion information, while no information about their location in image coordinates is provided. The motion cues are embedded by the GRU network, i.e. the hidden units are gathered for each time step. Afterwards, the difference between two embedded motion vectors $embed_{\partial A}$ and $embed_{\partial B}$ is computed as

$$d_{(embed_{\partial A}, embed_{\partial B})} = \sum_{i=1}^{h} (embed_{\partial A i} - embed_{\partial B i})^2, \tag{3}$$

where $h$ denotes the number of hidden units for each time step. The result of equation (3) is a vector which is consequently given to two fully connected layers

and the final similarity value is computed by a Sigmoid function. Therefore, for each pair of trajectories given to the Siamese [28] GRU network, it provides a measure of their likelihood to belong to the same motion pattern.

The joint life-time of the two trajectories could in practice be different from pair to pair and the GRU network requires a fixed number of time steps as input ($N$). This problem is dealt as follows:

If the joint life-time of the two trajectories is less than $N$, each trajectory is padded with its respective final partial derivative value in the intersection part. Otherwise, when the joint life-time has more than $N$ time steps, the time step $t$ with maximum motion distance, similar to equation (4), is determined for the entire lifetime,

$$t_{A,B} = \arg\max_{t} d_t(A, B). \tag{4}$$

The trajectory values are extracted before and after $t$ so that the required number of time steps $N$ is reached. The reason for doing this is that the important part of the trajectories are not lost. Consider a case where an object does not move in the first $x$ frames and starts moving from frame $x + 1$, the most important information will be available around frames $x$ and $x + 1$ and it is better not to lose such information by cutting this part out.

In our approach, the frame-wise Euclidean distance of trajectory embeddings (extracted from the hidden units) of the GRU model is fed to a fully connected layer for dimensionality reduction and passed to a Sigmoid function for classification into the classes 0 (same label - pair of trajectories belong to the same motion pattern) or 1 (different label - pair of trajectories belong to different motion pattern) using a mean squared error (MSE) loss.

To train the Siamese-GRU model two labels are considered for each pair of trajectories, label 0 where the trajectory pair correspond to the same motion pattern and label 1 otherwise (the trajectories belong to different motion patterns). To produce the ground-truth labeling to train the Siamese-GRU model, a subset of the edges in the produced graph $G = (V, E)$ by the method of Keuper et al. [5] are sampled (information about the graph is provided in the next paragraph). For each edge, which corresponds to a pair of trajectories, we look into each trajectory and its label in the provided ground-truth segmentation. We only take those trajectories which belong to exactly one motion pattern in the provided ground-truth. Some trajectories change their labels while passing through frames which are considered as unreliable. Furthermore, the same amount of edges with label 0 (same motion pattern) and label 1 (different motion pattern) are sampled to have a balanced training signal. At test time, costs for each edge $E$ in graph $G = (V, E)$ are generated by the trained Siamese-GRU network.

**Trajectory Clustering** yields a grouping of trajectories according to their modeled or learned motion affinities. We formalize the motion segmentation problem as minimum cost multicut problem [5]. It aims to optimally decompose a graph $G = (V, E)$, where trajectories are represented by nodes in $V$ and their affinities define costs on the edges $E$. In our approach the costs are assigned using the Siamese-GRU model.

While the multicut problem is APX-hard [33], heuristic solvers [34,35,36,37,38] are expected to provide practical solutions. We use the Kernighan Lin [34] implementation of [39]. This solver is proved to be practically successful in motion segmentation [5,6], image decomposition and mesh segmentation [35] scenarios.

### 3.2 Deep Learning Model for Sparse to Dense Segmentation

We use the sparse motion segmentation annotated video data as described in Section 3.1 for our deep learning based sparse-to-dense motion segmentation model. Specifically, the training data consist of input images (video frames) or edge maps and their sparse motion segmentations, which we use as annotations. Although, the loss function only applies at the sparse labels, the network learns to predict dense segmentations.

**Deep Learning Model** We use a U-Net [11] type architecture for dense segmentation, which is known for its high quality predictions in tasks such as semantic segmentation [40,41,42]. A U-Net is an encoder-decoder network with skip connections. During encoding, characteristic appearance properties of the input are extracted and are learnt to be associated with objectness. In the decoding phase, the extracted properties are traced back to locations causing the observed effect, while details from the downsampling phase are taken into account to ease the localisation (see Fig. 3 for details). The output is a dense (pixel-wise) segmentation of objects, i.e., a function $u\colon \Omega \to \{1, \ldots, K\}$, where $\Omega$ is the image domain and $K$ is the number of classes which corresponds to number of trajectory labels. This means that, after clustering the trajectories each cluster takes a label and overall we have class-agnostic motion segmentation of sparse trajectories. Such labels are only used during training.

**Loss Function** The U-Net is trained via the Binary Cross Entropy (BCE) and Cross Entropy (CE) loss function for the single and multiple object case, respectively. As labels are only present at a sparse subset of pixels, the loss function is restricted to those pixels. Intuitively, since the label locations where the loss is evaluated are unknown to the network, it is forced to predict a label at every location. (A more detailed discussion is provided below.)

**Dense Predictions with Sparse Loss Functions** At first glance, a sparse loss function may not force the network to produce a meaningful dense segmentation. Since trajectories are generated according to a simple deterministic criterion, namely extreme points of the local structure tensor [7], the network could internally reproduce this generation criterion and focus on labeling such points only. Actually, we observed exactly the problematic behaviour mentioned above, and, therefore, suggest variants for the learning process employing either RGB images or (deterministic) Sobel edge maps [43] as input. One remedy is to alleviate the local structure by smoothing the input RGB-image, making it

harder for the network to pick up on local dominant texture and to stimulate the usage of globally extracted features that can be associated with movable object properties.

**Conditional Random Filed (CRF) Segmentation Refinement** To build the fine-grained segmentation maps from the blurred images, we employ Conditional Random Fields (CRF): We compare

- the fully connected pre-trained CRF layer (dense-CRF) [44], with parameters learnded from pixel-level segmentation [45] and
- a CRFasRNN [46] model which we train using the output of our U-Net model as unaries on our same sparse annotations. To generate a training signal even in case the U-Net output perfectly fits the sparse labels, we add Gaussian noise to the unaries.

**Discussion: Sparse Trajectories vs. Dense Segmentation** The handling of sparse labels could be avoided if dense unsupervised motion segmentations were given. Although, in principle, dense trajectories can be generated by the motion segmentation algorithm, the clustering algorithm does not scale linearly with the number of trajectories and the computational cost explodes. Instead of densely tracking pixels throughout the video, a frame-wise computationally affordable densification step, for example based on variational or inpainting strategies [2], could be used. However, some sparse labels might be erroneous, an issue that can be amplified by the densification. Although some erroneous labels can also be corrected by [2], especially close to object boundaries, we observed that the negative effect prevails when it comes to learning from such unsupervised annotations. Moreover, variational methods often rely on local information to steer the propagation of label information in the neighborhood. In contrast, the U-Net architecture can incorporate global information and possibly objectness properties to construct its implicit regularization.

## 4 Experiments

We evaluate the performance of the proposed models on the two datasets $\text{DAVIS}_{16}$ [14] and FBMS59 [2], which contain challenging video sequences with high quality segmentation annotations of moving objects.

### 4.1 Datasets and Implementation Details

**Datasets** The $\text{DAVIS}_{16}$ dataset [14] is produced for high-precision binary object segmentation tracking of rigidly and non-rigidly moving objects. It contains 30 train and 20 validation sequences. The pixel-wise binary ground truth segmentation is provided per frame for each sequence. The evaluation metrics are Jaccard index (also known as Intersection over Union) and F-measure. Even

Table 1: The trajectories are segmented by 1. the method of Keuper et al. [5] and 2. our Siamese-GRU model. The densified results are generated based on 1. the method of Ochs et al. [2] and 2. the proposed U-Net model. The results are provided for the validation set of $\text{DAVIS}_{16}$.

| Traj. Seg. Method | Densification Method | Jaccard Index |
|---|---|---|
| Keuper et al. [5] | Ochs et al. [2] | 55.3 |
| Keuper et al. [5] | U-Net model (ours) | 58.5 |
| Siamese-GRU (ours) | Ochs et al. [2] | 57.7 |
| Siamese-GRU (ours) | U-Net model (ours) | **66.2** |

though this dataset is produced for object segmentation, it is commonly used to evaluate motion segmentation because only one object is moving in each sequence which makes the motion pattern of the foreground object to be different from the background motion.

The FBMS59 [2] dataset is specifically designed for motion segmentation and consists of 29 train and 30 test sequences. The sequences cover camera shaking, rigid/non-rigid motion as well as occlusion/dis-occlusion of single and multiple objects with ground-truth segmentations given for a subset of the frames. We evaluate precision, recall and F-measure.

**Implementation Details** Our Siamese-GRU model with 2 hidden units ($h = 2$, equation 3) and experimentally selected 25 time steps ($N = 25$, for more information refer to section 3.1) is trained for 3 epochs, a batch size of 256 and a learning rate of 0.001 where the trajectories are produced by large displacement optical flow (LDOF) [47] at 8 pixel sampling on the training set of $\text{DAVIS}_{16}$ [14].

We employ two different strategies of using the sparse motion segmentations of the resulting trajectories as labels, depending whether we face binary ($\text{DAVIS}_{16}$) or multi-label (FBMS59) problems. In case of single label, the most frequent trajectory label *overall frames* and second most frequent label *per frame* are considered as background and foreground, respectively. For multi-label cases, the most frequent class-agnostic labels are selected, i.e. we take only those labels which are frequent enough compared to the other labels.

Our U-Net model is trained in a sequence specific way. Such model can be used for example for label densification and is trained using color and edge-map data with a learning rate of 0.01 and batch size of 1 for 15 epochs. The overall train and prediction process takes around (maximally) 30 minutes per sequence on a NVIDIA Tesla V100 GPU. The CRFasRNN [46] is trained with learning rate of 0.0001, batch size 1 and 10 epochs with 5 and 10 inference iterations at train and test time, respectively.

### 4.2 Sparse Trajectory Motion-Model

We first evaluate our GRU model for sparse motion segmentation on the validation set of $\text{DAVIS}_{16}$ [14]. Therefore, we produce, in a first iteration, densified

Table 2: Sparse Motion Segmentation trained on $DAVIS_{16}$ (all seq.) and evaluated on FBMS59 (train set). We compare to [5] and their variant only using motion cues.

| | Precision | Recall | F-measure | #of extracted objs. |
|---|---|---|---|---|
| Keuper et al. [5] (motion cues) | 83.88 | 69.97 | 76.30 | 20 |
| Keuper et al. [5] (full) | 83.69 | 73.17 | **78.07** | **27** |
| Siamese-GRU (ours - transfer) | 81.01 | 70.07 | 75.14 | 24 |

segmentations using the variational approach from [2]. The results are given in Tab. 1 (line 3) and show an improvement over the motion model from Keuper et al. [5] by 2% in Jaccard index. In the following experiments on $DAVIS_{16}$, we thus use sparse labels from our Siamese GRU approach.

**Knowledge Transfer** Next, we investigate the generalization ability of this motion model. We evaluate the $DAVIS_{16}$-trained GRU-model on the train set of FBMS59 [2]. Results are given in Tab. 2. While this model performs below the hand-tuned model of [5] on this dataset, results are comparable, especially considering that our GRU model does not use color information. In further experiments on FBMS59, we use sparse labels from [5].

### 4.3 Dense Segmentation of Moving Objects

Next, we evaluate our self-supervised dense segmentation framework with sequence specific training on the color images as well as edge maps on the validation set of $DAVIS_{16}$ [14]. Tab. 1 shows that this model, trained on the GRU based labels, outperforms the model trained on the motion models from [5] as well as the densification of Ochs et al. [2] by a large margin. Tab. 3 (top) provides a comparison between different versions of our model, the densification model of Ochs et al. [2], and the per-frame evaluation of Tokmakov et al. [3] on $DAVIS_{16}$. [3] use large amounts of data for pre-training. Their better performing model variants require optical flow and full sequence information to be given at test time. Our results based on RGB and edge maps are better than those solely using edge maps. We also compare the different CRF versions:

- pre-trained dense-CRF [44],
- our trained CRFasRNN [46] model trained per sequence *(CRF-per-seq)*,
- CRFasRNN [46] trained on all frames in the train set of $DAVIS_{16}$ *(CRF-general)* with sparse labels

All CRF versions improve the F-measure. The CRF-general performs on par with dense-CRF by only using our sparse labels for training. See Fig. 4 and Fig. 5 for qualitative results of our model on $DAVIS_{16}$ [14] and FBMS59 [2] datasets, respectively. We show on par performance with the layered optimization approach by [20] in $DAVIS_{16}$ with Jaccard's index of 68.3 versus *ours:* 67.1.

Table 3: Evaluation of self supervised training on sequences from $DAVIS_{16}$ validation and comparison with other methods is provided. Effect of adding color information (RGB) to the edge maps (sobel) is studied (*ours*) and comparison between (pretrained) dense-CRF (*dense*), CRF-per-seq (*per-seq*) and CRF-general (*general*) is provided (for different versions of CRF refer to 4.3). We studied the effect of our best model while training it only on 50%, 70% and 90% of the frames in the last three rows.

| | % of frames | Jaccard Index | F-measure |
|---|---|---|---|
| variational [2] | 100 | 57.7 | 57.1 |
| appearance + GRU [3] | 100 | 59.6 | - |
| sobel + *dense* | 100 | 62.6 | 54.0 |
| sobel + RGB (*ours*) | 100 | 61.3 | 49.0 |
| *ours* + *dense* | 100 | **67.1** | 60.2 |
| *ours* + *per-seq* | 100 | 66.2 | 60.3 |
| *ours* + *general* | 100 | 66.2 | **62.1** |
| *ours* + *general* | 50 | 59.6 | 50.4 |
| *ours* + *general* | 70 | 62.3 | 53.5 |
| *ours* + *general* | 90 | 63.4 | 55.4 |

Table 4: We evaluate our densification method on FBMS59 (train) using sparse motion segmentations from Keuper et al. [5]. The sparse trajectories are produced with different flow estimation methods (LDOF [47] and FlowNet2 [48]) and densified with our proposed U-Net model (using edge maps (sobel) and color information (RGB) (*ours*)). Further, we study on different CRF methods, (pretrained) dense-CRF (*dense*) and CRF-general (*general*). For more details about different versions of CRF refer to section 4.3.

| | Precision | Recall | F-measure |
|---|---|---|---|
| Ochs et al. [2] | 85.31 | 68.70 | 76.11 |
| Lao et al. [20] | 90.04 | 65.09 | 76.02 |
| LDOF + *ours* + *dense* | 89.35 | 67.67 | 77.01 |
| FlowNet2 + *ours* + *dense* | 89.59 | 68.29 | **77.56** |
| FlowNet2 + *ours* + *general* | 89.27 | 68.20 | 77.33 |

**Partly trained Model** We further evaluate how well our model works for video frames for which no sparse labels are given during training. Please note that, throughout the sequences, the appearance of an object can change drastically. In Tab. 3 (bottom), we report results for the sequence specific U-Net model + *CRF-general* trained on the first 50%, 70% and 90% of the frames and evaluated on the remaining frames. While there is some loss in Jaccard's index compared to the model evaluated on seen frames (above), the performance only drops slightly as smaller portions of the data are used for training.

Fig. 4: Exemplary single-label motion segmentation results showing the five frames and their sparse and dense segmentation for two different sequences, generated using the proposed U-Net model. The images are from the sequences on the validation set of $\text{DAVIS}_{16}$ [14] dataset.

**Densification on FBMS59** Next, we evaluate our sequence specific model for label densification on FBMS59 [2]. We study on two different variants of optical flow (FlowNet2 [48] and Large Displacement Optical Flow (LDOF) [47]) for trajectory generation and sparse motion segmentation [5]. The results in Tab. 4 show that the proposed approach outperforms the approach of Ochs et al. [2] as well as the geometric, layered optimization approach by [20]. Improved optical flow leads to improved results overall. The different CRF versions do not provide significantly different results.

## 5 Conclusion

In this paper, we have addressed the segmentation of moving objects from single frames. To that end, we proposed a GRU-based trajectory embedding to produce high quality sparse segmentations automatically. Furthermore, we closed the gap between sparse and dense results by providing a self-supervised U-Net model trained on sparse labels and relying only on edge maps and color information. The trained model on sparse points provides single and multi-label

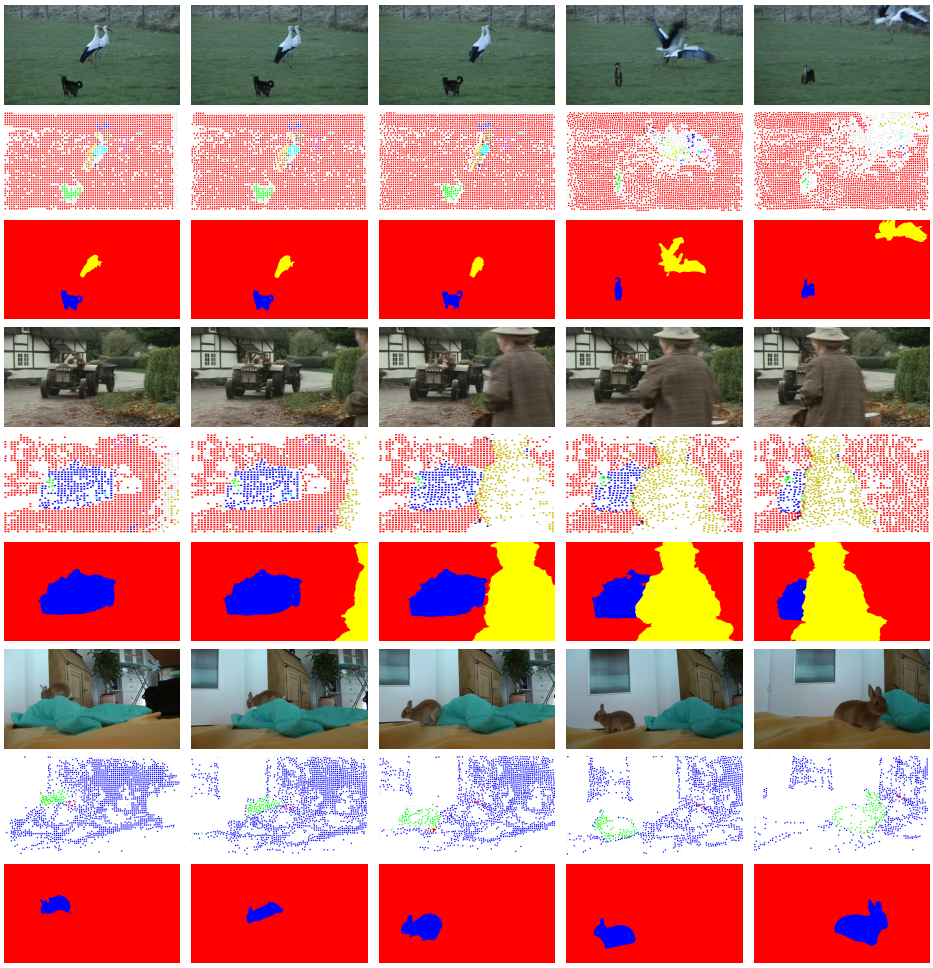

Fig. 5: Exemplary single- and multi-label motion segmentation results showing the image and its sparse results as well as dense segmentation for five frames in three different sequences, generated using the proposed U-Net model. The images are from the FBMS59 [2] dataset. Segmentations with fine details are produced even when training labels were scarce, notice how scarce the labels are for "rabbit" images in the 8th row. White areas are parts without any label.

dense segmentations. The proposed approach generalizes to unseen sequences from FBMS59 and $\text{DAVIS}_{16}$ and provides competitive and appealing results.

**Acknowledgment** We acknowledge funding by the DFG project KE 2264/1-1 and thank the NVIDIA Corporation for the donation of a Titan Xp GPU.

## References


1. Koffka, K.: Principles of Gestalt Psychology. Hartcourt Brace Jovanovich, NewYork (1935) 1
2. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE TPAMI **36** (2014) 1187 – 1200 2, 3, 4, 6, 9, 10, 11, 12, 13, 14
3. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: ICCV. (2017) 2, 3, 11, 12
4. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: CVPR. (2017) 2, 3
5. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: IEEE International Conference on Computer Vision (ICCV). (2015) 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13
6. Keuper, M.: Higher-order minimum cost lifted multicuts for motion segmentation. In: The IEEE International Conference on Computer Vision (ICCV). (2017) 2, 4, 8
7. T.Brox, J.Malik: Object segmentation by long term analysis of point trajectories. In: European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, Springer (2010) 2, 5, 6, 8
8. Fragkiadaki, K., Zhang, W., Zhang, G., Shi, J.: Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In: ECCV. (2012) 2
9. Shi, F., Zhou, Z., Xiao, J., Wu, W.: Robust trajectory clustering for motion segmentation. In: ICCV. (2013) 2
10. Rao, S.R., Tron, R., Vidal, R., Yi Ma: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8 2
11. Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). Volume 9351 of LNCS., Springer (2015) 234–241 (available on arXiv:1505.04597 [cs.CV]). 2, 5, 8
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09. (2009) 3
13. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV). (2015) 3
14. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. (2016) 724–732 3, 9, 10, 11, 13
15. Maczyta, L., Bouthemy, P., Meur, O.: Cnn-based temporal detection of motion saliency in videos. Pattern Recognition Letters **128** (2019) 3, 4
16. Fragkiadaki, K., Arbelaez, P., Felsen, P., Malik, J.: Learning to segment moving objects in videos. (2015) 4083–4090 3, 4
17. Siam, M., Mahgoub, H., Zahran, M., Yogamani, S., Jagersand, M., El-Sallab, A.: Modnet: Motion and appearance based moving object detection network for autonomous driving. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). (2018) 2859–2864 3

18. Bideau, P., Learned-Miller, E.: It's moving! a probabilistic model for causal motion segmentation in moving camera videos. Volume 9912. (2016) 433–449 4
19. Cremers, D.: A variational framework for image segmentation combining motion estimation and shape regularization. In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'03, Washington, DC, USA, IEEE Computer Society (2003) 53–58 4
20. Lao, D., Sundaramoorthi, G.: Extending layered models to 3d motion. In: ECCV. (2018) 4, 11, 12, 13
21. Hu, Y.T., Huang, J.B., Schwing, A. In: Unsupervised Video Object Segmentation Using Motion Saliency-Guided Spatio-Temporal Propagation: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I. (2018) 813–830 4
22. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2016) 4
23. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. 2013 IEEE International Conference on Computer Vision (2013) 1777–1784 4
24. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: ICCV. (2013) 4
25. P.Ochs, T.Brox: Higher order motion models and spectral clustering. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 4
26. Jianbo Shi, Malik, J.: Motion segmentation and tracking using normalized cuts. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). (1998) 1154–1160 4
27. Fragkiadaki, K., Shi, J.: Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In: CVPR. (2011) 4
28. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Volume 1. (2005) 539–546 vol. 1 4, 7
29. Bhattacharyya, A., Fritz, M., Schiele, B.: Long-term on-board prediction of people in traffic scenes under uncertainty. (2018) 4
30. Müller, S., Ochs, P., Weickert, J., Graf, N.: Robust interactive multi-label segmentation with an advanced edge detector. In: German Conference on Pattern Recognition (GCPR). Volume 9796 of LNCS., Springer (2016) 117–128 4
31. Shi, J.: Video segmentation by tracing discontinuities in a trajectory embedding. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR '12, Washington, DC, USA, IEEE Computer Society (2012) 1846–1853 4
32. Andres, B., Kröger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Köthe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: ECCV. (2012) 5
33. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning **56** (2004) 89–113 8
34. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal **49** (1970) 291–307 8
35. Keuper, M., Levinkov, E., Bonneel, N., Lavoue, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. In: IEEE International Conference on Computer Vision (ICCV). (2015) 8
36. Beier, T., Andres, B., Köthe, U., Hamprecht, F.A.: An efficient fusion move algorithm for the minimum cost lifted multicut problem. In: ECCV. (2016) 8

37. Kardoost, A., Keuper, M.: Solving minimum cost lifted multicut problems by node agglomeration. In: ACCV 2018, 14th Asian Conference on Computer Vision, Perth, Australia (2018) 8
38. Bailoni, A., Pape, C., Wolf, S., Beier, T., Kreshuk, A., Hamprecht, F.: A generalized framework for agglomerative clustering of signed graphs applied to instance segmentation. (2019) 8
39. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: ICCV. (2015) 8
40. Siam, M., Gamal, M., Abdel-Razek, M., Yogamani, S., Jagersand, M.: Rtseg: Real-time semantic segmentation comparative study. In: 2018 25th IEEE International Conference on Image Processing (ICIP). (2018) 1603–1607 8
41. Siam, M., Elkerdawy, S., Jagersand, M., Yogamani, S.: Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). (2017) 1–8 8
42. Fu, J., Liu, J., Wang, Y., Zhou, J., Wang, C., Lu, H.: Stacked deconvolutional network for semantic segmentation. IEEE Transactions on Image Processing (2019) 1–1 8
43. Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the sobel operator. IEEE Journal of solid-state circuits **23** (1988) 358–367 8
44. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. CoRR **abs/1210.5644** (2012) 9, 11
45. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. (2015) 9
46. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: International Conference on Computer Vision (ICCV). (2015) 9, 10, 11
47. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2009) 10, 12, 13
48. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 12, 13