The Best of Both Worlds: Combining CNNs and Geometric Constraints for Hierarchical Motion Segmentation

Pia Bideau Aruni RoyChowdhury Rakesh R. Menon Erik Learned-Miller
College of Information and Computer Science
University of Massachusetts Amherst
{pbideau, arunirc, rrmenon, elm}@cs.umass.edu

Abstract

Traditional methods of motion segmentation use powerful geometric constraints to understand motion, but fail to leverage the semantics of high-level image understanding. Modern CNN methods of motion analysis, on the other hand, excel at identifying well-known structures, but may not precisely characterize well-known geometric constraints. In this work, we build a new statistical model of rigid motion flow based on classical perspective projection constraints. We then combine piecewise rigid motions into complex deformable and articulated objects, guided by semantic segmentation from CNNs and a second "object-level" statistical model. This combination of classical geometric knowledge combined with the pattern recognition abilities of CNNs yields excellent performance on a wide range of motion segmentation benchmarks, from complex geometric scenes to camouflaged animals.

1. Introduction

Understanding motion is fundamental to our understanding of the world, from predicting the immediate future to understanding actions and interactions, and even in defining objectness itself. In this work we revisit the classic problem of motion segmentation in moving camera videos, a first step in understanding and interpreting motion.

Motion segmentation is an intriguing problem in that it combines subareas of vision in which geometry is a powerful constraint–the understanding of how images will change under camera motion–with "messy" problems like segmentation and the deformation of flexible moving objects, in which there are virtually no hard geometric constraints. This has given rise to a range of methods–some that use mostly geometric techniques while largely ignoring appearance [16, 3, 48], and others that try to learn the entire pipeline using CNN architectures [43, 44, 18] attempting to learn both the image patterns and the flow patterns in CNNs.

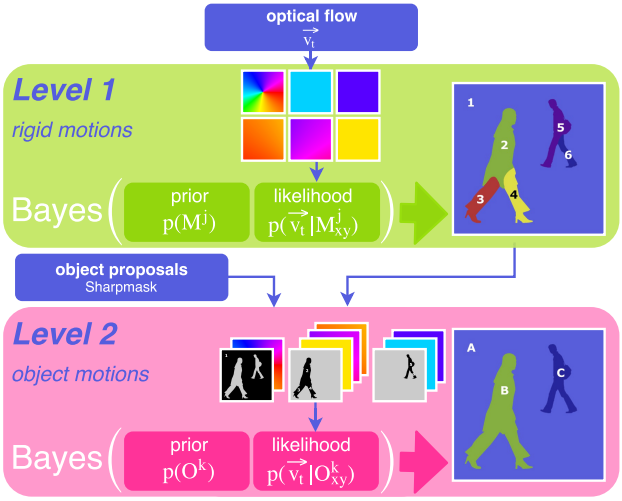

Figure 1: **A hierarchical model for Motion Segmentation.** The first level of our method estimates rigid motion components from optical flow. The second level groups these components based upon object proposals from SharpMask [35] to form object motion models.

Methods that use motion cues alone, without appearance models of moving objects, are likely to fail in cases where flow is noisy, ambiguous, or hard to determine. Such purely "geometric" approaches are often not sufficient to understand motion well. The appearance of what is moving must also be considered. This suggests using deep learning methods to solve the problem.

Of course, CNNs are excellent at modeling the appearance of objects [23, 40, 14]. They excel at finding objects in static images and videos [12, 25, 37]. They are also very good at segmenting objects [8, 49, 24, 13, 26, 38], exceeding performance of pre-CNN methods. However, there are cases where *appearance alone* is simply not enough to segment well. Such cases are highlighted by the Camouflaged Animal motion segmentation data set [3], in which moving objects are virtually invisible in many of the static frames.

In this paper, we combine careful motion modeling using classical ideas with a modern CNN for appearance modeling, yielding excellent results. Towards this end, we design a hierarchical motion segmentation system in which the first phase identifies simple rigid motion components, and the second phase assembles these rigid motion components into full objects, guided by a semantic segmentation of each frame [35] (see Figure 1). This new hierarchical system allows the first low-level phase to focus on the geometry of perspective projection, segmenting the frame into its rigid motions. Then, in the second phase, deformable and articulated objects, like pedestrians and animals, are modeled as a combination of a number of rigid motion components, as suggested by the semantic segmentation results. While neither the motion analysis nor the semantic segmentations are error free, their combination results in a significant improvement in performance on the multi-label motion segmentation problem. Our contributions include:

- A new hierarchical model for motion segmentation with two steps:
  1. segmenting a frame into rigid motions;
  2. using *objectness* knowledge from SharpMask [35] to combine these rigid parts into object models that describe the motions of articulated and deformable objects such as people or animals.
- A new statistical model for optical flow as a noisy measurement of the underlying motion field. We set noise distribution parameters using statistics of the Sintel database [6].
- A Bayesian approach to computing the likelihood of a *3D motion direction* associated with an optical flow vector, in which we integrate over the unobservable motion field magnitude. This allows us to assign pixels to rigid motion models in a fashion consistent with perspective projection and our statistical model.

On the definition of motion segmentation. In this work, we focus on the classical definition of *motion segmentation* [45], which is essentially about segmenting all objects which are moving (in 3D) relative to the background. Even so, there are subtleties that need to be addressed to make it clear what the ground truth should be. For example, should a bush that is barely moving in the wind be segmented as a moving object? These questions are discussed extensively in [2], and this work adopts their definitions of ground truth for motion segmentation.

We report results on three motion segmentation benchmarks that are consistent with the classical definition of motion segmentation: Freiburg-Berkeley Motion [4], Complex Background [27], and Camouflaged Animals [3]. The Davis data set [34, 36] is a popular *video segmentation* benchmark which focuses on segmenting prominent objects rather than all moving objects. While our method is not designed for such benchmarks, we discuss the relationship between object segmentation and motion segmentation and discuss our results on that benchmark in the supplementary material.

2. Related work

Many motion segmentation papers focus on the problem of *binary motion segmentation*, where pixels are classified as either moving or part of the background, but no distinction is made between separate moving objects [3, 27, 32, 10]. Others [42, 20, 11], like this paper, address *multi-label motion segmentation*, where a separate label is given to each independently moving object. In the remainder of this section, we do not distinguish between binary and multi-label motion segmentation.

Information about motion is mostly derived from matched pixels across consecutive frames. This could be in the form of either *sparse point trajectories* or *optical flow*. Methods based on *point trajectories* [20, 4, 11, 28, 19] have shown good results for motion segmentation. Given the optical flow, pixels showing similar displacements are grouped into objects. These objects are then tracked so that they are segmented consistently over time. However, trajectory-based methods often segment non-moving objects. Pixel displacement from one frame to the next is a function of depth and motion (see Eq. 1). Thus motion-trajectory based clustering methods form clusters not only for independently moving objects, but also for objects at different depths. Methods based on occlusions [31, 42] are subject to similar depth-related problems.

Other methods rely on *optical flow* and seek to find coherent motion patterns. These methods can be grouped into those that use projective geometry approximations [48, 45], those that use perspective projection [27, 3, 31, 16], and methods that learn motion patterns using CNNs [43, 44, 47, 33, 18, 21].

Approaches based on perspective projection are in general more accurate than those based on projective geometry, since the latter omits certain constraints (such as orthogonality constraints) in modeling image transformations [15, 3]. Bideau et al. [3] developed a fully automatic motion segmentation method based on optical flow. Following the geometry of perspective projection, a frame is segmented based on the optical flow's direction. Assuming that the underlying motion field magnitude is *equal* to the optical flow magnitude, they use the motion field magnitude to model the informativeness of the direction of each flow vector. Unlike Bideau et al., we integrate over the unknown motion field magnitude in a Bayesian fashion, rather than assuming its value is equal to the flow magnitude.

Some concurrent works have leveraged both object motion and semantic information in video object segmentation [9, 46] and optical flow estimation [39, 1]. Distinct

from these approaches, our work combines object-level semantic knowledge with ideas from classical perspective geometry for accurate segmentation of moving objects.

3. Approach

Our motion segmentation system is (automatically) initialized with an estimate of the background region and a set of rigid objects. We adopt the publicly available initialization code of [3] for this purpose.

Throughout our system, we consider two separate notions of movement:

- **Rigid motions**: motions that can be described by translating rigid 3D regions.[1]
- **Object motions**: motions of real objects (e.g., pedestrians or cars) that are modeled as compositions of one or more rigid motions.

Throughout the video, we maintain a set of rigid motions. This set may be expanded, to contain newly discovered motions, or contracted, if we find there is no more evidence of a previously seen rigid motion. Multiple rigid motions together can describe highly complex object motions. We maintain a set of such object motions, which typically correspond to real world objects such as cars, pedestrians, or animals. The "background", which is typically the static environment, can be modeled with a single rigid motion.

Algorithm 1 gives the overview of our main loop. Several types of information from the previous time step are used as prior information for the current time step. This includes the (soft probabilistic) masks of each rigid motion component, the (soft probabilistic) masks of each object, and the history of how rigid components have so far been assigned to objects. In addition, we incorporate new information from the current optical flow (Sun et al. [41]), and *region proposals* from SharpMask [35].

The main steps of our method are (1) removing rotational flow (Sec. 3.1), (2) estimating rigid motion components and assigning pixels in each frame to rigid motion components (Sec. 3.2), (3) grouping rigid motion components into sets to form object models (Sec. 3.3) and (4) assigning the pixels in each frame to objects for a final segmentation (Sec. 3.4).

3.1. Removing rotational flow

We seek a camera rotation such that, after subtracting off this rotation from the optical flow, the remaining flow corresponds to purely translational motion. This will be true for a static background region, since here the pixel displacement is *only* influenced by the camera's motion and *not* by independently moving objects. For the first frame we do not possess an estimate of background regions, so this rotation is found via RANSAC (details in [3]). For subsequent frames, we can get an estimate of the background regions from the previous time step, and estimate camera rotation only from those background pixels. Unless specified otherwise, all remaining optical flows discussed in the paper refer to the *translational component* of optical flow, i.e. the optical flow after camera rotation has been removed.

[1] Object rotations are not modeled.

Algorithm 1: Estimate motion models and segment frame into objects

Input:
- Optical flow.
- Rigid components of previous frame.
- Moving objects of previous frame.
- Assignment history of rigid motions to objs.
- SharpMask object proposals for current frame.

Output:
- Current rigid components.
- Current moving objects.

```
// Estimate rotational flow and remove it 3.1.
// Estimate rigid motion components 3.2.
for each rigid component region from prev. frame do
    Est. current rigid motion model for that region.
end
for each pixel in current image do
    Assign it to a rigid motion model.
end
// Grouping rigid motion components 3.3.
for moving object mask in object proposals do
    Assign rigid motion models to object mask.
    Check consistency with assignment history.
end
Create object motion models
//Assign pixels to moving objects 3.4.
for each pixel in current image do
    Assign it to an object motion model.
end
```

3.2. Estimating rigid motion components

The next step of our system is to estimate a set of J rigid motion models M^j, $j = 1 \dots J$, and to assign each pixel in the current image to one of the motion models. Intuitively, we want to discover the "set of motions" of rigid structures in the image, and then to determine which pixels belong to each motion, as shown in Figure 1(a).

The rigid motion model. Our rigid motion model describes the *direction* of the 3D motion of a rigid object (or scene), but not the magnitude of this motion. Let (U, V, W) be the translational motion of the camera relative to an object. Let (X, Y, Z) be the real world coordinates in 3D of a point that projects to (x, y) in the image. Let f be the camera's focal length. The motion field vector (u, v) at the

image location (x, y) due to a translational motion is given by

$$u = \frac{-fU + xW}{Z}; \quad v = \frac{-fV + yW}{Z}. \tag{1}$$

The 2D translational motion direction at each point in the image is then given by the angle of the motion field vector (u, v) at image location (x, y):

$$M_{xy}(f, U, V, W) = \texttt{atan}(-fV + yW, -fU + xW). \tag{2}$$

This leads us to our rigid motion model M, which is an $h \times w$ matrix (h and w are the image height and width), defined by a 3D translational motion (U, V, W). The elements of this matrix are the motion directions at each pixel location (x, y) in the image.

Note that the rigid motion model M is dependent upon the focal length. However we show in the supplementary material that the *set of all rigid motion models* M is independent of focal length, and that for each focal length, there is a unique mapping from 3D translational motions to rigid motion models. By establishing that the *set of rigid motion models* does not depend upon focal length, we develop a method that can separate different motions without identifying their exact form (which requires focal length).

An advantage of using our rigid motion models M as central building blocks is that they are independent of the scene depth Z (in Eq. 2 M_{xy} is not a function of Z). Methods that depend upon motion field magnitudes implicitly depend upon the scene depth, and thus can result in depth dependent segmentations by confusing image motion due to camera motion and image motion due to object motion.

Estimating a rigid motion model for each rigid motion segment. The next step in our method is to examine the regions from the rigid motion segmentation of the previous frame and estimate a rigid motion model that describes the current optical flow in each rigid motion region. First, we "flow forward" the previous frame's rigid motion regions to obtain the approximate positions of the same rigid structures in the current frame. We then use the optical flow vectors in each region to estimate the motion model by using Horn's method [5], which gives a closed form solution for the best fit to the current translational flow of each region using a least-squares estimation procedure. Eq. 3 describes the squared difference between observed optical flow (u_t, v_t) (after removing camera rotation from Sun et al. [41]'s optical flow) and a pure-translational motion vector (u, v) from Eq. 1. The parameters (U, V, W) are estimated such that Eq. 3 is minimized:

$$[\hat{U}, \hat{V}, \hat{W}] = \underset{U,V,W}{\arg\min} \sum_{i=1}^{N} \left(\left(u_t^i - u^i\right)^2 + \left(v_t^i - v^i\right)^2 \right). \tag{3}$$

Given the estimates $[\hat{U}, \hat{V}, \hat{W}]$ for each region, we can substitute them into Eq. 2 to obtain a set of rigid motion models for the current frame. Note that $i = 1 \ldots N$ indexes over the pixels in a region.

Assigning pixels to rigid motion models. Given the set of rigid motion models M^j, $j = 1 \ldots J$, we re-assign each pixel to one of the estimated rigid motion models. Let $\vec{v_t} = (u_t, v_t)$ be an observed translational flow vector at a particular pixel position (x, y), containing only motion due to camera translation and object motion. The current goal is to choose from among J motion models at each pixel location the one with highest probability given the observed flow vector:

$$L_{rigid} = \arg\max_j p(M_{xy}^j \mid \vec{v_t}). \tag{4}$$

Each pixel in the image will be assigned to this maximum a posteriori motion model, resulting in the segmentation of a frame into its J rigid motion components. We compute these posteriors using Bayes' rule as

$$p(M_{xy}^j \mid \vec{v_t}) \propto p(\vec{v_t} \mid M_{xy}^j) \cdot p(M_{xy}^j). \tag{5}$$

To compute this posterior, we introduce a new model for the flow likelihood $p(\vec{v_t}|M_{xy}^j)$ and the prior $p(M_{xy}^j)$, details of which are described in Section 4.

3.3. From rigid motions to object motions

The segmentation L_{rigid} (Eq. 4) segments a frame into its rigid motion components. Non-rigid moving objects are often composed of multiple rigid motions. To be able to model the motion of an object accurately we use a CNN to produce object proposal masks leveraging the semantics of high level image understanding. According to the generated object proposals we join rigid motion models into sets that belong to a specific object. Thus a set of rigid motion models is used to model an object's motion.

Given the segmentation L_{rigid} (Eq.4) of a frame into J rigid motions and a set of object proposal masks, we form mutually exclusive subsets $\mathcal{M}^k$ of the rigid motion models M^j. Each $\mathcal{M}^k$, $k = 1 \ldots K$ comprises a set of rigid motion models belonging to a specific object's motion. The steps are as follows:

1. Generate object proposals using the SharpMask segmentation method [35] to create candidate masks of objects and select masks corresponding to moving objects only.
2. Join rigid motion models into sets that belong to a specific object motion guided by semantic segmentations of [35].

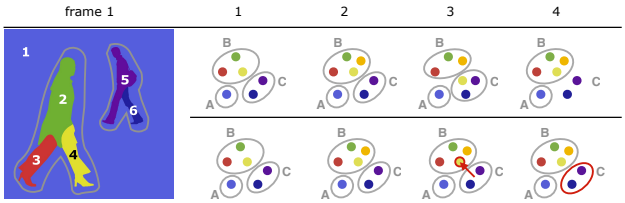

Figure 2: **Grouping rigid motion models with temporal consistency.** *Top row:* tracking objects over time: (i) Two rigid motion components, dark blue and violet assigned to Object C previously, become isolated in frame 4; (ii) The yellow component suddenly shifts from Object B to C in frame 3. *Bottom row:* time consistent assignment of rigid motions to object motions addresses both of these issues.

Generating moving object proposals. We first generate a large set of *object proposals* and *objectness scores* using the SharpMask segmentation method [35], and keep the top 100 proposals (based on objectness score). We analyze these object proposal masks and select a subset that best covers the non-background portions of the image, the latter being estimated from the rigid motion models.

Joining rigid motion models into sets that describe a specific object motion. Given moving object proposal masks and the segmentation L_{rigid}, we can simply assign each motion model M^j to the object proposal mask that has the highest intersection with the rigid motion region corresponding to M^j. However the object proposal masks are not necessarily time consistent – they might arise, disappear or cover part of other objects in single frames. Thus we require a more sophisticated approach than simply assigning each motion model M^j to its object proposal mask. To achieve a *temporally consistent segmentation*, we address the following two consistency requirements: (1) tracking objects over time and (2) time consistent assignment of motion components M^j to objects.

We track objects over time by evaluating shared rigid motion components among objects detected by SharpMask in the current frame and objects detected in the past. Let Q be the number of object proposal masks (i.e., the output of SharpMask at the current frame) and $q = 1, .., Q$ its index. Let K be the number of all moving objects detected till the current video frame T, indexed by $k = 1, ..., K$. Given the Q object proposal masks, segmentation of all frames into rigid motion components $\{L^t_{rigid}\}_{t=1,..,T}$, and object segmentations from all previous frames $\{L^t_{object}\}_{t=1,...,T-1}$,[2] the problem is to find the lowest-cost way to assign each object proposal mask at the current frame to its corresponding object segmentation. This problem can be represented in a matrix of the *component similarity* - the number of common rigid motion components between the object k and a motion mask q. This leads to a $Q \times K$ matrix. Then the Hungarian algorithm is used to find the best matching such that the component similarity is maximized.

The second consistency requirement we address is time consistent assignment of rigid motion models M^j of the current frame to the K objects detected in the video sequence so far. We assign a rigid motion component M^j to an object according to its conditional probability,

$$p(M^j \mid \mathcal{M}^k_T) = \frac{\sum_{t=1}^{T} \mathbb{1}[M^j_t \in \mathcal{M}^k_t]}{T}. \quad (6)$$

In words, the probability that a rigid motion M^j is part of the set $\mathcal{M}^k_T$ (set of rigid motions that define a specific object's motion of the current frame T) is the number of frames t, with $t = 1, ..., T$, where M^j was assigned to $\mathcal{M}^k_t$, out of the total number of frames seen so far, T.

In summary we first assign rigid motions to Q motion masks of the current frame based on its *component similarity* (top row of Figure 2). We then re-assign rigid motions to the K moving objects that have been seen so far (bottom row of Figure 2).

The object motion model. $\mathcal{M}^k_T$ is a set of rigid motion models belonging to a specific object's motion. Each rigid motion model describes part of that object's motion at the current frame T. Let r be the index over elements (rigid motions) in the set $\mathcal{M}^k_T$. We now explain how a new high level object motion model O^k is generated from a set of rigid motion models $M^r \in \mathcal{M}^k_T$.

Similar to a rigid motion model M^j, an object motion model O^k determines a motion direction at each pixel location. M^j often models just a *part* of an object's motion due to its rigidity constraint, whereas the *high level object motion model* overcomes this limitation by modeling the entire object's direction of motion as a whole.

The object motion model O^k is a MAP-estimate at each pixel over the set of rigid motion models in $\mathcal{M}^k_T$. We compute the probability of each rigid motion $M^r \in \mathcal{M}^k_T$ given the observed flow $\vec{v_t}$ at a particular pixel position (x, y) (Eq. 7) and assign the most likely motion model to that pixel (Eq. 8). An example of this is shown in Figure 3.

$$p(M^r_{xy}|\vec{v_t}) = \frac{p(\vec{v_t}|M^r_{xy}) \cdot p(M^r_{xy})}{p(\vec{v_t})} \quad (7)$$

$$O^k_{xy} = \arg\max_{M^r_{xy}}(p(M^r_{xy}|\vec{v_t})) \quad (8)$$

3.4. Assigning pixels to moving objects

Given the object motion models O^k we segment a frame into its independently moving objects. Similar to how we

[2] We do not have L^T_{object}, the object segmentation of the current frame, at this point.

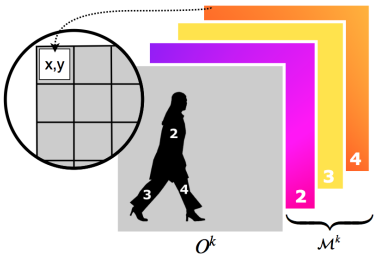

Figure 3: **The object motion model.** In this figure, the k-th object's motion *(walking person)* is described by three rigid motion models, forming the set $\mathcal{M}^k = \{M^2, M^3, M^4\}$. The object motion model O^k is a MAP-estimate at each pixel (x, y) over rigid motion models in $\mathcal{M}^k$.

assign pixels to rigid motion models (Eq. 4), the goal is now to choose among K high level object motion models at each pixel location (x, y), the one with highest probability given the optical flow vector $\vec{v_t}$:

$$p(O^k_{xy}|\vec{v_t}) = \frac{p(\vec{v_t}|O^k_{xy}) \cdot p(O^k_{xy})}{p(v_t)}. \tag{9}$$

This leads to a moving object segmentation,

$$L_{object} = \arg\max_k (p(O^k_{xy}|\vec{v_t})). \tag{10}$$

Likelihoods and priors are computed similarly to the segmentation procedure of a frame into rigid motion components 3.2 and are derived in the following section.

4. Flow likelihood and prior

Let $\vec{q} = (r, \theta)$ be the *true translational motion field vector* (with magnitude r and angle θ), representing the total motion field less the component due to camera rotation. Let $\vec{v_t}$ be the translational component of the *observed*[3] *optical flow vector* $\vec{v}$. We model $\vec{v_t}$ as a noisy observation of $\vec{q}$:

$$\vec{v_t} = \vec{q} + \vec{n}. \tag{11}$$

Inspired by [17], we model flow noise $\vec{n} = (n_u, n_v)$ as a product of Laplacian distributions (for the u and v components), where the parameters depend upon the motion field magnitude r:

$$\vec{n} \sim \mathsf{Laplace}(b_{n_u}(r)) \cdot \mathsf{Laplace}(b_{n_v}(r)). \tag{12}$$

With these assumptions we derive our new flow likelihood, the probability of $\vec{v_t}$ given a rigid motion model M^j (or given an object motion model O^k, respectively):[4]

$$p(\vec{v_t} \mid M^j_{xy}) = \int_0^\infty p(\vec{v_t}, r \mid M^j_{xy})\, dr \tag{13}$$

$$= \int_0^\infty p(\vec{v_t} \mid r, M^j_{xy})\; p(r \mid M^j_{xy})\, dr \tag{14}$$

$$\overset{(a)}{=} \int_0^\infty p(\vec{v_t} \mid \vec{q})\; p(r \mid M^j_{xy})\, dr \tag{15}$$

$$\overset{(b)}{=} \int_0^\infty p(\vec{n}; r)\; p(r \mid M^j_{xy})\, dr. \tag{16}$$

The equality (a) follows since the motion field vector $\vec{q}$ is just a combination of the motion field magnitude r and the motion direction M^j_{xy}. The final equality (b) expresses the fact that the only uncertainty in $\vec{v_t}$ is due to the flow noise $\vec{n}$. The noise variance depends upon r. Parameters of the flow noise distribution are estimated from the Sintel database [6], details of which can be found in the supplementary material.

$p(r \mid M^j_{xy})$ is the probability of flow magnitude r given a particular motion direction M^j_{xy}. We assume that $p(r)$ is independent of the flow direction θ and approximate it as an exponential distribution with parameter b_r:

$$p(r \mid M^j_{xy}) \approx \mathsf{Exp}(r; b_r). \tag{17}$$

The scale parameter b_r is learned using the FBMS-59 training data set [4, 3]. We discuss the relationship between the variance of the flow noise and the magnitude r of the motion field in the supplementary material.

Prior. The prior $p(M^j_{xy})$ on a particular rigid motion model at each pixel includes information about the posterior probability of each motion from the previous frame (the *motion prior*) and another factor that restricts the position of that component in the next frame to a position close to its expected position (the *location prior*).

Motion prior. To get a rough estimate about the motion modeled by M^j we proceed as follows: (1) We propagate the posterior of $p(M^j_{xy}|\vec{v_t})$ from the previous frame along the previous frame's optical flow. (2) We interpolate regions of disocclusion by iteratively smoothing from adjacent unoccluded regions. (3) Then we spatially distribute the probability that each motion component is present by smoothing the prior with a 7x7 Gaussian.

Location prior. The location prior restricts the location of a motion component to being near its former location. If there are multiple objects with similar motion, it is important that each object motion be described by its own set of rigid motion components. A rigid motion model cannot be shared among multiple objects. Therefore we propagate the hard segmentation from the previous frame spatially in a manner similar to the motion prior.

[3] We refer to the flow vector as "observed", but it is the output of an optical flow algorithm which has access to a pair of frames.

[4] We define the likelihood of a "new motion" that was not observed before to be $p(\vec{v_t} \mid M^{new}) = \frac{1}{2\pi}\int_0^{2\pi} p(\vec{v_t} \mid M)\, dM$. The likelihood of a new motion direction is the average likelihood over all possible motion directions.

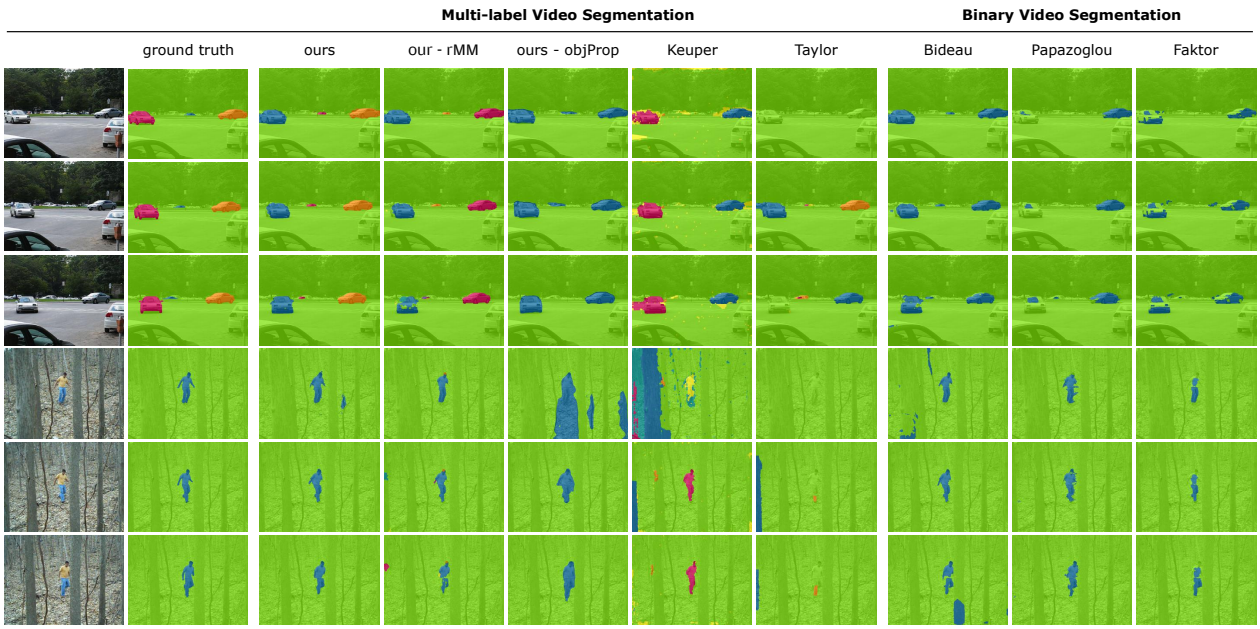

Figure 4: **Sample results.** Top to bottom: The first three rows show the video *cars5* from the FBMS-59 test set. Rows four to six show results on the video *forest* of the complex background data set. For both videos we show frames 1, 10 and 20. We show results on our final version of our algorithm ("ours") as well as for intermediate results of our final algorithm ("ours - rMM" and "ours - objProp"). Comparisons to state of the art methods on multi-label segmentation and binary segmentation are shown in rows 6-10 [20, 42, 3, 32, 10].

Multi-label Video Segmentation

	Testset: FBMS-motion (30 sequences)				**complex background** (5 sequences)				**camouflaged animals** (9 sequences)				**all** (44 sequences)			
	P	**R**	**F**	**ΔObj**	P	R	F	**ΔObj**	P	R	F	**ΔObj**	P	R	F	**ΔObj**
[20]	**74.64**	**62.03**	**63.59**	**7.7**	**67.62**	**58.28**	**60.27**	**3.4**	77.78	**68.10**	**69.97**	**5.7**	**74.48**	**62.84**	**64.52**	**6.8**
[42]	72.69	54.36	56.32	11.7	60.79	44.74	45.83	**3.4**	**84.71**	59.40	61.52	22.2	73.80	54.29	56.19	12.9
ours+CRF	**74.23**	**63.07**	**64.97**	**4**	**64.85**	**67.28**	**65.60**	**3.4**	**83.84**	**69.99**	**72.15**	**5**	**75.13**	**64.96**	**66.51**	**4.1**

Binary Video Segmentation

	Testset: FBMS-motion (30 sequences)				**complex background** (5 sequences)				**camouflaged animals** (9 sequences)				**all** (44 sequences)			
	P	**R**	**F**	**ΔObj**	P	R	F	**ΔObj**	P	R	F	**ΔObj**	P	R	F	**ΔObj**
[3]	79.94	80.76	77.33	-	84.31	**91.74**	**86.56**	-	**81.86**	**74.55**	**76.31**	-	80.83	**80.74**	**78.17**	-
[32]	83.86	79.96	79.56	-	**87.57**	84.95	80.64	-	73.31	56.65	60.38	-	82.12	75.76	75.76	-
[10]	86.24	76.25	77.33	-	79.91	69.31	73.65	-	**82.34**	68.45	72.48	-	84.72	73.92	75.91	-
[43]	**87.29**	72.19	74.79	-	86.78	77.49	78.19	-	77.82	62.03	64.84	-	**85.30**	70.71	73.14	-
[44]	**92.40**	**85.07**	**86.96**	-	74.58	77.02	70.52	-	77.62	51.08	50.82	-	**87.35**	77.20	77.67	-
ours+CRF	85.53	**83.14**	**81.85**	-	**87.69**	**93.13**	**90.11**	-	80.37	**75.21**	**75.95**	-	84.72	**82.65**	**81.49**	-

Table 1: **Comparison to state-of-the-art.** We compare to binary [10, 3, 32, 43] as well as multi-label video segmentation approaches [20, 42]. The top results are highlighted in **green** and the second-best results in **blue**.

5. Experiments

We evaluated our work on three motion segmentation data sets: FBMS-59 [4], the Complex Background data set [27], and the Camouflaged Animals data set [3]. As discussed in [2], FBMS-59 shows a significant number of annotation errors. We use a corrected version of the dataset that is linked on the original dataset's web site. Our main results are for multi-label segmentation, but we also convert our results to a binary segmentation form for comparison with previous work on binary motion segmentation. In addition, we show segmentation results of each stage of our moving object segmentation algorithm – segmentation into rigid motion models (*rMM*), segmentation of the video using object proposals mask of SharpMask directly (*objP*), segmentation of the video using a constant variance of the optical flow error for all flow magnitudes (*cVar*) and results of our final moving object segmentation algorithm *(ours)*. Videos are available in the supplementary material.

	Multi-label Video Segmentation				Binary Video Segmentation		
	all (44 sequences)				**all** (44 sequences)		
	P	**R**	**F**	**ΔObj**	P	R	F
cVar	**76.43**	62.19	64.86	**3.4**	**85.78**	81.09	81.15
rMM	76.01	50.11	52.69	85.88	81.05	81.81	78.91
objP	-	-	-	-	77.15	**85.03**	78.78
ours	74.75	64.70	66.45	4.3	83.66	82.68	81.27
ours+CRF	75.13	**64.96**	**66.51**	4.1	84.72	82.65	**81.49**

Table 2: **Ablation study.** We compare five versions of our algorithm to show how each part of the algorithm affects the performance of the overall motion segmentation method.

Evaluation scheme. We adopted the multi-label evaluation scheme from [30] and add an additional measure ΔObj that represents the accuracy of the segmented object count. ΔObj is the average absolute difference of the ground truth object count in each frame and the number of objects identified by the algorithm. A drawback of the evaluation scheme proposed by [30] is that it does not penalize algorithms much for large numbers of unnecessary (additional) segmented objects. Thus, the F-score of [30] alone does not entirely capture whether the algorithm has an accurate count of the number of objects and the additional ΔObj measure is necessary for a representative evaluation.

Ablation study. To show the contribution of each part of our algorithm separately, we evaluate intermediate results of our method as well as specific adaptations, shown in Tab. 2:

1. *Constant variance (cVar):* Modeling the variance of optical flow error as a function of the optical flow magnitude leads to an improvement of about 2% over all data sets. Regarding precision and ΔObj, we outperform our final motion segmentation approach – cVar segments fewer objects and, due to less false positives, the precision increases. However, the overall performance is worse due to low recall.
2. *Segmentation into rigid motion models (rMM):* Simple rigid motion models are not sufficient to model complex object motion. After the first stage – segmentation of a frame into its rigid motion models – complex motion patterns are broken into multiple simple rigid motion models. Thus, it is not surprising that ΔObj increases dramatically to 85.55.
3. *Segmentation into moving object proposals (objP):* Moving object proposals are generated from a subset of the object proposals out of SharpMask[35]. In Figure 4 ("ours - objProp"), it can be seen that the obtained proposals are covering the object completely (high recall); however the object boundaries are very rough. Those inaccurate boundaries – where a large part of the static background is segmented along with a moving object – lead to low performance. Therefore a composed motion model for modeling the motion of an object accurately is necessary and leads to an improved performance.[5]
4. *Conditional Random Field (ours+CRF):* We add a fully-connected CRF [22] on top of our method to refine the segmentations [43, 7]. The CRF hyperparameters were set by cross-validation on the FBMS Training set.

Multi-label experiments. We outperform [20, 42] by significant margins on FBMS-59, Complex Background and Camouflaged Animals datasets (see Tab. 1). The Complex Background dataset shows videos with high variance in depth, which is particularly challenging for trajectory based motion segmentation approaches such as [20], as well as for occlusion-based object segmentation approaches [42]. Over all the videos in these datasets combined, we gain an average improvement of 2% in F-score compared to the second best performing segmentation method [20]. Our ΔObj results are on par or better for Complex Background and Camouflaged Animals; on FBMS, we are more accurate than either of the other methods in segmenting the correct number of objects (Fig.4 for qualitative results).

Binary experiments. In these experiments, we segment each frame into either static background or moving objects, but do not distinguish among the moving objects, enabling us to compare to other methods that address the binary segmentation problem. We outperform other methods based on overall F-score and recall, and on all three performance metrics on the Complex Background dataset. On FBMS we are in second place behind Tokmakov et al. [44] and on Camouflaged Animals the method from Bideau et al. [3] is slightly better (0.36%) than ours. On average over all videos we have a lead of 3.32% over the next best method [3].

6. Discussion

Many previous methods have shown impressive results in motion segmentation using just low-level or low and mid-level cues [3, 20, 11, 32, 42, 31, 29, 48, 10]. Like recent work in optical flow [39] that uses the power of CNNs to condition optical flow on semantic regions, it seems logical to incorporate this type of high-level information into motion segmentation. We presented a hierarchical statistical method that leverages perspective geometry to model low level parts and semantic segmentation results from a CNN, and combines these parts in a logical way to form higher level objects. We demonstrated best average results across three major motion segmentation datasets and showed strong performance on a wide variety of challenging videos.

[5] Since the object proposal masks of SharpMask might be overlapping or describe the same object, an evaluation of multi-label segmentation is not directly possible for *objP*.

References

[1] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *Proc. ECCV*, pages 154–170. Springer, 2016. 2

[2] P. Bideau and E. Learned-Miller. A detailed rubric for motion segmentation. *arXiv preprint arXiv:1610.10033*, 2016. 2, 7

[3] P. Bideau and E. Learned-Miller. It's moving! A probabilistic model for causal motion segmentation in moving camera videos. In *Proc. ECCV*, pages 433–449. Springer, 2016. 1, 2, 3, 6, 7, 8

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proc. ECCV*, pages 282–295. Springer, 2010. 2, 6, 7

[5] A. R. Bruss and B. K. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20, 1983. 4

[6] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. ECCV*, pages 611–625. Springer-Verlag, 2012. 2, 6

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proc. ICLR*, 2015. 8

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv preprint arXiv:1606.00915*, 2016. 1

[9] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *Proc. ICCV*, pages 686–695. IEEE, 2017. 2

[10] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proc. BMVC*, volume 2, page 8, 2014. 2, 7, 8

[11] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Proc. CVPR*, pages 1846–1853, 2012. 2, 8

[12] R. Girshick. Fast R-CNN. In *Proc. ICCV*, pages 1440–1448, 2015. 1

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. CVPR*, pages 447–456, 2015. 1

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 1

[15] B. K. Horn. Projective geometry considered harmful, 1999 (accessed 2018-03-18). http://people.csail.mit.edu/bkph/articles/Harmful.pdf. 2

[16] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *PAMI*, 20(6):577–589, 1998. 1, 2

[17] A. Jaegle, S. Phillips, and K. Daniilidis. Fast, robust, continuous monocular egomotion computation. In *Proc. ICRA*, pages 773–780. IEEE, 2016. 6

[18] S. Jain, B. Xiong, and K. Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *Proc. CVPR*, 2017. 1, 2

[19] M. Keuper. Higher-order minimum cost lifted multicuts for motion segmentation. In *Proc. ICCV*, 2017. 2

[20] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proc. ICCV*, pages 3271–3279, 2015. 2, 7, 8

[21] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proc. CVPR*, pages 7417–7425. IEEE, 2017. 2

[22] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. NIPS*, pages 109–117, 2011. 8

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1097–1105, 2012. 1

[24] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *arXiv preprint arXiv:1611.07709*, 2016. 1

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proc. ECCV*, pages 21–37. Springer, 2016. 1

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, pages 3431–3440, 2015. 1

[27] M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Proc. ICCV*, pages 1577–1584, 2013. 2, 7

[28] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *Proc. ICCV*, pages 1583–1590. IEEE, 2011. 2

[29] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *Proc. CVPR*, pages 614–621. IEEE, 2012. 8

[30] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 36(6):1187–1200, 2014. 8

[31] A. S. Ogale, C. Fermüller, and Y. Aloimonos. Motion segmentation using occlusions. *PAMI*, 27(6):988–992, 2005. 2, 8

[32] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proc. ICCV*, pages 1777–1784, 2013. 2, 7, 8

[33] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proc. CVPR*, 2017. 2

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. CVPR*, pages 724–732, 2016. 2

[35] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *Proc. ECCV*, 2016. 1, 2, 3, 4, 5, 8

[36] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, pages 779–788, 2016. 1

[38] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pages 234–241. Springer, 2015. 1

[39] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *Proc. CVPR*, pages 3889–3898, 2016. 2, 8

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[41] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proc. CVPR*, pages 2432–2439. IEEE, 2010. 3, 4

[42] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *Proc. CVPR*, pages 4268–4276, 2015. 2, 7, 8

[43] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *Proc. CVPR*, 2017. 1, 2, 7, 8

[44] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proc. ICCV*, 2017. 1, 2, 7, 8

[45] P. H. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998. 2

[46] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *Proc. CVPR*, pages 3899–3908, 2016. 2

[47] J. Vertens, A. Valada, and W. Burgard. SMSnet: Semantic motion segmentation using deep convolutional neural networks. In *Proc. IROS*, 2017. 2

[48] D. Zamalieva and A. Yilmaz. Background subtraction for the moving camera: A geometric approach. *CVIU*, 127:73–85, 2014. 1, 2, 8

[49] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 1