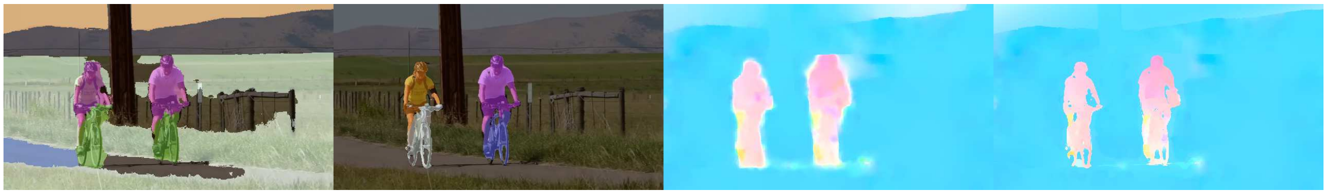
Optical Flow with Semantic Segmentation and Localized Layers

Laura Sevilla-Lara[1] Deqing Sun[2,3] Varun Jampani[1] Michael J. Black[1]

[1]MPI for Intelligent Systems
{laura.sevilla, varun.jampani, black}@tuebingen.mpg.de

[2]NVIDIA, [3]Harvard University
deqings@nvidia.com



(a) Initial segmentation [9] (b) Our segmentation (c) DiscreteFlow [38] (d) Semantic Optical Flow

Figure 1: (a) Semantic segmentation breaks the image into regions such as road, bike, person, sky, etc. (c) Existing optical flow algorithms do not have access to either the segmentations or the semantics of the classes. (d) Our semantic optical flow algorithm computes motion differently in different regions, depending on the semantic class label, resulting in more precise flow, particularly at object boundaries. (b) The flow also helps refine the segmentation of the foreground objects.

Abstract

Existing optical flow methods make generic, spatially homogeneous, assumptions about the spatial structure of the flow. In reality, optical flow varies across an image depending on object class. Simply put, different objects move differently. Here we exploit recent advances in static semantic scene segmentation to segment the image into objects of different types. We define different models of image motion in these regions depending on the type of object. For example, we model the motion on roads with homographies, vegetation with spatially smooth flow, and independently moving objects like cars and planes with affine motion plus deviations. We then pose the flow estimation problem using a novel formulation of localized layers*, which addresses limitations of traditional layered models for dealing with complex scene motion. Our* semantic flow *method achieves the lowest error of any published monocular method in the KITTI-2015 flow benchmark and produces qualitatively better flow and segmentation than recent top methods on a wide range of natural videos.*

1. Introduction

The accuracy of optical flow methods is improving steadily, as evidenced by results on several recent datasets [8, 13]. However, even state-of-the-art optical flow methods still perform poorly with fast motions, in areas of low texture, and around object (occlusion) boundaries (Fig. 1 (c)). Here we address these issues and improve the estimation of optical flow by using semantic image segmentation. Like flow, the field of semantic segmentation is also making rapid progress, driven by convolutional neural networks (CNNs) and large amounts of labeled data. Here we use a state-of-the-art method [9] (Fig. 1 (a)) and find that existing semantic segmentation methods, while not perfect, are good enough to significantly improve flow estimation.

We use semantic image segmentation in multiple ways. First, it provides information about object boundaries. Second, different objects move differently; roads are flat, cars move independently, and trees sway in the wind. This means that our prior expectations about the image motion should vary between regions with different class labels. Third, the spatial relationships between objects provide information about the relative local depth ordering of regions. Reasoning about depth order is typically challenging and we use the semantics to simplify this, improving flow estimates at occlusion boundaries. Fourth, object identities are constant over time, providing a cue that we exploit to encourage temporal consistency of the optical flow.

To model complex scene motions and to deal well with motion boundaries, we adopt a layered approach [4, 11, 18, 21, 24, 49, 54, 55, 56]. Layered models, however are typically global and cannot represent complex occlusion rela-

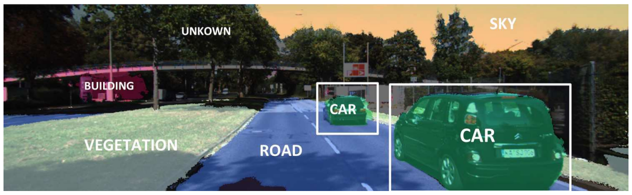


Figure 2: **Localized layered model.** An image is segmented into semantic regions (color coded). Different regions are assigned different motion models. Independently moving objects are shown with a box around them. These regions require reasoning about occlusion because such objects move in front of the background. Within each such region, we make the assumption that two motions are present (the background and the foreground object). The formulation is similar to previous layered models but here the spatial extent of each layer may vary.

tionships. There have been attempts to formulate locally layered models [25, 47], but these methods are still spatially homogenous. Here we propose a new model of *localized layers* in which the number of layers in the scene varies spatially. Any pixel of the scene may belong to one or more layers and these layers may have varying spatial extent. Local layered models are used as needed to capture the motion of relevant objects. In regions corresponding to objects that can move, we may find two motions – the foreground motion of the object against a background motion. Here we use local two-layer models. Rather than a small number of global layers, the result is a patchwork of smaller layered regions on top of background regions as illustrated in Fig. 2. The approach keeps the complexity and optimization manageable by using at most two layers within any patch. And because we can use as many patches as needed, the approach can model complex motions. This adaptive, spatially heterogeneous approach extends layered models to more complex scenes and uses them where they are most valuable.

Each layer or region is represented by a motion model and the type of model varies depending on the semantic label of the region. For regions that are likely to be planar we model their motion with a homography; this includes roads, sky, and water. For regions corresponding to independently moving objects, we treat their motion as affine but allow it to deviate from this assumption; these classes include objects like cars, planes, boats, horses, bicycles, and people. There are still other classes like vegetation and buildings that are diverse in their 3D shape and motion and are consequently not well modeled by a simple parametric motion. Consequently, we model these classes with a classical spatially varying dense flow field. The motion of the scene is then described by composing the motions of all the semantic regions (Fig. 4).

We call the algorithm *semantic optical flow (SOF)* because it exploits scene semantics to improve flow estimation. The approach achieves the lowest error on the KITTI-2015 flow dataset [37], when compared with all published monocular flow methods.[1] We also test the method on a challenging range of sequences from the Internet. There are several reasons for the improvements. First our motion models provide a form of long-range regularization in areas like roads. Since these are well modeled by a homography, accuracy improves. Second, this region-based regularization helps flow estimation in homogeneous regions, which contain few motion cues. Third, the localized layer formulation improves the segmentation and flow around motion boundaries. Key here is that the object segmentation gives a good initialization for layered flow segmentation and gives a good hypothesis for which surface is in front and which is behind; this improves occlusion estimation.

While we focus on improving optical flow, we note that motion can also help with scene segmentation. While current semantic segmentation methods are good, they still struggle to separate object boundaries from appearance boundaries (Fig. 1 (a)). Layered optical flow estimation segments the region and provides additional information about object boundaries (Fig. 1 (b)). When computed over several frames, this segmentation can be quite precise.

In summary, we make two contributions. First, we present the first optical flow method that uses semantic information about scenes, objects, and their segmentation, producing the lowest error among all monocular methods on the KITTI flow benchmark. Second, we show how layered optical flow estimation can be extended to cope with complex scenes. Our results confirm that knowing what and where things are helps the estimation of how they move.

2. Related Work

Motion estimation and segmentation. There is a long history of simultaneously estimating optical flow and its segmentation [36, 40]. Many methods focus on segmentation using motion information alone; we do not consider these here. More relevant are methods that use image segmentation to aid optical flow. Previous work [7, 59] segments the scene into patches according to color or other cues, and then fits parametric flow models within these. Like us they vary the type of model in each region but we go beyond this to use semantic information to determine the appropriate model. Sun et al. [47] first segment the scene into superpixels and then reason about the occlusion relationships between neighboring superpixels (cf. [58]). These methods are generic in the sense that they do not know anything about the objects being segmented but rather seek a partitioning of the scene into coherently moving regions.

[1] The most accurate methods use stereo motion sequences and exploit the stereo to estimate scene structure.

Combining flow models. Here we use different flow models to represent the motion of different parts of the scene. These are combined within our localized layer formulation to define the flow for the whole image. Previous work has explored the combination of different flow algorithms [31, 34]. Irani and Anandan [19] develop a theory for modeling motion in general scenes with varying levels of complexity. The above methods, however, are generic in the sense that they do not use any semantic information about objects to select among the possible models.

Occlusion reasoning and figure-ground. One goal of optical flow estimation is the detection of motion discontinuities that may signal the presence of an object (surface) boundary (see [52] for an overview). Previous methods focus on generic constraints without taking into account object-specific information [6, 46, 50, 52]. In these cases the goal is to detect boundaries that may be useful later for object detection. We turn this around by performing object detection and then using this to detect motion boundaries more accurately.

Layered optical flow. Layered flow estimation has a long history [4, 11, 18, 21, 24, 54, 55] and recent improvements have made the approach more competitive on standard benchmarks [49] and more computationally tractable [56]. The most recent work integrates image segmentation cues with motion cues to produce an accurate segmentation at motion boundaries. In particular, we build on [49], which uses a fully connected graphical model (cf. [26]) to exploit long-range image cues for layer segmentation. Unlike previous work, we apply the model locally within image patches around segmented objects that can move.

Traditional layered models have limitations and are most applicable to simple scenes with a small number of moving objects. Occlusion relationships in the world are complex and 2D motion layers are too restrictive to capture the 3D spatial occlusion relationships in real scenes. Also, while the depth order of layers is important, this may be ambiguous in two frames [48]. Reasoning about layer depth order is combinatorial (K! for K layers), which becomes infeasible in realistic scenarios. To address these issues, locally layered models of motion have been proposed [25, 47]. These models, again, are generic and do not know about objects. Here we find the problem of depth order reasoning is often simplified when we have semantic information. For example, we assume that independently moving objects like cars are in front of static objects like roads. When the assumption holds, as it often does, this simplifies layered flow estimation and produces accurate motion boundaries.

Several methods decompose scenes into layers corresponding to objects [22, 24, 28, 53, 60]. What these methods mean by "object," however, is a region of the image that moves coherently and differently from the background; there is no notion of what this object is. In contrast, Isola and Liu [20] represent static images of scenes as a patchwork of objects layered on top of each other but they do not consider image motion.

Video segmentation. There is significant and increasing interest in the field [12, 14, 32, 41, 42, 57] but the definition of the problem varies between identifying coherent motions or coherent objects regions. Like the approaches above, these methods are generic in that they focus on bottom-up analysis of regions and motion. They typically use optical flow as a cue to track superpixels over time to establish temporal coherence. They usually do not use high-level object recognizers or try to improve optical flow. Taylor *et al*. [51] incorporate object detections and use temporal information to reason about occlusions to improve their segmentation results, but do not compute optical flow. Lalos *et al*. [30] compute optical flow for an object of interest using a tracking-by-detection approach. Unlike us, they only estimate object displacement (not full flow), ignore background motion, and do not take object identity into account.

Semantic segmentation in other low-level vision problems. Object class influences the way things move, but also influences their shape. Recent work uses semantic segmentation to resolve ambiguities in stereo [15], to guide 3D reconstruction [16, 29], and to constrain the motion of the 3D scene by enforcing class label coherence over time [44].

3. Model and Methods

Using a semantic segmentation of the scene allows us to model the motion of different regions of the image differently. We define the motion in the scene compositionally in terms of the motion of the regions. Below we discuss how we compute the motion for each segmented region and then how we combine these into a coherent flow field.

Classes. We define three classes of objects (Things, Planes, and Stuff) that exhibit different types of motion (see Fig. 2). *(1) Things* [2, 17] correspond to objects with a defined spatial extent, that can move independently, are typically seen in the foreground and may be rigid or non-rigid. Things include aeroplane, bicycle, bird, boat, bus, car, cat, cow, dog, horse, motorbike, sheep, train and person. *(2) Planes* are regions like 'roads' that have a broad spatial extent, are roughly planar, and are typically in the background. Other classes that we treat as planes are 'sky' and 'water'. Water is treated as a plane because the air/water boundary is often planar. *(3) Stuff* [3] corresponds to classes that exhibit textural motion or objects like 'buildings' and 'vegetation' that may have a complicated 3D shape, exhibit complex parallax, and for which we have no compact motion representation. Regions of unknown class are modeled as Stuff.

3.1. Preprocessing

Segmentation. We used Caffe [23] to train the semantic segmentation model DeepLab [9], substituting all fully-

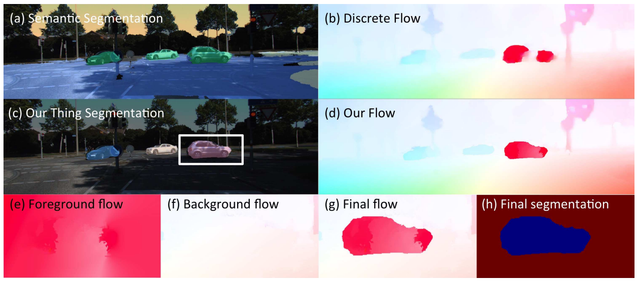


Figure 3: **The method in pictures.** (a) Image with the segmentation into road (blue), car (green), sky (yellow), grass (grey), and "unknown" (clear) superimposed. (b) Initial dense flow computed with DiscreteFlow [38]. The following images show intermediate results in the extracted car region. (c) Our final Thing segmentation. (d) Our final flow. (e) Estimated foreground motion. (f) Estimated background motion. (g) Estimated flow for the localized region. (h) Final layer segmentation (blue is foreground).

connected layers in the VGG network [45] with convolutional layers. We modified the output layer to predict the 22 classes described above and used the atrous [35] algorithm to get denser predictions. We initialized the network with the VGG model and fine-tuned it with standard stochastic gradient descent using a fixed momentum of 0.9 and weight decay of 0.0005 during 200K iterations. The learning rate is 0.0001 for the first 100K iterations and is reduced by 0.1 after every 50K steps. To improve performance [9, 27] we used a densely connected conditional random field (Dense-CRF). The unaries are the CNN output and the pairwise potentials are a position kernel and a bilateral kernel with both position and RGB values. The standard deviation of the filter kernels and their relative weights are cross-validated. The inference in the Dense-CRF model is performed using 10 steps of meanfield. To train the network, we selected 22 of the 540 classes from the Pascal-Context dataset [39].

Thing matching. Given the segmentation in each frame, we compute connected components to obtain regions containing putative objects (Things). Regions smaller than 200 pixels are treated as Stuff. For each Thing found in the first frame, we find its corresponding region in subsequent frames and create a bounding box for layered flow estimation that fully surrounds the object regions across all frames. This defines the spatial extent of the layered flow estimation (Fig. 3). Below we estimate the flow of Things using $T = 5$ frames at a time unless otherwise stated. Figure 2 shows a few Thing regions in one frame. If a Thing region is not found over the entire sub-sequence, it is treated as Stuff.

Initial flow. We also compute an initial dense flow field, $\hat{\mathbf{u}}$ using the DiscreteFlow method [38] based on [43]. We use this in several ways as described below.

3.2. Motion Models

The motion of Planes. We model planar regions using homographies. Given the initial flow vectors $\hat{\mathbf{u}}(\mathbf{x})$, $\mathbf{x} \in R_i$ in region i, we use RANSAC to robustly estimate the parameters, $\mathbf{h}_i$ of the homography. The planar motion then defines the flow $\mathbf{u}_{\text{Plane}}(\mathbf{x}; \mathbf{h}_i)$ for every pixel $\mathbf{x} \in R_i$.

The motion of Stuff. For Stuff we have no class-specific motion model and set the flow in every Stuff region i to be the initial flow; that is $\mathbf{u}_{\text{Stuff}}(\mathbf{x}) = \hat{\mathbf{u}}(\mathbf{x})$ for $\mathbf{x} \in R_i$.

The motion of Things. In Thing regions we expect occlusions and disocclusions, complex geometry, and deformations. Thus, we assume the motion of a Thing can be described as affine plus a smooth deformation from affine. This may sound restrictive but we build on the work of [49], where they show positive results applying this motion model to the entire scene. Our Thing regions are much smaller than the entire scene, and the motion within this region is more likely to satisfy the assumptions. We allow the motion of Things to deviate from affine and the amount of deviation depends on the object class. For example, cars are more rigid than people and their motion is more affine. Consequently we assume that the motion of cars will be more affine and penalize deviations from this assumption more.

While we are interested in the motion of the Thing, because we assume Things are in front of backgrounds, it is actually important to also consider the motion of the background. Specifically, estimating an accurate foreground segmentation requires that we reason about the motion of both foreground and background. We do this using a local layered model based on [49].

Formally, given a sequence of images $\{\mathbf{I}_t, 1 \leq t \leq T\}$, we want to jointly estimate the motion $(\mathbf{u}_{tk}, \mathbf{v}_{tk})$ for every pixel, in each layer, at every frame, as well as group pixels that move together into layers denoted by $\mathbf{g}_{tk}$, where $k \in \{1, 2\}$. We only consider two layers, and thus we only need to estimate the foreground segmentation, $\mathbf{g}_{t1}$, as the background layer is constant. We formulate the local layered energy term (Eq. 1) similar to Sun *et al.* with some modifications described below and refer the reader to [49] for further details. The method estimates the motion of both layers and the segmentation of the foreground region.

The general formulation incorporates occlusion reasoning in the motion estimation using layered segmentation (data term), enforces temporal consistency of layer segmentation (time term) according to the motion, couples semantic segmentation and layered segmentation (layer term), and encourages spatial contiguity of layered segmentation using a fully-connected CRF model (space term).

The **data term** imposes appearance constancy when corresponding pixels are visible at the same layer, and a constant penalty otherwise. It reasons about occlusions by com-

$$E_{\text{Thing}}(\mathbf{u},\mathbf{v},\mathbf{g},\Theta;\mathbf{I},\hat{\mathbf{g}})=\sum_{k=1}^{2}\Bigg\{\sum_{t=1}^{T-1}\{E_{\text{data}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{tk};\mathbf{I}_t,\mathbf{I}_{t+1})+\lambda_{\text{motion}}E_{\text{motion}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{tk},\Theta_{tk}) \quad (1)$$

$$+\lambda_{\text{time}}E_{\text{time}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{t,k},\mathbf{g}_{t+1,k})\}+\sum_{t=1}^{T}\{\lambda_{\text{layer}}E_{\text{layer}}(\mathbf{g}_{tk};\hat{\mathbf{g}}_{tk})+\lambda_{\text{space}}E_{\text{space}}(\mathbf{g}_{tk})\}\Bigg\}.$$

paring the layer assignment of corresponding pixels:

$$E_{\text{data}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{tk};\mathbf{I}_t,\mathbf{I}_{t+1}) = \sum_p \rho_D(I_t^p - I_{t+1}^q)\delta(g_{t1}^p = g_{t+1,1}^q) + \lambda_D\delta(g_{t1}^p \neq g_{t+1,1}^q), \quad (2)$$

where $q=(x+u_{tk}^p, y+v_{tk}^p)$ denotes the corresponding pixel according to the motion for pixel p, for every pixel in the image, ρ_D is a robust penalty function, and λ_D is a constant penalty for occluded pixels and pixels of different objects. The indicator function $\delta(x)$ is 1 if the expression x is true, and 0 otherwise.

The **motion term** encodes two assumptions. First, neighboring pixels should have similar motion if they belong to the same layer. Second, pixels from each layer k should share a global motion model $\bar{\mathbf{u}}(\Theta_{tk})$, where Θ_{tk} are parameters that change over time and depend on the object class k:

$$E_{\text{motion}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{tk},\Theta_{tk}) = \sum_p \sum_{r\in\mathcal{N}_p} \rho(u_{tk}^p - u_{tk}^r)\delta(g_{tk}^p = g_{tk}^r) + \lambda_{\text{aff}} \sum_p \rho_{\text{aff}}(u_{tk}^p - \bar{u}^p(\Theta_{tk})) \quad (3)$$

where the set $\mathcal{N}_p$ contains the four nearest neighbors of pixel p. The motion term for the vertical flow field $\mathbf{v}_t$ is defined similarly.

The **time term** encourages corresponding pixels over time to have the same layer label

$$E_{\text{time}}(\mathbf{u}_{tk},\mathbf{v}_{tk},\mathbf{g}_{tk},\mathbf{g}_{t+1k}) = \sum_p \delta(g_{tk}^p \neq g_{t+1k}^q), \quad (4)$$

where q is the corresponding pixel at the next frame for p according to the motion $(\mathbf{u}_{tk},\mathbf{v}_{tk})$.

The **space term** encourages spatial contiguity of layer segmentation:

$$E_{\text{space}}(\mathbf{g}_{tk}) = \sum_p \sum_{r\neq p} w_r^p \delta(g_{tk}^p \neq g_{tk}^r), \quad (5)$$

where the weight w_r^p is the same as in Sun et al. [49]. This term fully connects each pixel with all other pixels in the localized region. In our implementation, we modify the approach in [49] and apply this, not over the whole frame, but over a detected object region.

The major difference from Sun *et al.* [49] is that we have a semantic segmentation for the foreground and this segmentation is usually reasonably good. Consequently we define a new **coupling term**, E_{layer}, that enforces similarity between the foreground layer segmentation and the semantic segmentation:

$$E_{\text{layer}}(\mathbf{g}_{tk};\hat{\mathbf{g}}_{tk}) = \sum_p \delta(g_{tk}^p \neq \hat{g}_{tk}^p), \quad (6)$$

where $\hat{g}_t$ is the segmentation mask of the foreground Thing.

Initialization and optimization. The layer method requires an initialization of the foreground region $\mathbf{g}$, an initial flow $\hat{\mathbf{u}}$, and parametric motions of both layers $\bar{\mathbf{u}}(\Theta)$.

The initial flow is typically inaccurate at the boundaries and we do not want this to corrupt the initialization. Consequently we compute the initial affine motion ignoring the pixels close to the object boundary both in the background and foreground. We then optimize Eq. 1 using the method in [49]. This refines the flow of each layer and the segmentation (Fig. 3). The segmentation is quite accurate because it uses backward and forward flow and image evidence with the fully connected model in the region (see [49]). The method [49] uses heuristics to reason about depth ordering. Here we use the class category to decide the depth ordering and assume that Things are always foreground.

3.3. Composing the Flow Field

Each Plane and Stuff region gives exactly one flow value per pixel. If these pixels are not occluded by a localized layer, then their flow becomes the final flow value. The localized layers estimate the flow of the foreground and background pixels within an object region. These regions may extend over Plane and Stuff regions, giving multiple possible flow values for these overlapped pixels. We select a single value for each such pixel as follows (Fig. 4). The foreground flow is directly pasted onto the flow field (blue region). When the background region of a localized layer overlaps a Plane, we keep the planar motion (yellow region). When the background overlaps a Stuff region, we take a weighted average of the Stuff flow and the layer flow

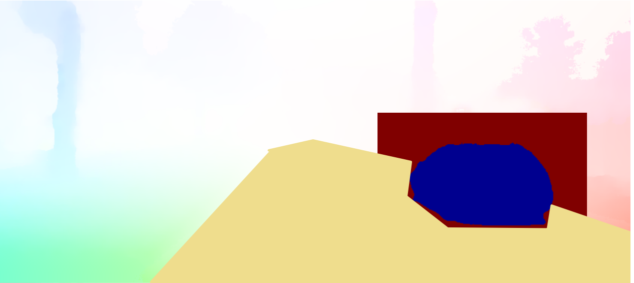


Figure 4: **Compositing the flow.** The motion of Stuff, Planes (yellow) and regions around Things (red and blue) is composited to produce the final flow estimation.

(red region). The weight for the layer flow is high near the foreground and decays to zero at the region boundary. Thus we favor the layered flow estimate near the foreground because it tends to be more accurate at boundaries. We found this approach faster and better than FusionFlow [31].

4. Experiments

We test our Semantic Optical Flow (SOF) method in two different datasets: natural Youtube sequences and KITTI 2015 [37]. Standard optical flow benchmarks do not contain the variety of objects that a semantic segmentation method can recognize. Thus, we collected a suite of natural videos from YouTube, containing objects of the Pascal VOC classes that move. Although there is no ground truth to provide a quantitative analysis, the difference of quality is clearly visible in planar regions and at motion boundaries. All sequences will be made publicly available [1]. In addition, we test our method on the KITTI 2015 dataset, where existing semantic segmentation methods perform reasonably well. We do not include results on the Sintel dataset because semantic segmentation does not produce reasonable results. This is probably due to the fact that the statistics of synthetically generated images are different from those of natural images, like the ones in the enriched Pascal VOC dataset. We tried training the same network using the Sintel training set (manually annotated), and we found that the network did not perform well, presumably due to a shortage of training data. In the Middlebury dataset [5] the semantic segmentation results produce mostly the 'unknown' class, or they correspond to classes without a specific motion model (*i.e.* building), or they are very small regions and we do not consider them. Thus, on Middlebury our results are identical to the initial flow (DiscreteFlow) in all sequences but in one, where our accuracy is 0.004 better.

4.1. KITTI 2015

We quantitatively evaluate our method on the KITTI 2015 benchmark (Fig. 5) using $T = 2$ frames as input.

Method	**Fl-all** (All px)	Fl-bg (All px)	Fl-fg (All px)	**Fl-all** (Nocc)	Fl-bg (Nocc)	Fl-fg (Nocc)
Full	24.26%	23.09%	30.11%	15.35 %	12.97%	26.10%
Discrete	22.38%	21.53%	26.68%	12.18%	9.96%	22.17%
SOF	16.81%	14.63%	27.73%	10.86%	8.11%	23.28%

Table 1: Results for the test set of KITTI 2015. We compare with DiscreteFlow [38] and FullFlow [10], which is the next most accurate published monocular method.

A numerical comparison between DiscreteFlow, FullFlow [10], and our method is shown in Table 1. Our method significantly reduces the overall percentage of outliers compared with DiscreteFlow (from 22.38% to 16.81%). The improvements mainly come from 1) our refined motion for the Planes; and 2) correctly interpolated motion for the occluded background regions. Figure 6 shows several examples where our method fixes large errors of the foreground cars in the initial DiscreteFlow results.

Our method has a slightly higher percentage of outliers in the foreground region. This reveals a tradeoff between segmentation and flow accuracy. The more we restrict the foreground to affine motion, the better the segmentation but the worse the flow estimate. Also our method only assumes two major motions are present in the detected region, and it may fail when the assumption does not hold (Fig. 7). This is due to our segmentation method giving a class segmentation and grouping multiple objects together. To address this, we either need instance-level segmentation of Things or a formulation that deals with more than two layers [48].

The execution time of our method depends on the size of the image, the number of objects, and the size of these. An upper bound for the total time is 6 minutes for a frame of KITTI 2015. Specifically, the initial semantic segmentation takes 10 seconds, the initial motion estimation from DiscreteFlow takes 3 minutes, the motion of Planes takes 2 seconds, and the motion of Things depends on the size of the object, but takes on average 1-2 minutes.

4.2. Natural Sequences.

Figure 8 shows examples on natural sequences downloaded from YouTube. We estimate the flow using non-overlapping 5-frame subsequences. Our method improves over the state-of-the-art optical flow estimation method. It corrects errors in large planar regions and produces more accurate motion boundaries. It is also able to refine the semantic segmentation, especially at object boundaries and in thin regions. These results demonstrate the benefits of our approach when reliable semantic segmentation is available.

5. Conclusion and Future work

We have defined a method for using semantic segmentation to improve optical flow estimation. Our semantic op-

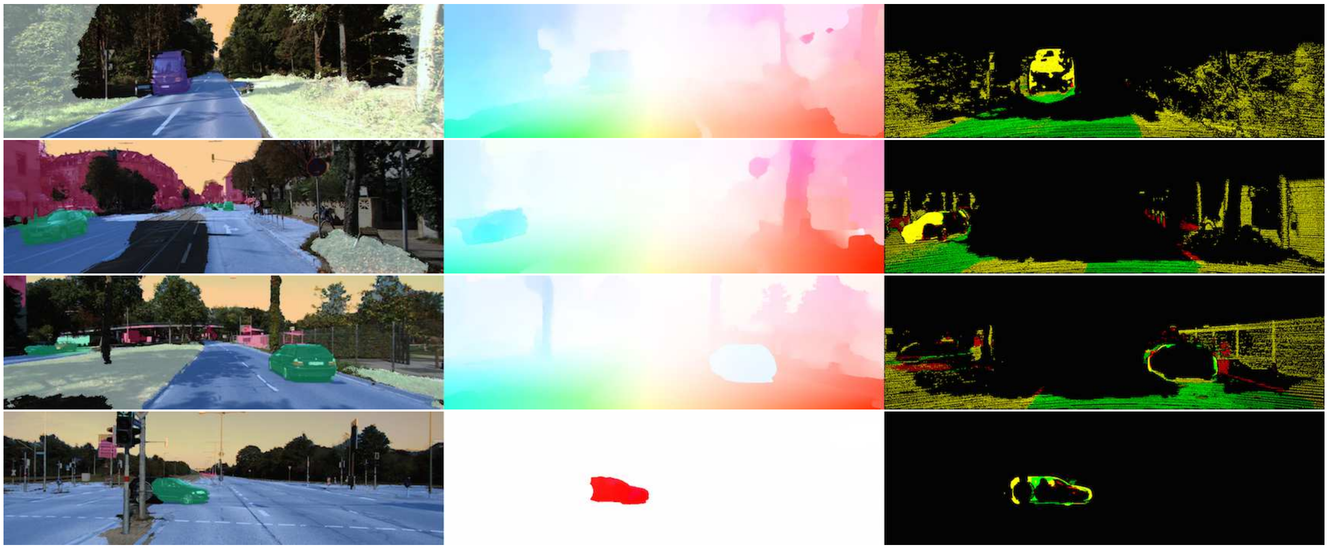


Figure 5: **Examples of Semantic Optical Flow on KITTI 2015.** From left to right: Initial segmentation; Optical flow estimation from SOF; Comparison of outliers between DiscreteFlow and SOF (black pixels indicate neither algorithm produced an outlier in that location, yellow pixels indicate both methods produced an outlier, green pixels indicate DiscreteFlow was incorrect SOF was correct, and red pixels indicate DiscreteFlow was correct but SOF was not). Notice that much of the gain from SOF is on the road, especially at occluded regions, and on the areas close to cars.

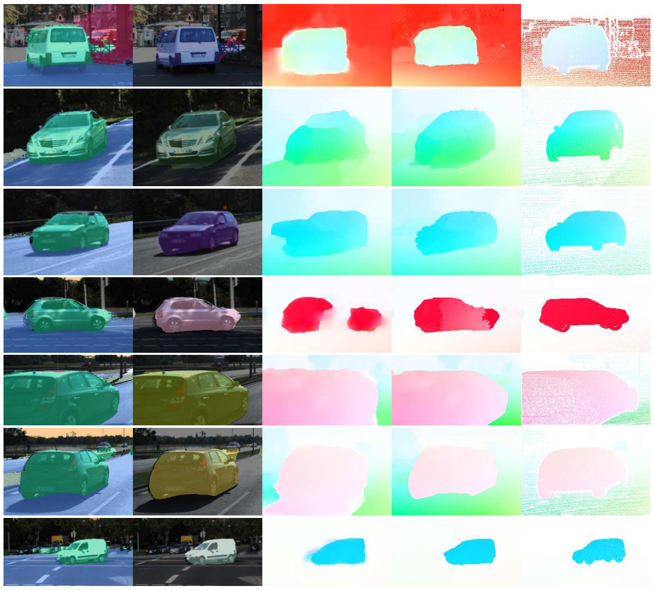


Figure 6: **Comparison of details recovered by Semantic Optical Flow.** From left to right: Initial segmentation; SOF segmentation; Optical flow estimation from DiscreteFlow; Optical flow estimation from SOF; Ground truth flow.

tical flow method uses object class labels to determine the appropriate motion model to apply in each region. We classify a scene into Things, which move independently, Planes, which are large, roughly planar regions, and Stuff, which is everything else. We focus on the estimation of Things using a localized layer model in which we only apply layered optical flow in constrained regions around objects of interest. We introduce a novel constraint to prefer layered segmentations that resemble our semantic segmentation. A key insight is that a detected object region is likely to contain at most two motions and the object is likely to be in front. We show that using motion we are able to visually improve the segmentation, sometimes dramatically. We tested the method on the KITTI-2015 flow benchmark and have the lowest error of any monocular method by a significant margin at the time of writing. We also tested on a wide range of other videos containing more varied classes and see clear qualitative improvement in terms of flow and segmentation. This work confirms the benefit of using high quality segmentation for optical flow and for exploiting knowledge of the class labels in estimating flow. This opens several doors for future work. In particular, it may be possible to formulate our localized layer model as a single objective function and optimize it as such; this may improve results further.

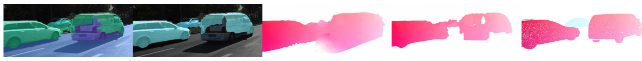


Figure 7: **Failure case.** From left to right: Initial segmentation; SOF segmentation; Flow estimation from DiscreteFlow; Flow estimation from SOF; Ground truth flow. Our layered method assumes two dominant motions in the region, failing if there are more than two motions.

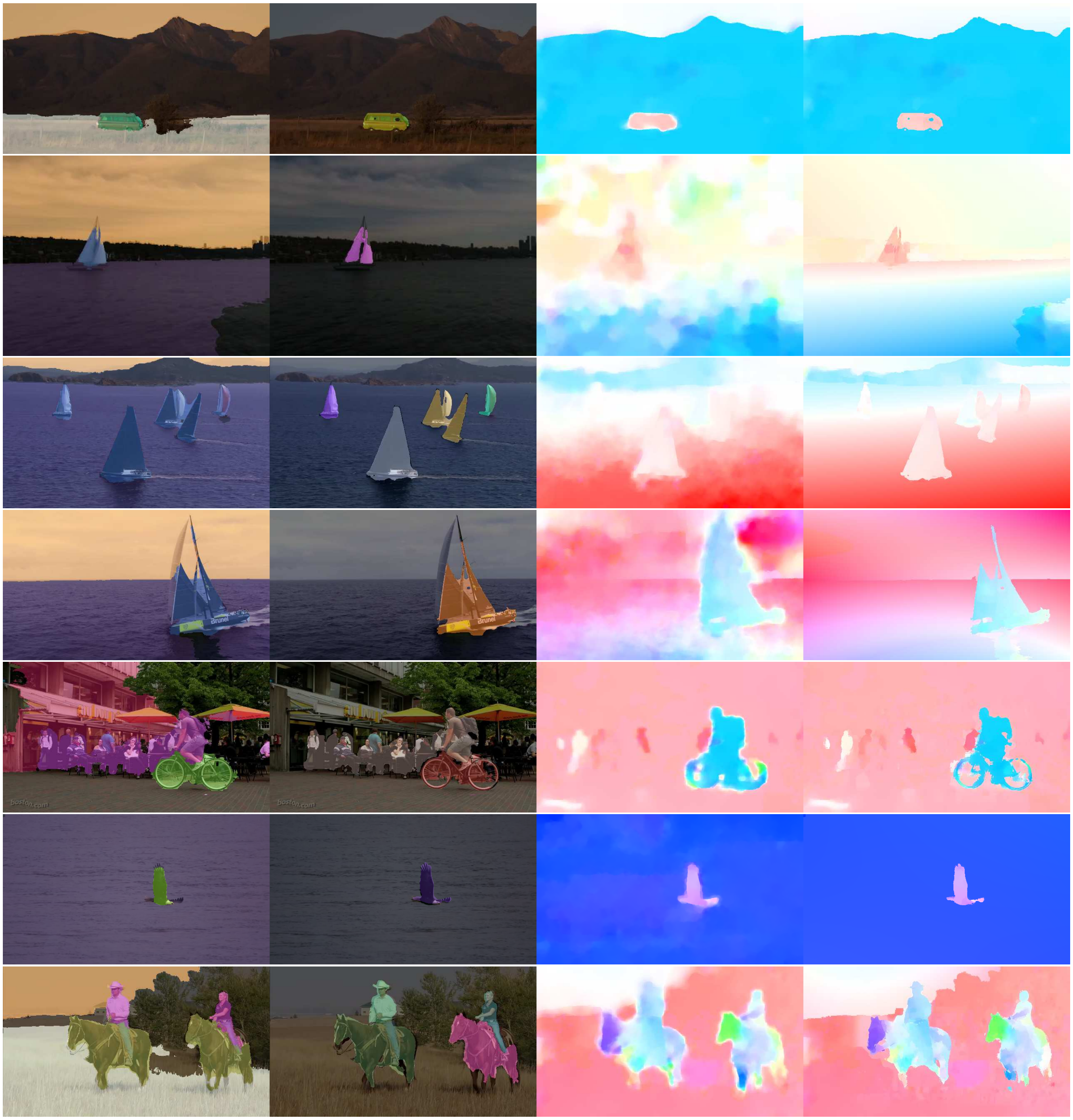


Figure 8: **Qualitative analysis of Semantic Optical Flow.** We show a few representative examples from the YouTube dataset. From left to right: Initial segmentation, SOF segmentation, optical flow estimation from DiscreteFlow, optical flow estimation from SOF. More examples can be found at [1].

Additionally it would be useful, but challenging, to integrate flow estimation with semantic segmentation. Flow information may even help with class recognition in addition to segmentation.

Acknowledgments. We thank Martin Kiefel and Jonas Wulff for their help and insight. We thank Raquel Urtasun for helpful comments on the manuscript.

References

[1] https://ps.is.tuebingen.mpg.de/research_projects/semantic-optical-flow 6, 8

[2] E. Adelson and J. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. Movshon, editors, *Computational Models of Visual Processing*, pages 1–20. MIT Press, 1991. 3

[3] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Photonics West 2001-Electronic Imaging*, pages 1–12. International Society for Optics and Photonics, 2001. 3

[4] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *ICCV*, pages 777–784, Jun 1995. 1, 3

[5] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, Mar. 2011. 6

[6] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, July 2000. 3

[7] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, Oct. 1996. 2

[8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, IV, pages 611–625, 2012. 1

[9] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 1, 3, 4

[10] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. *CVPR*, 2016. 6

[11] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Workshop on Visual Motion*, pages 173–178, 1991. 1, 3

[12] F. Galasso, N. Nagaraja, T. Jimenez Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, pages 3527–3534, Dec 2013. 3

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231 – 1237, Sept. 2013. 1

[14] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. *CVPR*, pages 2141–2148, 2010. 3

[15] F. Gney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR*, 2015. 3

[16] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *CVPR*, pages 97–104, 2013. 3

[17] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, pages 30–43. Springer, 2008. 3

[18] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representations. In *ICPR*, pages A:743–746, 1994. 1, 3

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

[20] P. Isola and C. Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *CVPR*, pages 3048–3055, 2015. 3

[21] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, 1993. 1, 3

[22] A. Jepson, D. Fleet, and M. Black. A layered motion representation with occlusion and compact spatial support. In *ECCV*, volume I, pages 692–706, 2002. 3

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. 3

[24] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *CVPR*, pages I:199–206, 2001. 1, 3

[25] S. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR*, pages 307–314, June 1996. 2, 3

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011. 3

[27] P. Krähenbühl and V. Koltun. Efficient nonlocal regularization for optical flow. In *ECCV*, pages I:356–369, 2012. 4

[28] M. Kumar, P. Torr, and A. Zisserman. Learning layered motion segmentations of video. *IJCV*, 76(3):301–319, March 2008. 3

[29] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2011. 3

[30] C. Lalos, H. Grabner, L. Van Gool, and T. Varvarigou. Object flow: Learning object displacement. In *ACCV*, pages 133–142, 2011. 3

[31] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, pages 1–8, June 2008. 3, 6

[32] B. Liu and X. He. Multiclass semantic video segmentation with object-level active inference. In *CVPR*, pages 4286–4294, 2015. 3

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

[34] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *PAMI*, 35(5):1107–1120, May 2013. 3

[35] S. Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008. 4

[36] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. Im. Proc.*, 7(5):703–719, May 1988. 2

[37] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, pages 3061–3070, 2015. 2, 6

[38] M. Menze, C. Heipke, and A. Geiger. Discrete optimization for optical flow. In *German Conference on Pattern Recognition (GCPR)*, volume 9358, pages 16–28. Springer International Publishing, 2015. 1, 4, 6

[39] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 4

[40] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *PAMI*, 9(2):220–228, March 1987. 2

[41] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 36(6):1187–1200, June 2014. 3

[42] X. Ren, T. Han, and Z. He. Ensemble video object cut in highly dynamic scenes. In *CVPR*, pages 1947–1954, 2013. 3

[43] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR*, 2015. 4

[44] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. Stixmantics: A medium-level model for real-time semantic scene understanding. In *ECCV*, pages 533–548. Springer, 2014. 3

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[46] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, 82(3):325–357, May 2009. 3

[47] D. Sun, C. Liu, and H. Pfister. Local layering for joint motion estimation and occlusion detection. In *CVPR*, pages 1098–1105, 2014. 2, 3

[48] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775, 2012. 3, 6

[49] D. Sun, J. Wulff, E. Sudderth, H. Pfister, and M. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, pages 2451–2458, June 2013. 1, 3, 4, 5

[50] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240, 2011. 3

[51] B. Taylor, A. Ayvaci, A. Ravichandran, and S. Soatto. Semantic video segmentation from occlusion relations within a convex optimization framework. In *EMMCVPR*, volume 8081, pages 195–208. Springer Berlin Heidelberg, 2013. 3

[52] W. B. Thompson. Exploiting discontinuities in optical flow. *IJCV*, 30(3):163–173, Dec. 1998. 3

[53] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *ICCV*, pages 747–754, 2009. 3

[54] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. IP*, 3(5):625–638, 1994. 1, 3

[55] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR*, pages 520–526, 1997. 1, 3

[56] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, pages 120–130, 2015. 1, 3

[57] C. Xu, C. Xiong, and J. Corso. Streaming hierarchical video segmentation. In *ECCV*, volume 7577, pages 626–639. Springer Berlin Heidelberg, 2012. 3

[58] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. *CVPR*, 0:1862–1869, 2013. 2

[59] J. Yang and H. Li. Dense, accurate optical flow estimation with piecewise parametric model. In *CVPR*, pages 1019–1027, June 2015. 2

[60] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *ICCV*, pages 1079–1085, 2003. 3