Cascaded Scene Flow Prediction using Semantic Segmentation

Zhile Ren
Brown University
ren@cs.brown.edu

Deqing Sun
NVIDIA
deqings@nvidia.com

Jan Kautz
NVIDIA
jkautz@nvidia.com

Erik B. Sudderth
UC Irvine
sudderth@uci.edu

Abstract

Given two consecutive frames from a pair of stereo cameras, 3D scene flow methods simultaneously estimate the 3D geometry and motion of the observed scene. Many existing approaches use superpixels for regularization, but may predict inconsistent shapes and motions inside rigidly moving objects. We instead assume that scenes consist of foreground objects rigidly moving in front of a static background, and use semantic cues to produce pixel-accurate scene flow estimates. Our cascaded classification framework accurately models 3D scenes by iteratively refining semantic segmentation masks, stereo correspondences, 3D rigid motion estimates, and optical flow fields. We evaluate our method on the challenging KITTI autonomous driving benchmark, and show that accounting for the motion of segmented vehicles leads to state-of-the-art performance.

1. Introduction

The *scene flow* [30] is the dense 3D geometry and motion of a dynamic scene. Given images captured by calibrated cameras at two (or more) frames, a 3D motion field can be recovered by projecting 2D motion (optical flow) estimates onto a depth map inferred via binocular stereo matching. Scene flow algorithms have many applications, ranging from driver assistance [22] to 3D motion capture [10].

The problems of optical flow estimation [28, 2] and binocular stereo reconstruction [26] have been widely studied in isolation. Recent scene flow methods [19, 36, 32] leverage 3D geometric cues to improve stereo and flow estimates, as evaluated on road scenes from the challenging KITTI scene flow benchmark [21]. State-of-the-art scene flow algorithms [33, 21] assume superpixels are approximately planar and undergo rigid 3D motion. Conditional random fields then provide temporal and spatial regularization for 3D motion estimates. Those methods generally perform well on background regions of the scene, but are significantly less accurate for moving foreground objects.

Estimating the geometry of rapidly moving foreground objects is difficult, especially near motion boundaries. Vehicles are particularly challenging because painted surfaces have little texture, windshields are transparent, and reflections violate the brightness constancy assumptions underlying stereo and flow likelihoods. However, accurate estimation of vehicle geometry and motion is critical for autonomous driving applications. To improve accuracy, it is natural to design models that separately model the motion of objects and background regions [24, 21].

Several recent methods for the estimation of optical flow [1, 14, 27, 24] have used semantic cues to improve accuracy. While motion segmentation using purely bottom-up cues is challenging, recent advances in semantic segmentation [38, 6] make it possible to accurately segment traffic scenes given a single RGB image. Given segmented object boundaries, object-specific 3D motion models may then be used to increase the accuracy of optical flow methods.

In this paper, we use instance-level semantic segmentations [6] and piecewise-rigid scene flow estimates [32] as inputs, and integrate them via a cascade of *conditional random fields* (CRFs) [17]. We define pixel-level CRFs relating dense segmentation masks, stereo depth maps, optical flow fields, and rigid 3D motion estimates for foreground objects. Due to the high dimensionality of these variables, we refine them iteratively using a cascaded classification model [12], where each stage of the cascade is tuned via structural SVM learning algorithms [15]. We evaluate using previous scene flow annotations [21] of the challenging KITTI autonomous driving benchmark [11], and improve on the state-of-the-art in two-frame scene flow estimation. Our work demonstrates the importance of semantic cues in the recovery of the geometry and motion of 3D scenes.

2. Related Methods for Scene Flow Estimation

Vedula *et al*. [30] first defined the scene flow as the dense 3D motion of all points in an observed scene, and recovered voxel-based flow estimates using 2D optical flow fields from several calibrated cameras. Huguet and Devernay [13] then proposed a variational approach and jointly solved for stereo and optical flow, while Wedel *et al*. [35] decoupled the stereo and flow problems for efficiency. These classic algorithms only improve marginally over modern, state-of-

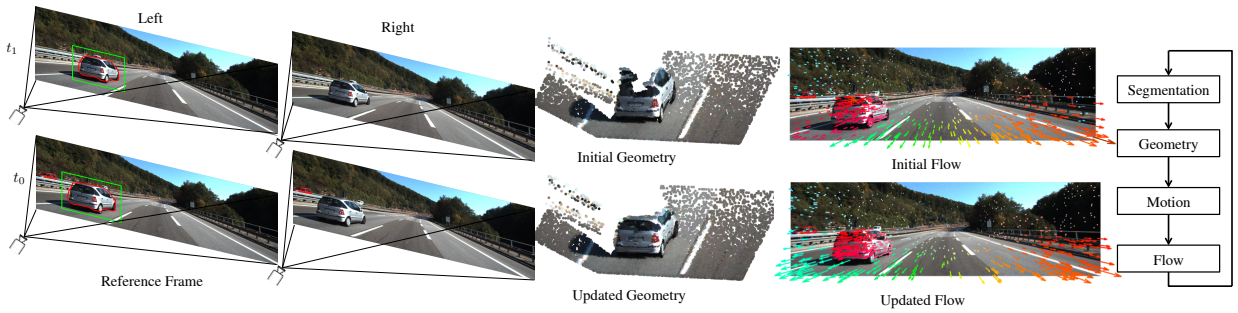


Figure 1. An illustration of our method for scene flow estimation. Given two frames from a pair of stereo cameras, and initial geometry and optical flow estimates provided by a non-semantic scene flow algorithm [32], we use semantic segmentation cues [6] to identify foreground vehicles. Our cascade of CRFs (right) then iteratively refines the inferred segmentation, geometry, 3D motion, and flow. In this example, our updated geometry estimate reduces flow errors in the windshield of the car and the adjacent road (lower left).

the-art stereo and optical flow methods.

Although scene flow algorithms require more input images than standard optical flow or stereo reconstruction methods, the task is still challenging due to the high dimensionality of the output disparity and motion fields. To reduce the solution space, Vogel *et al*. [32] introduced a *piecewise rigid scene flow* (PRSF) model and used superpixels to constrain scene flow estimation. For the first time, they showed that scene flow methods could outperform stereo and optical flow methods by a large margin on the challenging KITTI dataset [11]. In follow-up work [33] they extended their formulation to multiple frames and improved accuracy. However, because the PRSF model relies on bottom-up cues for superpixel segmentation, it tends to over-segment foreground objects such as cars. Over-segmented parts are allocated independent motion models, so global information cannot be effectively shared.

Inspired by the success of Vogel *et al*. [32], Menze and Geiger [21] annotated a new KITTI dataset with dynamic foreground objects for scene flow evaluation. They proposed an *object scene flow* (OSF) algorithm that segments the scene into independently moving regions, and encourages the superpixels within each region to have similar 3D motion. Although the performance of OSF improved on baselines, the "objects" in their model are assumed to be planar and initialized via bottom-up motion estimation, so physical objects are often over-segmented. The inference time required for the OSF method is also significantly longer than most competing methods.

The successes of *convolutional neural networks* (CNNs) for high-level vision tasks has motivated CNN-based regression methods for low-level vision. Dosovitskiy *et al*. [7] introduced a denoising autoencoder network, called FlowNet, for estimating optical flow. Mayer *et al*. [20] extended the FlowNet to disparity and scene flow estimation with a large synthetic dataset. While CNN models generate scene flow predictions rapidly, networks trained on synthetic data are not competitive with state-of-the-art methods on the real-world KITTI scene flow benchmark [21].

Some related work integrates automatic motion segmentation with optical flow prediction [24, 29], but assumes large differences between the motion of objects and cameras, and requires multiple input frames. Exploiting the recent success of CNNs for semantic segmentation [6, 38], semantic cues have been shown to improve optical flow estimation [1, 14, 27]. Concurrent work [4] also shows that semantic cues can improve scene flow estimation. In this paper, we propose a coherent model of semantic segmentation, scene geometry, and object motion. We use a cascaded prediction framework [12] to efficiently solve this high-dimensional inference task. We evaluate our algorithm on the challenging KITTI dataset [21] and show that using semantic cues leads to state-of-the-art scene flow estimates.

3. Modeling Semantic Scene Flow

Given two consecutive frames I, J and their corresponding stereo pairs I', J', our goal is to estimate the segmentation mask, stereo disparity, and optical flow for each pixel in the reference frame (Fig. 1). Let $p_i = (d_i^{(1)}, s_i^{(1)}, m_i, f_i)$ denote the variables associated with pixel i in the reference frame, where $d_i^{(1)} \in \mathbb{R}^+$ is its disparity, $s_i^{(1)} \in \{0, 1, \ldots\}$ is a semantic label (0 is background, positive integers are foreground object instances), $m_i \in SE(3)$ is its 3D rigid motion (translation and rotation), and $f_i = [u_i, v_i]$ is its optical flow. We denote the disparity and semantic segmentation for each pixel in the second frame by $q_i = (d_i^{(2)}, s_i^{(2)})$. We only use two frames to estimate scene flow, and thus need not explicitly model motion in the second frame.

Existing scene flow algorithms make predictions at the superpixel level without explicitly modeling the semantic content of the scene [21, 33]. Predictions inside each semantic object may thus be noisy or inconsistent. In this work, we assume that the scene contains foreground objects (vehicles, for our autonomous driving application) rigidly moving across a static background. Given an accurate se-

mantic segmentation of some foreground object, the geometry of the pixels within that segment should be spatially and temporally consistent, and the optical flow should be consistent with the underlying 3D rigid motion.

Due to the high dimensionality of the scene flow problem, we refine our estimates using a cascade of discriminative models [12], with parameters learned via a structural SVM [15]. Every stage of the cascade makes a targeted improvement to one scene variable, implicitly accounting for uncertainty in the current estimates of other scene variables. We initialize our semantic segmentation $\mathcal{S}$ using an instance-level segmentation algorithm [6], and our disparities $\mathcal{D}$ and optical flow fields $\mathcal{F}$ using the PRSF method [33]. We discuss their cascaded refinement next.

3.1. Refinement of Semantic Segmentation

The initial single-frame segmentation is unreliable in regions with shadows and reflections. Given stereo inputs, however, our depth estimates provide a strong cue to improve the segmentation. Therefore for each segmentation instance, we define a CRF on the pixels in its enclosing bounding box B_i. We seek to estimate the foreground segmentation s given an initial noisy segmentation $\hat{s}$.

Our data term encourages the inferred segmentation s to be close to the initial segmentation $\hat{s}$. The KITTI scene flow dataset [21] generates "ground truth" segmentations by aligning approximate CAD models, and these annotations are often inaccurate at object boundaries, thus violating that assumption that foreground and background objects typically have distinct color and geometry. To add robustness, we define a feature by computing the signed distance of pixel i to the original segmentation border and using a sigmoid function to map these distances to $[0, 1]$, denoted by $\phi_{\text{dist}}(i, \hat{s})$. The data energy for our CRF model is then

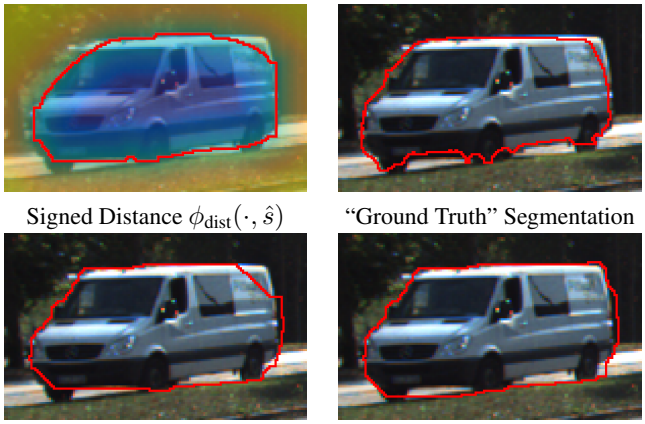
$$E_{\text{seg}}^{\text{data}}(\mathcal{S}) = \sum_{i \in B_i} \Big[\lambda_1 + \lambda_2 \phi_{\text{dist}}(i, \hat{s})\Big] \delta(s_i = 0, \hat{s}_i = 1) + \Big[\lambda_3 + \lambda_4 \phi_{\text{dist}}(i, \hat{s})\Big] \delta(s_i = 1, \hat{s}_i = 0). \tag{1}$$

We demonstrate the benefits of our signed distance feature $\phi_{\text{dist}}(i, \hat{s})$ in Fig. 2. By allowing the CRF to reduce confidence in $\hat{s}$ near boundaries, this feature allows other image-based cues to improve segmentation accuracy.

To allow spatial regularization, we add edges $\mathcal{E}$ to our CRF connecting each pixel to its 8 spatial neighbors:

$$E_{\text{seg}}^{\text{space}}(\mathcal{S}) = \sum_{(i,j) \in \mathcal{E}} \Big[\lambda_5 + \lambda_6 \rho_{\text{img}}(I_i, I_j) + \lambda_7 \rho_{\text{disp}}(d_i, d_j)\Big] \delta(s_i \neq s_j). \tag{2}$$

Here, $\rho_{\text{img}}(I_i, I_j) = \exp\{-\frac{||I_i - I_j||}{\sigma_{\text{img}}}\}$ measures RGB color similarity, and $\rho_{\text{disp}}(d_i, d_j) = \exp\{-\frac{|d_i - d_j|}{\sigma_{\text{disp}}}\}$ measures similarity of the current (approximate) disparity estimates.



Signed Distance $\phi_{\text{dist}}(\cdot, \hat{s})$ | "Ground Truth" Segmentation

CRF Excluding ϕ_{dist} | CRF Including ϕ_{dist}

Figure 2. The "true" KITTI segmentations [21] are approximate (top right). By incorporating a signed distance feature $\phi_{\text{dist}}(i, \hat{s})$ (top left), CRF segmentation accuracy improves (bottom).

To learn the parameters $\lambda = [\lambda_1, \ldots, \lambda_7]$, we use a structured SVM [15] with loss equal to the average label error within bounding box B_i [9]. Feature bandwidths $\sigma_{\text{img}}, \sigma_{\text{disp}}$ are tuned using validation data. To perform inference on $E_{\text{seg}}^{\text{data}} + E_{\text{seg}}^{\text{space}}$, we use an efficient implementation of tree-reweighted belief propagation [34, 16]. Because pixel labels are binary, inference takes less than 0.5s. To apply our CRF model to the scene flow problem, we independently estimate the segmentation for each instance and frame.

3.2. Estimation of Scene Geometry

Given a disparity map $\mathcal{D}$ and camera calibration parameters, a 3D point cloud representation of the scene may be constructed. Standard stereo estimation algorithms ignore semantic cues, and often perform poorly on surfaces that are shadowed, reflective, or transparent. As illustrated in Fig. 3, for autonomous driving applications the depth estimates for vehicle windshields are especially poor. Because inaccurate depth estimates lead to poor motion and flow estimates, we design a model that enforces local smoothness of depths within inferred segmentation masks.

We define a CRF model of the pixels within each semantic segment previously inferred by our cascaded model. For each pixel i in the left camera with disparity hypothesis d_i, we denote its corresponding pixel in the right camera as $P_d(i, d_i)$. The data term is defined to penalize the difference in smooth census transform between pixel i and $P_d(i, d_i)$:

$$E_{\text{geom}}^{\text{data}}(\mathcal{D}) = \sum_{\{i | s_i = s\}} \rho_{\text{CSAD}}(I_i, I'_{P_d(i,d_i)}). \tag{3}$$

Here, $\rho_{\text{CSAD}}(.,.)$ is the CSAD cost [31] for matched pixels in different images. The CSAD difference is a convex approximation of the census transform [37] that gives reliable pixel correspondences for many datasets [31].

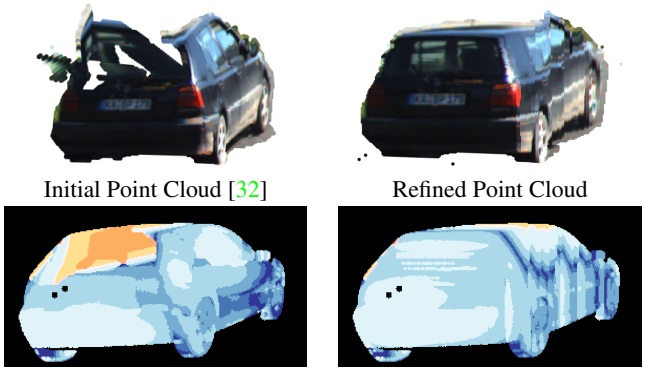


Figure 3. 3D point clouds (top) and corresponding disparity errors (blue small, orange large) for the initial PRSF depth estimates [32], and the refined depth estimates produced by our CRF model.

We encourage piecewise-smooth depth maps by penalizing the absolute difference of neighboring pixel depths:

$$E^{\text{space}}_{\text{geom}}(\mathcal{D}) = \tau_1 \sum_{(i,j)\in\mathcal{E}_s} \rho_{\text{depth}}(d_i, d_j). \quad (4)$$

Here $\mathcal{E}_s$ contains neighboring pixels within segment s, $\rho_{\text{depth}}(d_i, d_j) = |\frac{C}{d_i} - \frac{C}{d_j}|$, and C is a camera-specific constant that transforms disparity d into depth $\frac{C}{d}$. We enforce consistency of pixel depths because the scale of disparities varies widely with the distance of objects from the camera.

If naively applied to the full image, simple CRF models are often inaccurate at object boundaries [32]. However as illustrated in Fig. 3, although our stereo CRF uses standard features, it is effective at resolving uncertainties in challenging regions of foreground objects and it is much better able to capture depth variations within a single object. Moreover, because our pairwise distances depend only on the absolute value of depth differences, distance transforms [8] may be used for efficient inference in minimizing $E^{\text{data}}_{\text{geom}} + E^{\text{space}}_{\text{geom}}$. On average, it takes less than 5s to perform inference in a 200×200 region with 200 disparity candidates. We refine the disparities for each frame independently.

3.3. Estimation of 3D Motion

If the segmentation mask and disparity estimates for each object instance were perfect, we could apply 3D rigid motion to the 3D point cloud for each segment, and project back to the image plane to recover the 2D optical flow. We let (x_i, y_i) denote the *motion flow* constructed in this way. Although our imperfect geometry estimates will cause the motion flow to differ from the true optical flow (u_i, v_i), each still provides valuable cues for the estimation of the other.

For each detected segment, we let $M = (R, t)$ denote its 3D relative motion between the first and second frames. The motion M has 6 degress of freedom: t is a translation vector, and $R = (\alpha, \beta, \gamma)$ is a rotation represented by three

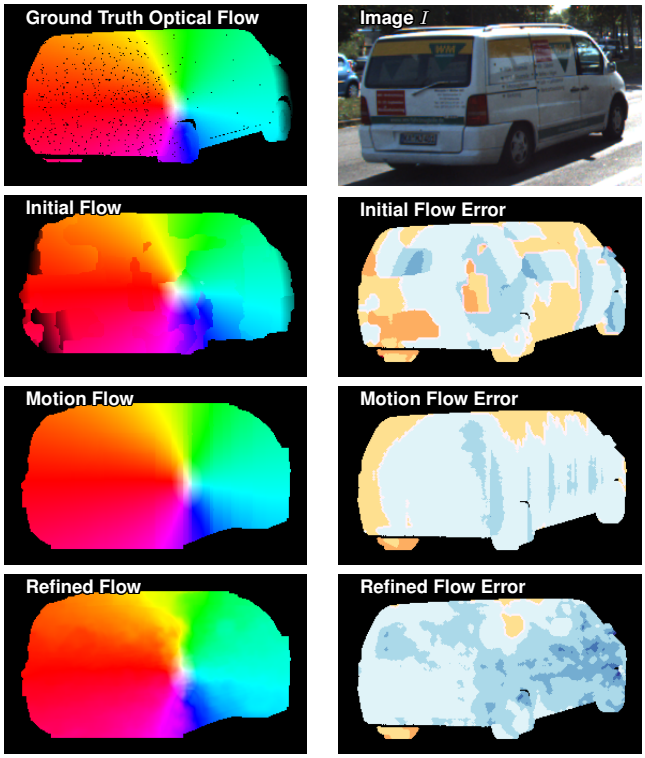


Figure 4. Visualization of estimated flow fields (left, hue encodes orientation [28]) and their error (right, blue small, orange large). A rigid 3D motion flow captures the dominant object motion, and the refined estimates from our CRF model further improve accuracy.

axis-aligned rotation angles. We match the rigid motion M to the current flow field estimate (u, v) by minimizing the following energy function:

$$E_{\text{motion}}(M) = \nu(\rho(\alpha) + \rho(\beta) + \rho(\gamma)) + \sum_{\{i|s_i=s\}} |x_i(M, d_i) - u_i| + |y_i(M, d_i) - v_i|, \quad (5)$$

where $(x_i(M, d_i), y_i(M, d_i))$ is the motion flow computed from disparity d_i, 3D motion M, and the camera calibration. We let $\rho(a) = \sqrt{a^2 + \epsilon^2}$ be the Charbonnier penalty, a smooth function similar to the L_1 penalty that provides effective regularization for motion estimation tasks [28]. We regularize R to avoid unrealistically large rotation estimates. We set the regularization constant ν using validation data, and use gradient descent to find the optimal value for M. We visualize an example motion flow map in Fig. 4.

3.4. Estimation of 2D Optical Flow

The estimated motion flow from the previous stage provides valuable cues for optical flow estimation. As in the example in Fig. 4, motion flow errors are primarily caused by imperfect geometries (or disparities). We thus seek a flow field $f_i = (u_i, v_i)$ such that the corresponding pixel $P_f(i, f_i)$ in the next frame matches pixel i, and f_i does not

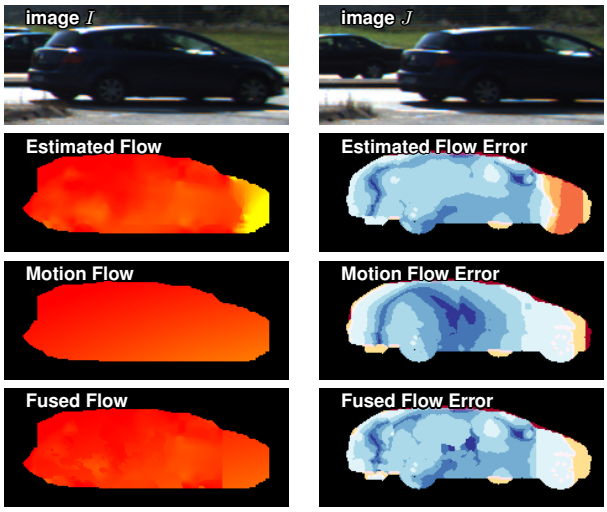


Figure 5. Visualization of our flow fusion CRF to reduce motion errors (blue small, orange large) for out-of-border pixels.

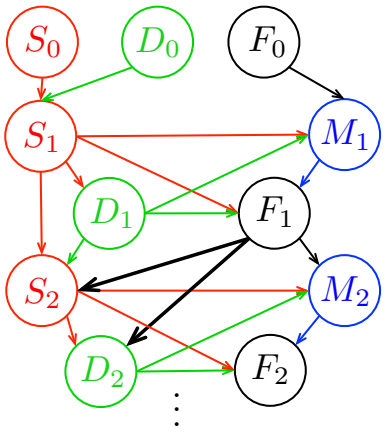


Figure 6. A directed graph summarizing our cascaded approach to the estimation of object segmentations S, disparities D, 3D rigid motions M, and optical flow F. Subscripts indicate different stages of the cascade. The **bold arrows** represent additional temporal dependencies added for stages two and later.

deviate too much from (x_i, y_i). We define a CRF model of the pixels within segment s in frame 1, with likelihood

$$E^{\text{data}}_{\text{flow}}(\mathcal{F}) = \sum_{\{i|s_i=s\}} \rho_{\text{CSAD}}(I_i, J_{P_f(i,f_i)}) + \eta_1(|u_i - x_i| + |v_i - y_i|). \quad (6)$$

We also encourage spatially smooth flow field estimates:

$$E^{\text{space}}_{\text{flow}}(\mathcal{F}) = \sum_{(i,j)\in\mathcal{E}_s} \eta_2(|u_i - u_j| + |v_i - v_j|). \quad (7)$$

While many optical flow methods use superpixel approximations to make inference more efficient [27], max-product belief propagation can be efficiently implemented for our pixel-level CRF using distance transforms [8, 5]. As shown in Fig. 4, our refined optical flow improves the initial flow by smoothly varying across the segment, while simultaneously capturing details that are missed by the motion flow.

To limit the memory consumption of our optical flow algorithm, we perform inference on a down-sampled image and then use the EpicFlow [25] algorithm to interpolate back to the full image resolution. Other recent optical flow algorithms have used a similar approximation [5, 1].

Motion Estimation for Out-of-Frame Pixels We notice that the EpicFlow interpolation tends to produce significant errors for pixels that move outside of the image border. Outside of the camera's field of view, optical flow can only be predicted using the known 3D rigid motion, and we thus propose a flow fusion CRF [18] to combine the estimated optical flow and motion flow for partially occluded objects.

In particular, we use a binary CRF to determine whether the optical flow (u_i, v_i) or motion flow (x_i, y_i) provides a better estimate of the true flow (U_i, V_i) for each pixel i. Intuitively, for within-border pixels we should use the matching cost to compare flow fields, while out-of-border pixels should be biased towards the motion flow interpolation:

$$E^{\text{data}}_{\text{fuse}}(\mathcal{F}) = \omega_1(|U_i - x_i| + |V_i - y_i|)\delta[P_f(i, f_i) \text{ outside}] + \sum_{f=\{(u,v),(x,y)\}} \sum_{\{i|s_i=s\}} \rho_{\text{CSAD}}(I_i, J_{P_f(i,f_i)})\delta[P_f(i, f_i) \text{ inside}].$$

Spatial smoothness is encouraged for neighboring pixels:

$$E^{\text{space}}_{\text{fuse}}(\mathcal{F}) = \sum_{(i,j)\in\mathcal{E}} \omega_2(|U_i - U_j| + |V_i - V_j|). \quad (8)$$

We tune parameters ω_1, ω_2 using validation data, and minimize the energy using tree-reweighted belief propagation [16]. We show in Fig. 5 that the fused flow estimate retains many details of the optical flow, while using the motion flow to better interpolate in occluded regions. We also apply our flow fusion technique to update the noisy background flow predictions. See Fig. 8 for additional examples of our final optical flow estimates.

4. Cascaded Scene Flow Prediction

The CRF models defined in Sec. 3 refine the various components of our scene model greedily, by estimating each one given the current best estimates for all others. However, this approach does not fully utilize the temporal relationships between the segmentation and geometry at different frames. Also, when the initial optical flow contains major errors, our motion flow estimates will be inaccurate. To better capture the full set of geometric and temporal relationships, we thus use multiple stages of cascaded prediction [12] to further refine our scene flow estimates. The inputs and outputs for each stage of our cascade are summarized by the directed graph in Fig. 6.

Temporal Segmentation Consistency Rather than segmenting each video frame independently, in the second

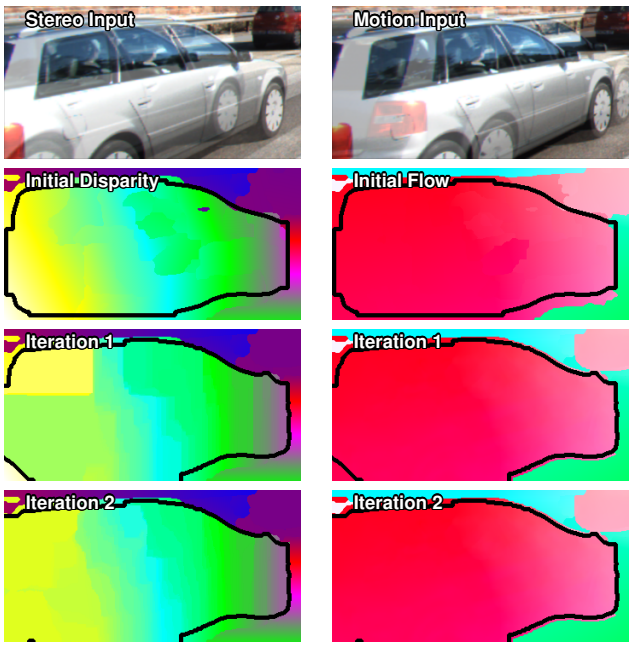


Figure 7. From top to bottom, we visualize input frames, initial disparity (left) and flow (right) predictions, and the refined disparity and flow after the first and second stages of the cascade. Our refined flow estimates from stage 1 (note object boundaries) lead to improved stereo estimates in stage 2 (upper left).

stage of our cascade, we use the inferred flow field f to encourage temporal consistency. Each pixel i in frame 1 is linked to matched pixel $P_f(i, f_i)$ in frame 2:

$$E^{\text{time}}_{\text{seg}}(\mathcal{S}) = \lambda_8 \delta(s^{(1)}_i = 0, s^{(2)}_{P_f(i,f_i)} = 1) + \lambda_9 \delta(s^{(1)}_i = 1, s^{(2)}_{P_f(i,f_i)} = 0) + \sum_i \left[\lambda_{10} + \lambda_{11} \rho_{\text{CSAD}}(I_i, J_{P_f(i,f_i)})\right] \delta(s^{(1)}_i = s^{(2)}_{P_f(i,f_i)}).$$

We again use S-SVM learning of CRF parameters λ on $E^{\text{data}}_{\text{seg}} + E^{\text{space}}_{\text{seg}} + E^{\text{time}}_{\text{seg}}$, and infer segmentations using tree-reweighted belief propagation.

Temporal Geometric Consistency As in our temporal segmentation model, we also extend the stereo CRF of Sec. 3.2 to encourage smooth changes for the depths of pixels linked by our optical flow estimates:

$$E^{\text{time}}_{\text{geom}}(\mathcal{D}) = \tau_2 \sum_{\{i | s^{(1)}_i = s\}} \rho_{\text{depth}}(d_i(m_i), d_{P_f(i,f_i)}). \quad (9)$$

Here, $d_i(m_i)$ denotes the disparity value of pixel i in the second frame when rigid motion m_i is applied. The parameters τ are learned using validation data, and efficient distance transformation [8] is also used to solve $E^{\text{data}}_{\text{geom}} + E^{\text{space}}_{\text{geom}} + E^{\text{time}}_{\text{geom}}$. Fig. 7 shows an example of the improved disparity and flow estimates produced across multiple stages of our cascade.

Recovery from a Poor Optical Flow Initialization If the initial noisy optical flow is very inaccurate, our cascade cannot recover the correct 3D motions of objects because we assume motion flow should match optical flow. Since our updated semantic segmentation masks $s^{(1)}$ and $s^{(2)}$ are typically very accurate, when applying rigid motion M to pixels in $s^{(1)}$, the shape of the new segmentation mask $s(M)$ should be similar to $s^{(2)}$. We measure this similarity via a cost defined on the second-frame bounding box B:

$$\frac{1}{|B|} \sum_{i \in B} \alpha S(M)_i \cdot C(S^{(2)}_i) + (1-\alpha) C(S(M)_i) \cdot S^{(2)}_i. \quad (10)$$

Here, $C(\cdot)$ is the Chamfer difference map and $\alpha = 0.5$. This cost function is widely used for human pose estimation [3]. If this cost exceeds 0.5, we replace the first term in Eq. (5) with this silhouette cost. By optimizing this modified objective in Eq. (10) using standard gradient-descent, we can recover from bad motion estimates. An illustration is in the supplementary material.

Second Frame Disparities For the KITTI scene flow dataset [21], the ground truth disparity for the second frame is represented as per-pixel disparity changes with respect to the first frame. To predict this quantity for evaluation, we apply our estimated 3D rigid motion for each pixel to its estimated geometry in the first frame. The accuracy of these disparity estimates is thus strongly dependent on the performance of our motion estimation algorithm.

Global Energy Function The global energy function implicitly minimized by our cascade of CRFs can be constructed by adding all energy terms together. Our iterative optimization of subsets of variables (as in Fig. 6) can be seen as block coordinate descent, where the cascaded prediction framework refines the energy function to reflect the typical accuracy of previous stages. This cascaded framework enables efficient, adaptive discretization of a large state space for flow and disparity, and is a principled way of optimizing a limited number of inference iterations [12].

5. Experiments

We test our semantic scene flow algorithm (**SSF**) with 3 iterations of cascaded prediction on the challenging KITTI 2015 benchmark [21]. We evaluate the performance of our disparity estimates for two frames (**D1**, **D2**), flow estimates (**Fl**) for the reference frame, and scene flow estimates (**SF**) for foreground pixels (**fg**), background pixels (**bg**), and all pixels (**all**). See Table 1 for experimental results on all pixels, and Table 2 for non-occluded pixels. We evaluate SSF cascades learned to refine PRSF [32] initializations (**SSF-P**) and also apply the learned parameters to OSF [21] initializations (**SSF-O**). Our cascaded approach is superior to the published two-frame scene flow algorithms with respect to all evaluation metrics. SSF-P is about 60% more accurate than the two-frame PRSF method; SSF-P is overall 2%

	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all	Time
PRSF	4.74	13.74	6.24	11.14	20.47	12.69	11.73	27.73	14.39	13.49	31.22	16.44	2.5min
CSF	4.57	13.04	5.98	7.92	20.76	10.06	10.40	30.33	13.71	12.21	33.21	15.71	1.3min
OSF	4.54	12.03	5.79	5.45	19.41	7.77	5.62	22.17	8.37	7.01	26.34	10.23	50min
SSF-O	4.30	8.72	5.03	5.13	15.27	6.82	5.42	17.24	7.39	6.95	25.78	10.08	52.5min
SSF-P	3.55	8.75	4.42	4.94	17.48	7.02	5.63	14.71	7.14	7.18	24.58	10.07	5min
ISF	4.12	6.17	4.46	4.88	11.34	5.95	5.40	10.29	6.22	6.58	15.63	8.08	10min
OSFTC	4.11	9.64	5.03	5.18	15.12	6.84	5.76	16.61	7.57	7.08	22.55	9.65	50min
PRSM	3.02	10.52	4.27	5.13	15.11	6.79	5.33	17.02	7.28	6.61	23.60	9.44	10min

Table 1. Scene flow results on all pixels for the KITTI test set. Under most evaluation metrics, our SSF algorithm outperforms baseline two-frame scene flow methods such as PRSF [32], CSF [19], and OSF [21]. OSFTC and PRSM take multiple frames as input. ISF [4] is concurrent work (to appear at ICCV 2017) that benefits from additional training data and corrected KITTI training labels.

	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all
CSF	4.03	11.82	5.32	6.39	16.75	8.25	8.72	26.98	12.03	10.26	28.68	13.56
PRSF	4.41	13.09	5.84	6.35	16.12	8.10	6.94	23.64	9.97	8.35	26.08	11.53
OSF	4.14	11.12	5.29	4.49	16.33	6.61	4.21	18.65	6.83	5.52	22.31	8.52
SSF-O	3.98	7.82	4.62	4.26	12.31	5.70	4.04	13.18	5.70	5.44	21.11	8.25
SSF-P	3.30	7.74	4.03	4.12	14.57	5.99	4.20	10.81	5.40	5.70	19.93	8.25
ISF	3.74	5.46	4.02	4.06	9.04	4.95	4.21	6.83	4.69	5.31	11.65	6.45
PRSM	2.93	10.00	4.10	4.13	12.85	5.69	4.33	14.15	6.11	5.54	20.16	8.16
OSFTC	3.79	8.66	4.59	4.18	12.06	5.59	4.34	12.86	5.89	5.52	18.02	7.76

Table 2. Scene flow results on non-occluded pixels for the KITTI test set. SSF also outperforms published methods with two-frame inputs.

more accurate than OSF, while 10 times faster. At the time of submission, the only published work that performed better than our SSF approach were the multi-frame PRSM [33] and OSFTC [23] methods, which require additional data. The concurrently developed ISF method [4] uses external training data for instance segmentation and disparity estimation, leading to further improvements over our approach at the cost of slower speed.

We visualize the qualitative performance of our **SSF-P** method on training data in Fig. 8. In Table 3, we evaluate the performance gain provided by each stage of the cascade on the training set. There is an improvement at the first stage of the cascade when modeling segmentation and geometry independently at each frame, followed by another improvement at the second stage when temporal consistency is introduced. At the third stage, performance starts to saturate.

Speed Scene flow estimation is a computationally demanding task, and efficient algorithms [19] usually sacrifice accuracy for speed. Although the number of variables in our scene flow representation is huge and we make pixel-level predictions, our cascaded algorithm with MATLAB/C++ implementation on a single-core 2.5 Ghz CPU remains efficient. The main reason is that we disentangle the output space, and utilize efficient message-passing algorithms [8, 5] to solve each high-dimensional inference problem. Most of the computation time is spent on feature evaluation, and could be accelerated using parallelization.

Failure Cases As shown in Fig. 8, in challenging cases where the semantic segmentation algorithm fails to detect vehicle boundaries, our scene flow estimates can be inaccurate. As previously studied for semantic optical flow methods [1], we conducted an experiment using ground truth segmentation masks and witnessed a significant performance gain; see Table 3. Our framework for cascaded scene flow estimation will immediately benefit from future advances in semantic instance segmentation.

	Seg	D1-fg	D1-bg	Fl-fg	Fl-bg
PRSF	10.1	5.00	3.27	8.54	4.04
Iter 1	9.79	3.69	3.18	8.38	3.67
Iter 2	8.41	3.50	3.15	8.20	3.65
Iter 3	8.40	3.49	3.15	8.20	3.65
GT Seg	0	2.19	3.05	7.61	3.56

Table 3. Results of SSF-P on the KITTI validation set. Each stage of the cascade makes further improvements to the noisy PRSF initialization. In the last row, we show that when given a perfect segmentation mask, predictions are improved by a large margin.

6. Conclusion

In this paper, we utilize semantic cues to identify rigidly moving objects, and thereby produce more accurate scene flow estimates for real-world scenes. Our cascaded prediction framework allows computationally efficient recovery of high-dimensional motion and geometry estimates, and can flexibly utilize cues from sophisticated semantic segmentation algorithms. We improve on the state-of-the-art for the challenging KITTI scene flow benchmark [21, 11]. While our experiments have focused on using vehicle detections to

Figure 8. Visualization of our semantic scene flow estimates for four sequences from the KITTI training set, using a cascade initialized with noisy PRSF estimates. The initial or updated segmentation mask (black lines) is overlaid on the disparity and optical flow estimates. The last sequence is a failure case where an imperfect segmentation leads to large disparity and flow errors.

improve scene flow estimates for autonomous driving, our cascaded scene flow framework is directly applicable to any category of objects with near-rigid motion.

Acknowledgements This research supported in part by ONR Award Number N00014-17-1-2094. Some work was completed by Zhile Ren during an internship at NVIDIA.

References

[1] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *ECCV*, pages 154–170. Springer, 2016. 1, 2, 5, 7

[2] 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, March 2011. 1

[3] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *CVPR*, pages 1–8. IEEE, 2007. 6

[4] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3D scene flow estimation in autonomous driving scenarios? In *ICCV*, 2017. 2, 7

[5] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. *CVPR*, 2016. 5, 7

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1, 2, 3

[7] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006. 4, 5, 6, 7

[9] T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *ICML*. ACM, 2008. 3

[10] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. In *Image and Geometry Processing for 3-D Cinematography*. Springer, 2010. 1

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 1, 2, 7

[12] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, pages 641–648, 2009. 1, 2, 3, 5, 6

[13] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 1

[14] J. Hur and S. Roth. Joint optical flow and temporally consistent semantic segmentation. In *ECCV*. Springer, 2016. 1, 2

[15] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 1, 3

[16] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10), 2006. 3, 5

[17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Morgan Kaufmann, 2001. 1

[18] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008. 5

[19] Z. Lv, C. Beall, P. F. Alcantarilla, F. Li, Z. Kira, and F. Dellaert. A continuous optimization approach for efficient and accurate scene flow. In *ECCV*. Springer, 2016. 1, 7

[20] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2

[21] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, pages 3061–3070, 2015. 1, 2, 3, 6, 7

[22] T. Müller, J. Rannacher, C. Rabe, and U. Franke. Feature- and depth-supported modified total variation optical flow for 3d motion field estimation in real scenes. In *CVPR*, 2011. 1

[23] M. Neoral and J. ochman. Object scene flow with temporal consistency. In *22nd Computer Vision Winter Workshop (CVWW)*. Pattern Recongition and Image Processing Group, TU Wien & PRIP Club, Vienna, Austria, Feb. 2017. ISBN: 978-3-200-04969-7. 7

[24] E. L.-M. Pia Bideau. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, 2016. 1, 2

[25] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR*, 2015. 5

[26] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002. 1

[27] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, 2016. 1, 2, 5

[28] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*. IEEE, 2010. 1, 4

[29] T. Taniai, S. N. Sinha, and Y. Sato. Fast multi-frame stereo scene flow with motion segmentation. In *CVPR*, 2017. 2

[30] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV*, 1999. 1

[31] C. Vogel, S. Roth, and K. Schindler. An evaluation of data costs for optical flow. In *GCPR*. Springer, 2013. 3

[32] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *ICCV*, pages 1377–1384, 2013. 1, 2, 4, 6, 7

[33] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *IJCV*, 115(1), 2015. 1, 2, 3, 7

[34] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE transactions on information theory*, 51(11):3697–3717, 2005. 3

[35] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *ECCV*, pages 739–751. Springer, 2008. 1

[36] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, pages 756–771. Springer, 2014. 1

[37] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994. 3

[38] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, pages 669–677, 2016. 1, 2