Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation

Koichiro Yamaguchi[1], David McAllester[2] and Raquel Urtasun[3]

[1]Toyota Central R&D Labs., Inc.
[2]Toyota Technological Institute at Chicago
[3] University of Toronto

Abstract. In this paper we propose a slanted plane model for jointly recovering an image segmentation, a dense depth estimate as well as boundary labels (such as occlusion boundaries) from a static scene given two frames of a stereo pair captured from a moving vehicle. Towards this goal we propose a new optimization algorithm for our SLIC-like objective which preserves connecteness of image segments and exploits shape regularization in the form of boundary length. We demonstrate the performance of our approach in the challenging stereo and flow KITTI benchmarks and show superior results to the state-of-the-art. Importantly, these results can be achieved an order of magnitude faster than competing approaches.

1 Introduction

Most autonomous vehicles rely on active sensing (e.g., lidar) to construct point cloud representations of the environment. However, passive computer vision holds out the potential to provide richer geometric representations at lower cost. In this paper we are interested in the problem of recovering image segmentations, dense depth, and segment boundary labels from stereo video — a sequence of stereo image pairs taken over time from a moving vehicle. This is an important estimation problem as it is a fundamental step to perform navigation and recognition tasks such as path planning, obstacle avoidance, semantic segmentation and object detection.

Current leading techniques are slanted plane methods, which assume that the 3D scene is piece-wise planar and the motion is rigid or piece-wise rigid [30, 31, 26]. Unfortunately, these slanted plane methods have involved time-consuming optimization algorithms (several minutes per frame) such as particle belief propagation [30, 31] or algorithms based on plane proposals with fusion moves and iterated cut-based segmentations [26]. This makes to date slanted plane methods non-practical for robotics applications such as autonomous driving.

To address this issue, in this paper we propose a fast and accurate slanted plane algorithm that operates on three images — a stereo pair and an image from the left stereo camera at a later point in time. Our approach exploit the fact that in autonomous driving scenarios most of the scene is static and utilizes the stereo and video pairs to produce a joint estimate of depth, an image segmentation as

well as boundary labels in the reference image. Importantly, it does so at least an order of magnitude faster than existing slanted plane methods [30, 31, 26], while outperforming the state-of-the-art on the challenging KITTI benchmark [9].

Following [30, 31], our algorithm first uses semi global block matching (SGM) [13] to construct a semi-dense depth map on the reference image. A contribution here is the development of an SGM algorithm based on the joint evidence of the stereo and video pairs. The semi-dense SGM depth map is then used as input to our slanted plane method for inferring the segmentation, planes and boundary labels.

Our new inference algorithm is a form of block-coordinate descent on a total energy involving the segmentation, the planes assigned to the segments, an "outlier-flag" at each pixel, and a line label assigned to each pair of neighboring segments giving the occlusion-status of the boundary between those segments. In particular, each slanted plane can be optimized by a closed-form least-squares fit holding the segmentation, outlier-flags, and line-labels fixed. The line labels can be optimized holding the segments, planes and outlier flags fixed. The segmentation and the outlier flags are optimized jointly. The segmentation objective is an extension of the SLIC energy to handle both color and depth as well as a shape prior regularizing the length of the boundary. Importantly, our segmentation optimization subroutine uses unit-time single pixel moves restricted to the boundaries of segments, preserving the invariant that each segment is simply connected (connected and without holes).

Our block-coordinate descent algorithm is guaranteed to converge as the optimization over each set of variables (including the segmentation) is guaranteed to reduce the total energy. Importantly, this objective can be optimized over all unknowns on a single core in as little as 3s, while achieving state-of-the-art results. As a byproduct, when ignoring the depth energy term, our topology preserving segmentation subroutine can be used to create superpixels from single images.

2 Related Work

Recovering depth from a stereo and a video pair with a common reference image is a special case of the more general structure from motion problem, where scene geometry is recovered from multiple images taken from different camera angles. There is a very large literature on structure from motion, for example see [23, 7, 10, 21]. Here, we are interested in a particular three-image setting. The three image case has been studied from the perspective of the tri-focal tensor — a generalization of the fundamental matrix to three images [11]. In our setting we are given the calibration between the two images of the stereo pair and for this reason we chose to work with the single fundamental matrix defined by the ego-motion underlying the video pair.

Although we assume a static scene, it is useful to review work on scenes with moving objects such as pedestrians and cars. The widely cited Tomasi-

Kanade matrix factorization method for structure from motion [23] has been generalized to the case of scenes containing moving objects [6]. This algorithm groups points (correspondences) into rigid objects and assigns both a position in space and a six dimensional motion to each rigid object. However, it assumes that correspondences are given and the cameras are projective.

The term "scene flow" was introduced in [24] for the problem of assigning both positions and motions to a dense set of points on the surface of objects in the scene. While an object has a six dimentional motion, a point does not rotate and thus only three degrees of freedom are necessary (a flow). Several papers have tackled the problem of estimating the 3D flow-field [28, 17, 14, 2]. To date, good performance has not yet been shown in challenging real-world scenarios.

Vogel et. al. [26] handles scenes with moving objects using a segmentation of a reference image with both a planar surface and a six dimensional rigid motion associated with each image segment. They incorporate the rigid-scene assumption using a soft bias, while it is a hard constraint in our approach. Both systems do inference by minimizing an energy defined on planes associated with segments, however, our method is an order of magnitude faster and achieves greater accuracy on the KITTI benchmark for both stereo and flow.

Our approach is also related to the stereo and motion-stereo algorithms of Yamaguchi et. al. [30, 31]. As in [31], our approach first computes a semi-dense SGM depth map which then undergoes slanted-plane smoothing. The difference is that our SGM depth map is derived by joint inference from a stereo and a video pair and that our slanted-plane algorithm is roughly three orders of magnitude faster. Our system spends 25 seconds computing SGM fields and as little as 10 seconds on the slanted plane smoothing. Furthermore, the smoothing time can be reduced to 3 seconds with very little loss of accuracy. Slanted plane models for stereo have a long history going back at least to [3]. They have proved quite successful on the Middleburry [20, 15, 4, 27] and KITTI [30] stereo benchmarks.

The topology-preserving segmentation algorithm proposed here is related to SLIC superpixels [1]. However, our segmentation algorithm preserves the invariant that segments remain simply connected. This eliminates the need for the post-processing step in the SLIC algorithm to simplify segments. This is important as this post-processing step can result in large increases of the total energy. Furthermore, this speeds-up inference, as only boundary pixels are considered at each iteration. Our segmentation method also incorporates a length of boundary energy for shape regularization, as well as the evidence from the stereo and video pairs, which SLIC does not.

3 SGM for Joint Stereo and Flow

Our approach first estimates a semi-dense depth map on the reference image $\mathcal{I}_{L,t}$ using a variant of SGM [13] which integrates evidence from both a stereo pair $\{\mathcal{I}_{L,t}, \mathcal{I}_{R,t}\}$ and a video pair $\{\mathcal{I}_{L,t}, \mathcal{I}_{L,t+1}\}$. We then smooth these results to create a dense field using a slanted plane method, which we explain in the next section. An overview of our approach is shown in Fig. 1.

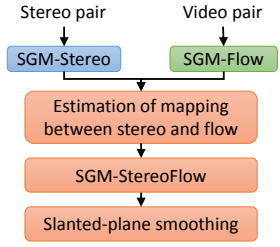

Fig. 1. Processing flow of our approach

Following [31], we first use semi-global matching (SGM) [13] to independently compute a semi-dense disparity field from the stereo pair — SGM-stereo — and a semi-dense epipolar flow field from the motion pair — SGM-flow. These two fields are then used to estimate a scaling relationship between stereo and flow. More specifically, let b be the distance between the stereo cameras (the stereo baseline), let f be the focal length of the cameras, and let $Z_{\mathbf{p}}$ be the Z coordinate of the point in the scene imaged at pixel p in the coordinate system defined by the reference image $\mathcal{I}_{L,t}$. The stereo disparity field, which is estimated by SGM-stereo, is defined by the following equation

$$d_{\mathbf{p}} = \frac{b}{Z_{\mathbf{p}}} f \tag{1}$$

Let v_z be the distance that the left camera moved in the Z direction (as defined by the reference image) from time t to $t+1$. The SGM-flow field [31] is an estimate of the following "V over Z" field, also called VZ-ratio

$$\omega_{\mathbf{p}} = \frac{v_z}{Z_{\mathbf{p}}} \tag{2}$$

When the scene is static, we get a constant (across pixels) relationship between these two fields $\alpha = \omega_p / d_p = v_z/(bf)$. However, due to errors in calibration and registration, we formulate α as a linear function of the image coordinates

$$\omega_{\mathbf{p}} = \alpha(\mathbf{p}) d_{\mathbf{p}} = (\alpha_x p_x + \alpha_y p_y + \alpha_c) d_{\mathbf{p}} \tag{3}$$

In practice, we robustly estimate $\alpha = (\alpha_x, \alpha_y, \alpha_c)$ using RANSAC from the set of pixels from which we have both an estimate of flow and stereo.

Given an estimate of $(\alpha_x,\ \alpha_y,\ \alpha_c)$, we formulate an SGM algorithm to jointly estimate stereo and flow by making use of Eq. (3). For the SGM algorithm we define the energy of the system to be the sum of a data energy $C_{\mathbf{sf}}$ and a smoothness energy $S_{\mathbf{sf}}$

$$E_{\mathrm{sf}}(\mathbf{d}) = \sum_{\mathbf{p}} C_{\mathrm{sf}}(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}} S_{\mathrm{sf}}(d_{\mathbf{p}}, d_{\mathbf{q}}) \tag{4}$$

where $\mathbf{d}$ is a field assigning a disparity to each reference pixel.

We say that a pixel is occluded in the flow (stereo) field, if the SGM-flow (SGM-stereo) does not return an estimate for that pixel. We define the unary cost of a depth at a pixel to be the average of the costs of the flow and stereo matchings. When the pixel is flagged as an outlier by a field, the cost function is simply computed using only the other's field evidence. In particular, we employ the Census transform and gradient information to compute the cost function of the stereo pair as follows

$$C_{\text{st}}(\mathbf{p}, d_{\mathbf{p}}) = \sum_{\mathbf{q}\in\mathcal{W}(\mathbf{p})} \{|\mathcal{G}_{L,t}(\mathbf{q}, \mathbf{h}(\mathbf{q})) - \mathcal{G}_{R,t}(\mathbf{q'}_{\text{st}}(\mathbf{q}, d_{\mathbf{q}}), \mathbf{h}(\mathbf{q}))| + \lambda_{\text{cen}} H(\mathcal{T}_{L,t}(\mathbf{q}), \mathcal{T}_{R,t}(\mathbf{q'}_{\text{st}}(\mathbf{q}, d_{\mathbf{q}}))\} \quad (5)$$

where $\mathcal{G}(\cdot,\cdot)$ is the directional derivative in the image and $\mathbf{h}(\mathbf{p})$ is the epipolar line passing through pixel $\mathbf{p}$. $\mathcal{T}(\cdot)$ is the Census transform and $H(\cdot,\cdot)$ is the Hamming distance between two binary descriptors, with λ_{cen} a constant parameter, and $\mathbf{q'}_{\text{st}}(\mathbf{q}, d_{\mathbf{q}})$ the corresponding pixel in the right image whose disparity is $d_{\mathbf{q}}$, that is $\mathbf{q'}_{\text{st}}(\mathbf{q}, d_{\mathbf{q}}) = (q_x - d_{\mathbf{q}}, q_y)$. In a similar manner, we define the cost function of the motion pair

$$C_{\text{fl}}(\mathbf{p}, d_{\mathbf{p}}) = \sum_{\mathbf{q}\in\mathcal{W}(\mathbf{p})} \{|\mathcal{G}_{L,t}(\mathbf{q}, \mathbf{e'}(\mathbf{q})) - \mathcal{G}_{L,t+1}(\mathbf{q'}_{\mathbf{fl}}(\mathbf{q}, d_{\mathbf{q}}), \mathbf{e'}(\mathbf{q}))| + \lambda_{\text{cen}} H(\mathcal{T}_{L,t}(\mathbf{q}), \mathcal{T}_{L,t+1}(\mathbf{q'}_{\mathbf{fl}}(\mathbf{q}, d_{\mathbf{q}}))\} \quad (6)$$

where $\mathbf{e'}(\mathbf{q})$ is the epipolar line of pixel $\mathbf{q}$ and $\mathbf{q'}_{\mathbf{fl}}(\mathbf{q}, d_{\mathbf{q}})$ is the corresponding pixel in the left image at time $t+1$ whose VZ-ratio is $\omega_{\mathbf{q}} = \alpha(\mathbf{q}) d_{\mathbf{q}}$.

The smoothness term $S(d_{\mathbf{p}}, d_{\mathbf{q}})$ is defined to be 0, if $d_{\mathbf{p}} = d_{\mathbf{q}}$, and two different penalties ($0 \geq \lambda_{s1} \geq \lambda_{s2}$) depending whether they are 1 or more integers apart. This scheme permits adapting to slanted or curved surfaces.

The motion and stereo fields can then be estimated jointly by solving for the disparities $\{d_{\mathbf{p}}\}$ by minimizing the energy in Eq. (4). While this global minimization is NP hard, we adopt the strategy of [13] and aggregate the matching cost in 1D from all directions equally

$$L(\mathbf{p}, d_{\mathbf{p}}) = \sum_j L_j(\mathbf{p}, d_{\mathbf{p}})$$

with L_j the cost of direction j. This can be done efficiently by employing dynamic programming and recursively computing

$$L_j(\mathbf{p}, d_{\mathbf{p}}) = C(\mathbf{p}, d_{\mathbf{p}}) + \min_i \{L_j(\mathbf{p} - \mathbf{j}, i) + S_{sf}(d_p, i)\}$$

After minimizing Eq.(4) with respect to $\mathbf{d}$, we refine the disparity map by subpixel estimation and removing spurious regions. We called this algorithm *SGM-StereoFlow*. While effective, SGM-StereoFlow provides only semi-dense estimations of both fields. Furthermore, it employs very local regularization, which exploits only the relationships between neighboring pixels. In the next section we derive an efficient and effective slanted plane method which estimates dense flow and stereo fields while reasoning about segmentation, occlusion and outliers.

4 Slanted Plane Smoothing

The slanted-plane smoothing constructs an image segmentation, a slanted plane foreach segment, an outlier flag for each pixel, and a line label for each pair of neighboring segments. This is done by performing a form of block-coordinate descent on a joint energy involving all these latent structures. In particular, our algorithm is very efficient and it only updates the necessary components in an online fashion when possible.

4.1 Energy Definition

We denote our overall energy as $E(s, \theta, f, o, \mathcal{I}, d)$ where s is a segmentation, θ assigns a plane to each segment, f assigns an "outlier flag" to each pixel, o assigns a line label to each pair of neighboring segments, $\mathcal{I}$ is the reference image (for defining mean segment colors), and d is the semi-dense depth field being smoothed. Let s_p be the index of the segment that segmentation s assigns to pixel p, and let μ_i and c_i be the mean position and color respectively of segment i. In our implementation to be computationally efficient, the mean positions and colors are maintained incrementally as pixels shift between segments. Let $\theta_i = (A_i, B_i, C_i)$ be the disparity plane that θ assigns to segment i. At each pixel the disparity can be computed as

$$\hat{d}(\mathbf{p}, \theta_i) = A_i p_x + B_i p_y + C_i, \tag{7}$$

where (p_x, p_y) are the coordinates of pixel $\mathbf{p}$. We use the disparity estimate $\hat{d}(p, \theta_{s_p})$ at pixel p, where the plane is indexed by the variable s_p. In the following we will use the terms superpixel and segment interchangeably. Further, let $f_p \in \{0, 1\}$ be the outlier flag of pixel $\mathbf{p}$.

We define the energy of the system to be the sum of energies encoding appearance, location, disparity, smoothness and boundary energies as follows

$$\begin{aligned} E(s, \theta, f, o, \mathcal{I}, d) = & \underbrace{\sum_{\mathbf{p}} E_{\text{col}}(\mathbf{p}, c_{s_p})}_{color-data} + \underbrace{\lambda_{\text{pos}} \sum_{\mathbf{p}} E_{\text{pos}}(\mathbf{p}, \mu_{s_p})}_{location} + \underbrace{\lambda_{\text{depth}} \sum_{\mathbf{p}} E_{\text{depth}}(\mathbf{p}, \theta_{s_p}, f_p)}_{depth-data} \\ & + \underbrace{\lambda_{\text{smo}} \sum_{\{i,j\} \in \mathcal{N}_{\text{seg}}} E_{\text{smo}}(\theta_i, \theta_j, o_{i,j})}_{plane-smoothness} + \underbrace{\lambda_{com} \sum_{\{i,j\} \in \mathcal{N}_{\text{seg}}} E_{\text{prior}}(o_{i,j})}_{label-prior} \\ & + \underbrace{\lambda_{\text{bou}} \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}_8} E_{\text{bou}}(s_p, s_q)}_{boundary-length} \end{aligned} \tag{8}$$

where we have left the dependence on $\mathcal{I}$ and d implicit and where $\mathcal{N}_8$ is the set of pairs of 8-neighbor pixels. We now define the energy components in more detail.

Location: We define an energy term that prefers well-shaped segments

$$E_{\text{pos}}(\mathbf{p}, \mu_{s_p}) = ||\mathbf{p} - \mu_{s_p}||_2^2, \tag{9}$$

Appearance: This term simply encourages pixels to be in a superpixel if they agree on their color

$$E_{\text{col}}(\mathbf{p}, c_{s_p}) = ||\mathcal{I}_{L,t}(\mathbf{p}) - c_{s_p}||_2^2 \tag{10}$$

Disparity: This term encourages the plane estimates to agree with the image evidence (i.e., SGM-StereoFlow estimate). When the pixel is an outlier, we simply pay a constant factor λ_d. This prevents the trivial solution where all pixels are outliers. Thus

$$E_{\text{depth}}(\mathbf{p}, \theta_{s_p}, f_p) = \begin{cases} (d(\mathbf{p}) - \hat{d}(\mathbf{p}, \theta_{s_p}))^2 & \text{if } f_p = 0 \\ \lambda_d & \text{otherwise} \end{cases} \tag{11}$$

where λ_d is a constant parameter.

Complexity: We encourage simple explanations (i.e., co-planarity) by defining

$$E_{\text{prior}}(o_{i,j}) = \begin{cases} \lambda_{\text{occ}} & \text{if } o_{i,j} = lo \ \vee \ o_{i,j} = ro \\ \lambda_{\text{hinge}} & \text{if } o_{i,j} = hi \\ 0 & \text{if } o_{i,j} = co \end{cases} \tag{12}$$

where $\lambda_{\text{occ}}, \lambda_{\text{hinge}}$ are constants with $\lambda_{\text{occ}} > \lambda_{\text{hinge}} > 0$. In the absence of this term discontinuous solutions are preferred.

Boundary-Plane Agreement: The plane smoothness energy encourages the planes of adjacent segments to be similar if they belong to the same object. Therefore the smoothness energy between adjacent planes depends on the line label between them: If two neighboring segments are co-planar then the two planes should agree in the full segment, if they form a hinge, they should agree in the boundary, and if they form an occlusion boundary, the occluder should be closer in depth to the camera (i.e., higher disparity). We thus write

$$E_{\text{smo}}(\theta_i, \theta_j, o_{i,j}) = \begin{cases} \phi_{\text{occ}}(\theta_i, \theta_j) & \text{if } o_{i,j} = lo \\ \phi_{\text{occ}}(\theta_j, \theta_i) & \text{if } o_{i,j} = ro \\ \frac{1}{|\mathcal{B}_{i,j}|} \sum_{\mathbf{p} \in \mathcal{B}_{i,j}} \left(\hat{d}(\mathbf{p}, \theta_i) - \hat{d}(\mathbf{p}, \theta_j) \right)^2 & \text{if } o_{i,j} = hi \\ \frac{1}{|\mathcal{S}_i \cup \mathcal{S}_j|} \sum_{\mathbf{p} \in \mathcal{S}_i \cup \mathcal{S}_j} \left(\hat{d}(\mathbf{p}, \theta_i) - \hat{d}(\mathbf{p}, \theta_j) \right)^2 & \text{if } o_{i,j} = co \end{cases} \tag{13}$$

where $\mathcal{B}_{i,j}$ is the set of pixels on the boundary between segments i, j, $\mathcal{S}_i$ is the set of pixels in segment i, and $\phi_{\text{occ}}(\theta_{\text{front}}, \theta_{\text{back}})$ is a function which penalizes occlusion boundaries that are not supported by the plane parameters

$$\phi_{\text{occ}}(\theta_{\text{front}}, \theta_{\text{back}}) = \begin{cases} \lambda_{\text{pen}} & \text{if } \sum_{\mathbf{p} \in \mathcal{B}_{\text{front,back}}} (\hat{d}(\mathbf{p}, \theta_{\text{front}}) - \hat{d}(\mathbf{p}, \theta_{\text{back}}) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Boundary length: This term encourages super pixels to be regular, preferring straight boundaries

$$E_{\text{bou}}(s_p, s_q) = \begin{cases} 0 & \text{if } s_p = s_q \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

Algorithm 1: Our Block Coordinate Descent algorithm

```
Init segmentation to a regular grid.
Compute μ_i and c_i for each segment i.
Init assigments by running TPS (Algorithm 3)
forall the segments i do
    Initialize θ_i using RANSAC to approximately minimize
    Σ_{p|s_p=i} E_depth(p, θ_i)
end
for k = 1 to out − iters do
    Obtain s, f by running ETPS (i.e., Algorithm 2)
    for j = 1 to in − iters do
        forall the boundaries (i, j) do
            o_{i,j} = argmin_{o_{i,j}} E(s, μ, c, θ, o, f)
        end
        forall the segments i do
            θ_i = argmin_{θ_i} E(s, μ, c, θ, o, f)
        end
    endfor
endfor
```

4.2 Efficient Block Coordinate Descent Inference

The minimization of Eq. (8) is NP-hard. Furthermore, it is particularly challenging as it involves inference in a Markov random field (MRF) containing a large number of both discrete (i.e., $\{s, f, o\}$) and continuous variables (i.e., $\{\theta, \mu, c\}$). Previous work employed particle methods to solve continuous-discrete problems by forming a sequence of discrete MRFs, which can be minimized using message passing algorithms [30, 31] or fusion moves with QPBO [26]. This however is computationally very expensive.

In contrast, in this paper we derive a simple yet effective block coordinate descent algorithm which is several orders of magnitude faster than particle methods. Our approach alternates three steps: (i) jointly solving for the pixel-wise outlier flags f, the pixel-to-segment assignments s, as well as the location μ and appearance descriptions c of the segments, (ii) estimating the segment boundary labels o, and (iii) estimating the plane parameters θ. Algorithm 1 summarizes our block coordinate descent algorithm including the initialization of the latent information. We now describe these three steps and initialization in more detail.

Extended Topology-Preserving Segmentation: Our first step optimizes jointly over the segmentation, pixel-wise outlier flags as well as the appearance and location of the segments, while enforcing that each segment is composed of a single connected component with no holes. Note that this is in contrast with segmentation algorithms such as SLIC [1], which require a post processing step to guarantee connectivity and hole-free solutions. Towards this goal, we derive a novel algorithm, called Extended Topology Preserving Segmentation (ETPS), which works as follows. We initialize the stack to contain all boundary

Algorithm 2: ETPS: Extended Topology Preserving Segmentation

Initialize the stack to contain all boundary pixels.
while *not empty stack* **do**
  Take pixel $\mathbf{p}$ off the stack.
  if valid_connectivity$(\mathbf{p}) = 0$ **then**
    continue
  end
  $\{f_p, s_p\} = \text{argmin}_{\{f_p, s_p \in \cup\{s_{\mathcal{N}_4(\mathbf{p})}\}\}} E(s, \mu, c, \theta, o, f)$
  if s_p *updated* **then**
    incrementally update μ and c for the two altered segments.
    Push the boundary pixels in $\mathcal{N}_4(\mathbf{p})$ onto the stack.
  end
end

Algorithm 3: TPS: Topology Preserving Segmentation

Initialize the stack to contain all boundary pixels.
while *not empty stack* **do**
  Take pixel $\mathbf{p}$ off the stack.
  if valid_connectivity$(\mathbf{p}) = 0$ **then**
    continue
  end
  $s_p = \text{argmin}_{\{s_p \in \cup\{s_{\mathcal{N}_4(\mathbf{p})}\}\}} E_{\text{col}}(\mathbf{p}, c_{s_p}) + \lambda_{\text{pos}} E_{\text{pos}}(\mathbf{p}, \mu_{s_p}) +$
  $\lambda_{bou} \sum_{p,q \in N8} E_{bou}(s_p, s_q)$
  if s_p *updated* **then**
    incrementally update μ and c for the two altered segments.
    Push the boundary pixels in $\mathcal{N}_4(\mathbf{p})$ onto the stack.
  end
end

pixels. While the stack is not empty, we take a pixel from the stack and check whether changing its segment assignment will break connectivity. If not, we update the assignment and the outlier flag for that pixel, as well as the location and appearance of the two segments with membership changes (i.e., the segment that pixel $\mathbf{p}$ was assigned in the previous iteration as well as the new assigned segment). This can be done very efficiently using the incremental mean equation, i.e., given the previous estimate m_{n-1} and a new element a_n the mean can computed as

$$m_n = m_{n-1} + \frac{a_n - m_{n-1}}{n}$$

We then push onto the stack the new boundary pixels using a 4-neighborhood around $\mathbf{p}$, as the boundary has changed due to the change of assignment of pixel $\mathbf{p}$. We refer the reader to Algorithm 2 for a summary of ETPS.

Boundary and Slanted Planes: We solve for the superpixel boundaries (second step) by iteratively computing the maximal argument for each boundary.

	> 2 pixels		> 3 pixels		> 4 pixels		> 5 pixels		End-Point	
	Non-Occ	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**
ALTGV [16]	7.88 %	9.30 %	5.36 %	6.49 %	4.17 %	5.07 %	3.42 %	4.17 %	1.1 px	1.2 px
iSGM [12]	7.94 %	10.00 %	5.11 %	7.15 %	3.84 %	5.82 %	3.13 %	5.02 %	1.2 px	2.1 px
ATGV [18]	7.08 %	9.05 %	5.02 %	6.88 %	3.99 %	5.76 %	3.33 %	5.01 %	1.0 px	1.6 px
wSGM [22]	7.27 %	8.72 %	4.97 %	6.18 %	3.88 %	4.89 %	3.25 %	4.11 %	1.3 px	1.6 px
PR-Sceneflow [26]	6.26 %	7.36 %	4.36 %	5.22 %	3.43 %	4.10 %	2.85 %	3.40 %	0.9 px	1.1 px
PCBP [30]	6.08 %	7.62 %	4.04 %	5.37 %	3.14 %	4.29 %	2.64 %	3.64 %	0.9 px	1.1 px
PR-Sf+E [26]	5.79 %	6.88 %	4.02 %	4.87 %	3.15 %	3.82 %	2.62 %	3.17 %	0.9 px	1.0 px
StereoSLIC [31]	5.76 %	7.20 %	3.92 %	5.11 %	3.04 %	4.04 %	2.49 %	3.33 %	0.9 px	1.0 px
PCBP-SS [31]	5.19 %	6.75 %	3.40 %	4.72 %	2.62 %	3.75 %	2.18 %	3.15 %	0.8 px	1.0 px
Ours (Stereo only)	**4.98** %	**6.28** %	**3.39** %	**4.41** %	2.72 %	**3.52** %	2.33 %	**3.00** %	0.9 px	1.0 px
Ours (Joint)	**4.30** %	**5.39** %	**2.83** %	**3.64** %	**2.24** %	**2.89** %	**1.90** %	**2.46** %	**0.8** px	**0.9** px

Table 1. Stereo: Comparison with the state-of-the-art on the test set of KITTI. We highlight in bold when our approach outperforms the state-of-the-art. Ours (stereo only) stands for our slanted plane algorithm when using only the stereo pair, Ours (joint) utilizes both stereo and video pairs.

Solving for the plane parameters (third step) can be done in closed form as the energy is the sum of quadratic functions, including the disparity energy in Eq. (11) and the boundary-plane agreement energy in Eq. (13).

Initialization: As our approach is guaranteed to converge to a local optima, initialization is important. We first initialize the segmentation to form a regular grid, and compute in closed form the mean appearance and location of the superpixels. We then derive a version of ETPS which takes into account the image appearance and not the disparity, and returns superpixels forming a single hole-free connected component. We call this algorithm Boundary-Aware segmentation (TPS). We refer the reader to Algorithm 3. The disparity planes are initialized by minimizing the scene flow energy in Eq. (11) using RANSAC. For each pixel $\mathbf{p}$, f_p is set to 0 when the distance between the initial plane and SGM-StereoFlow estimate is less than a threshold. We then run the iterative algorithm given this initialization as summarized in Algorithm 1.

5 Experimental Evaluation

We performed our experimentation on the challenging KITTI dataset [9], which consists of 194 training and 195 test high-resolution real-world images. The ground truth is semi-dense covering approximately 30 % of the pixels. We employ two different metrics to evaluate our approach: the average number of pixels whose error is bigger than a fixed threshold, as well as the end-point error. We report this for two settings, when only non-occluded pixels are considered as well as predicting all pixels. Unless otherwise stated, we employ the same parameters for all experiments. We set the number of superpixels $n = 1000$, $\lambda_{\text{pos}} = 500$, $\lambda_{\text{dis}} = 2000$, $\lambda_{\text{smo}} = \lambda_{\text{com}} = 400$, $\lambda_{\text{bou}} = 1000$, $\lambda_d = 9$, $\lambda_{\text{occ}} = 15$, $\lambda_{\text{hinge}} = 5$, $\lambda_{\text{pen}} = 30$ and use 10 inner and outer loop iterations.

Comparison to State-of-the-art: We begin our experimentation by comparing our approach to the state-of-the-art. As show in Fig. 1 and Fig. 2 our approach significantly outperforms all stereo, flow and scene flow approaches in the test set of KITTI. The improvements are particularly significant in terms of

	> 2 pixels		> 3 pixels		> 4 pixels		> 5 pixels		End-Point	
	Non-Occ	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**
CRTflow [8]	13.11 %	22.83 %	9.43 %	18.72 %	7.79 %	16.51 %	6.86 %	15.06 %	2.7 px	6.5 px
TVL1-HOG [19]	12.06 %	23.06 %	7.91 %	18.90 %	6.20 %	16.83 %	5.26 %	15.45 %	2.0 px	6.1 px
DeepFlow [29]	9.31 %	20.44 %	7.22 %	17.79 %	6.08 %	16.02 %	5.31 %	14.69 %	1.5 px	5.8 px
Data-Flow [25]	9.16 %	17.41 %	7.11 %	14.57 %	6.05 %	12.91 %	5.34 %	11.72 %	1.9 px	5.5 px
TGV2ADCSIFT [5]	8.04 %	17.87 %	6.20 %	15.15 %	5.24 %	13.43 %	4.60 %	12.27 %	1.5 px	4.5 px
MotionSLIC [31]	5.68 %	13.20 %	3.91 %	10.56 %	3.10 %	9.08 %	2.60 %	8.04 %	0.9 px	2.7 px
PR-Sceneflow [26]	5.67 %	10.32 %	3.76 %	7.39 %	2.96 %	5.98 %	2.52 %	5.14 %	1.2 px	2.8 px
PCBP-Flow [31]	5.28 %	10.62 %	3.64 %	8.28 %	2.90 %	7.01 %	2.46 %	6.16 %	0.9 px	2.2 px
PR-Sf+E [26]	5.58 %	10.13 %	3.57 %	7.07 %	2.69 %	5.48 %	2.17 %	4.49 %	0.9 px	1.6 px
Ours (Flow only)	**5.01** %	12.41 %	**3.38** %	10.06 %	**2.69** %	8.79 %	2.28 %	**7.90** %	**0.9** px	2.9 px
Ours (Joint)	**4.75** %	**8.69** %	**2.82** %	**5.61** %	**2.03** %	**4.10** %	**1.61** %	**3.26** %	**0.8** px	**1.3** px

Table 2. Flow: Comparison with the state-of-the-art on the test set of KITTI. We highlight in bold when our approach outperforms the state-of-the-art. Ours (flow only) stands for our slanted plane algorithm when using only the video pair, Ours (joint) utilizes both stereo and video pairs.

	Stereo									
	> 2 pixels		> 3 pixels		> 4 pixels		> 5 pixels		End-Point	
	Non-Occ	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**
SGM-Stereo	7.42 %	8.62 %	4.93 %	5.89 %	3.73 %	4.50 %	3.01 %	3.65 %	0.9 px	1.1 px
SGM-StereoFlow	6.21 %	7.42 %	4.06 %	5.03 %	3.04 %	3.85 %	2.46 %	3.13 %	0.8 px	1.0 px
Slanted Plane	4.83 %	5.87 %	3.18 %	3.99 %	2.50 %	3.19 %	2.12 %	2.71 %	0.8 px	0.9 px
	Flow									
	> 2 pixels		> 3 pixels		> 4 pixels		> 5 pixels		End-Point	
	Non-Occ	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**	**Non-Occ**	**All**
SGM-Flow	5.55 %	15.31 %	3.67 %	12.62 %	2.83 %	10.97 %	2.35 %	9.74 %	0.9 px	2.9 px
SGM-StereoFlow	5.03 %	8.81 %	3.14 %	5.90 %	2.29 %	4.41 %	1.82 %	3.51 %	0.7 px	1.2 px
Slanted Plane	4.40 %	7.69 %	2.67 %	4.93 %	1.96 %	3.61 %	1.58 %	2.87 %	0.7 px	1.1 px

Table 3. Performance of each step on the training set of KITTI. Jointly estimating stereo and motion fields improves significantly performance over the independent baselines. By incorporating segmentation and explicit occlusion reasoning our slanted plane method improves even further.

the occluded pixels, demonstrating the benefit of having a joint energy which reasons about outliers at the level of the pixels and occlusions boundaries between superpixels. Note that our slanted plane method can also be used with only the stereo or the video pair. This is shown in Fig. 1 and Fig. 2 under Ours (Stereo only) and Ours (Flow only). Note that utilizing both pairs results in much better estimation particularly for occluded pixels in flow.

Importance of Every Step: In the next experiment we look at the importance of every step. As shown in Table 3, reasoning jointly about stereo and flow (last two entries) brings large performance improvements with respect to independently estimating each field (i.e., SGM-Stereo and SGM-Flow). This is particularly significant for occluded pixels in flow. Furthermore, our slanted plane algorithm significantly improves over our intermediate steps.

Number of Iterations: In the next experiment we look at performance of our slanted plane method as a function of the number of outer loop iterations for different number of inner loop iterations. As shown in Fig. 2, very good performance can be achieved with a small number of both inner and outer loop iterations. Fig. 3 (right) depicts convergence of the energy in Eq. 8 as a function of the number of inner and outer loop iterations. Note that we can converge extremely quickly with 1 inner loop iteration.

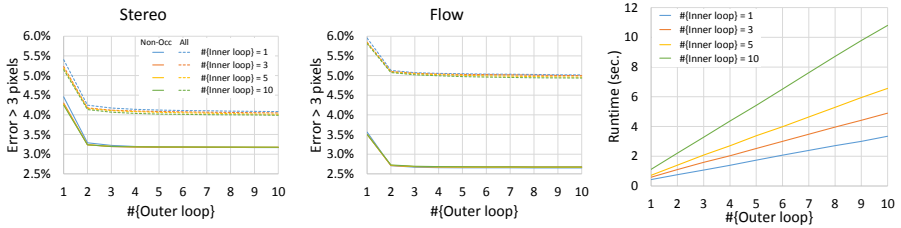

Fig. 2. Performance as a function of the number of iterations

	Joint	Stereo only	Flow only
SGM-Stereo	1.5	1.5	-
Camera motion est.	3.7	-	3.7
SGM-Flow	4.0	-	4.0
Alpha estimation	1.0	-	-
SGM-StereoFlow	12.8	-	-
Slanted plane	3.3	3.3	3.3
Total	26.3 s	4.8 s	11.0 s

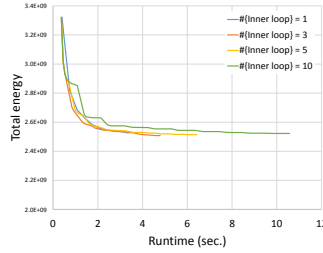

Fig. 3. Runtime (in seconds) of each step of our approach as well total energy for different number of inner loop iterations as a function of time.

Running Time: We next evaluate the running time of our approach in a single core machine. Fig. 3 (left) illustrates the average running time for each step of the algorithm. Note that results superior to the state of the art can be achieved in as little as 3 seconds for our slanted plane algorithm. In comparison, slanted plane methods such as [26, 30, 31] take more than 10 minutes in a single core. Thus, our approach is between 2 to 3 orders of magnitude faster.

Number of superpixels: Fig. 4 shows results as a function of the number of superpixels. Note that performance saturates around 1000 superpixels. This is the number we used for all other experiments.

Sensitivity to Parameters: As shown in Fig. 5, our approach is fairly insensitive to the choice of parameters.

Sensitivity to Motion Magnitude: As shown in Fig. 6 (left) our slanted plane method is not very sensitive to the magnitude of the ego-motion. Furthermore, as observed in in Fig. 6 (center) there is no correlation between the magnitude of motion and the improvement of using stereo and flow w.r.t. to only using stereo. Fig. 6 (right) shows the gain of using stereo and flow as a function of the errors of using only stereo information. Joint inference using flow and stereo helps particularly to correct large errors.

Qualitative Results: Fig. 7 illustrates qualitative results on KITTI. Note that our approach is able to estimate occlusion boundaries as well as hinge labels very accurately.

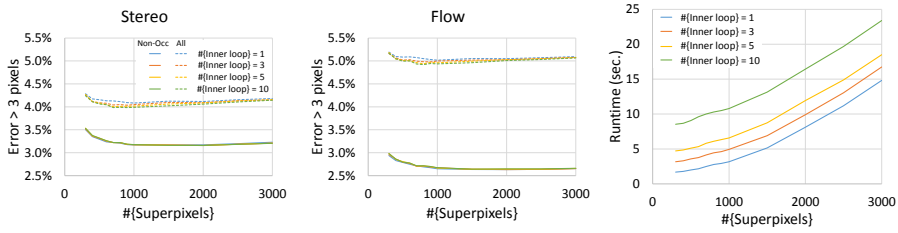

Fig. 4. Performance as a function of the number of superpixels: The performance saturates after 1000 superpixels.

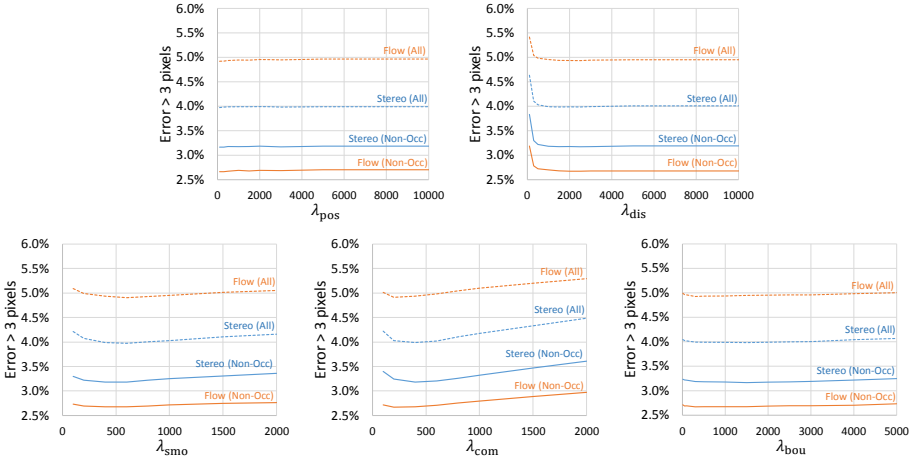

Fig. 5. Sensitivity to Parameters: Our slanted plane model is not sensitive to parameters.

6 Conclusion

We have proposed a fast and accurate algorithm to recover dense depth and motion from stereo video under the assumption that the scene is static. We have demonstrated the effectiveness of our approach in the challenging KITTI dataset, showing state-of-the-art result. Importantly, our approach achieves one order of magnitude speed-ups over current slanted plane methods. We are currently investigating parallel implementations of our approach that can run in real-time in the autonomous driving platform. Furthermore, we believe that the extension to moving objects by employing motion segmentation is also a very interesting venue of future work.

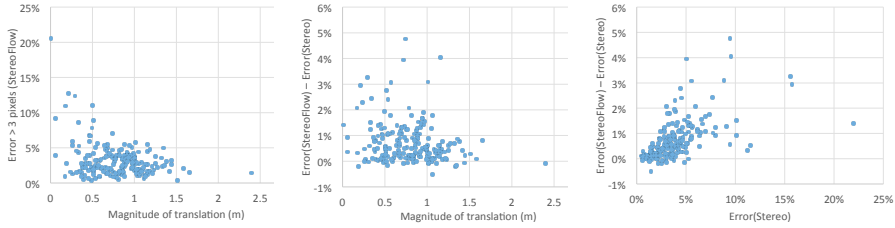

Fig. 6. Sensitivity to Motion Magnitude: (left) Error of our slanted plane method as a function of magnitude of motion. (center) Improvement of using stereo and flow w.r.t. only using stereo as a function of motion magnitude. As observed in the figure there is no correlation between the magnitude of motion and the improvement of using stereo + flow w.r.t. only using stereo. (right) Gain of using stereo + flow as a function of the errors of using stereo alone. The gain of using both flow and stereo is large when the error of using only stereo is also large.

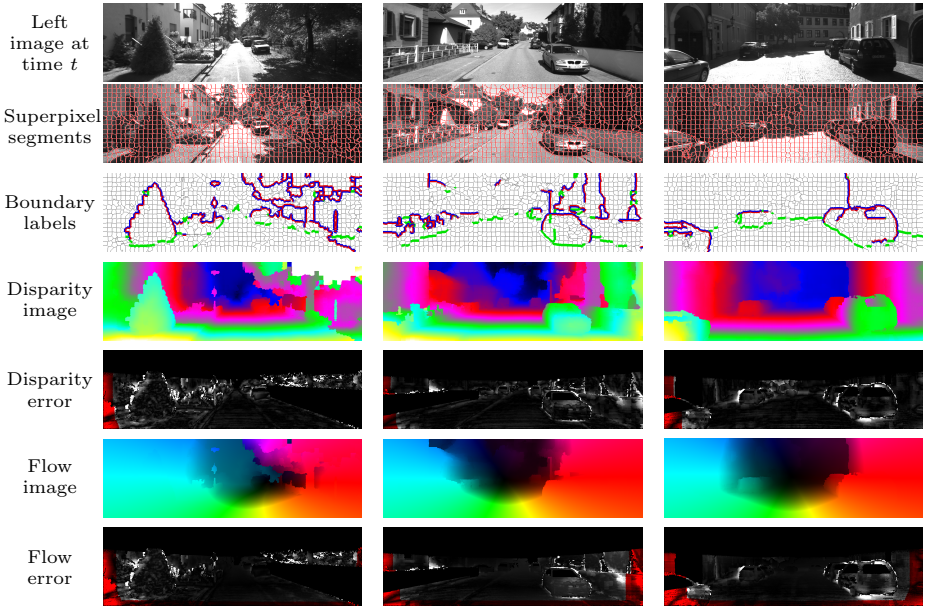

Fig. 7. Qualitative results for our slanted plane model: From top to bottom we show the original left image, our segmentation, boundary labels and the corresponding disparity and flow estimates with their errors. Note that our approach can accurately estimate occlusion boundaries (red/blue) as well as hinge (green) and coplanar (gray) relations.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(11), 2274–2282 (2012)
2. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: A view centered variational approach. International journal of computer vision 101(1), 6–21 (2013)
3. Birchfield, S., Tomasi, C.: Multiway cut for stereo and motion with slanted surfaces. In: CVPR. vol. 1, pp. 489–495. IEEE (1999)
4. Bleyer, M., Gelautz, M.: A layered stereo matching algorithm using image segmentation and global visibility constraints. ISPRS Journal of Photogrammetry and Remote Sensing 59(3), 128–150 (2005)
5. Braux-Zin, J., Dupont, R., Bartoli, A.: A general dense image matching framework combining direct and feature-based costs. In: ICCV (2013)
6. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving objects. International Journal of Computer Vision 29(3), 159–179 (1998)
7. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure from motion without correspondence. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. vol. 2, pp. 557–564. IEEE (2000)
8. Demetz, O., Hafner, D., Weickert, J.: The complete rank transform: A tool for accurate and morphologically invariant matching of structure. In: BMVC (2013)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
10. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
11. Hartley, R.I.: Lines and points in three views and the trifocal tensor. International Journal of Computer Vision 22(2), 125–140 (1997)
12. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In: ACCV (2012)
13. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 807–814. IEEE (2005)
14. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–7. IEEE (2007)
15. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 3, pp. 15–18. IEEE (2006)
16. Kuschk, G., Cremers, D.: Fast and accurate large-scale stereo reconstruction using variational methods. In: ICCV Workshop on Big Data in 3D Computer Vision (2013)
17. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: Computer Vision–ECCV 2010, pp. 582–595. Springer (2010)
18. Ranftl, R., Pock, T., Bischof, H.: Minimizing TGV-based Variational Models with Non-Convex Data terms. In: ICSSVM (2013)

19. Rashwan, H.A., Mohamed, M.A., Garcia, M.A., Mertsching, B., Puig, D.: Illumination robust optical flow model based on histogram of oriented gradients. In: German Conference on Pattern Recognition (GCPR) (2013)
20. Scharstein, D., Szeliski, R.: Middlebury stereo vision page. Online at http://www.middlebury. edu/stereo (2002)
21. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM transactions on graphics (TOG) 25(3), 835–846 (2006)
22. Spangenberg, R., Langner, T., Rojas, R.: Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In: CAIP (2013)
23. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9(2), 137–154 (1992)
24. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. vol. 2, pp. 722–729. IEEE (1999)
25. Vogel, C., Roth, S., Schindler, K.: An evaluation of data costs for optical flow. In: German Conference on Pattern Recognition (GCPR) (2013)
26. Vogel, C., Roth, S., Schindler, K.: Piecewise rigid scene flow. In: ICCV (2013)
27. Wang, Z.F., Zheng, Z.G.: A region based stereo matching algorithm using cooperative optimization. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
28. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. Springer (2008)
29. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: ICCV (2013)
30. Yamaguchi, K., Hazan, T., McAllester, D., Urtasun, R.: Continuous markov random fields for robust stereo estimation. In: ECCV (2012)
31. Yamaguchi, K., McAllester, D., Urtasun, R.: Robust monocular epipolar flow estimation. In: CVPR (2013)