A Continuous Optimization Approach for Efficient and Accurate Scene Flow

Zhaoyang Lv[1(✉)], Chris Beall[1], Pablo F. Alcantarilla[3], Fuxin Li[4], Zsolt Kira[2], and Frank Dellaert[1]

[1] Georgia Institute of Technology, Atlanta, USA
{zlv30,cbeal3}@gatech.edu, dellaert@cc.gatech.edu
[2] Georgia Tech Research Institute, Atlanta, USA
zkira@gatech.edu
[3] iRobot Corporation, London, UK
palcantarilla@irobot.com
[4] Oregon State University, Corvallis, USA
lif@eecs.oregonstate.edu

Abstract. We propose a continuous optimization method for solving dense 3D scene flow problems from stereo imagery. As in recent work, we represent the dynamic 3D scene as a collection of rigidly moving planar segments. The scene flow problem then becomes the joint estimation of pixel-to-segment assignment, 3D position, normal vector and rigid motion parameters for each segment, leading to a complex and expensive discrete-continuous optimization problem. In contrast, we propose a purely continuous formulation which can be solved more efficiently. Using a fine superpixel segmentation that is fixed a-priori, we propose a factor graph formulation that decomposes the problem into photometric, geometric, and smoothing constraints. We initialize the solution with a novel, high-quality initialization method, then independently refine the geometry and motion of the scene, and finally perform a global non-linear refinement using Levenberg-Marquardt. We evaluate our method in the challenging KITTI Scene Flow benchmark, ranking in third position, while being 3 to 30 times faster than the top competitors (x37 [10] and x3.75 [24]).

Keywords: Scene flow · Stereo · Optical flow · Factor graph · Continuous optimization

1 Introduction

Understanding the geometry and motion within urban scenes, using either monocular or stereo imagery, is an important problem with increasingly relevant applications such as autonomous driving [15], urban scene understanding

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46484-8_46) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 757–773, 2016.
DOI: 10.1007/978-3-319-46484-8_46

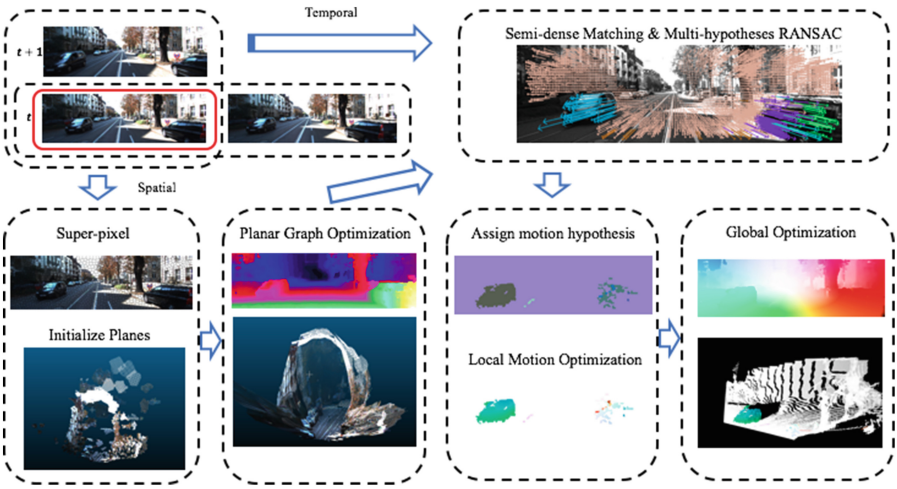

Fig. 1. An overview of our system: we estimate the 3D scene flow w.r.t. the reference image (the red bounding box), a stereo image pair and a temporal image pair as input. Image annotations show the results at each step. We assign a motion hypothesis to each superpixel as an initialization and optimize the factor graph for more accurate 3D motion. Finally, after global optimization, we show a projected 2D flow map in the reference frame and its 3D scene motion (static background are plotted in white). (Color figure online)

[13,15,26], video analysis [7], dynamic reconstruction [12,14], etc. In contrast to separately modeling 3D geometry (stereo) and characterizing the movement of 2D pixels in the image (optical flow), the scene flow problem is to characterize the 3D motion of points in the scene [20] (Fig. 1). Scene flow in the context of stereo sequences was first investigated by Huguet et al. [6]. Recent work [10,19,23] has shown that explicitly reasoning about the scene flow can in turn improve both stereo and optical flow estimation.

Early approaches to scene flow ranged from directly estimating 3D displacement from stereo [30], using volumetric representations [20,21] in a many-camera setting, to re-casting the problem as a 2D disparity flow [6,8] in motion stereo settings. A joint optimization is often leveraged to solve an energy model with all spatio-temporal constraints, e.g. [1,6,10,23], but [19] argues for solving scene and camera motion in an alternating fashion. [25] claims that a decomposed estimation of disparity and motion field can be advantageous as each step can use a different optimization technique to solve the problem more efficiently. A real-time semi-dense scene flow can be achieved without loss of accuracy.

However, efficient and accurate estimation of scene flow is still an unsolved problem. Both dense stereo and optical flow are challenging problems in their own right, and reasoning about the 3D scene must still cope with an equivalent aperture problem [20]. In particular, in scenarios where the scene scale is much larger than the stereo camera baseline, scene motion and depth are hardly

distinguishable. Finally, when there is significant motion in the scene there is a large displacement association problem, an unsolved issue for optical flow algorithms.

Recently, approaches based on a rigid moving planar scene assumption have achieved impressive results [10,22,23]. In these approaches, the scene is represented using planar segments which are assumed to have consistent motion. The scene flow problem is then posed as a discrete-continuous optimization problem which associates each pixel with a planar segment, each of which has continuous rigid 3D motion parameters to be optimized. Vogel et al. [23] view scene flow as a discrete labeling problem: assign the best label to each super-pixel plane from a set of moving plane proposals. [22] additionally leverages a temporal sequence to achieve consistency both in depth and motion estimation. Their approach casts the entire problem into a discrete optimization problem. However, joint inference in this space is both complex and computationally expensive. Menze and Geiger [10] partially address this by parameter-sharing between multiple planar segments, by assuming the existence of a finite set of moving objects in the scene. They solve the candidate motion of objects with continuous optimization, and use discrete optimization to assign the label of each object to each superpixel. However, this assumption does not hold for scenes with non-rigid deformations. Piece-wise continuous planar assumption is not limited to 3D description. [29] achieves state-of-art optical flow results using planar models.

In contrast to this body of work, we posit that it is better to solve for the scene flow in the continuous domain. We adopt the same rigid planar representation as [23], but solve it more efficiently with high accuracy. Instead of reasoning about discrete labels, we use a fine superpixel segmentation that is fixed a-priori, and utilize a robust nonlinear least-squares approach to cope with occlusion, depth and motion discontinuities in the scene. A central assumption is that once a fine enough superpixel segmentation is used as a priori, there is no need to jointly optimize the superpixel segmentation within the system. The rest of the scene flow problem, being piecewise continuous, can be optimized entirely in continuous domain. A good initialization is obtained by leveraging *DeepMatching* [27]. We achieve fast inference by using a sparse nonlinear least squares solver and avoid discrete approximation. To utilize Census cost for fast robust cost evaluation in continuous optimization, we proposes a differentiable Census-based cost, similar to but not same as the approach in [2].

This work makes the following contributions: first, we propose a factor-graph formulation of the scene flow problem that exposes the inherent sparsity of the problem, and use a state of the art sparse solver that directly optimizes over the manifold representations of the continuous unknowns. Compared to the same representation in [23], we achieve better accuracy and faster inference. Second, instead of directly solving for all unknowns, we propose a pipeline to decompose geometry and motion estimation. We show that this helps us cope with the highly nonlinear nature of the objective function. Finally, as initialization is crucial for nonlinear optimization to succeed, we use the DeepMatching algorithm from [27] to obtain a semi-dense set of feature correspondences from which we

initialize the 3D motion of each planar segment. As in [10], we initialize planes from a restricted set of motion hypotheses, but optimize them in the continuous domain to cope with non-rigid objects in the scene.

2 Scene Flow Analysis

We follow [23] in assuming that our 3D world is composed of locally smooth and rigid objects. Such a world can be represented as a set of rigid planes moving in 3D, $\mathcal{P} = \{\bar{\mathbf{n}}, \mathcal{X}\}$, with parameters representing the plane normal $\bar{\mathbf{n}}$ and motion $\mathcal{X}$. In the ideal case, a slanted plane projects back to one or more superpixels in the images, inside of which the appearance and geometry information are locally similar. The inverse problem is then to infer the 3D planes (parameters $\bar{\mathbf{n}}$ and $\mathcal{X}$), given the images and a set of pre-computed superpixels.

3D Plane. We denote a plane as $\bar{\mathbf{n}}$ in 3-space, specified by its normal coordinates in the reference frame. For any 3D point $\mathbf{x} \in \mathbf{R}^3$ on $\bar{\mathbf{n}}$, the plane equation holds as $\bar{\mathbf{n}}^\top \mathbf{x} + 1 = 0$. We choose this parameterization for ease of optimization on its manifold (refer to Sect. 2.3.)

Plane Motion. A rigid plane transform $\mathcal{X} \in \mathbf{SE}(3)$ comprising rotation and translation is defined by

$$\mathcal{X} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R} \in \mathbf{SO}(3), \mathbf{t} \in \mathbf{R}^3 \tag{1}$$

Superpixel Associations. We assume each superpixel S_i is a one-to-one mapping from the reference frame to a 3D plane. The boundary between adjacent superpixels S_i and S_j is defined as $\mathcal{E}_{i,j} \in \mathbf{R}^2$.

2.1 Transformation Induced by Moving Planes

For any point $\mathbf{x}$ on $\bar{\mathbf{n}}$, its homogeneous representation is $[\mathbf{x}^\top, -\bar{\mathbf{n}}^\top \mathbf{x}]$. From $\mathbf{x}_0$ in the reference frame, its corresponding point $\mathbf{x}_1$ in an observed frame is:

$$\begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{t}_0^1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ -\bar{\mathbf{n}}^T \mathbf{x}_0 \end{bmatrix} \tag{2}$$

where $[\mathbf{R}_0^1 | \mathbf{t}_0^1]$ is the transform from reference frame to the observed image frame (referred to as $\mathcal{T}_0^1$) and $[\mathbf{R}_i | \mathbf{t}_i]$ is the plane motion in the reference frame (referred to as $\mathcal{X}_i$). Suppose the camera intrinsic matrix as $\mathbf{K}$, A homography transform can thus be induced as:

$$\begin{aligned} \mathbf{H}(\mathcal{P}_i, \mathcal{T}_0^1) &= \mathbf{K}[\mathbf{A} - \mathbf{a}\bar{\mathbf{n}}]\mathbf{K}^{-1} \\ \begin{bmatrix} \mathbf{A} & a \\ 0 & 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{t}_0^1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3}$$

In stereo frames where planes are static, the homography from reference frame to the right frame is simply:

$$\mathbf{H}(\bar{\mathbf{n}}, \mathcal{T}_0^r) = \mathbf{K}(\mathbf{R}_0^r - \mathbf{t}_0^r \bar{\mathbf{n}})\mathbf{K}^{-1} \tag{4}$$

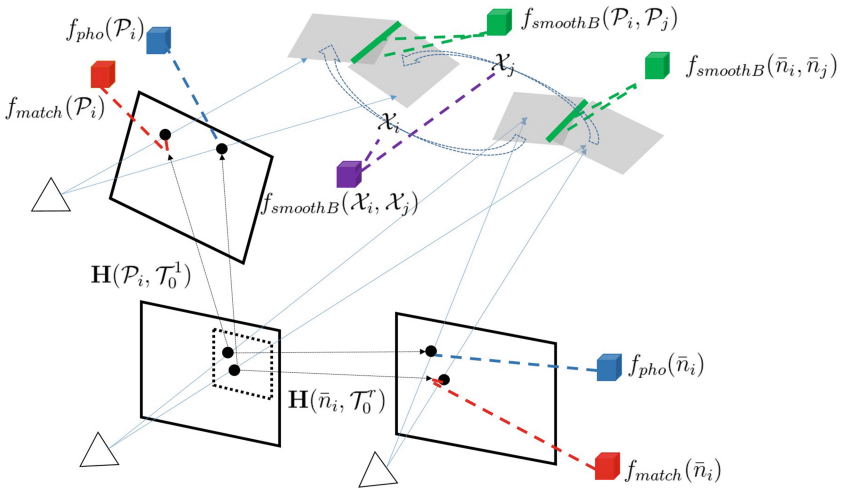

Fig. 2. The proposed factor graph for this scene flow problem. The unary factors are set up based on the homography transform relating two pixels, given $\mathcal{P}$. Binary factors are set up based on locally smooth and rigid assumptions. In this graph, a three-view geometry is used to explain factors for simplicity. Any other views can be constrained by incorporating the same temporal factors in this graph.

We will only use $\mathcal{T}_0^r$ to represent the transform of reference frame to the other stereo frame, while $\mathcal{T}_0^1$ is applicable from reference frame to any other frames, whether the planes are static or moving.

2.2 A Factor Graph Formulation for Scene Flow

For all images $I' : \Omega \rightarrow \mathbf{R}$ relative to the reference image $I : \Omega \rightarrow \mathbf{R}$, we want to estimate all of the planes $\Theta = \{\bar{\mathbf{n}}_{\{1\ldots N\}}, \mathcal{X}_{\{1\ldots N\}}\}$ observed in I. Besides raw image measurements, we also assume that a set of sparsely matched point pairs $M \in \mathbf{R}^2$ is available. As mentioned above, we assume an a-priori fixed superpixel segmentation S, along with its boundaries $\mathcal{E}$. We denote these as our measurements $\mathcal{M} = \{I, I', M, S, \mathcal{E}\}$.

We begin by defining parameters $\theta = \{\bar{\mathbf{n}}, \mathcal{X}\}$, in which $\bar{\mathbf{n}}$ and $\mathcal{X}$ are independent to each other. We also assume dependencies only exist between superpixels across common edges. The joint probability distribution of Θ can then be:

$$\begin{aligned} \mathbf{P}(\Theta, \mathcal{M}) &\propto \prod_{i \in N} \mathbf{P}(\theta_i|\mathcal{M}) \prod_{j \in N \setminus \{i\}} \mathbf{P}(\theta_i, \theta_j|\mathcal{M}) \\ \mathbf{P}(\theta_i|\mathcal{M}) &\propto \mathbf{P}(I', M|\bar{\mathbf{n}}_i, \mathcal{X}_i, S_i, I)\mathbf{P}(\bar{\mathbf{n}}_i)\mathbf{P}(\mathcal{X}_i) \\ \mathbf{P}(\theta_i, \theta_j|\mathcal{M}) &= \mathbf{P}(\bar{\mathbf{n}}_i, \bar{\mathbf{n}}_j|S_i, S_j, \mathcal{E}_{i,j})\mathbf{P}(\mathcal{X}_i, \mathcal{X}_j|S_i, S_j, \mathcal{E}_{i,j}), \end{aligned} \tag{5}$$

Factor graphs (see e.g., [9]) are convenient probabilistic graphical models for formulating the scene flow problem:

$$G(\Theta) = \prod_{i \in N} f_i(\theta_i) \prod_{i,j \in N} f_{ij}(\theta_i, \theta_j), \tag{6}$$

Typically $f(\theta_i)$ encodes a prior or a single measurement constraint at unknown θ, and $f_{i,j}$ relate to measurements or constraints between θ_i, θ_j. In this paper, we assume each factor is a least-square error term with Gaussian noises. To fully represent the measurements and constraints in this problem, we will use multiple factors for $G(\Theta)$ (see Fig. 2), which will be illustrated below.

Unary Factors. A point p, associated with a particular superpixels, can be associated with the homography transform $\mathbf{H}(\mathcal{P}_i, \mathcal{T}_s)$ w.r.t. its measurements. For a stereo camera, the transformation of a point from one image to the other is simply $\mathbf{H}(\bar{\mathbf{n}}, \mathcal{T}_s)$ in Eq. 4. For all the pixels p in superpixel S_i, their photometric costs given $\mathcal{P}\{\bar{\mathbf{n}}_i, \mathcal{X}_i\}$ is described by factor $f_{pho}(\mathcal{P}_i)$:

$$f_{pho}(\mathcal{P}_i) \propto \prod_{p \in S_i} f\big(C(p'), C(\mathbf{H}(\mathcal{P}_i, \mathcal{T}_0^1)p\big), \tag{7}$$

where $C(\cdot)$ is the Census descriptor. This descriptor is preferred over intensity error for its robustness against noise and edges. Similarly, using the homography transform and with sparse matches we can estimate the geometric error of match m by measuring its consistency with the corresponding plane motion:

$$f_{match}(\mathcal{P}_i) \propto \prod_{p \in S_i} f\big(p + m, \mathbf{H}(\mathcal{P}_i, \mathcal{T}_0^1)p\big), \tag{8}$$

Pairwise Factors. The pairwise factors relate the parameters based on their constraints. $f_{smoothB}(\cdot, \cdot)$ describes the locally smooth assumption that adjacent planes should share similar boundary connectivity:

$$f_{smoothB}(\bar{\mathbf{n}}_i, \bar{\mathbf{n}}_j) \propto \prod_{p \in \mathcal{E}_{i,j}} f\big(D^{-1}(\bar{\mathbf{n}}_i, p), D^{-1}(\bar{\mathbf{n}}_j, p)\big), \tag{9}$$

where $D^{-1}(\bar{\mathbf{n}}, p)$ represents the inverse depth of pixel p on $\bar{\mathbf{n}}$. This factor describes the distance of points over the boundary of two static planes. After plane motion, we expect the boundary to still be connected after the transformation:

$$f_{smoothB}(\mathcal{P}_i, \mathcal{P}_j) \propto \prod_{p \in \mathcal{E}_{i,j}} f\big(D^{-1}(\mathcal{P}_i, p), D^{-1}(\mathcal{P}_j, p)\big), \tag{10}$$

With our piece-wise smooth motion assumption, we also expect that two adjacent superpixels should share similar motion parameters, described by $f_{smoothM}$, which is a *Between* operator of $\mathbf{SE}(3)$:

$$f_{smoothM}(\mathcal{X}_i, \mathcal{X}_j) \propto f\big(\mathcal{X}_i, \mathcal{X}_j\big). \tag{11}$$

Each factor is created as a Gaussian noise model: $f(x;m) = \exp(-\rho(h(x) - m)_\Sigma)$ for unary factor and $f(x_1, x_2) = \exp(-\rho(h_1(x_1) - h_2(x_2))_\Sigma)$ for binary factor. $\rho(\cdot)_\Sigma$ is the Huber robust cost which measures the Mahalanobis norm. It incorporates the noise effect of each factor and down-weights the effect of outliers. Given a decent initialization, this robust kernel helps us to cope with occlusions, depth and motion discontinuities properly.

2.3 Continuous Optimization of Factor Graph on Manifold

The factor graph in Eq. 5 can be estimated via maximum a posteriori (MAP) as a non-linear least square problem, and solved with standard non-linear optimization methods. In each step, we linearize all the factors at $\theta = \{\bar{\mathbf{n}}_\theta, \mathcal{X}_\theta\}$. On manifold, the update is a *Retraction* $\mathcal{R}_\theta$. The retraction for $\{\bar{\mathbf{n}}, \mathcal{X}\}$ is:

$$\mathcal{R}_\theta(\delta\bar{\mathbf{n}}, \delta\mathcal{X}) = (\bar{\mathbf{n}} + \delta\bar{\mathbf{n}}, \mathcal{X}\mathrm{Exp}(\delta x)), [\delta\bar{\mathbf{n}} \in \mathbf{R}^3, \delta x \in \mathbf{R}^6] \tag{12}$$

For $\bar{\mathbf{n}} \in \mathbf{R}^3$, it has the same value of its tangent space at any value $\hat{n}$. This explains our choice of plane representation: it is the most convenient for manifold optimization in all of its families in 3-space. For motion in $\mathbf{SE}(3)$, the retraction is an exponential map.

Although the linearized factor graph can be thought of as a huge matrix, it is actually quite sparse in nature: pairwise factors only exist between adjacent superpixels. Sparse matrix factorization can solve this kind of problem very efficiently. We follow the same sparse matrix factorization which is discussed in detail in [4].

2.4 Continuous Approximation for Census Transform

In Eq. 7, there are two practical issues: first, we cannot get a sub-pixel Census Transform; and second, the Hamming distance between the two descriptors is not differentiable. To overcome these problems, we use bilinear interpolated distance as the census cost (see Fig. 3). The bilinear interpolation equation is differentiable w.r.t. the image coordinate, from which we can approximately get the Jacobian of Census Distance w.r.t. to a sub-pixel point. We use a 9×7 size Census, and set up Eq. 7 over a pyramid of images. In evaluation, we will discuss how this process helps us to achieve better convergence purely with a data-cost.

3 Scene Flow Estimation

The general pipeline of our algorithms consists of five steps (see Fig. 1). We summarize each step and provide detailed descriptions in the subsections below.

Initialization. We initialize the superpixels for the reference frame. For both of the stereo pairs, we estimate a depth map as priors. The 3D plane is initialized from the depth map using RANSAC.

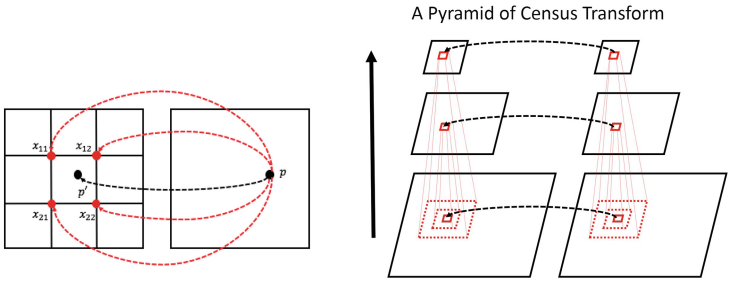

Fig. 3. The left figure shows how to use bilinear interpolation to achieve a differentiable cost of Census Transform. In the right figure, a census descriptor is extracted at different pyramid levels of the images. When evaluating its distance w.r.t. another pixel, we also use bilinear interpolation to evaluate census cost at lower resolution images.

Planar Graph Optimization. We solve the factor graph composed of factors in Eqs. 7, 8 and 9. The result is the estimation of plane geometry parameter $\bar{\mathbf{n}}$ w.r.t. reference frame.

Estimation of Motion Hypotheses. We first estimate a semi-dense matching from reference frame to the next temporal frame and associate them with our estimated 3D plane to get a set of 3D features. We use RANSAC to heuristically find a set of motion hypothesis. In each RANSAC step, we find the most likely motion hypothesis of Eq. 3 by minimizing the re-projection errors of 3D features in two temporally consecutive frames. A set of motion hypotheses are generated by iterating this process.

Local Motion Graph Optimization. We initialize the motion of superpixels from the set of motion hypotheses, framed as a Bayesian classification problem. For all of the superpixels assigned to one single motion hypothesis, we estimate both the plane $\bar{\mathbf{n}}$ and its motion $\mathcal{X}$, by incorporating factors in Eqs. 7, 10 and 11.

Global Graph Optimization. In this step, the set of all unknowns $\mathcal{P}$ is estimated globally. All factors from Eqs. 7–11 are used.

3.1 Initialization

The superpixels in the reference frame are initialized with the sticky-edge superpixels introduced in [31]. Since the urban scene is complex in appearance, the initialized superpixel number needs to be large to cope with tiny objects, while too many superpixels can cause an under-constrained condition for some plane parameters. Empirically, we find generating 2,000 superpixels is a good balance (refer to our superpixel discussion in supplement materials.)

We use the stereo method proposed in [28] to generate the stereo prior, and initialize the 3D planes with a plane-fitting RANSAC algorithm. The plane is initialized as frontal parallel if the RANSAC inlier percentage is below a certain

threshold (50 % in our setting), or the plane induces a degenerated homography transform (where the plane is parallel to the camera focal axis).

We sample robust matches $\mathcal{M}$ from the disparity map, and use it to set up the matching factor in Eq. 8. The samples are selected from the Census Transform which share a maximum distance of 3 bits, given the disparity matching.

3.2 Planar Graph Optimization

In the stereo factor graph, we only estimate the planes $\bar{\mathbf{n}}$ from the factors in Eq. 7, i.e. we constrain the motion $\mathcal{X}$ to be constant (Eqs. 8 and 9). Suppose for each Gaussian noise factor, r is its residual: $f(x) = \exp(-r(x))$. We can obtain the maximum a posterior (MAP) of the factor graph by minimizing the residuals in the least-square problem:

$$\begin{aligned} \bar{\mathbf{n}}^{\star} &= \text{argmax}_{\bar{\mathbf{n}}} \prod f_{pho}(\bar{\mathbf{n}}_i) \cdot \prod f_{match}(\bar{\mathbf{n}}_i) \cdot \prod f_{smoothB}(\bar{\mathbf{n}}_i, \bar{\mathbf{n}}_j) \\ &= \text{argmin}_{\bar{\mathbf{n}}} \sum r_{pho}(\bar{\mathbf{n}}_i) + \sum r_{match}(\bar{\mathbf{n}}_i) + \sum r_{smoothB}(\bar{\mathbf{n}}_i, \bar{\mathbf{n}}_j) \end{aligned} \tag{13}$$

Levenberg-Marquardt can be used to solve this equation as a more robust choice (e.g. compared to Gauss-Newton), trading off efficiency for accuracy.

3.3 Semi-dense Matching and Multi-hypotheses RANSAC

We leverage the state-of-art matching method [27] to generate a semi-dense matching field, which has the advantage of being able to associate across large displacements in the image space. To estimate the initial motion for superpixels, we chose RANSAC similar to [10]. We classify putatives as inliers based on their re-projection errors. The standard-deviation $\sigma = 1$ is small to ensure that bad hypotheses are rare. All hypotheses with more than 20 % inliers in each step are retained. Compared to the up-to-5 hypotheses in [10], we found empirically that our RANSAC strategy can retrieve 10–20 hypotheses in complex scenes, which ensures a high recall of even small moving objects, or motion patterns on non-rigid objects (e.g. pedestrians and cyclists). This process can be quite slow when noisy matches are prominent and inliers ratios are low. To cope with this effect, we use superpixels as a prior in RANSAC. We evaluate the inlier superpixels (indicated by inlier feature matches through non-maximum suppression), and reject conflicting feature matches as outliers. This prunes the number of motion hypotheses, and substantially speeds up this step. See Fig. 4 for an illustration of the motion hypotheses.

Since the most dominant transform in the scene is induced by the camera transform, we can get an estimate of the incremental camera transform in the first iteration. After each iteration, the hypothesis is refined by a weighted least squares optimization, solved efficiently by Levenberg-Marquardt.

Fig. 4. A visualization of motion hypothesis (left), optical flow (middle), and scene motion flow (right). Camera motion is explicitly removed from scene motion flow. In the image of the cyclist we show that although multiple motion hypotheses are discovered by RANSAC (in two colors), a final smooth motion over this non-rigid entity is estimated with continuous optimization. (Color figure online)

3.4 Local Motion Estimation

After estimation of the plane itself, we initialize the motion $\mathcal{X}_i$ of each individual plane from the set of motion hypotheses. At this step, given the raw image measurements $I_{0,1}$, a pair of estimated depth maps in both frames $D_{0,1}$, and the sparse point-matching field F, the goal is to estimate the most probable hypothesis $l^\star$ for each individual superpixel. We assume a set of conditional independencies among $I_{0,1}$, $D_{0,1}$, and F, given the superpixel. The label l for each superpixel can therefore be inferred from the Bayes rule:

$$\begin{aligned} P(l|F, I_{0,1}, D_{0,1}) &\propto P(F, I_{0,1}, D_{0,1}|l)P(l) \\ &\propto P(I_{0,1}|l)P(D_{0,1}|l)P(F, I_0, D_0|l)P(l), \end{aligned} \tag{14}$$

Assuming each motion hypothesis has equally prior information, a corresponding MAP estimation to the above equation can be presented as:

$$l^\star = \text{argmax}_{l^\star}\, \mathbf{E}_{depth}(l) + \alpha \mathbf{E}_{photometric}(l) + \beta \mathbf{E}_{cluster}(l), \tag{15}$$

where $\mathbf{E}_{depth}(l)$ represents the depth error between the warped depth and transformed depth, given a superpixel and its plane; $\mathbf{E}_{photometric}(l)$ represents the photometric error between the superpixel and its warped superpixel; $\mathbf{E}_{cluster}(l)$ represents the clustering error of a superpixel, w.r.t. its neighborhood features:

$$\begin{aligned} \mathbf{E}_{depth}(l) &= \sum_{p_i \in S} (D_1(\mathbf{H}p_i)) - z(\mathbf{H}p_i))^2, \\ \mathbf{E}_{photometric}(l) &= \sum_{p_i \in S} (I(p_i) - I(\mathbf{H}p_i))^2, \\ \mathbf{E}_{cluster}(l) &= \sum_{p_i \in S} \sum_{p_k \in F_l} \exp(-\frac{\bigtriangledown I_{i,k}^2}{\sigma_i^2}) \exp(-\frac{\bigtriangledown D_{i,k}^2}{\sigma_D^2}), \end{aligned} \tag{16}$$

where $\mathbf{H}$ is the homography transform and $z(p)$ is the depth at pixel p. $\bigtriangledown I_{i,k}^2$ and $\bigtriangledown D_{i,k}^2$ describes the color and depth difference of a pixel $p_i \in S$ to a feature point $p_k \in F_l$ belonging to hypothesis l. σ_I and σ_D are their variances.

A local motion optimization is done for each hypothesis by incorporating the factors 7, 8, 10, 11 with pre-estimated planes values as:

$$\mathcal{X}^{\star} = \underset{\mathcal{X}}{\operatorname{argmin}} \sum r_{pho}(\mathcal{X}_i) + \sum r_{match}(\mathcal{X}_i) + \sum r_{smoothB}(\mathcal{X}_i, \mathcal{X}_j) + \sum r_{smoothM}(\mathcal{X}_i, \mathcal{X}_j) + \sum r_{prior}(\mathcal{M}). \tag{17}$$

Similar to Eq. 13, r is the residual for each factor. We add a prior factor $f_{prior}(\cdot)$ to enforce an L_2 prior centered at 0. It works as a diagonal term to improve the condition numbers in the matrix factorization. The prior factor has small weights and in general do not affect the accuracy or speed significantly.

3.5 Global Optimization

Finally, we estimate the global factor graph, with the complete set of parameters $\mathcal{P} = \{\bar{\mathbf{n}}, \mathcal{X}\}$ in the reference frame. The factors in this stage are set using measurements in all of the other three views, w.r.t. reference image.

$$\mathcal{P}^{\star} = \underset{\mathcal{P}}{\operatorname{argmin}} \sum r_{pho}(\mathcal{P}_i) + \sum r_{match}(\mathcal{P}_i) + \sum r_{smoothB}(\mathcal{P}_i, \mathcal{P}_j) + \sum r_{smoothM}(\mathcal{P}_i, \mathcal{P}_j) + \sum r_{prior}(\mathcal{P}_i) \tag{18}$$

4 Experiments and Evaluations

Our factors and optimization algorithm are implemented using GTSAM [3]. As input to our method, we use super-pixels generated from [31], a fast stereo prior from [28], and the DeepMatching method in [27]. The noise models and robust kernel thresholds of the Gaussian factors are selected based on the first 100 training images in KITTI. In the next subsections, we discuss the results as well as optimization and individual factor contribution to the results.

4.1 Evaluation over KITTI

We evaluate our algorithm on the challenging KITTI Scene Flow benchmark [10], which is a realistic benchmark in outdoor environments. In the KITTI benchmark, our method ranks *3rd in Scene Flow test* while being significantly faster than close competitors, as well as *3nd in the KITTI Optical Flow test* and 11th in the stereo test which we did not explicitly target. We show our quantitative scene flow results in Table 1 and qualitative visualizations in Fig. 6.

Table 1 shows a comparison of our results against the other top 4 publicly-evaluated scene flow algorithms. In addition, we also added [6] (which proposed the four-image setting in scene flow) as a general comparison. In all of these results, the errors in disparity and flow evaluation are counted if the disparity or flow estimation exceeds 3 pixels and 5 % of its true value. In the Scene Flow evaluation, the error is counted if any pixel in any of the three estimates (two

Table 1. Quantitative Results on KITTI Scene Flow Test Benchmark. We show the disparity errors reference frame (D1) and second frame (D2), flow error (Fl), and the scene flow (SF) in 200 test images on KITTI. The errors are reported as background (bg), foregound (fg), and all pixels (bg+fg), OCC for errors over all areas, NOC only for errors non-occluded areas.

Method	Occlusion (OCC) error												
	D1			D2			Fl			SF			time
	bg %	fg %	all %	bg %	fg %	all %	bg %	fg %	all %	bg %	fg %	all %	
PRSM [24]	**3.02**	**10.52**	**4.27**	**5.13**	**15.11**	**6.79**	**5.33**	**17.02**	**7.28**	**6.61**	**23.60**	**9.44**	300 s
OSF [10]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	22.17	8.37	7.01	28.76	10.63	50 min
PRSF [23]	4.74	13.74	6.24	11.14	20.47	12.69	11.73	27.73	14.39	13.49	33.72	16.85	150 s
SGM+SF [5]	5.15	15.29	6.84	14.10	23.13	15.60	20.91	28.90	22.24	23.09	37.12	25.43	45 min
SGM+C+NL [18]	5.15	15.29	6.84	28.77	25.65	28.25	34.24	45.40	36.10	38.21	53.04	40.68	4.5 min
VSF [6]	27.73	21.72	26.38	59.51	44.93	57.08	50.06	47.57	49.64	67.69	64.03	67.08	125 min
Ours	4.57	13.04	5.98	7.92	20.76	10.06	10.40	30.33	13.71	12.21	36.97	16.33	**80 s**
Method	Non-Occlusion (NOC) error												
	D1			D2			Fl			SF			time
	bg %	fg %	all %	bg %	fg %	all %	bg %	fg %	all	bg %	fg %	all	
PRSM [24]	**2.93**	**10.00**	**4.10**	**4.13**	**12.85**	**5.69**	4.33	**14.15**	**6.11**	**5.54**	**20.16**	**8.16**	300 s
OSF [10]	4.14	11.12	5.29	4.49	16.33	6.61	**4.21**	18.65	6.83	5.52	24.58	8.93	50 min
PRSF [23]	4.41	13.09	5.84	6.35	16.12	8.10	6.94	23.64	9.97	8.35	28.45	11.95	150 s
SGM+SF [5]	4.75	14.22	6.31	8.34	18.71	10.20	13.36	25.21	15.51	15.28	32.33	18.33	45 min
SGM+C+NL [18]	4.75	14.22	6.31	15.72	20.79	16.63	23.03	41.92	26.46	26.22	48.61	30.23	4.5 min
VSF [6]	26.38	19.88	25.31	52.30	40.83	50.24	41.15	44.16	41.70	61.14	60.38	61.00	125 min
Ours	**4.03**	11.82	5.32	6.39	16.75	8.25	8.72	26.98	12.03	10.26	32.58	14.26	**80 s**

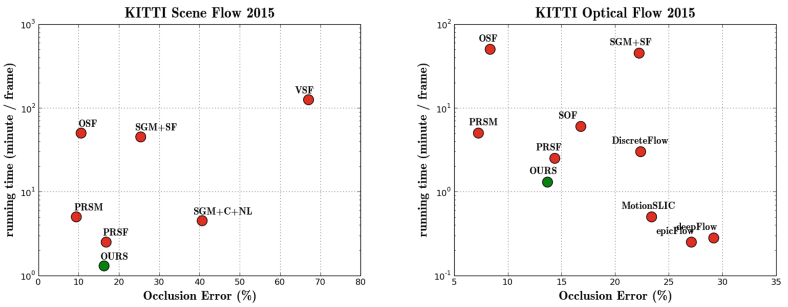

Fig. 5. Occlusion error-vs-time on KITTI. The running time axis is plotted in log scale. Our method is highlighted as green, which achieves top performance both in accuracy and computation speed. (Color figure online)

stereo frame disparity images and flow image) exceed the criterion. We plot a error-vs-time figure in Fig. 5, which shows that our method achieves state-of-art performance, when considering both efficiency and accuracy.

Our results show a small difference in occlusion-errors, although occlusion is not directly handled as discrete labels. We follow the same representation in [23] and achieved better performance in overall pixel errors and faster inference.

Compared to all of these methods, our method is the fastest. Detailed test results are presented in our supplementary materials.

Table 2. Quantitative Results on KITTI Optical Flow 2015 Dataset. The errors are reported as background error(Fl-bg), foreground error (Fl-fg), and all pixels (Fl-bg+Fl-fg), NOC for non-occluded areas error and OCC for errors over all pixels. Methods that use stereo information are shown as *italic*.

Method	OCC error			NOC error			time
	Fl-bg %	Fl-fg %	all %	Fl-bg %	Fl-fg %	all %	
PRSM [24]	**5.33**	**17.02**	**7.28**	4.33	**14.15**	**6.11**	300 s
OSF [10]	5.62	22.17	8.37	**4.21**	18.65	6.83	50 min
PRSF [23]	11.73	27.32	14.39	6.94	23.64	9.97	150 s
SOF [17]	14.63	27.73	16.81	8.11	23.28	10.86	6 min
SGM SF[5]	20.91	28.90	22.24	13.36	25.21	15.51	45 min
DiscreteFlow [11]	21.53	26.68	22.38	9.96	22.17	12.18	3 min
MotionSLIC [28]	14.86	66.21	23.40	6.19	64.82	16.83	30 s
epicFlow [16]	25.81	33.56	27.10	15.00	29.39	17.61	**15 s**
deepFlow [27]	27.96	35.28	29.18	16.47	31.25	19.15	17s
ours	10.40	30.33	13.71	8.72	26.98	12.03	80 s

Table 2 shows our method compared to state-of-art optical flow methods. Methods using stereo information are shown in italic. The deepFlow [27] and epicFlow [16] methods are also presented; these also leverage DeepMatching for data-association. Our method is third best for all-pixels estimation.

4.2 Parameter Discussions

In Table 3, we evaluate the choice of each factor and their effects in the results. During motion estimation, we see that multi-scale Census has an important positive effect in improving convergence towards the optima. Note that the best choice of weights for each factor was tuned by using a similar analysis. A more detailed parameter analyses is presented in the supplement materials.

5 Conclusions

We present an approach to solve the scene flow problem in continuous domain, resulting in a high accuracy (3rd) on the KITTI Scene Flow benchmark at a large computational speedup. We show that faster inference is achievable by rethinking the solution as a non-linear least-square problem, cast within a factor graph formulation. We then develop a novel initialization method, leveraging a multi-scale differentiable Census-based cost and DeepMatching. Given this

Table 3. Evaluation over factors. The non-occlusion error are used from 50 images of KITTI training set. The corresponding factors (in braces) are in Sect. 2.2

Stereo error % (Noc)				Flow error % (Noc)			
Factors	D1-bg %	D1-fg %	D1-all %	Factors	F-bg %	F-fg %	F-all %
Census (7)	9.21	19.22	12.31	Census raw only (7)	10.9	34.25	14.20
Matching (8)	5.95	15.20	7.62	Census multi-scale (7)	9.3	30.13	12.45
Census + matching (7, 8)	5.66	15.01	6.93	Matching only (8)	10.5	33.40	13.20
Census + continuity (7, 9)	4.85	14.22	5.94	Census + piecewise motion (7, 11)	9.0	29.01	12.45
All (7, 8, 9)	4.13	10.20	4.85	Census + continuity (7, 10)	9.2	30.15	12.44
				All (7, 8, 11, 10)	8.92	28.92	12.31

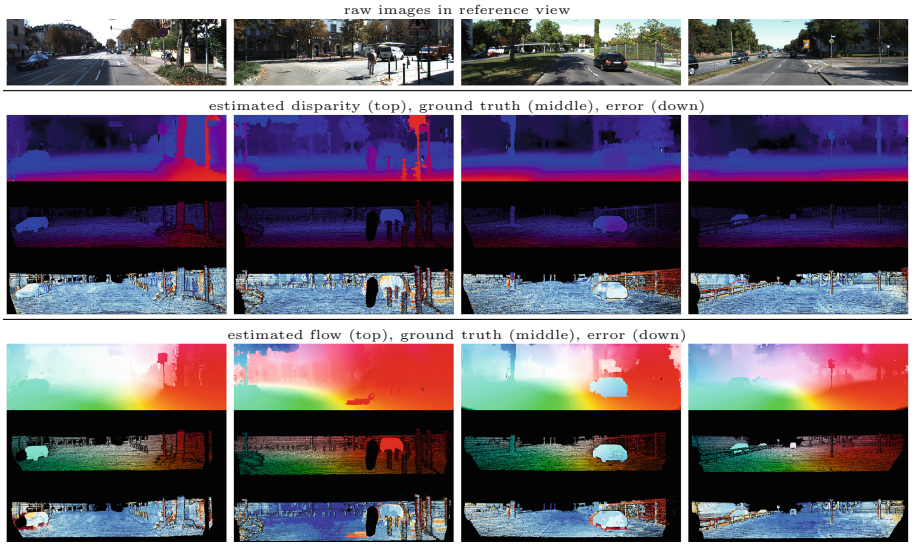

Fig. 6. Qualitative Results in KITTI. We show the disparity and flow estimation against the ground truth results in Kitti Scene Flow training set.

initialization, we individually optimize geometry (stereo) and motion (optical flow) and then perform a global refinement using Levenberg-Marquardt. Analysis shows the positive effects of each of these contributions, ultimately leading to a fast and accurate scene flow estimation.

The proposed method has already achieved significant speed and accuracy, and several enhancements are possible. For example, there are several challenging points and failure cases that we do not cope with so far, such as photometric inconsistency in scenes and areas with aperture ambiguity. To address these problems, we expect to explore more invariant constraints than the current unary factors, and more prior knowledge to enforce better local consistency. Finally, it is possible that additional speed-ups could be achieved through profiling and optimization of the code. Such improvements in both accuracy and speed would enable a host of applications related to autonomous driving, where both are crucial factors.

Acknowledgments. This work was supported by the National Science Foundation and National Robotics Initiative (grant # IIS-1426998). Fuxin Li was partially supported by NSF # 1320348.

References

1. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: a view centered variational approach. Int. J. Comput. Vis. **101**, 6–21 (2012)
2. Vogel, C., Roth, S., Schindler, K.: An evaluation of data costs for optical flow. In: Weickert, J., Hein, M., Schiele, B. (eds.) GCPR 2013. LNCS, vol. 8142, pp. 343–353. Springer, Heidelberg (2013)
3. Dellaert, F.: Factor graphs and GTSAM: a hands-on introduction. Technical report, GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012
4. Dellaert, F., Kaess, M.: Square Root SAM: simultaneous localization and mapping via square root information smoothing. Intl. J. Robot. Re. **25**(12), 1181–1203 (2006)
5. Hornacek, M., Fitzgibbon, A., Rother, C.: SphereFlow: 6 DOF scene flow from RGB-D pairs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
6. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: International Conference on Computer Vision (ICCV). IEEE (2007)
7. Hung, C.H., Xu, L., Jia, J.: Consistent binocular depth and scene flow with chained temporal profiles. Intl. J. Comput. Vis. **102**(1–3), 271–292 (2013)
8. Isard, M., MacCormick, J.: Dense motion and disparity estimation via loopy belief propagation. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 32–41. Springer, Heidelberg (2006)
9. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theor. **47**(2), 498–519 (2001)
10. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

11. Menze, M., Heipke, C., Geiger, A.: Discrete optimization for optical flow. In: Gall, J., Gehler, P., Leibe, B. (eds.) GCPR 2015. LNCS, vol. 9358, pp. 16–28. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24947-6_2
12. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: reconstruction and tracking of non-rigid scenes in real-time. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
13. Pfeiffer, D., Franke, U.: Efficient representation of traffic scenes by means of dynamic stixels. In: Proceedings of the IEEE Intelligent Vehicles Symposium, San Diego, CA, pp. 217–224, June 2010
14. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Intl. J. Comput. Vis. **72**(2), 179–193 (2007)
15. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 582–595. Springer, Heidelberg (2010)
16. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: edge-preserving interpolation of correspondences for optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
17. Sevilla-Lara, L., Sun, D., Jampani, V., Black, M.J.: Optical flow with semantic segmentation and localized layers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
18. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. Intl. J. Comput. Vis. **106**(2), 115–137 (2014). doi:10.1007/s11263-013-0644-x
19. Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C., Theobalt, C.: Joint estimation of motion, structure and geometry from stereo sequences. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 568–581. Springer, Heidelberg (2010)
20. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: International Conference on Computer Vision (ICCV), vol. 2, pp. 722–729 (1999)
21. Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. IEEE Trans. Pattern Anal. Machine Intell. **27**(3), 475–480 (2005)
22. Vogel, C., Roth, S., Schindler, K.: View-consistent 3D scene flow estimation over multiple frames. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 263–278. Springer, Heidelberg (2014)
23. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: International Conference on Computer Vision (ICCV), pp. 1377–1384 (2013)
24. Vogel, C., Schindler, K., Roth, S.: 3D scene flow estimation with a piecewise rigid scene model. Intl. J. Comput. Vis. **115**(1), 1–28 (2015)
25. Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., Cremers, D.: Stereoscopic scene flow computation for 3D motion understanding. Intl. J. Comput. Vis. **95**(1), 29–51 (2011)
26. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
27. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: large displacement optical flow with deep matching. In: International Conference on Computer Vision (ICCV) (2013)

28. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 756–771. Springer, Heidelberg (2014)
29. Yang, J., Li, H.: Dense, accurate optical flow estimation with piecewise parametric model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1019–1027 (2015)
30. Zhang, Z., Faugeras, O.D.: Estimation of displacements from two 3-D frames obtained from stereo. IEEE Trans. Pattern Anal. Mach. Intell. **14**(12), 1141–1156 (1992). http://dx.doi.org/10.1109/34.177380
31. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 391–405. Springer, Heidelberg (2014)