Fast Multi-frame Stereo Scene Flow with Motion Segmentation

Tatsunori Taniai*
RIKEN AIP

Sudipta N. Sinha
Microsoft Research

Yoichi Sato
The University of Tokyo

Abstract

We propose a new multi-frame method for efficiently computing scene flow (dense depth and optical flow) and camera ego-motion for a dynamic scene observed from a moving stereo camera rig. Our technique also segments out moving objects from the rigid scene. In our method, we first estimate the disparity map and the 6-DOF camera motion using stereo matching and visual odometry. We then identify regions inconsistent with the estimated camera motion and compute per-pixel optical flow only at these regions. This flow proposal is fused with the camera motion-based flow proposal using fusion moves to obtain the final optical flow and motion segmentation. This unified framework benefits all four tasks – stereo, optical flow, visual odometry and motion segmentation leading to overall higher accuracy and efficiency. Our method is currently ranked third on the KITTI 2015 scene flow benchmark. Furthermore, our CPU implementation runs in 2-3 seconds per frame which is 1-3 orders of magnitude faster than the top six methods. We also report a thorough evaluation on challenging Sintel sequences with fast camera and object motion, where our method consistently outperforms OSF [30], which is currently ranked second on the KITTI benchmark.

(a) Left input frame (reference) (b) Zoom-in on stereo frames

(c) Ground truth disparity (d) Estimated disparity $\mathcal{D}$

(e) Ground truth flow (f) Estimated flow $\mathcal{F}$

(g) Ground truth segmentation (h) Estimated segmentation $\mathcal{S}$

Figure 1. Our method estimates dense disparity and optical flow from stereo pairs, which is equivalent to stereoscopic scene flow estimation. The camera motion is simultaneously recovered and allows moving objects to be explicitly segmented in our approach.

1. Introduction

Scene flow refers to 3D flow or equivalently the dense 3D motion field of a scene [38]. It can be estimated from video acquired with synchronized cameras from multiple viewpoints [28, 29, 30, 43] or with RGB-D sensors [18, 20, 15, 33] and has applications in video analysis and editing, 3D mapping, autonomous driving [30] and mobile robotics.

Scene flow estimation builds upon two tasks central to computer vision – stereo matching and optical flow estimation. Even though many existing methods can already solve these two tasks independently [24, 16, 35, 27, 17, 46, 9], a naive combination of stereo and optical flow methods for computing scene flow is unable to exploit inherent redundancies in the two tasks or leverage additional scene information which may be available. Specifically, it is well known that the optical flow between consecutive image pairs for stationary (rigid) 3D points are constrained by their depths and the associated 6-DOF motion of the camera rig. However, this idea has not been fully exploited by existing scene flow methods. Perhaps, this is due to the additional complexity involved in simultaneously estimating camera motion and detecting moving objects in the scene.

Recent renewed interest in stereoscopic scene flow estimation has led to improved accuracy on challenging benchmarks, which stems from better representations, priors, optimization objectives as well as the use of better optimization methods [19, 45, 8, 30, 43, 28]. However, those state of the art methods are computationally expensive which limits their practical usage. In addition, other than a few exceptions [40], most existing scene flow methods process ev-

*Work done during internship at Microsoft Research and partly at the University of Tokyo.

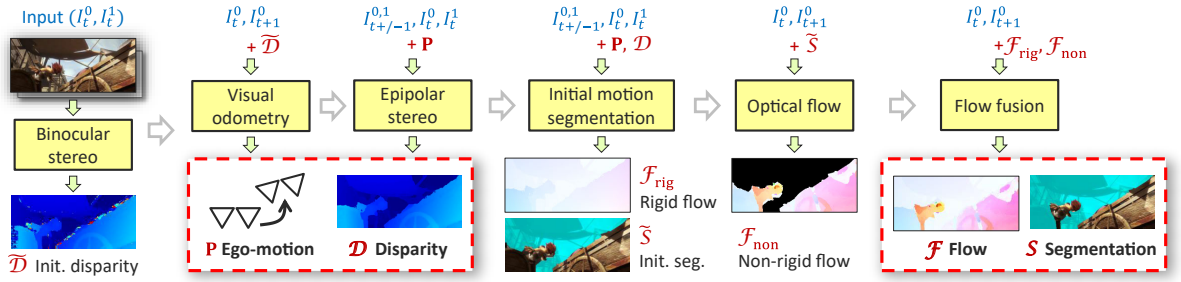


Figure 2. Overview of the proposed method. In the first three steps, we estimate the disparity $\mathcal{D}$ and camera motion $\mathbf{P}$ using stereo matching and visual odometry techniques. We then detect moving object regions by using the rigid flow $\mathcal{F}_{\text{rig}}$ computed from $\mathcal{D}$ and $\mathbf{P}$. Optical flow is performed only for the detected regions, and the resulting non-rigid flow $\mathcal{F}_{\text{non}}$ is fused with $\mathcal{F}_{\text{rig}}$ to obtain final flow $\mathcal{F}$ and segmentation $\mathcal{S}$.

ery two consecutive frames independently and cannot efficiently propagate information across long sequences.

In this paper, we propose a new technique to estimate scene flow from a multi-frame sequence acquired by a calibrated stereo camera on a moving rig. We simultaneously compute dense disparity and optical flow maps on every frame. In addition, the 6-DOF relative camera pose between consecutive frames is estimated along with a per-pixel binary mask that indicates which pixels correspond to either rigid or non-rigid independently moving objects (see Fig. 1). Our sequential algorithm uses information only from the past and present, thus useful for real-time systems.

We exploit the fact that even in dynamic scenes, many observed pixels often correspond to static rigid surfaces. Given disparity maps estimated from stereo images, we robustly compute the 6-DOF camera motion using visual odometry robust to outliers (moving objects in the scene). Given the ego-motion estimate, we improve the depth estimates at occluded pixels via epipolar stereo matching. Then, we identify image regions inconsistent with the camera motion and compute an explicit optical flow proposal for these regions. Finally, this flow proposal is fused with the camera motion-based flow proposal using fusion moves to obtain the final flow map and motion segmentation.

While these four tasks – stereo, optical flow, visual odometry and motion segmentation have been extensively studied, most of the existing methods solve these tasks independently. As our primary contribution, we present a single unified framework where the solution to one task benefits the other tasks. In contrast to some joint methods [43, 30, 28, 42] that try to optimize single complex objective functions, we decompose the problem into simpler optimization problems leading to increased computational efficiency. Our method is significantly faster than top six methods on KITTI taking about 2–3 seconds per frame (on the CPU), whereas state-of-the-art methods take 1–50 minutes per-frame [43, 30, 28, 42]. Not only is our method faster but it also explicitly recovers the camera motion and motion segmentation. We now discuss how our unified framework benefits each of the four individual tasks.

Optical Flow. Given known depth and camera motion, the 2D flow for rigid 3D points which we refer to as *rigid flow* in the paper, can be recovered more efficiently and accurately compared to generic *non-rigid flow*. We still need to compute non-rigid flow but only at pixels associated with moving objects. This reduces redundant computation. Furthermore, this representation is effective for occlusion. Even when corresponding points are invisible in consecutive frames, the rigid flow can be correctly computed as long as the depth and camera motion estimates are correct.

Stereo. For rigid surfaces in the scene, our method can recover more accurate disparities at pixels with left-right stereo occlusions. This is because computing camera motions over consecutive frames makes it possible to use multi-view stereo matching on temporally adjacent stereo frames in addition to the current frame pair.

Visual Odometry. Explicit motion segmentation makes camera motion recovery more robust. In our method, the binary mask from the previous frame is used to predict which pixels in the current frame are likely to be outliers and must be downweighted during visual odometry estimation.

Motion Segmentation. This task is essentially solved for free in our method. Since the final optimization performed on each frame fuses rigid and non-rigid optical flow proposals (using MRF fusion moves) the resulting binary labeling indicates which pixels belong to non-rigid objects.

2. Related Work

Starting with the seminal work by Vedula *et al*. [38, 39], the task of estimating scene flow from multiview image sequences has often been formulated as a variational problem [32, 31, 3, 45]. These problems were solved using different optimization methods – Pons *et al*. [32, 31] proposed a solution based on level-sets for volumetric representations whereas Basha *et al*. [3] proposed view-centric representations suiltable for occlusion reasoning and large motions. Previously, Zhang *et al*. [47] studied how image segmentation cues can help recover accurate motion and depth discontinuities in multi-view scene flow.

Subsequently, the problem was studied in the binocular stereo setting [26, 19, 45]. Huguet and Devernay [19] proposed a variational method suitable for the two-view case and Li and Sclaroff [26] proposed a multiscale approach that incorporated uncertainty during coarse to fine processing. Wedel *et al*. [45] proposed an efficient variational method suitable for GPUs where scene flow recovery was decoupled into two subtasks – disparity and optical flow estimation. Valgaerts *et al*. [36] proposed a variational method that dealt with stereo cameras with unknown extrinsics.

Earlier works on scene flow were evaluated on sequences from static cameras or cameras moving in relatively simple scenes (see [30] for a detailed discussion). Cech *et al*. proposed a seed-growing method for sterescopic scene flow [8] which could handle realistic scenes with many moving objects captured by a moving stereo camera. The advent of the KITTI benchmark led to further improvements in this field. Vogel *et al*. [41, 42, 40, 43] recently explored a type of 3D regularization – they proposed a model of dense depth and 3D motion vector fields in [41] and later proposed a piecewise rigid scene model (PRSM) in two [42] and multi-frame settings [40, 43] that treats scenes as a collection of planar segments undergoing rigid motions. While PRSM [43] is the current top method on KITTI, its joint estimation of 3D geometries, rigid motions and superpixel segmentation using discrete-continuous optimization is fairly complex and computationally expensive. Lv *et al*. [28] recently proposed a simplified approach to PRSM using continuous optimization and fixed superpixels (named CSF), which is faster than [43] but is still too slow for practical use.

As a closely related approach to ours, object scene flow (OSF) [30] segments scenes into multiple rigidly-moving objects based on fixed superpixels, where each object is modeled as a set of planar segments. This model is more rigidly regularized than PRSM. The inference by max-product particle belief propagation is also very computationally expensive taking 50 minutes per frame. A faster setting of their code takes 2 minutes but has lower accuracy.

A different line of work explored scene flow estimation from RGB-D sequences [15, 33, 18, 20, 21, 44]. Meanwhile, deep convolutional neural network (CNN) based supervised learning methods have shown promise [29].

3. Notations and Preliminaries

Before describing our method in details, we define notations and review basic concepts used in the paper.

We denote relative camera motion between two images using matrices $\mathbf{P} = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$, which transform homogeneous 3D points $\hat{\mathbf{x}} = (x, y, z, 1)^T$ in camera coordinates of the source image to 3D points $\mathbf{x}' = \mathbf{P}\hat{\mathbf{x}}$ in camera coordinates of the target image. For simplicity, we assume a rectified calibrated stereo system. Therefore, the two cameras have the same known camera intrinsics matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ and the left-to-right camera pose $\mathbf{P}^{01} = [I| - B\mathbf{e}_x]$ is also known. Here, I is the identity rotation, $\mathbf{e}_x = (1, 0, 0)^T$, and B is the baseline between the left and right cameras.

We assume the input stereo image pairs have the same size of image domains $\Omega \in \mathbb{Z}^2$ where $\mathbf{p} = (u, v)^T \in \Omega$ is a pixel coordinate. Disparity $\mathcal{D}$, flow $\mathcal{F}$ and segmentation $\mathcal{S}$ are defined as mappings on the image domain Ω, *e.g*., $\mathcal{D}(\mathbf{p}) : \Omega \to \mathbb{R}^+$, $\mathcal{F}(\mathbf{p}) : \Omega \to \mathbb{R}^2$ and $\mathcal{S}(\mathbf{p}) : \Omega \to \{0, 1\}$.

Given relative camera motion $\mathbf{P}$ and a disparity map $\mathcal{D}$ of the source image, pixels $\mathbf{p}$ of stationary surfaces in the source image are warped to points $\mathbf{p}' = w(\mathbf{p}; \mathcal{D}, \mathbf{P})$ in the target image by the rigid transformation [14] as

$$w(\mathbf{p}; \mathcal{D}, \mathbf{P}) = \pi\left(\mathbf{K}\mathbf{P}\begin{bmatrix}\mathbf{K}^{-1} & \mathbf{0}\\ \mathbf{0}^T & (fB)^{-1}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{p}}\\ \mathcal{D}(\mathbf{p})\end{bmatrix}\right). \quad (1)$$

Here, $\hat{\mathbf{p}} = (u, v, 1)^T$ is the 2D homogeneous coordinate of $\mathbf{p}$, the function $\pi(u, v, w) = (u/w, v/w)^T$ returns 2D non-homogeneous coordinates, and f is the focal length of the cameras. This warping is also used to find which pixels $\mathbf{p}$ in the source image are visible in the target image using z-buffering based visibility test and whether $\mathbf{p}' \in \Omega$.

4. Proposed Method

Let I_t^0 and I_t^1, $t \in \{1, 2, \cdots, N + 1\}$ be the input image sequences captured by the left and right cameras of a calibrated stereo system, respectively. We sequentially process the first to N-th frames and estimate their disparity maps $\mathcal{D}_t$, flow maps $\mathcal{F}_t$, camera motions $\mathbf{P}_t$ and motion segmentation masks $\mathcal{S}_t$ for the left (reference) images. We call moving and stationary objects as foreground and background, respectively. Below we focus on processing the t-th frame and omit the subscript t when it is not needed.

At a high level, our method is designed to implicitly minimize image residuals

$$E(\mathbf{\Theta}) = \sum_{\mathbf{p}} \|I_t^0(\mathbf{p}) - I_{t+1}^0(w(\mathbf{p}; \mathbf{\Theta}))\| \quad (2)$$

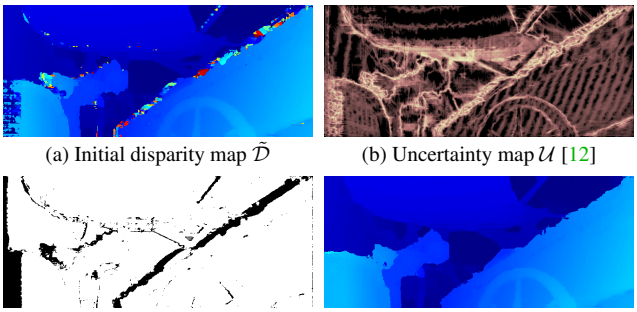
by estimating the parameters $\mathbf{\Theta}$ of the warping function w

$$\mathbf{\Theta} = \{\mathcal{D}, \mathbf{P}, \mathcal{S}, \mathcal{F}_{\text{non}}\}. \quad (3)$$

The warping function is defined, in the form of the flow map $w(\mathbf{p}; \mathbf{\Theta}) = \mathbf{p} + \mathcal{F}(\mathbf{p})$, using the binary segmentation $\mathcal{S}$ on the reference image I_t^0 as follows.

$$\mathcal{F}(\mathbf{p}) = \begin{cases} \mathcal{F}_{\text{rig}}(\mathbf{p}) & \text{if } \mathcal{S}(\mathbf{p}) = \textit{background} \\ \mathcal{F}_{\text{non}}(\mathbf{p}) & \text{if } \mathcal{S}(\mathbf{p}) = \textit{foreground} \end{cases} \quad (4)$$

Here, $\mathcal{F}_{\text{rig}}(\mathbf{p})$ is the rigid flow computed from the disparity map $\mathcal{D}$ and the camera motion $\mathbf{P}$ using Eq. (1), and $\mathcal{F}_{\text{non}}(\mathbf{p})$ is the non-rigid flow defined non-parametrically. Directly estimating this full model is computationally expensive. Instead, we start with a simpler rigid motion model computed



(a) Initial disparity map $\tilde{\mathcal{D}}$ (b) Uncertainty map $\mathcal{U}$ [12]

(c) Occlusion map $\mathcal{O}$ (d) Final disparity map $\mathcal{D}$

Figure 3. Binocular and epipolar stereo. (a) Initial disparity map. (c) Uncertainity map [12] (darker pixels are more confident). (b) Occlusion map (black pixels are invisible in the right image). (d) Final disparity estimate by epipolar stereo.

from the reduced model parameters $\mathbf{\Theta} = \{\mathcal{D}, \mathbf{P}\}$ (Eq. (1)), and then increase the complexity of the motion model by adding non-rigid motion regions $\mathcal{S}$ and their flow $\mathcal{F}_{\text{non}}$. Instead of directly comparing pixel intensities, at various steps of our method, we robustly evaluate the image residuals $\|I(\mathbf{p}) - I'(\mathbf{p}')\|$ by truncated normalized cross-correlation

$$\text{TNCC}_\tau(\mathbf{p}, \mathbf{p}') = \min\{1 - \text{NCC}(\mathbf{p}, \mathbf{p}'), \tau\}. \quad (5)$$

Here, NCC is normalized cross-correlation computed for 5×5 grayscale image patches centered at $I(\mathbf{p})$ and $I'(\mathbf{p}')$, respectively. The thresholding value τ is set to 1.

In the following sections, we describe the proposed pipeline of our method. We first estimate an initial disparity map $\tilde{\mathcal{D}}$ (Sec. 4.1). The disparity map $\tilde{\mathcal{D}}$ is then used to estimate the camera motion $\mathbf{P}$ using visual odometry recovery (Sec. 4.2). This motion estimate $\mathbf{P}$ is used in the epipolar stereo matching stage, where we improve the initial disparity to get the final disparity map $\mathcal{D}$ (Sec. 4.3). The $\mathcal{D}$ and $\mathbf{P}$ estimates are used to compute a rigid flow proposal $\mathcal{F}_{\text{rig}}$ and recover an initial segmentation $\tilde{\mathcal{S}}$ (Sec. 4.4). We then estimate non-rigid flow proposal $\mathcal{F}_{\text{non}}$ for only the moving object regions of $\tilde{\mathcal{S}}$ (Sec. 4.5). Finally we fuse the rigid and non-rigid flow proposals $\{\mathcal{F}_{\text{rig}}, \mathcal{F}_{\text{non}}\}$ and obtain the final flow map $\mathcal{F}$ and segmentation $\mathcal{S}$ (Sec. 4.6). All the steps of the proposed method are summarized in Fig. 2.

4.1. Binocular Stereo

Given left and right images I^0 and I^1, we first estimate an initial disparity map $\tilde{\mathcal{D}}$ of the left image and also its occlusion map $\mathcal{O}$ and uncertainty map $\mathcal{U}$ [12]. We visualize example estimates in Figs. 3 (a)–(c).

As a defacto standard method, we estimate disparity maps by using semi-global matching (SGM) [16] with a fixed disparity range of $[0, 1, \cdots, D_{\text{max}}]$. Our implementation of SGM uses 8 cardinal directions and NCC-based matching costs of Eq. (5) for the data term. The occlusion map $\mathcal{O}$ is obtained by left-right consistency check. The uncertainty map $\mathcal{U}$ is computed during SGM as described in [12] without any computational overhead. We also define a fixed confidence threshold τ_{u} for $\mathcal{U}$, *i.e.*, $\tilde{\mathcal{D}}(\mathbf{p})$ is considered unreliable if $\mathcal{U}(\mathbf{p}) > \tau_{\text{u}}$. More details are provided in the supplementary material.

4.2. Stereo Visual Odometry

Given the current and next image I_t^0 and I_{t+1}^0 and the initial disparity map $\tilde{\mathcal{D}}_t$ of I_t^0, we estimate the relative camera motion $\mathbf{P}$ between the current and next frame. Our method extends an existing stereo visual odometry method [1]. This is a direct method, *i.e.*, it estimates the 6-DOF camera motion $\mathbf{P}$ by directly minimizing image intensity residuals

$$E_{\text{vo}}(\mathbf{P}) = \sum_{\mathbf{p} \in T} \omega_{\mathbf{p}}^{\text{vo}} \rho \left(|I_t^0(\mathbf{p}) - I_{t+1}^0(w(\mathbf{p}; \tilde{\mathcal{D}}_t, \mathbf{P}))| \right) \quad (6)$$

for some target pixels $\mathbf{p} \in T$, using the rigid warping w of Eq. (1). To achieve robustness to outliers (*e.g.*, by moving objects, occlusion, incorrect disparity), the residuals are scored using the Tukey's bi-weight [4] function denoted by ρ. The energy E_{vo} is minimized by iteratively re-weighted least squares in the inverse compositional framework [2].

We have modified this method as follows. First, to exploit motion segmentation available in our method, we adjust the weights $\omega_{\mathbf{p}}^{\text{vo}}$ differently. They are set to either 0 or 1 based on the occlusion map $\mathcal{O}(\mathbf{p})$ but later downweighted by $1/8$, if $\mathbf{p}$ is predicted as a moving object point by the previous mask $\mathcal{S}_{t-1}$ and flow $\mathcal{F}_{t-1}$. Second, to reduce sensitivity of direct methods to initialization, we generate multiple diverse initializations for the optimizer and obtain multiple candidate solutions. We then choose the final estimate $\mathbf{P}$ such that best minimizes weighted NCC-based residuals $E = \sum_{\mathbf{p} \in \Omega} \omega_{\mathbf{p}}^{\text{vo}} \text{TNCC}_\tau(\mathbf{p}, w(\mathbf{p}; \tilde{\mathcal{D}}_t, \mathbf{P}))$. For diverse initializations, we use (a) the identity motion, (b) the previous motion $\mathbf{P}_{t-1}$, (c) a motion estimate by feature-based correspondences using [25], and (d) various forward translation motions (about 16 candidates, used only for driving scenes).

4.3. Epipolar Stereo Refinement

As shown in Fig. 3 (a), the initial disparity map $\tilde{\mathcal{D}}$ computed from the current stereo pair $\{I_t^0, I_t^1\}$ can have errors at pixels occluded in right image. To address this issue, we use the multi-view epipolar stereo technique on temporarily adjacent six images $\{I_{t-1}^0, I_{t-1}^1, I_t^0, I_t^1, I_{t+1}^0, I_{t+1}^1\}$ and obtain the final disparity map $\mathcal{D}$ shown in Fig. 1 (d).

From the binocular stereo stage, we already have computed a matching cost volume of I_t^0 for I_t^1, which we denote as $C_{\mathbf{p}}(d)$, with some disparity range $d \in [0, D_{\text{max}}]$. The goal here is to get a better cost volume $C_{\mathbf{p}}^{\text{epi}}(d)$ as input to SGM, by blending $C_{\mathbf{p}}(d)$ with matching costs for each of the four target images $I' \in \{I_{t-1}^0, I_{t-1}^1, I_{t+1}^0, I_{t+1}^1\}$.

Since the relative camera poses of the current to next frame $\mathbf{P}_t$ and previous to current frame $\mathbf{P}_{t-1}$ are already estimated by the visual odometry in Sec. 4.2, the relative poses from I_t^0 to each target image can be estimated as $\mathbf{P}' \in \{\mathbf{P}_{t-1}^{-1}, \mathbf{P}^{01}\mathbf{P}_{t-1}^{-1}, \mathbf{P}_t, \mathbf{P}^{01}\mathbf{P}_t\}$, respectively. Recall $\mathbf{P}^{01}$ is the known left-to-right camera pose. Then, for each target image I', we compute matching costs $C'_{\mathbf{p}}(d)$ by projecting points $(\mathbf{p}, d)^T$ in I_t^0 to its corresponding points in I' using the pose $\mathbf{P}'$ and the rigid transformation of Eq. (1). Since $C'_{\mathbf{p}}(d)$ may be unreliable due to moving objects, we here lower the thresholding value τ of NCC in Eq. (5) to $1/4$ for higher robustness. The four cost volumes are averaged to obtain $C^{\text{avr}}_{\mathbf{p}}(d)$. We also truncate the left-right matching costs $C_{\mathbf{p}}(d)$ at $\tau = 1/4$ at occluded pixels known by $\mathcal{O}(\mathbf{p})$.

Finally, we compute the improved cost volume $C^{\text{epi}}_{\mathbf{p}}(d)$ by linearly blending $C_{\mathbf{p}}(d)$ with $C^{\text{avr}}_{\mathbf{p}}(d)$ as

$$C^{\text{epi}}_{\mathbf{p}}(d) = (1 - \alpha_{\mathbf{p}})C_{\mathbf{p}}(d) + \alpha_{\mathbf{p}}C^{\text{avr}}_{\mathbf{p}}(d), \tag{7}$$

and run SGM with $C^{\text{epi}}_{\mathbf{p}}(d)$ to get the final disparity map $\mathcal{D}$. The blending weights $\alpha_{\mathbf{p}} \in [0, 1]$ are computed from the uncertainty map $\mathcal{U}(\mathbf{p})$ (from Sec. 4.1) normalized as $u_{\mathbf{p}} = \min\{\mathcal{U}(\mathbf{p})/\tau_u, 1\}$ and then converted as follows.

$$\alpha_{\mathbf{p}}(u_{\mathbf{p}}) = \max\{u_{\mathbf{p}} - \tau_{\text{c}}, 0\}/(1 - \tau_{\text{c}}). \tag{8}$$

Here, τ_{c} is a confidence threshold. If $u_{\mathbf{p}} \leq \tau_{\text{c}}$, we get $\alpha_{\mathbf{p}} = 0$ and thus $C^{\text{epi}}_{\mathbf{p}} = C_{\mathbf{p}}$. When $u_{\mathbf{p}}$ increases from τ_{c} to 1, $\alpha_{\mathbf{p}}$ linearly increases from 0 to 1. Therefore, we only need to compute $C^{\text{avr}}_{\mathbf{p}}(d)$ at $\mathbf{p}$ where $u_{\mathbf{p}} > \tau_{\text{c}}$, which saves computation. We use $\tau_c = 0.1$.

4.4. Initial Segmentation

During the initial segmentation step, the goal is to find a binary segmentation $\tilde{\mathcal{S}}$ in the reference image I_t^0, which shows where the rigid flow proposal $\mathcal{F}_{\text{rig}}$ is inaccurate and hence optical flow must be recomputed. Recall that $\mathcal{F}_{\text{rig}}$ is obtained from the estimated disparity map $\mathcal{D}$ and camera motion $\mathbf{P}$ using Eq. (1). An example of $\tilde{\mathcal{S}}$ is shown in Fig. 4 (f). We now present the details.

First, we define binary variables $s_{\mathbf{p}} \in \{0, 1\}$ as proxy of $\tilde{\mathcal{S}}(\mathbf{p})$ where 1 and 0 correspond to foreground (moving objects) and background, respectively. Our segmentation energy $E_{\text{seg}}(\mathbf{s})$ is defined as

$$E_{\text{seg}} = \sum_{\mathbf{p}\in\Omega} \left[C^{\text{ncc}}_{\mathbf{p}} + C^{\text{flo}}_{\mathbf{p}} + C^{\text{col}}_{\mathbf{p}} + C^{\text{pri}}_{\mathbf{p}}\right] \bar{s}_{\mathbf{p}} + E_{\text{potts}}(\mathbf{s}). \tag{9}$$

Here, $\bar{s}_{\mathbf{p}} = 1 - s_{\mathbf{p}}$. The bracketed terms $[\,\cdot\,]$ are data terms that encode the likelihoods for mask $\tilde{\mathcal{S}}$, *i.e.*, positive values bias $s_{\mathbf{p}}$ toward 1 (moving foreground). E_{potts} is the pairwise smoothness term. We explain each term below.

Appearance term $\mathbf{C^{ncc}_p}$: This term finds moving objects by checking image residuals of rigidly aligned images. We compute NCC-based matching costs between $I = I_t^0$ and $I' = I_{t+1}^0$ as

$$C'^{\text{ncc}}_{\mathbf{p}}(I, I') = \text{TNCC}_{\tau}(\mathbf{p}, \mathbf{p}'; I, I') - \tau_{\text{ncc}} \tag{10}$$

where $\mathbf{p}' = \mathbf{p} + \mathcal{F}_{\text{rig}}(\mathbf{p})$ and $\tau_{\text{ncc}} \in (0, \tau)$ is a threshold. However, TNCC values are unreliable at texture-less regions (see the high-residual tarp in Fig. 4 (a)). Furthermore, if $\mathbf{p}'$ is out of field-of-view, $C'^{\text{ncc}}_{\mathbf{p}}$ is not determined (yellow pixels in Fig. 4 (a)). Thus, similarly to epipolar stereo, we match I_t^0 with $I' \in \{I_{t-1}^0, I_{t-1}^1, I_{t+1}^0, I_{t+1}^1\}$ and compute the average of valid matching costs

$$C^{\text{ncc}}_{\mathbf{p}} = \lambda_{\text{ncc}} w^{\text{var}}_{\mathbf{p}} \,\text{Average}_{I'}\left[C'^{\text{ncc}}_{\mathbf{p}}(I, I')\right]. \tag{11}$$

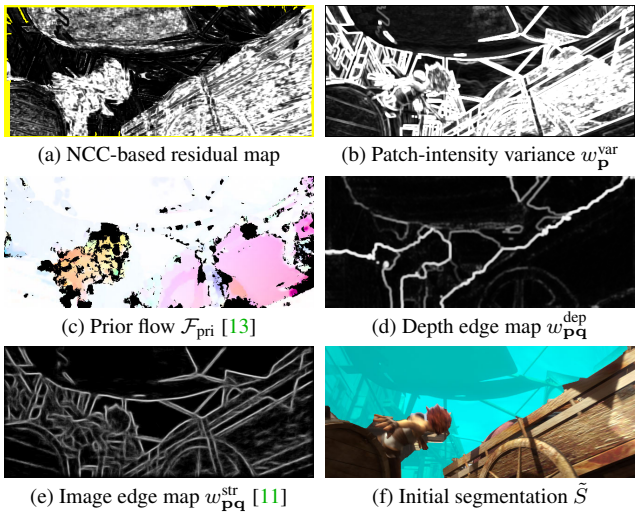
Matching with many images increases the recall for detecting moving objects. To improve matching reliability, $C^{\text{ncc}}_{\mathbf{p}}$ is weighted by $w^{\text{var}}_{\mathbf{p}} = \min(\text{StdDev}(I), \tau_w)/\tau_w$, the truncated standard deviation of the 5×5 patch centered at $I(\mathbf{p})$. The weight map $w^{\text{var}}_{\mathbf{p}}$ is visualized in Fig. 4 (b). We also truncate $C'^{\text{ncc}}_{\mathbf{p}}(I, I')$ at 0, if $\mathbf{p}'$ is expected to be occluded in I' by visibility test. We use $(\lambda_{\text{ncc}}, \tau_{\text{ncc}}, \tau_w) = (4, 0.5, 0.005)$.

Flow term $\mathbf{C^{flo}_p}$: This term evaluates flow residuals $r_{\mathbf{p}} = \|\mathcal{F}_{\text{rig}}(\mathbf{p}) - \mathcal{F}_{\text{pri}}(\mathbf{p})\|$ between the rigid flow $\mathcal{F}$ and (non-rigid) prior flow $\mathcal{F}_{\text{pri}}$ computed by [13] (see Fig. 4 (c)). Using a threshold $\tau^{\text{flo}}_{\mathbf{p}}$ and the patch-variance weight $w^{\text{var}}_{\mathbf{p}}$, we define $C^{\text{flo}}_{\mathbf{p}}$ as

$$C^{\text{flo}}_{\mathbf{p}} = \lambda_{\text{flo}} w^{\text{var}}_{\mathbf{p}} \left[\min(r_{\mathbf{p}}, 2\tau^{\text{flo}}_{\mathbf{p}}) - \tau^{\text{flo}}_{\mathbf{p}}\right]/\tau^{\text{flo}}_{\mathbf{p}}. \tag{12}$$

The part after $w^{\text{var}}_{\mathbf{p}}$ normalizes $(r_{\mathbf{p}} - \tau^{\text{flo}}_{\mathbf{p}})$ to lie within $[-1, 1]$. The threshold $\tau^{\text{flo}}_{\mathbf{p}}$ is computed at each pixel $\mathbf{p}$ by

$$\tau^{\text{flo}}_{\mathbf{p}} = \max(\tau^{\text{flo}}, \gamma\|\mathcal{F}_{\text{rig}}(\mathbf{p})\|). \tag{13}$$



(a) NCC-based residual map (b) Patch-intensity variance $w^{\text{var}}_{\mathbf{p}}$ (c) Prior flow $\mathcal{F}_{\text{pri}}$ [13] (d) Depth edge map $w^{\text{dep}}_{\mathbf{pq}}$ (e) Image edge map $w^{\text{str}}_{\mathbf{pq}}$ [11] (f) Initial segmentation $\tilde{S}$

Figure 4. Initial segmentation. We detect moving object regions using clues from (a) image residuals weighted by (b) patch-intensity variance and (c) prior flow. We also use (d) depth edge and (e) image edge information to obtain (f) initial segmentation.

This way the threshold is relaxed if the rigid motion $\mathcal{F}_{\text{rig}}(\mathbf{p})$ is large. If prior flow $\mathcal{F}_{\text{pri}}(\mathbf{p})$ is invalidated by bi-directional consistency check (black holes in Fig. 4 (c)), $C_{\mathbf{p}}^{\text{flo}}$ is set to 0. We use $(\lambda_{\text{flo}}, \tau^{\text{flo}}, \gamma) = (4, 0.75, 0.3)$.

Prior term $\mathbf{C}_{\mathbf{p}}^{\text{pri}}$: This term encodes segmentation priors based on results from previous frames or on scene context via ground plane detection. Sec. 4.7 for the details.

Color term $\mathbf{C}_{\mathbf{p}}^{\text{col}}$: This is a standard color-likelihood term [6] for RGB color vectors $\mathbf{I}_{\mathbf{p}}$ of pixels in the reference image $I_t^0(\mathbf{p})$:

$$C_{\mathbf{p}}^{\text{col}} = \lambda_{\text{col}} \Big[\log \theta_1(\mathbf{I}_{\mathbf{p}}) - \log \theta_0(\mathbf{I}_{\mathbf{p}}) \Big]. \quad (14)$$

We use $\lambda_{\text{col}} = 0.5$ and 64^3 bins of histograms for the color models $\{\theta_0, \theta_1\}$.

Smoothness term $\mathbf{E}_{\text{potts}}$: This term is based on the Potts model defined for all pairs of neighboring pixels $(\mathbf{p}, \mathbf{q}) \in N$ on the 8-connected pixel grid.

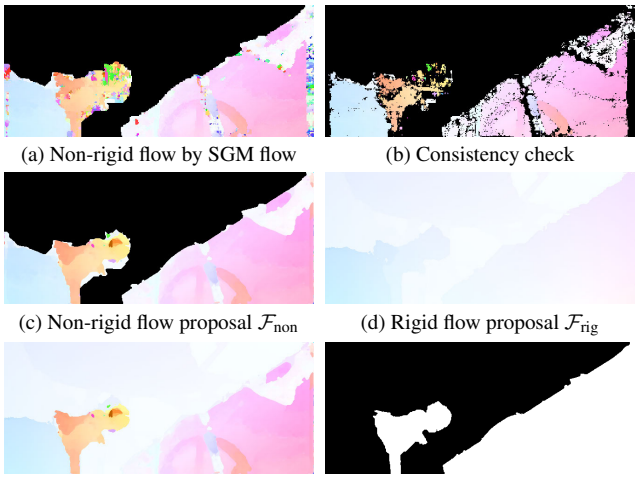
$$E_{\text{potts}}(\mathbf{s}) = \lambda_{\text{potts}} \sum_{(\mathbf{p},\mathbf{q}) \in N} (\omega_{\mathbf{pq}}^{\text{col}} + \omega_{\mathbf{pq}}^{\text{dep}} + \omega_{\mathbf{pq}}^{\text{str}}) |s_{\mathbf{p}} - s_{\mathbf{q}}|. \quad (15)$$

We use three types of edge weights. The color-based weight $\omega_{\mathbf{pq}}^{\text{col}}$ is computed as $\omega_{\mathbf{pq}}^{\text{col}} = e^{-\|\mathbf{I}_{\mathbf{p}} - \mathbf{I}_{\mathbf{q}}\|_2^2 / \kappa_1}$ where κ_1 is estimated as the expected value of $2\|\mathbf{I}_{\mathbf{p}} - \mathbf{I}_{\mathbf{q}}\|_2^2$ over $(\mathbf{p}, \mathbf{q}) \in N$ [34]. The depth-based weight $\omega_{\mathbf{pq}}^{\text{dep}}$ is computed as $\omega_{\mathbf{pq}}^{\text{dep}} = e^{-|L_{\mathbf{p}} + L_{\mathbf{q}}| / \kappa_2}$ where $L_{\mathbf{p}} = |\Delta D(\mathbf{p})|$ is the absolute Laplacian of the disparity map $\mathcal{D}$. The κ_2 is estimated similarly to κ_1. The edge-based weight $\omega_{\mathbf{pq}}^{\text{str}}$ uses an edge map $e_{\mathbf{p}} \in [0, 1]$ obtained by a fast edge detector [11] and is computed as $\omega_{\mathbf{pq}}^{\text{str}} = e^{-|e_{\mathbf{p}} + e_{\mathbf{q}}| / \kappa_3}$. Edge maps of $\omega_{\mathbf{pq}}^{\text{dep}}$ and $\omega_{\mathbf{pq}}^{\text{str}}$ (in the form of $1 - w_{\mathbf{pq}}$) are visualized in Figs. 4 (d) and (e). We use $(\lambda_{\text{potts}}, \kappa_3) = (10, 0.2)$.

The minimization of $E_{\text{seg}}(\mathbf{s})$ is similar to the GrabCut [34] algorithm, *i.e.*, we alternate between minimizing $E_{\text{seg}}(\mathbf{s})$ using graph cuts [5] and updating the color models $\{\theta_1, \theta_0\}$ of $C_{\mathbf{p}}^{\text{col}}$ from segmentation $\mathbf{s}$. We run up to five iterations until convergence using dynamic max-flow [22].

4.5. Optical Flow

Next, we estimate the non-rigid flow proposal $\mathcal{F}_{\text{non}}$ for the moving foreground regions estimated as the initial segmentation $\tilde{\mathcal{S}}$. Similar to Full Flow [9], we pose optical flow as a discrete labeling problem where the labels represent 2D translational shifts with in a 2D search range (see Sec. 4.7 for range estimation). Instead of TRW-S [23] as used in [9], we apply the SGM algorithm as a discrete optimizer. After obtaining a flow map from SGM as shown in Fig. 5 (a), we filter it further by 1) doing bi-directional consistency check (see Fig. 5 (b)), and 2) filing holes by weighted median filtering to get the non-rigid flow proposal $\mathcal{F}_{\text{non}}$. The flow consistency map $\mathcal{O}^{\text{flo}}(\mathbf{p})$ is passed to the next stage. Our extension of SGM is straightforward and is detailed in our supplementary material as well as the refinement scheme.



(a) Non-rigid flow by SGM flow (b) Consistency check

(c) Non-rigid flow proposal $\mathcal{F}_{\text{non}}$ (d) Rigid flow proposal $\mathcal{F}_{\text{rig}}$

(e) Final flow map $\mathcal{F}$ (f) Final segmentation mask $\mathcal{S}$

Figure 5. Optical flow and flow fusion. We obtain non-rigid flow proposal by (a) performing SGM followed by (b) consistency filtering and (c) hole filing by weighted median filtering. This flow proposal is fused with (d) the rigid flow proposal to obtain (e) the final flow estimate and (f) motion segmentation.

4.6. Flow Fusion and Final Segmentation

Given the rigid and non-rigid flow proposals $\mathcal{F}_{\text{rig}}$ and $\mathcal{F}_{\text{non}}$, we fuse them to obtain the final flow estimate $\mathcal{F}$. This fusion step also produces the final segmentation $\mathcal{S}$. These inputs and outputs are illustrated in Figs. 5 (c)–(f).

The fusion process is similar to the initial segmentation. The binary variables $s_{\mathbf{p}} \in \{0, 1\}$ indicating the final segmentation $\mathcal{S}$, now also indicate which of the two flow proposals $\{\mathcal{F}_{\text{rig}}(\mathbf{p}), \mathcal{F}_{\text{non}}(\mathbf{p})\}$ is selected as the final flow estimate $\mathcal{F}(\mathbf{p})$. To this end, the energy E_{seg} of Eq. (9) is modified as follows. First, $C_{\mathbf{p}}^{\text{ncc}}$ is replaced by

$$C_{\mathbf{p}}^{\text{ncc}} = \lambda_{\text{ncc}} w_{\mathbf{p}}^{\text{var}} [\text{TNCC}_\tau(\mathbf{p}, \mathbf{p}'_{\text{rig}}) - \text{TNCC}_\tau(\mathbf{p}, \mathbf{p}'_{\text{non}})], \quad (16)$$

where $\mathbf{p}'_{\text{rig}} = \mathbf{p} + \mathcal{F}_{\text{rig}}(\mathbf{p})$ and $\mathbf{p}'_{\text{non}} = \mathbf{p} + \mathcal{F}_{\text{non}}(\mathbf{p})$. Second, the prior flow $\mathcal{F}_{\text{pri}}(\mathbf{p})$ in $C_{\mathbf{p}}^{\text{flo}}$ is replaced by $\mathcal{F}_{\text{non}}(\mathbf{p})$. When $\mathbf{p}'_{\text{rig}}$ is out of view or $\mathcal{F}_{\text{non}}(\mathbf{p})$ is invalidated by the flow occlusion map $\mathcal{O}^{\text{flo}}(\mathbf{p})$, we set $C_{\mathbf{p}}^{\text{ncc}}$ and $C_{\mathbf{p}}^{\text{flo}}$ to 0.

The fusion step only infers $s_{\mathbf{p}}$ for pixels labeled foreground in the initial segmentation $\tilde{\mathcal{S}}$, since the background labels are fixed. The graph cut optimization for fusion is typically very efficient, since the pixels labeled foreground in $\tilde{\mathcal{S}}$ is often a small fraction of all the pixels.

4.7. Implementation Details

Disparity range reduction. For improving the efficiency of epipolar stereo, the disparity range $[0, D_{\text{max}}]$ is reduced by estimating D_{max} from the initially estimated $\tilde{\mathcal{D}}(\mathbf{p})$. We compute D_{max} robustly by making histograms of non-occluded disparities of $\tilde{\mathcal{D}}(\mathbf{p})$ and ignoring bins whose frequency is less than 0.5%. D_{max} is then chosen as the max

bin from remaining valid non-zero bins.

Flow range estimation. The 2D search range $R = ([u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}])$ for SGM flow is estimated as follows. For the target region $\tilde{\mathcal{S}}$, we compute three such ranges from feature-based sparse correspondences, the prior flow and rigid flow. For the latter two, we robustly compute ranges by making 2D histograms of flow vectors and ignoring bins whose frequency is less than one-tenth of the max frequency. Then, the final range R is the range that covers all three. To make R more compact, we repeat the range estimation and subsequent SGM for individual connected components in $\tilde{\mathcal{S}}$.

Cost-map smoothing. Since NCC and flow-based cost maps $C_{\mathbf{p}}^{\text{ncc}}$ and $C_{\mathbf{p}}^{\text{flo}}$ used in the segmentation and fusion steps are noisy, we smooth them by averaging the values within superpixels. We use superpixelization of approximately 850 segments produced by [37] in OpenCV.

Segmentation priors. We define $C_{\mathbf{p}}^{\text{pri}}$ of Eq. (9) as $C_{\mathbf{p}}^{\text{pri}} = \lambda_{\text{mask}} C_{\mathbf{p}}^{\text{mask}} + C_{\mathbf{p}}^{\text{pcol}}$. Here, $C_{\mathbf{p}}^{\text{mask}} \in [-0.1, 1]$ is a signed soft mask predicted by previous mask $\mathcal{S}_{t-1}$ and flow $\mathcal{F}_{t-1}$. Negative background regions are downweighted by 0.1 for better detection of new emerging objects. We use $\lambda_{\text{mask}} = 2$. $C_{\mathbf{p}}^{\text{pcol}}$ is a color term similar to Eq. (14) with the same λ_{col} but uses color models updated online as the average of past color models. For road scenes, we additionally use the ground prior such as shown in Fig. 6 as a cue for the background. It is derived by the ground plane detected using RANSAC. See the supplementary material for more details.

Figure 6. Segmentation ground prior. For road scenes (left), we compute the ground prior (middle) from the disparity map (right).

Others. We run our algorithm on images downscaled by a factor of 0.4 for optical flow and 0.65 for the other steps (each image in KITTI is 1242×375 pixels). We do a subpixel refinement of the SGM disparity and flow maps via standard local quadratic curve fitting [16].

5. Experiments

We evaluate our method on the KITTI 2015 scene flow benchmark [30] and further extensively evaluate on the challenging Sintel (stereo) datasets [7]. On Sintel we compare with the top two state of the art methods – PRSM [43] and OSF [30]. PRSM is a multi-frame method like ours. Although OSF does not explicitly distinguish moving objects from static background in segmentation, the dominant rigid motion bodies are assigned the first object index, which we regarded as background in evaluations. Our method was implemented in C++ and running times were measured on a computer with a quadcore 3.5GHz CPU. All parameter settings were determined using KITTI training data for validation. Only two parameters were re-tuned for Sintel.

5.1. KITTI 2015 Scene Flow Benchmark

We show a selected ranking of KITTI benchmark results in Table 1, where our method is ranked third. Our method is much faster than all the top methods and more accurate than the fast methods [10, 8]. See Fig. 8 for the per-stage running times. The timings for most stages of our method are small and constant, while for optical flow they vary depending on the size of the moving objects. Motion segmentation results are visually quite accurate (see Fig. 7). As shown in Table 2, epipolar stereo refinement using temporarily adjacent stereo frames improves disparity accuracy even for non-occluded pixels. By visual inspection of successive images aligned via the camera motion and depth, we verified that there was never any failure in ego-motion estimation.

5.2. Evaluation on Sintel Dataset

Unlike previous scene flow methods, we also evaluated our method on Sintel and compared it with OSF [30] and PRSM [43] (see Table 3 – best viewed in color). Recall, PRSM does not perform motion segmentation. Although OSF and PRSM are more accurate on KITTI, our method outperforms OSF on Sintel on all metrics. Also, unlike OSF, our method is multi-frame. Sintel scenes have fast, unpredictable camera motion, drastic non-rigid object motion and deformation unlike KITTI where vehicles are the only type of moving objects. While OSF and PRSM need strong rigid regularization, we employ per-pixel inference without requiring piecewise planar assumption. Therefore, our method generalizes more easily to Sintel. Only two parameters had to be modified as follows. $(\lambda_{\text{col}}, \tau_{\text{ncc}}) = (1.5, 0.25)$.

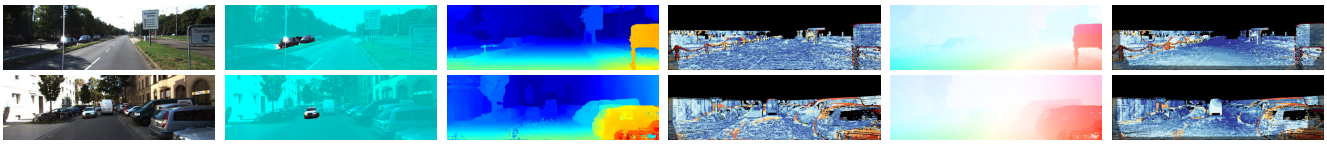
Limitations. The visual odometry step may fail when the scene is far away (see *mountain_1* in Fig. 9) due to subtle disparity. It may also fail when the moving objects dominate the field of view. Our motion segmentation results are often accurate but in the future we will improve temporal consistency to produce more coherent motion segmentation.

6. Conclusions

We proposed an efficient scene flow method that unifies dense stereo, optical flow, visual odometry, and motion segmentation estimation. Even though simple optimization methods were used in our technique, the unified framework led to higher overall accuracy and efficiency. Our method is currently ranked third on the KITTI 2015 scene flow benchmark after PRSM [43] and OSF [30] but is 1–3 orders of magnitude faster than the top six methods. On challenging Sintel sequences, our method outperforms OSF [30] and is close to PRSM [43] in terms of accuracy. Our efficient method could be used to initialize PRSM [43] to improve its convergence speed. We hope it will enable new, practical applications of scene flow.

Table 1. KITTI 2015 scene flow benchmark results [30]. We show the error rates (%) for the disparity on the reference frame (D1) and second frame (D2), the optical flow (Fl) and the scene flow (SF) at background (bg), foreground (fg) and all pixels. Disparity or flow is considered correctly estimated if the end-point error is < 3px or $< 5\%$. Scene flow is considered correct if D1, D2 and Fl are correct.

Rank	Method	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
1	PRSM [43]	**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
2	OSF [30]	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
3	**FSF+MS** (ours)	5.72	11.84	6.74	7.57	21.28	9.85	8.48	29.62	12.00	11.17	37.40	15.54	2.7 s
4	CSF [28]	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
5	PR-Sceneflow [42]	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
8	PCOF + ACTF [10]	6.31	19.24	8.46	19.15	36.27	22.00	14.89	62.42	22.80	25.77	69.35	33.02	0.08 s (GPU)
12	GCSF [8]	11.64	27.11	14.21	32.94	35.77	33.41	47.38	45.08	47.00	52.92	59.11	53.95	2.4 s



(a) Reference image (b) Motion segmentation $\mathcal{S}$ (c) Disparity map $\mathcal{D}$ (d) Disparity error map (e) Flow map $\mathcal{F}$ (f) Flow error map

Figure 7. Our results on KITTI testing sequences *002* and *006*. Black pixels in error heat maps indicate missing ground truth.

Table 2. Disparity improvements by epipolar stereo.

	all pixels			non-occluded pixels		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
Binocular ($\tilde{\mathcal{D}}$)	7.96	12.61	8.68	7.09	10.57	7.61
Epipolar ($\mathcal{D}$)	**5.82**	**10.34**	**6.51**	**5.57**	**8.84**	**6.06**

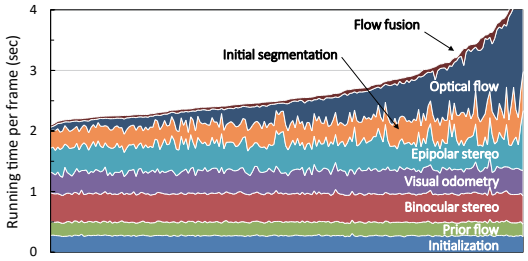
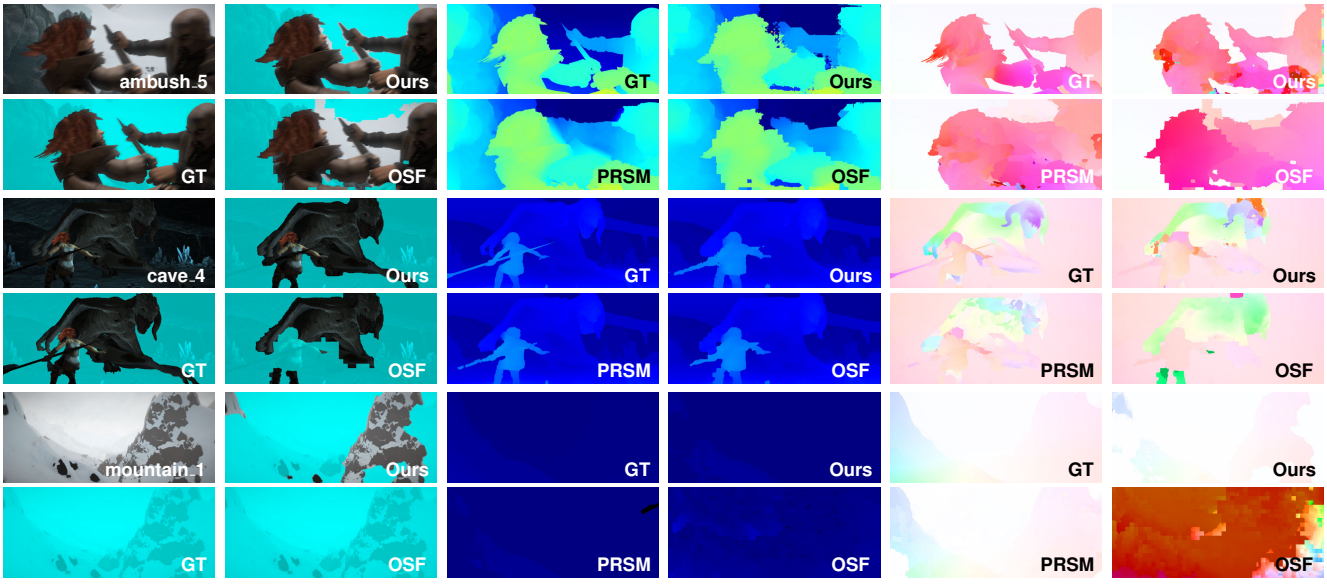


Figure 8. Running times on 200 sequences from KITTI. The average running time per-frame was 2.7 sec. Initialization includes edge extraction [11], superpixelization [37] and feature tracking.

Table 3. Sintel evaluation [7]: We show error rates (%) for disparity (D1), flow (Fl), scene flow (SF) and motion segmentation (MS) averaged over the frames. Cell colors in OSF [30] and PRSM [43] columns show performances relative to ours; blue shows where our method is better, red shows where it is worse. We outperform OSF most of the time.

	D1-all			Fl-all			SF-all			MS-all	
	Ours	OSF	PRSM	Ours	OSF	PRSM	Ours	OSF	PRSM	Ours	OSF
alley_1	5.92	**5.28**	7.43	2.11	7.33	**1.58**	**6.91**	10.04	7.90	**5.40**	17.45
alley_2	2.08	1.31	**0.79**	1.20	1.44	**1.08**	2.99	2.49	**1.63**	1.94	**1.31**
ambush_2	**36.93**	55.13	41.77	72.68	87.37	**51.33**	80.33	90.96	**61.92**	**1.72**	32.76
ambush_4	**23.30**	24.05	24.09	45.23	49.16	**41.99**	49.81	53.25	**46.14**	20.98	**19.82**
ambush_5	18.54	19.54	**17.72**	**24.82**	44.70	25.23	35.15	52.26	**34.12**	**2.50**	19.39
ambush_6	30.33	**26.18**	29.41	44.05	54.75	**41.98**	49.93	58.46	**47.08**	53.95	**24.98**
ambush_7	**23.47**	71.58	35.07	27.87	22.47	**3.35**	44.51	77.94	**36.92**	**26.77**	36.08
bamboo_1	9.67	9.71	**7.34**	4.11	4.04	**2.41**	11.05	10.81	**8.35**	4.43	**4.17**
bamboo_2	19.27	18.08	**17.06**	3.65	4.86	**3.58**	21.39	21.24	**19.23**	**4.08**	4.54
bandage_1	20.93	**19.37**	21.22	4.00	18.40	**3.30**	23.72	36.57	**23.37**	**33.32**	46.66
bandage_2	22.69	23.53	**22.44**	4.76	13.12	**4.06**	24.19	32.33	**23.62**	**16.37**	41.14
cave_4	6.22	5.86	**4.27**	**14.62**	33.94	16.32	17.53	36.04	**17.71**	**16.13**	16.92
market_2	6.81	6.61	**5.27**	5.17	10.08	**4.77**	10.38	14.52	**8.54**	**8.97**	13.90
market_5	**13.25**	13.67	15.38	**26.31**	29.58	28.38	**29.93**	31.60	32.00	**15.26**	15.33
market_6	10.63	10.29	**8.99**	13.13	16.39	**10.72**	18.07	20.18	**15.09**	**3.59**	37.63
mountain_1	**0.23**	0.78	0.42	17.05	88.60	**3.71**	17.05	88.61	**3.85**	31.63	**0.00**
shaman_2	**24.77**	28.27	25.49	**0.56**	1.67	**0.46**	**25.07**	29.43	25.75	30.98	**27.04**
shaman_3	**27.09**	52.22	33.92	**1.31**	11.45	1.75	27.61	55.51	34.43	**3.81**	29.64
sleeping_2	3.52	2.97	**1.74**	0.02	0.01	**0.00**	3.52	2.97	**1.74**	**0.00**	0.54
temple_2	5.96	5.54	**4.92**	9.66	10.52	**9.51**	**9.82**	10.55	9.87	**1.32**	4.13
temple_3	**10.65**	16.62	11.04	62.34	81.39	**32.10**	63.56	81.86	**34.60**	**4.20**	25.42
AVERAGE	**15.35**	19.84	15.99	18.32	28.16	**13.70**	27.26	38.93	**23.52**	**13.68**	19.95



Reference images / motion segmentation Disparity maps Flow maps

Figure 9. Comparisons on *ambush_5*, *cave_4* and *mountain_1* from Sintel: [LEFT] Motion segmentation results – ours, OSF and ground truth. [MIDDLE] Disparity and [RIGHT] Flow maps estimated by our method, PRSM and OSF and the ground truth versions.

References

[1] H. S. Alismail and B. Browning . Direct disparity space: Robust and real-time visual odometry. Technical Report CMU-RI-TR-14-20, Robotics Institute, Pittsburgh, PA, 2014.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int'l Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004.

[3] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. *Int'l Journal of Computer Vision (IJCV)*, pages 1–16, 2012.

[4] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 26(9):1124–1137, 2004.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

[7] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 611–625, 2012.

[8] J. Čech, J. Sanchez-Riera, and R. Horaud. Scene flow estimation by growing correspondence seeds. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3129–3136, 2011.

[9] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] M. Derome, A. Plyer, M. Sanfourche, and G. Le Besnerais. A prediction-correction approach for real-time optical flow computation using stereo. In *Proc. of German Conference on Pattern Recognition*, pages 365–376, 2016.

[11] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2015.

[12] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht. Semi-global matching: a principled derivation in terms of message passing. *Pattern Recognition*, pages 43–53, 2014.

[13] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proc. of the 13th Scandinavian Conference on Image Analysis*, SCIA'03, pages 363–370, 2003.

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[15] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 2276–2282, 2013.

[16] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 30(2):328–341, 2008.

[17] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[18] M. Hornacek, A. Fitzgibbon, and C. Rother. Sphereflow: 6 dof scene flow from rgb-d pairs. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3526–3533, 2014.

[19] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1–7, 2007.

[20] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 98–104, 2015.

[21] M. Jaimez, M. Souiai, J. Stueckler, J. Gonzalez-Jimenez, and D. Cremers. Motion cooperation: Smooth piece-wise rigid scene flow from rgb-d images. In *Proc. of the Int. Conference on 3D Vision (3DV)*, 2015.

[22] P. Kohli and P. H. S. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 29(12):2079–2088, 2007.

[23] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 28(10):1568–1583, 2006.

[24] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, volume 2, pages 508–515 vol.2, 2001.

[25] V. Lepetit, F.Moreno-Noguer, and P.Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int'l Journal of Computer Vision (IJCV)*, 81(2), 2009.

[26] R. Li and S. Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. *Computer Vision and Image Understanding*, 110(1):75–90, 2008.

[27] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Int'l Joint Conf. on Artificial Intelligence*, pages 674–679, 1981.

[28] 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 *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[29] N. Mayer, E. Ilg, P. Hausser, 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 *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015.

[31] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int'l Journal of Computer Vision (IJCV)*, 72(2):179–193, 2007.

[32] J.-P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 597–602, 2003.

[33] J. Quiroga, T. Brox, F. Devernay, and J. Crowley. Dense semi-rigid scene flow estimation from rgbd images. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 567–582, 2014.

[34] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graph.*, 23(3):309–314, 2004.

[35] T. Taniai, Y. Matsushita, and T. Naemura. Graph cut based continuous stereo matching using locally shared labels. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1613–1620, 2014.

[36] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 568–581, 2010.

[37] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 13–26, 2012.

[38] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, volume 2, pages 722–729, 1999.

[39] S. Vedula, S. Baker, P. Rander, R. T. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 27(3):475–480, 2005.

[40] C. Vogel, S. Roth, and K. Schindler. View-consistent 3d scene flow estimation over multiple frames. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 263–278, 2014.

[41] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a rigid motion prior. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1291–1298, 2011.

[42] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1377–1384, 2013.

[43] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *Int'l Journal of Computer Vision (IJCV)*, 115(1):1–28, 2015.

[44] Y. Wang, J. Zhang, Z. Liu, Q. Wu, P. A. Chou, Z. Zhang, and Y. Jia. Handling occlusion and large displacement through improved rgb-d scene flow estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1265–1278, 2016.

[45] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3d motion understanding. *Int'l Journal of Computer Vision (IJCV)*, 95(1):29–51, 2011.

[46] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 34(9):1744–1757, 2012.

[47] Y. Zhang and C. Kambhamettu. On 3d scene flow and structure estimation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–778, 2001.

Fast Multi-frame Stereo Scene Flow with Motion Segmentation
— Supplementary Material —

Tatsunori Taniai
RIKEN AIP

Sudipta N. Sinha
Microsoft Research

Yoichi Sato
The University of Tokyo

In the supplementary material we present details of our SGM stereo and flow implementations (used in Sec. 4.1 and Sec. 4.5) as well as the segmentation ground prior (used in Sec. 4.7) that were omitted from the main paper due to the limit on page length. We also discuss parameter settings and their effects on our method. Note that we review the SGM algorithm as proposed by Hirschmuller [3], but describe the algorithm using our own notation to be consistent with the main paper. We also provide additional qualitative results and comparisons with state-of-the-art methods in the supplementary video.

A. SGM Stereo

In the binocular and epipolar stereo stages (Sec. 4.1 and 4.3), we solve stereo matching problems using the semi-global matching (SGM) algorithm [3]. Here, stereo matching is cast as a discrete labeling problem, where we estimate the disparity map $\mathcal{D}_\mathbf{p} = \mathcal{D}(\mathbf{p}) : \Omega \to D$ (where $D = \{D_{\text{min}}, \cdots, D_{\text{max}}\}$ is the disparity range) that minimizes the following 2D Markov random field (MRF) based energy function.

$$E_{\text{stereo}}(\mathcal{D}) = \sum_{\mathbf{p}\in\Omega} C_\mathbf{p}(\mathcal{D}_\mathbf{p}) + \sum_{(\mathbf{p},\mathbf{q})\in N} cV_\mathbf{pq}(\mathcal{D}_\mathbf{p}, \mathcal{D}_\mathbf{q}). \tag{A1}$$

Here, $C_\mathbf{p}(\mathcal{D}_\mathbf{p})$ is the unary data term that evaluates photo-consistencies between the pixel $\mathbf{p}$ in the left image I^0 and its corresponding pixel $\mathbf{p}' = \mathbf{p} - (\mathcal{D}_\mathbf{p}, 0)^T$ at the disparity $\mathcal{D}_\mathbf{p}$ in the right image I^1. $V_\mathbf{pq}(\mathcal{D}_\mathbf{p}, \mathcal{D}_\mathbf{q})$ is the pairwise smoothness term defined for neighboring pixel pairs $(\mathbf{p}, \mathbf{q}) \in N$ on the 8-connected pixel grid. In SGM, this term is usually defined as

$$V_\mathbf{pq}(\mathcal{D}_\mathbf{p}, \mathcal{D}_\mathbf{q}) = \begin{cases} 0 & \text{if } \mathcal{D}_\mathbf{p} = \mathcal{D}_\mathbf{q} \\ P_1 & \text{if } |\mathcal{D}_\mathbf{p} - \mathcal{D}_\mathbf{q}| = 1 \\ P_2 & \text{otherwise} \end{cases}. \tag{A2}$$

Here, P_1 and P_2 ($0 < P_1 < P_2$) are smoothness penalties. The coefficient c in Eq. (A1) is described later.

While the exact inference of Eq. (A1) is NP-hard, SGM decomposes the 2D MRF into many 1D MRFs along 8 cardinal directions $\mathbf{r}$ and minimizes them using dynamic programming [3]. This is done by recursively updating the following cost arrays $L_\mathbf{r}(\mathbf{p}, d)$ along 1D scan lines in the directions $\mathbf{r}$ from the image boundary pixels.

$$L_\mathbf{r}(\mathbf{p}, d) = C_\mathbf{p}(d) + \min_{d'\in D} [L_\mathbf{r}(\mathbf{p} - \mathbf{r}, d') + V_\mathbf{pq}(d, d')] - \min_{d'\in D} L_\mathbf{r}(\mathbf{p} - \mathbf{r}, d'). \tag{A3}$$

Here, by introducing the following normalized scan-line costs

$$\bar{L}_\mathbf{r}(\mathbf{p}, d) = L_\mathbf{r}(\mathbf{p}, d) - \min_{d'\in D} L_\mathbf{r}(\mathbf{p}, d'), \tag{A4}$$

the updating rule of Eq. (A3) is simplified as follows.

$$L_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in D} \left[\bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V_{\mathbf{pq}}(d, d') \right] \tag{A5}$$

$$= C_{\mathbf{p}}(d) + \min\{\bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, \bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, P_2\} \tag{A6}$$

Then, the scan-line costs by the 8 directions are aggregated as

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d), \tag{A7}$$

from which the disparity estimate at each pixel $\mathbf{p}$ is retrieved as

$$\mathcal{D}_{\mathbf{p}} = \underset{d \in D}{\operatorname{argmin}}\, S(\mathbf{p}, d). \tag{A8}$$

Recently, Drory *et al*. [2] showed that the SGM algorithm is a variant of message passing algorithms such as belief propagation and TRW-T [9] that approximately optimize Eq. (A1). Here, the coefficient c in Eq. (A1) is a scaling factor that accounts for an overweighting effect on the data term during SGM ($c = 1/8$ when using 8 directions) [2].

Drory *et al*. [2] also proposed an uncertainty measure $\mathcal{U}$ that is computed as

$$\mathcal{U}(\mathbf{p}) = \min_{d} \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) - \sum_{\mathbf{r}} \min_{d} L_{\mathbf{r}}(\mathbf{p}, d). \tag{A9}$$

$\mathcal{U}(\mathbf{p})$ is lower-bounded by 0, and becomes 0 when minimizers of 8 individual scan-line costs agree. Since the first and second term in Eq. (A9) are respectively computed in Eqs. (A8) and (A3), the computation of $\mathcal{U}(\mathbf{p})$ essentially does not require computational overhead.

In our implementation of SGM, we use the data term $C_{\mathbf{p}}(\mathcal{D}_{\mathbf{p}})$ defined using truncated normalized cross-correlation in Eq. (5) in the main paper. The smoothness penalties P_1 and P_2 are defined as follows.

$$P_1 = \lambda_{\text{sgm}} / |\mathbf{p} - \mathbf{q}| \tag{A10}$$

$$P_2 = P_1 \left(\beta + \gamma w^{\text{col}}_{\mathbf{pq}} \right) \tag{A11}$$

Here, $w^{\text{col}}_{\mathbf{pq}}$ is the color edge-based weight used in Eq. (15) and we use parameters of $(\lambda_{\text{sgm}}, \beta, \gamma) = (200/255, 2, 2)$. The disparity range is fixed as $\{D_{\text{min}}, \cdots, D_{\text{max}}\} = \{0, \cdots, 255\}$ for the original image size of KITTI (since we downscale the images by a factor of 0.65, the disparity range is also downscaled accordingly). We also set the confidence threshold τ_{u} for the uncertainty map $\mathcal{U}$ to 2000 by visually inspecting $\mathcal{U}(\mathbf{p})$.

B. SGM Flow

We have extended the SGM algorithm for our optical flow problem in Sec. 4.5. Here, we estimate the flow map $\mathcal{F}_{\mathbf{p}} = \mathcal{F}(\mathbf{p}) : \Omega \rightarrow R$ (where $R = ([u_{\text{min}}, u_{\text{max}}] \times [v_{\text{min}}, v_{\text{max}}])$ is the 2D flow range) by minimizing the following 2D MRF energy.

$$E_{\text{flow}}(\mathcal{F}) = \sum_{\mathbf{p} \in \Omega} C'_{\mathbf{p}}(\mathcal{F}_{\mathbf{p}}) + \sum_{(\mathbf{p}, \mathbf{q}) \in N} c V'_{\mathbf{pq}}(\mathcal{F}_{\mathbf{p}}, \mathcal{F}_{\mathbf{q}}). \tag{A12}$$

Similarly to SGM stereo, we use the NCC-based matching cost of Eq. (5) for the data term $C'_{\mathbf{p}}(\mathcal{F}_{\mathbf{p}})$ to evaluate matching photo-consistencies between I^0_t and I^0_{t+1}. We also define the smoothness term as

$$V'_{\mathbf{pq}}(\mathcal{F}_{\mathbf{p}}, \mathcal{F}_{\mathbf{q}}) = \begin{cases} 0 & \text{if } \mathcal{F}_{\mathbf{p}} = \mathcal{F}_{\mathbf{q}} \\ P_1 & \text{if } 0 < \|\mathcal{F}_{\mathbf{p}} - \mathcal{F}_{\mathbf{q}}\| \leq \sqrt{2} \\ P_2 & \text{otherwise} \end{cases} . \tag{A13}$$

Since we use integer flow labels, the second condition in Eq. (A13) is equivalent to saying that the components of the 2D vectors $\mathcal{F}_{\mathbf{q}} = (u_{\mathbf{q}}, v_{\mathbf{q}})$ and $\mathcal{F}_{\mathbf{p}} = (u_{\mathbf{p}}, v_{\mathbf{p}})$ can at-most differ by 1. We use the same smoothness penalties $\{P_1, P_2\}$ and the parameter settings with SGM stereo.

The optimization of Eq. (A12) is essentially the same with SGM stereo, but the implementation of updating scan-line costs in Eq. (A3) was extended to handle the new definition of the pairwise term $V'_{\mathbf{pq}}$. Therefore, Eq. (A6) is modified using a flow label $\mathbf{u} = (u, v) \in R$ as follows.

$$L_{\mathbf{r}}(\mathbf{p}, \mathbf{u}) = C_{\mathbf{p}}(\mathbf{u}) + \min\{\bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, \mathbf{u}), \bar{L}_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, \mathbf{u} + \Delta_{\pm 1}) + P_1, P_2\} \tag{A14}$$

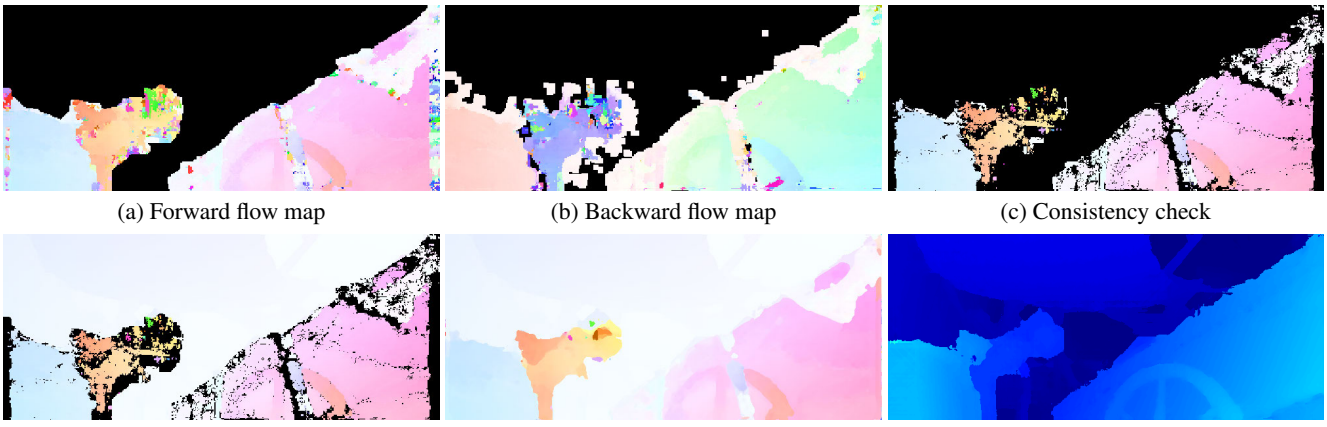
Here, $(\mathbf{u} + \Delta_{\pm 1})$ is enumeration of 8 labels neighboring to $\mathbf{u}$ in the 2D flow space.

C. Refinement of Flow Maps

In the optical flow stage of Sec. 4.5, we refine flow maps using consistency check and weighted median filtering. Similar schemes are commonly employed in stereo and optical flow methods such as [10, 4, 5]. Below we explain these steps.

We first estimate the forward flow map $\mathcal{F}^0$ (from I^0_t to I^0_{t+1}) by SGM for only the foreground pixels of the initial segmentation $\tilde{\mathcal{S}}$ such as shown in Fig. A1 (a). Then, using this flow $\mathcal{F}^0$ and the mask $\tilde{\mathcal{S}}$, we compute a mask in the next image I^0_{t+1} and estimate the backward flow map $\mathcal{F}^1$ (from I^0_{t+1} to I^0_t) for those foreground pixels. This produces a flow map such as shown in Fig. A1 (b). We filter out outliers in $\mathcal{F}^0$ using bi-directional consistency check between $\mathcal{F}^0$ and $\mathcal{F}^1$ to obtain a flow map with holes (Fig. A1 (c)), whose background is further filled by the rigid flow $\mathcal{F}_{\text{rig}}$ (see Fig. A1 (d)). Finally, weighted median filtering is applied for the hole pixels followed by median filtering for all foreground pixels to obtain the non-rigid flow estimate such as shown in Fig. A1 (e).

At the final weighted median filtering step, the filter kernel $\omega^{\text{geo}}_{\mathbf{pq}} = e^{-d_{\mathbf{pq}}/\kappa_{\text{geo}}}$ is computed using geodesic distance $d_{\mathbf{pq}}$ on



(a) Forward flow map (b) Backward flow map (c) Consistency check

(d) Background filling (e) Weighted median filtering (f) Disparity map as guidance image

Figure A1. Process of flow map refinement.

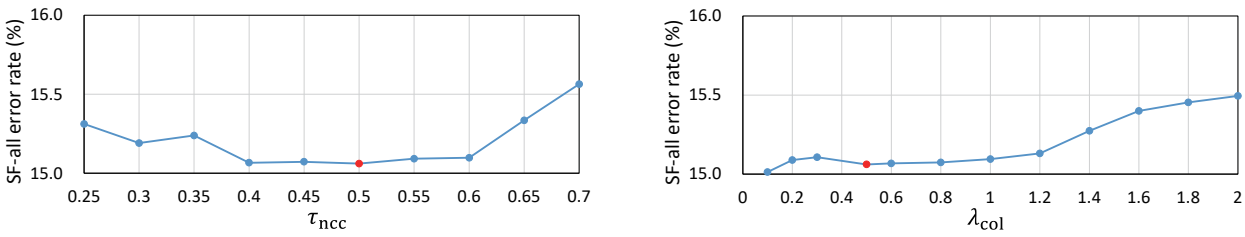


Figure A2. Profiles of scene flow accuracies with reference to parameters τ_{ncc} (left) and λ_{col} (right). The error rates are evaluated on 200 training sequences from KITTI. The scores with the default parameter settings are colored by red.

the disparity map $\mathcal{D}$ (Fig. A1 (f)) as the guidance image. For this, we define the distance between two adjacent pixels as

$$\text{dist}(\mathbf{p}_1, \mathbf{p}_2) = |\mathcal{D}(\mathbf{p}_1) - \mathcal{D}(\mathbf{p}_2)| + \|\mathbf{p}_1 - \mathbf{p}_2\|/100. \tag{A15}$$

The geodesic distance $d_{\mathbf{pq}}$ is then computed for the pixels in the filter window $\mathbf{q} \in W_{\mathbf{p}}$ as the cumulative shortest-path distance from $\mathbf{q}$ to the center pixel $\mathbf{p}$. This is efficiently computed using an approximate algorithm [8]. We use the filter window $W_{\mathbf{p}}$ of 31×31 size and $\kappa_{\text{geo}} = 2$. The subsequent (constant-weight) median filtering further reduces outliers [7], for which we use the window of 5×5 size.

D. Segmentation Ground Prior

The segmentation ground prior term mentioned in Sec. 4.7 is computed as follows. First, we detect the ground plane from the disparity map $\mathcal{D}(\mathbf{p})$. We use RANSAC to fit a disparity plane $[d = au + bv + c]$ defined on the 2D image coordinates. Here, we assume that the cameras in the stereo rig are upright. Therefore, during RANSAC we choose disparity planes whose b is positive and high and $|a|$ is relatively small. Then, we compute the disparity residuals between $\mathcal{D}$ and the ground plane as $r_{\mathbf{p}} = |\mathcal{D}_{\mathbf{p}} - (a\mathbf{p}_u + b\mathbf{p}_v + c)|$, where (a, b, c) are the obtained plane parameters. Our ground prior as a cue of background is then defined as follows.

$$C^{\text{gro}}_{\mathbf{p}} = \lambda_{\text{gro}} \Big(\min(r_{\mathbf{p}}, \tau_{\text{gro}})/\tau_{\text{gro}} - 1 \Big) \tag{A16}$$

When $r_{\mathbf{p}} = 0$, $C^{\text{gro}}_{\mathbf{p}}$ strongly favors background, and when $r_{\mathbf{p}}$ increases to τ_{gro}, it becomes 0. The thresholding value $\tau_{\mathbf{p}}$ is set to $0.01 \times D_{\text{max}}$. We use $\lambda_{\text{gro}} = 10$.

E. Parameter Settings

In this section, we explain our strategy of tuning parameters and also show effects of some parameters. Most of the parameters can be easily interpreted and tuned, and our method is fairly insensitive to parameter settings.

For example, the effects of the threshold τ_{u} for the uncertainty map $\mathcal{U}$ (Sec. 4.1), the threshold τ_w for the patch-variance weight $\omega^{\text{var}}_{\mathbf{p}}$ (Sec. 4.4), and κ_3 of the image edge-based weight $\omega^{\text{str}}_{\mathbf{pq}}$ (Sec. 4.4) can be easily analyzed by direct visualization as shown in Figure 3 (b), Figures 4 (b) and (e).

The parameters of SGM (discussed in Sec. A) can be tuned independently from the whole algorithm.

For the weights $(\lambda_{\text{ncc}}, \lambda_{\text{flo}}, \lambda_{\text{col}}, \lambda_{\text{potts}})$ in Sec. 4.4, we first tuned $(\lambda_{\text{ncc}}, \lambda_{\text{flo}}, \lambda_{\text{potts}})$ on a small number of sequences. Since the ranges of the NCC appearance term (Eq. (11)) and flow term (Eq. (12)) are limited to $[-1, 1]$, they are easy to interpret. Then, we tuned λ_{col} of the color term (Eq. (14)). Here, $\lambda_{\text{potts}}/\lambda_{\text{col}}$ is known to be usually around 10 - 60 from previous work [6, 1].

Even though we fine-tuned τ_{ncc} and λ_{col} for Sintel, they are insensitive on KITTI image sequences. We show the effects of these two parameters for KITTI training sequences in Figure A2. The threshold τ_{ncc} for NCC-based matching costs was adjusted for Sintel because its synthesized images have lesser image noise compared to real images of KITTI. Also, the

weight λ_{col} was adjusted for Sintel, to increase the weight on the prior color term (Sec. 4.7). For Sintel sequences, sometimes moving objects stop moving on a few frames and become stationary momentarily. In such cases, increasing λ_{col} improves the temporal coherence of the motion segmentation results. In the future we will improve the scheme for online learning of the prior color models, which will improve temporal consistency of motion segmentation and also will make λ_{col} more insensitive to settings.

References

[1] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

[2] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht. Semi-global matching: a principled derivation in terms of message passing. *Pattern Recognition*, pages 43–53, 2014.

[3] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 30(2):328–341, 2008.

[4] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 35(2):504–511, 2013.

[5] C. R. Michael Bleyer and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proc. of British Machine Vision Conf. (BMVC)*, pages 14.1–14.11, 2011.

[6] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graph.*, 23(3):309–314, 2004.

[7] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439, 2010.

[8] P. J. Toivanen. New geodesic distance transforms for gray-scale images. *Pattern Recogn. Lett.*, 17(5):437–450, 1996.

[9] M. Wainwright, T. Jaakkola, and A. Willsky. Map estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. *IEEE Trans. on Information Theory*, 51:3697–3717, 2002.

[10] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (wmf). In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2830–2837, 2014.