Completed Dense Scene Flow in RGB-D Space

Yucheng Wang[1,3], Jian Zhang[1], Zicheng Liu[2], Qiang Wu[1], Philip Chou[2], Zhengyou Zhang[2], and Yunde Jia[3]

[1] Advanced Analytics Institute, University of Technology, Sydney
[2] Microsoft Research, Redmond, WA, USA
[3] Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology
yucheng.wang@student.uts.edu.au

Abstract. Conventional scene flow containing only translational vectors is not able to model 3D motion with rotation properly. Moreover, the accuracy of 3D motion estimation is restricted by several challenges such as large displacement, noise, and missing data (caused by sensing techniques or occlusion). In terms of solution, there are two kinds of approaches: local approaches and global approaches. However, local approaches can not generate smooth motion field, and global approaches is difficult to handle large displacement motion. In this paper, a completed dense scene flow framework is proposed, which models both rotation and translation for general motion estimation. It combines both a local method and a global method considering their complementary characteristics to handle large displacement motion and enforce smoothness respectively. The proposed framework is applied on the RGB-D image space where the computation efficiency is further improved. According to the quantitative evaluation based on Middlebury dataset, our method outperforms other published methods. The improved performance is further confirmed on the real data acquired by Kinect sensor.

1 Introduction

Dense scene flow (3D motion) estimation is a challenging research task in computer vision. Consumer RGB-D cameras like Kinect, which provide relatively reliable depth information, promote a trend to estimate scene flow from RGB-D data. Unlike most conventional scene flow methods [1–5] generating only translation vectors, completed scene flow methods can acquire both rotation and translation information, which is more favorable for two main reasons. The first reason is that it can model the general 3D rotational motion in the physical world. The second reason is that it provides abundant temporal information for high-accuracy vision tasks (e.g. 3D reconstruction).

However, it is very challenging to estimate completed scene flow from RGB-D data. The first challenging problem is that large displacement motion. Large displacement motion often indicates the searching dimension and range for scene flow are both large. Without good initial or candidate values, it is difficult to obtain accurate and robust estimates. The second problem is that there usually exists noises and missing data in the captured RGB-D data. The RGB-D data

RGB-D scene flow	Conventional Scene Flow	Completed Scene Flow
Local method	Hadfield and Bowden [1]	Hornáček *et al.* [10]
Global method	Gottfried *et al.* [2]	Our work
	Herbst *et al.* [3]	
	Quiroga *et al.* [4]	
	Zhang *et al.* [5]	

Table 1. Classification for scene flow according to conventional or completed scene flow modeling, and local or global approach employed in the method.

may be affected or even disappeared from the reference image to the target image.

Currently, the solutions for motion estimation can be divided into two types: local approaches and global approaches. Local approaches only focus on feature consistency between the corresponding points (or their *local* supporting areas) on two neighboring frames on the time domain. Some local approaches [6–8] can address the displacement issue, since they can employ a random search strategy. However, they can not generate very accurate and smooth motion field. Global approaches are able to further consider the spatial relation of all points in the image, such as occlusion and smoothness. Since global approaches model the complex spatial relation, a limitation is that they often trap into local minima and require good initial values to achieve accurate performance [9].

In this paper, we propose a new scene flow estimation framework to address these challenging issues. Different from previous methods, our framework fully combines the complementary advantages of a local method and a global method, and avoid their corresponding drawbacks. The local method is utilized to provide good candidate values for the global method to overcome large displacement motion. The global method combines these candidate values by explicitly modeling occlusion and enforcing smoothness for good-quality results. In addition, we further handle the missing data issue caused by sensing techniques and occlusion. Our contributions can be summarized as follows: (1) We present a framework to combine the advantages of local and global approaches, i.e. handling large displacement and enforcing smoothness, respectively. (2) We give a new formulation of scene flow estimation that is able to further handle missing data caused by various reasons. (3) We propose compute the matching cost for each point in a 3D local supporting area with adaptive weights, which is more robust to noise. (4) We convert 2D motion as initial values and reduce the searching dimension in the optimization, which improves the accuracy and efficiency.

2 Related Work

Scene flow is 3D motion in the physical world. Compared with optical flow, scene flow has view-independent characteristics, which is preferred in many vision

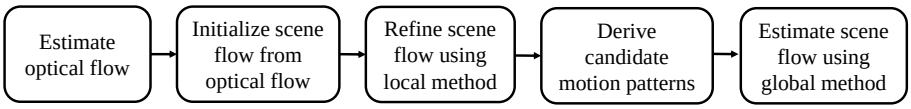

Fig. 1. Framework overview.

applications like action recognition [11]. We refer the readers to optical flow [12] and scene flow literatures [13] for more details about the similarity and difference.

Most scene flow methods [14–16] employ only RGB stereo images. Until recent years, some RGB-D image-based methods have emerged thanks to the development of consumer RGB-D cameras. The classification for existing scene flow methods are shown in Table 1. Zhang *et al.* [5] proposed a two-step framework consisting a global optimization and a bilateral filtering to compute scene flow. Hadfield and Bowden [1] estimated the scene flow using a particle filtering technique. Gottfried *et al.* [2] presented an extended optical flow framework for the estimation of range flow fields from RGB-D video sequences captured by Kinect. Herbst *et al.* [3] presented a variational method for dense 3D motion estimation for rigid motion segmentation. Quiroga *et al.* [4] solved the scene flow problem in a variational framework combining local and global constraints.

These conventional scene flow methods only employ translation in the motion modeling and are not able to handle large displacement motion, since large displacement motion usually contains complex components including rotation and translation. Completed scene flow of rotation and translation information can model 3D motion better and generate more precise results than using translation only. Recently, Hornáček *et al.* [10] proposed a completed scene flow method. However, this method estimated the scene flow relying heavily on a local method, which may introduce error in occlusion detection and can not generate very accurate motion field. Our framework only derive good initial and candidate values from the local method, and estimate the scene flow with explicitly modeling the occlusion and smoothness in the global method.

3 Our Framework

Figure 1 shows the overview of the framework. Our framework starts with a local optical flow named NRDC [17] to generate completed optical flow from RGB image pair. We transform the optical flow into scene flow as good initial values for our local scene method. Based on the initial values, our local method combine cross-modal RGB color and depth information to refine the scene flow. Next, we derive a set of candidate motion patterns from the local scene flow results. Finally, the set of candidate motion patterns are fused by further modeling occlusion and enforcing 3D smoothness in a global approach. Details will be given in following sections.

Given two RGB-D images $\{I, D\}$ and $\{I', D'\}$, we aim to compute motion from the reference image $\{I, D\}$ to the target image $\{I', D'\}$. Each pixel $\mathbf{p}$ in the

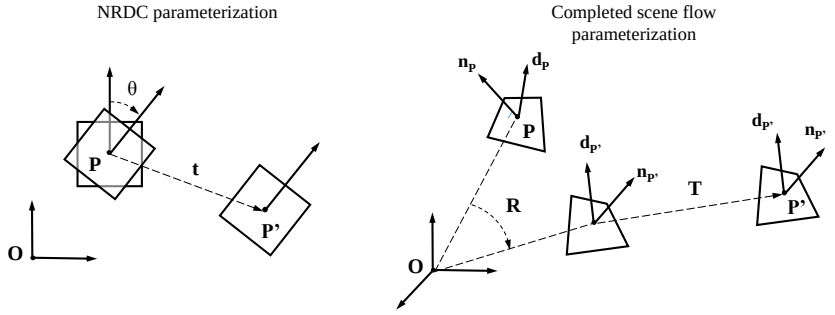

Fig. 2. NRDC and complete scene flow parameterization. To clearly see the rotation change for a point, we use a square patch with a principal direction vector to represent the point.

reference image has RGB color $I(\mathbf{p})$ and depth $D(\mathbf{p})$. A pixel $\mathbf{p}$ is considered to be valid if its depth value is provided in the depth map. Thus, each valid pixel $\mathbf{p}$ can be deemed as a 3D point $\mathbf{P}$ with color information in the scene.

The 3D coordinate of $\mathbf{P} = \{X_\mathbf{P}, Y_\mathbf{P}, Z_\mathbf{P}\}$ is compute by back-projecting $\mathbf{p}$ using its depth value $D(\mathbf{p})$ and intrinsic camera parameters $\mathbf{K}$ using $\mathbf{P} = \Pi^{-1}(\mathbf{p}) = D(\mathbf{p}) \cdot \mathbf{K}^{-1}\tilde{\mathbf{p}}$, and vice versa $\mathbf{p} = \Pi(\mathbf{P})$. Here, Π is the projection operation, while Π^{-1} means back-projection operation. Let $\mathbf{V} = (\mathbf{R}, \mathbf{T}) \in SE(3)$ denotes a 6-DoF (Degree of Freedom) motion in 3D, where $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$. This completed scene flow is employed in our framework. Our goal is to assign such a 6-DoF scene flow $\mathbf{V_P}$ to each point $\mathbf{P}$ in the reference RGB-D image. The predicted 3D position of point $\mathbf{P}$ with motion $\mathbf{V_P} = (\mathbf{R_P}, \mathbf{T_P})$ denotes $\mathbf{P}' = \mathbf{V_P}(\mathbf{P}) = \mathbf{R_P}\mathbf{P} + \mathbf{T_P}$.

3.1 Initialization from 2D Optical Flow

Some 2D optical flow methods deal with large displacement on the image plane, since the search dimension is smaller than the scene flow situations. Thus, we choose an efficient method named NRDC [17] to generate initial values. N-RDC can generate 2D motion field includes 2-DoF translational vectors $\mathbf{t_p}$ (see Figure 2). However, the required motion parameters for our method is $\mathbf{V_P} = \{\mathbf{R_P}, \mathbf{T_P}\}$. We give a simple approach to enable the conversion from 2D motion field into 3D completed scene flow.

In order to compute rotation matrix, we intuitively define each point $\mathbf{P}$ having corresponding 2D and 3D principal directions. 3D principal direction $\mathbf{d_P}$ on the 3D object surface which is orthogonal to its normal $\mathbf{n_P}$, and 2D principal direction is the projection of 3D principal direction on the image plane. Inspired by [18], we adopt the prominent orientation in SIFT feature detection [19] as the 2D principal directions, i.e. $[\sin(\theta_\mathbf{P}), \cos(\theta_\mathbf{P})]$ for the point $\mathbf{P}$ and $[\sin(\theta_{\mathbf{P}'}), \cos(\theta_{\mathbf{P}'})]$ for the point $\mathbf{P}'$. According to our definition, 3D principal

direction vectors can be then computed by

$$\mathbf{d_P} = \text{orthonorm}([\sin(\theta_\mathbf{P}), \cos(\theta_\mathbf{P}), 0]^\text{T}, \mathbf{n_P}), \tag{1}$$

$$\mathbf{d_{P'}} = \text{orthonorm}([\sin(\theta_{\mathbf{P}'}), \cos(\theta_{\mathbf{P}'}), 0]^\text{T}, \mathbf{n_{P'}}), \tag{2}$$

where $\text{orthonorm}(\cdot,\cdot)$ is the Gram-Schmidt orthonormalization procedure. The rotation variation of a point is reflected by the variations of its normal and principal directions: $\mathbf{n_{P'}} = \mathbf{R_P n_P}$ and $\mathbf{d_{P'}} = \mathbf{R_P d_{P'}}$. Thus, we can calculate the 3D rotation matrix $\mathbf{R_P}$ of the point $\mathbf{P}$ by

$$\mathbf{R_P} = [\mathbf{n_{P'}}, \mathbf{d_{P'}}, \mathbf{n_{P'}} \times \mathbf{d_{P'}}] \cdot [\mathbf{n_P}, \mathbf{d_P}, \mathbf{n_P} \times \mathbf{d_P}]^{-1}. \tag{3}$$

Once the rotation is obtained, the translational vector of the point $\mathbf{P}$ can also be simply computed by

$$\mathbf{T_P} = \mathbf{P'} - \mathbf{R_P} \cdot \mathbf{P}. \tag{4}$$

3.2 Refinement using Local Method

The motion initial value from the optical flow only concern RGB color information. Thus, the major principle for optimizing the local scene flow estimates is that multi-modal RGB-D features (descriptors) consistency for a point in the reference image and its corresponding position in the target image. To address the noise and missing data issue, for a point $\mathbf{P}$ with a motion $\mathbf{V_P}$, we aggregate cost values of points using adaptive weights and reliability in a corresponding 3D supporting area. Our goal in the local is to reduce overall matching cost of all the points in the reference image:

$$E_\text{local}(\mathbf{V}) = \sum_\mathbf{P} C_\text{local}(\mathbf{P}, \mathbf{V_P}). \tag{5}$$

where $C_\text{local}(\mathbf{P}, \mathbf{V_P})$ is the 3D supporting patch-based matching cost for the point $\mathbf{P}$ with the motion $\mathbf{V_P}$, and it is defined by

$$C_\text{local}(\mathbf{P}, \mathbf{V_P}) = \frac{\sum\limits_{\mathbf{Q}\in S(\mathbf{P})} \omega(\mathbf{P},\mathbf{Q}) \cdot R(\mathbf{Q}) \cdot R'(\mathbf{Q}') \cdot C(\mathbf{Q}, \mathbf{V_P})}{\sum\limits_{\mathbf{Q}\in S(\mathbf{P})} \omega(\mathbf{P},\mathbf{Q}) \cdot R(\mathbf{Q}) \cdot R'(\mathbf{Q}')}, \tag{6}$$

where $S(\mathbf{P})$ is the 3D supporting area for the point $\mathbf{P}$, $\omega(\mathbf{P},\mathbf{Q})$ is the weighting function which gives the probability of points $\mathbf{P}$ and $\mathbf{Q}$ on a same surface, R and R' are the reliability maps for the reference and target RGB-D image respectively, and $C(\mathbf{Q}, \mathbf{V_P})$ is the point-based matching cost for the point $\mathbf{P}$ with the motion $\mathbf{V_P}$.

3D Supporting Area Due to noises on RGB-D data, the features (descriptors) of a single point is usually unstable. To deal with noises, we assume local rigidity for each point, and aggregate cost values of local neighboring points on the same surface for a robust matching cost. Unlike [10], 3D geodesic distance is a better choice to judge whether 3D neighboring points are on a same surface than Euclidean distance. However, it is expensive to compute geodesic distance between all the points. We propose a new 3D patch representation as an approximation by using the normal information $\mathbf{n_P}$, which is capable of selecting such neighboring points on the same surface for a point. Our basic observation is that if $\mathbf{Q}$ is a neighboring point of $\mathbf{P}$ and they are on a same surface, the value of $(\mathbf{P}-\mathbf{Q})\cdot\mathbf{n_P}$ should close to 0.

Given the 3D world coordinates of a point $\mathbf{P}=\{X_\mathbf{P}, Y_\mathbf{P}, Z_\mathbf{P}\}$. Thus, the supporting patch of a 3D point $\mathbf{P}$ can be expressed as the set of the neighboring points satisfying

$$S(\mathbf{P}) = \{\mathbf{Q} \mid ||\mathbf{P}-\mathbf{Q}||_2 < \epsilon \cdot Z_\mathbf{P} \wedge (\mathbf{P}-\mathbf{Q})\cdot\mathbf{n_P} < \delta \cdot Z_\mathbf{P}\} \tag{7}$$

where ϵ is a threshold ratio using in the previous normal estimation, and δ is usually a small threshold ratio decided by the sensor noise.

Weighting Function In the supporting area of a point $\mathbf{P}$, its neighboring points $\mathbf{Q} \in S(\mathbf{P})$ should have higher probability if they are closer in the 3D space. Thus, we utilize an adaptive weight $\omega(\mathbf{P},\mathbf{Q})$ based on Euclidean distance to aggregate the cost values of neighboring points $\mathbf{Q}$ in the support area $S(\mathbf{P})$. The weighting function is

$$\omega(\mathbf{P},\mathbf{Q}) = \exp(-||\mathbf{P}-\mathbf{Q}||_2/\gamma). \tag{8}$$

where γ is a parameter to control the weight function.

Reliability Map Considering the fact that the depth channel of RGB-D data often contains missing data and noises, we introduce reliability of each pixel (point) in the RGB-D data. We observe that depth noise often occur predominantly near depth discontinuities. Therefore we apply an edge detector on the depth map, and use the 2D spatial distance $\rho_\mathbf{p}$ to the closest depth edge as a reliability measure for $\mathbf{p}$. The reliability of $\mathbf{p}$ is

$$R(\mathbf{p}) = \begin{cases} \exp(-\frac{2\cdot\min(\rho_{\max}-\rho_\mathbf{p},0)}{\rho_{\max}}) & \text{if } D(\mathbf{p}) \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\rho_{\max}$ are constant scaling parameters.

Point-based Matching Cost Given a point in $\mathbf{P}$ in the reference image and a 6-DoF motion $\mathbf{V_P}$, the motion is of high probability for this point if we can find a position with similar appearance and geometrical information in the target

image. We assume brightness constancy and use color difference $||I(\mathbf{p})-I'(\mathbf{p}')||_2$ to measure appearance similarity. For geometrical similarity, we use difference of depth values $||Z_{\mathbf{P}'} - D'(\mathbf{p}')||_2$ as an approximation of 3D Euclidean distance. The matching cost of one single point $\mathbf{P}$ with motion $\mathbf{V_P} = \{\mathbf{R_P}, \mathbf{T_P}\}$ is defined as

$$C(\mathbf{P}, \mathbf{V_P}) = ||I(\mathbf{p}) - I'(\mathbf{p}')||_2 + \alpha \cdot ||Z_{\mathbf{P}'} - D'(\mathbf{p}')||_2. \tag{10}$$

where α is the parameter to control the ratio of the two components.

3D Searching We modify the 2D PatchMatch method for our 3D scene flow case, due to its good characteristic to handle large displacement. Firstly, each point is assigned with the initial value provided by the local optical flow method. Next, we iteratively carry out two steps to refine the motion estimates for each point, i.e. spatial propagation and random search. In the spatial propagation, we use 6-DoF completed scene flow instead of 2-DoF translational optical flow. In the random search, the searching dimension is too large to efficiently obtain good results. We introduce a reduced-DoF random search by only generating a random 2-DoF translation $\mathbf{t_p}$. We compute the 2D principal direction vectors of $\mathbf{p}$ in the reference image and $\mathbf{p}+\mathbf{t_p}$ in the target image by adopting the prominent orientations in SIFT feature detection [19]. Then, the following computation is similar with the situation when we convert 2D motion field to 3D completed scene flow in the section *Initialization from Optical flow*. We can finally acquire a 6-DoF motion from a 2-DoF random guess using this reduced-DoF random search. Thus, the dimension of random searching for 3D scene flow case is then significantly reduced from six to two.

3.3 Estimation using Global Method

In spite of feature consistency assumption in the local method, we can further explicitly model the occlusion and enforce 3D smoothness in the global approach. The energy function of the global scene flow is

$$E_{\text{global}}(\mathbf{V}) = \sum_{\mathbf{P}} C_{\text{global}}(\mathbf{P}, \mathbf{V_P}) + \lambda \sum_{\mathbf{P},\mathbf{Q}} S_{\text{global}}(\mathbf{P}, \mathbf{Q}, \mathbf{V_P}, \mathbf{V_Q}) \tag{11}$$

where $\mathbf{Q} \in S(\mathbf{P}) \cap N(\mathbf{P})$, $S(\mathbf{P})$ is the set of points in the supporting patch of $\mathbf{P}$, and $N(\mathbf{P})$ is the set of 4(8) connected neighboring points on the image plane, $C_{\text{global}}(\mathbf{P}, \mathbf{V_P})$ is used for feature consistency, and $S_{\text{global}}(\mathbf{P}, \mathbf{Q}, \mathbf{V_P}, \mathbf{V_Q})$ promotes the 3D smoothness of the motion field. Note that C_{global} is different from C_{local} by further modeling occlusion.

Robust Matching Cost with Occlusion Modeling To address the occlusion issue, we incorporate the occlusion in our matching cost computation. We deem the occluded points as outliers when finding correspondence, and use a constant cost value for matching outliers. The robust matching cost in the global method is

$$C_{\text{global}}(\mathbf{P}, \mathbf{V_P}) = (1 - O(\mathbf{P})) \cdot C_{\text{local}}(\mathbf{P}, \mathbf{V_P}) + O(\mathbf{P}) \cdot \xi \tag{12}$$

where ξ is set to be a minimal value for matching outliers, and $O(\mathbf{P})$ is the occlusion status of the point $\mathbf{P}$.

Points in the target image are occluded if there exist other points in front of them from the camera view. Previous methods [5, 10] usually estimate the motion without considering occlusion first, then use consistency check to detect occlusion, and refine the motion in occluded region as a postprocessing. This may introduce error if the estimated motion is incorrect. Our method can explicitly model the occlusion using the depth order, since occlusion relationship has been directly reflected in depth values. For robust performance, we also consider depth noises in the occlusion modeling. We assume the depth noises following a Gaussian distribution with mean zero and standard derivation σ. The occlusion status of a point is defined as

$$O(\mathbf{P}) = [Z_{\mathbf{P}'} > D(\mathbf{P}') + 3 \cdot \sigma(\mathbf{p})], \tag{13}$$

where $[\cdot]$ is the Iverson bracket which denotes a number that is 1 if the condition in square brackets is satisfied, and 0 otherwise, the computation of $\sigma(\mathbf{p})$ is given in the experiment section.

3D Smoothness Instead of enforcing smoothness only on the translation vectors, we apply a 3D smoothness considering both translation and rotation. The basic idea is to promote one point to have similar 3D positions after applying the motion of itself or its neighbors. Thus, the smoothness term is the energy function can be expressed as

$$S_{\text{global}}(\mathbf{P}, \mathbf{Q}, \mathbf{V_P}, \mathbf{V_Q}) = \omega(\mathbf{P}, \mathbf{Q}) \cdot \left(||\mathbf{V_P}(\mathbf{P}) - \mathbf{V_Q}(\mathbf{P})||_2^2 + ||\mathbf{V_P}(\mathbf{Q}) - \mathbf{V_Q}(\mathbf{Q})||_2^2 \right) \tag{14}$$

where $\mathbf{Q} \in S(\mathbf{P}) \cap N(\mathbf{P})$, $S(\mathbf{P})$ is the set of points in the supporting patch of $\mathbf{P}$, and $N(\mathbf{P})$ is the set of 4 connected neighboring points on the image plane, $||\mathbf{V_P}(\mathbf{P}) - \mathbf{V_Q}(\mathbf{P})||_2$ is the Euclidean distance of the point $\mathbf{P}$ with motion patterns $\mathbf{V_P}$ and $\mathbf{V_Q}$, and $||\mathbf{V_P}(\mathbf{Q}) - \mathbf{V_Q}(\mathbf{Q})||_2$ is the Euclidean distance of the point $\mathbf{Q}$ with motion patterns $\mathbf{V_P}$ and $\mathbf{V_Q}$.

Optimization Given Eq. 12 and 14, we minimize our energy function in Eq. 11 via the FusionMoves [20] method using QPBO [21]. The FusionMoves can efficiently combine two proposal labelings (candidates) in a theoretically sound way, which is in practice often globally optima. The key of achieving good results is to generate high-quality motion proposals for FusionMoves. One direct way is to use existing motion pattern directly from the result of our local method. However, the number of different motion patterns in the local result of is usually very limited. Thus, we not only include the motion patterns from the local result as proposals, but also add some random slight perturbation on them as new proposals. The perturbation can be combinations of changing the translation by jumping to its neighboring points (3 DoF), altering the rotation axis (2 DoF), or modifying the rotation angle (1 DoF). The algorithm stops when energy change in a period is less than a threshold, and outputs the final result.

Methods	Venus			Cones			Teddy		
	RMS_O	RMS_Z	AAE	RMS_O	RMS_Z	AAE	RMS_O	RMS_Z	AAE
Brox2011 [9]	0.72	0.14	1.28	2.83	1.75	0.39	3.20	0.47	0.39
Xu2012 [22]	0.30	0.22	1.43	1.66	1.15	**0.21**	1.70	0.50	0.28
Huguet2007 [16]	0.31	N/A	0.98	1.10	N/A	0.69	1.25	N/A	0.51
Basa2013 [15]	0.16	N/A	1.58	0.58	N/A	0.39	0.57	N/A	1.01
Zhang2013 [5]	**0.15**	N/A	1.15	1.04	N/A	0.69	0.73	N/A	0.66
Quiroga2013 [4]	0.31	**0.00**	1.26	0.57	0.05	0.42	0.69	0.04	0.71
Hadfield2014 [1]	0.36	0.02	1.03	1.24	0.06	1.01	0.83	0.03	0.83
Hornáček2014 [10]	0.26	0.02	**0.53**	0.54	0.02	0.52	**0.35**	0.01	**0.15**
NRDC [17]	5.65	N/A	16.2	15.5	N/A	18.3	17.7	N/A	14.3
Our local SF	3.35	0.27	14.5	7.91	1.29	7.10	11.4	0.30	10.9
Our global SF	**0.15**	**0.00**	1.17	**0.33**	**0.00**	0.39	0.40	**0.00**	0.50

Table 2. The evaluated errors of compared methods.

4 Experimental Results

To analyze the performance of the proposed method, we apply our algorithm on the Middlebury dataset and some challenging RGB-D images captured by Kinect cameras as a complement. We use millimeter as the unit of distance measure, and $[0, 255]$ for color range. For the local optical flow NRDC [17], we use its default parameters. The threshold ratio ϵ for normal estimation is set to 0.05, the threshold ratio δ for 3D supporting area is 0.02, the ratio α in cost computation for a singe point is set to 1.0, the constant cost ξ for outliers is set to 30.0, the standard derivation of sensor noise $\sigma(\mathbf{p}) = k \cdot D(\mathbf{p})^2$, and the constant used in computing the weight of two point $\gamma = 10.0$. For Middlebury dataset, the ratio in the global optimization $\lambda = 100$, the parameters to model data reliability $\rho_{\max} = 2$, and the parameters to model depth noise $k = 1.5 \times 10^{-4}$. For Kinect RGB-D data, $\lambda = 1$, $\rho_{\max} = 4$ and $k = 1.5 \times 10^{-5}$.

4.1 Middlebury Dataset

We accordingly test the method on Middlebury dataset following Huguet and Devernay [16] in order to perform quantitative evaluation. The RGB-D images are captured by a set of cameras which are parallel and equally spaced along the X axis at the same time. The motion along Y and Z axis is always zero, and the ground truth of motion along X axis can be obtained from corresponding disparity, which is also available in the Middlebury dataset. We take the color

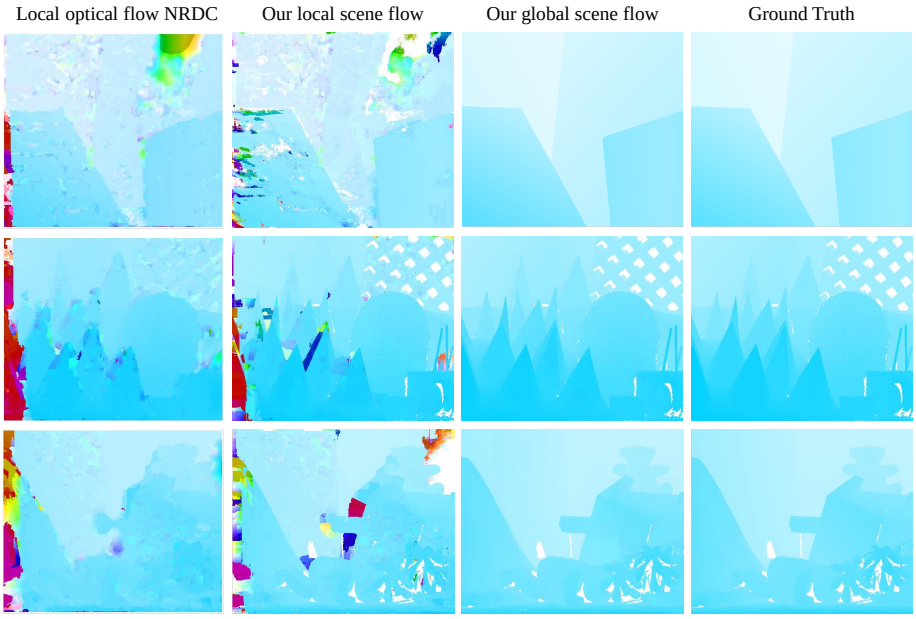

Fig. 3. 2D XY motion (optical flow) by projection of 3D displacements on image plane using middlebury color coding. From left to right, these images are the results of local optical flow, local scene flow and global scene flow in our motion estimation framework along with the ground truth. From up to down, these images are the results of Middlebury Cones, Teddy, and Venus. The optical flow maps are rendered using middlebury coloring method. For scene flow, 3D displacements are projected to image space to obtain 2D optical flow.

images and ground truth disparity maps of frames 2 and 6 of the Middlebury Cones, Teddy, and Venus as the reference and target RGB-D images.

Our approach is compared with three optical flow methods [9, 22, 17], two stereo-based scene flow methods [16, 15] and four RGB-D scene flow methods [5, 4, 1, 10]. Following [15, 16, 1], we use end point error (RMS_O), disparity change error (RMS_Z) and average angular error (AAE) as the error measurement criteria. Results were computed over all valid pixels. For stereo-based methods [16, 15], they jointly estimate the scene flow and disparity using frames 2, 4, 6 and 8 of the Middlebury Cones, Teddy, and Venus. For the two optical flow techniques [9, 22], RMS_Z was computed by estimating 3D translational flow by interpolating depth encoded at the start and end points given its 2D flow vector. The error values are given as reported in their papers or computed using provided codes with default parameters. From Table 2, our method is the top performer under most evaluation criteria among all the optical flow and scene flow algorithms.

An interesting observation is that the local optical flow (NRDC) and local scene flow employed in our framework perform worse than most competing methods while our global scene flow still can generate good-quality motion results.

This is consistent with the qualitative results shown in Figure 3. The estimated local optical flow by NRDC is quite false and noisy on the all the three RGB-D images pairs. The local scene flow improves the motion quality in some region, but the result is still incorrect especially on occlusion, textureless regions and repeated patterns. Our global scene flow can capture the correct motion patterns from the noisy input result of local scene flow, and overcome these issues to generate accurate results.

4.2 Kinect RGB-D Data

We also apply our algorithm on two frames of the RGB-D video sequence *Tshirt4* recorded by a Kinect camera from [23] and RGB-D data captured by us as a complement. We compare the performance of our algorithm to a scene flow method called *RGB-D flow* method [3], a 3D surface tracking method [23], a large displacement optical flow method [9] based on RGB color images. For the optical flow, the scene flow can be computed by back-projecting the 2D optical flow to 3D space domain using camera intrinsic parameters and depth values.

Qualitative Evaluation We visualize the motion results from different methods in two strategies for qualitative evaluation. The first strategy is that we create XY motion (Optical flow) map and Z motion map to show the motion projection on 2D image plane and the motion along depth Z direction respectively. These maps are illustrated based on middlebury color coding [12]. To visualize Z motion map, the values along x-axis of middlebury color coding map are employed. The second strategy is using the motion field to register the point cloud of reference image to the point cloud of the target image. A good motion estimation result should be able to register two point clouds to each other closely and smoothly.

Figure 4 shows the results of estimating the 3D motion field between the frame 58 and the frame 61 on the Tshirt4 sequence. As shown in the row 2 and 3, the RGB-flow method and the optical flow method fail in capturing the distinct motion of the right side of T-shirt marked by the green circle due to occlusion, the surface tracking method over-smooth the region, while our method robustly estimates the 3D motion field. This difference reflects the advantages of the completed scene flow parametrization and occlusion modeling employed by our framework. Row 4-6 depicts the registration results from three orthographic views. We can see that the other three methods fail in registering the reference point clouds to target point clouds smoothly on the regions marked by green circles, and our method works robustly on the deformable surfaces of T-shirt. The registration results are consistent with the XY- and Z-motion maps.

Figure 5 gives the motion results of two RGB-D images of a person waving his hands captured by us. The data is challenging since there is almost no texture on the clothes worn on the person. As shown in the row 2 and 3, other methods fail in estimating the motion field on the region marked by the green circle. In contrast, our method still works robustly against these competing methods.

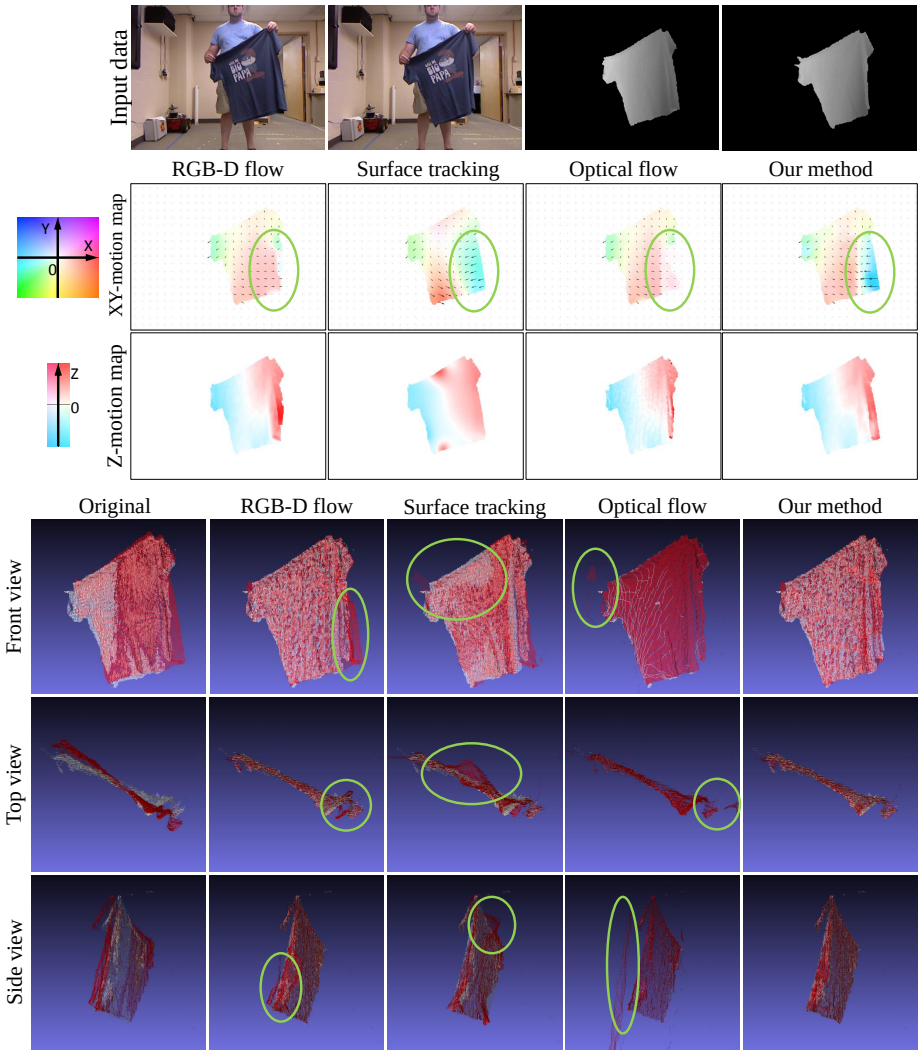

Fig. 4. Scene flow on two frames on Tshirt4 sequence. Row 1: Input reference and target RGB-D images. For clarity, we only show the depth values of the foreground. Row 2-3: XY-motion (optical flow) maps and Z-motion maps of the RGB-D flow, surface tracking, optical flow and our methods. Left images are extended Middlebury color coding maps for 3D motion visualization. Row 4-6: Three basic orthographic views of the two point clouds from reference and target data before and after registration using 3D motion field generated by different methods.

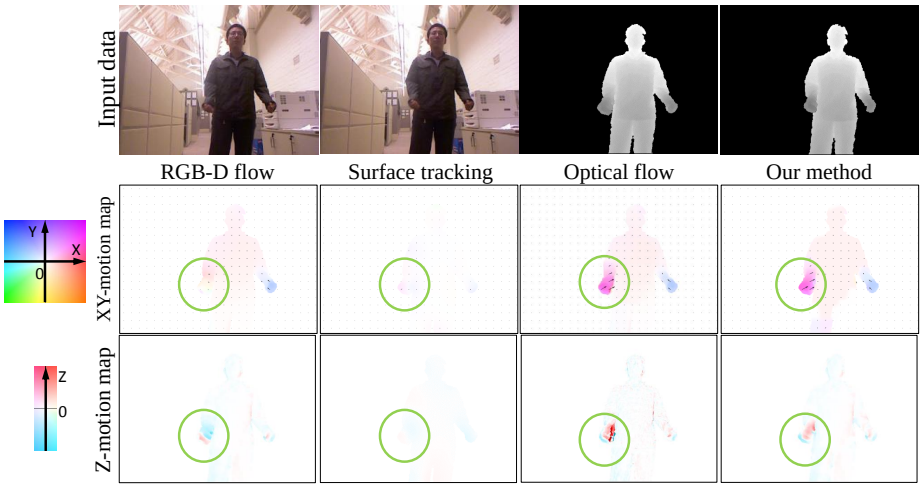

Fig. 5. 3D motion estimation on two RGB-D images captured by us. Row 1: Input reference and target RGB-D images. For clarity, we only show the depth values of the foreground. Row 2-3: XY-motion maps and Z-motion maps of the RGB-D flow, surface tracking, optical flow and our methods. Left images are extended Middlebury color coding maps for 3D motion visualization.

Quantitative Evaluation It is prohibitively expensive to label correspondences for every point in the two RGB-D sequences. Instead, we use a sparse set of hand-tracked points, approximately uniformly spaced in the first frame of each sequence. The position displacement of these points are served as ground truth to measure accuracy and robustness of the estimated motion results.

We evaluate the four methods on the two sequence under different two time intervals configuration of neighboring frames: $\Delta t = 1$ and $\Delta t = 3$. The motion displacement in the time interval configuration $\Delta t = 3$ is approximately 3 times larger than $\Delta t = 1$. Thus, we can discriminate between two time interval configuration by considering them as small displacement ($\Delta t = 1$) and large displacement ($\Delta t = 3$) scenarios, respectively. Table. 3 depicts mean and standard deviation of error (3D Euclidean distance from the ground truth) with small and large displacement scenarios in *Tshirt4* and *Human Hand Waving* sequences. From the table, we can observe that our method achieve comparative results compared with other three state-of-art methods in small displacement scenario. When it turns to the situation that there exists large displacement 3D motion in the scene, our method performs much better and reaches the lowest mean and standard deviation of error. This proves the ability of the proposed method in dealing with large displacement 3D motion estimation.

Methods	Tshirt4				Human Hand Waving			
	$\Delta t = 1$		$\Delta t = 3$		$\Delta t = 1$		$\Delta t = 3$	
	mean	std.	mean	std.	mean	std.	mean	std.
RGB-D flow [3]	3.5	2.0	18.8	56.4	6.6	**3.9**	11.3	14.9
Surface tracking [23]	7.4	4.2	71.9	128.8	11.0	15.1	39.0	129.0
Optical flow [9]	7.1	4.7	23.8	71.1	6.7	5.1	35.9	129.7
Our method	**2.8**	**1.8**	**8.9**	**4.2**	**5.9**	4.2	**7.7**	**10.4**

Table 3. The mean and standard deviation of error (mm) with small and large displacement in *Tshirt4* and *Human Hand Waving* sequences.

5 Conclusions

In this paper, we present a framework to address the challenging problems of scene flow estimation based on RGB-D data. In the framework, we efficiently initialize scene flow from a 2D motion method to address the large displacement motion problem, and then refine it using a local method to provide candidates, and fuse these motion candidates by considering occlusion and smoothness. In the local method, we propose calculate the matching cost using a 3D supporting area using adaptive weights which is robust to noise. In the global method, we explicitly model occlusion to jointly estimate occlusion and scene flow to address the occlusion problem. For the noise and missing data issues, RGB-D data reliability is also taken into account in the formulation. We showed compelling results on the Middlebury datasets as well as on challenging Kinect RGB-D data.

Acknowledgement. This work was supported by Microsoft Research, Redmond. We also acknowledge Minqi Li for recording the Kinect RGB-D data sequence in our experiment.

References

1. Hadfield, S., Bowden, R.: Scene particles: Unregularized particle based scene flow estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014) 564–576
2. Gottfried, J.M., Fehr, J., Garbe, C.S.: Computing range flow from multi-modal kinect data. In: Advances in Visual Computing. Springer (2011) 758–767
3. Herbst, E., Ren, X., Fox, D.: Rgb-d flow: Dense 3-d motion estimation using color and depth. In: IEEE International Conference on Robotics and Automation (ICRA), IEEE (2013)
4. Quiroga, J., Devernay, F., Crowley, J.L., et al.: Local/global scene flow estimation. In: ICIP-IEEE International Conference on Image Processing. (2013)
5. Zhang, X., Chen, D., Yuan, Z., Zheng, N.: Dense scene flow based on depth and multi-channel bilateral filter. In: Computer Vision–ACCV 2012. Springer (2013) 140–151
6. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics-TOG **28** (2009) 24

7. Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A.: The generalized patchmatch correspondence algorithm. In: Computer Vision–ECCV 2010. Springer (2010) 29–43
8. Korman, S., Avidan, S.: Coherency sensitive hashing. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1607–1614
9. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 500–513
10. Hornacek, M., Fitzgibbon, A., Carsten, R.: Sphereflow: 6 dof scene flow from rgb-d pairs. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014)
11. Junejo, I.N., Dexter, E., Laptev, I., Perez, P.: View-independent action recognition from temporal self-similarities. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 172–185
12. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92** (2011) 1–31
13. Vedula, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. Pattern Analysis and Machine Intelligence, IEEE Transactions on **27** (2005) 475–480
14. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013)
15. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: A view centered variational approach. International journal of computer vision **101** (2013) 6–21
16. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE (2007) 1–7
17. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Non-rigid dense correspondence with applications for image enhancement. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 70
18. Eshet, Y., Korman, S., Ofek, E., Avidan, S.: Dcsh-matching patches in rgbd images. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 89–96
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
20. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010) 1392–1405
21. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discrete applied mathematics **123** (2002) 155–225
22. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1744–1757
23. Willimon, B., Hickson, S., Walker, I., Birchfield, S.: An energy minimization approach to 3d non-rigid deformable surface estimation using rgbd data. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE (2012) 2711–2717