Handling Occlusion and Large Displacement through Improved RGB-D Scene Flow Estimation

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
wangyucheng@student.uts.edu.au

***Abstract*—The accuracy of scene flow is restricted by several challenges such as occlusion and large displacement motion. When occlusion happens, the positions inside the occluded regions lose their corresponding counterparts in preceding and succeeding frames. Large displacement motion will increase the complexity of motion modeling and computation. Moreover, occlusion and large displacement motion are highly related problems in scene flow estimation, e.g. large displacement motion often leads to considerably occluded regions in the scene.**

An improved dense scene flow method based on RGB-D data is proposed in this paper. To handle occlusion, we model the occlusion status for each point in our problem formulation, and jointly estimate the scene flow and occluded regions. To deal with large displacement motion, we employ an over-parameterized scene flow representation to model both the rotation and translation components of the scene flow, since large displacement motion cannot be well approximated using translational motion only. Furthermore, we employ a two-stage optimization procedure for this over-parameterized scene flow representation. In the first stage, we propose a new RGB-D PatchMatch method, which is mainly applied in the RGB-D image space to reduce the computational complexity introduced by the large displacement motion. According to the quantitative evaluation based on the Middlebury dataset, our method outperforms other published methods. The improved performance is also comprehensively confirmed on the real data acquired by Kinect sensor.

***Index Terms*—scene flow, RGB-D data, occlusion, large displacement motion, rotation**

I. Introduction

Scene flow is the 3D motion field in the physical world. Compared to optical flow, scene flow has view-independent characteristics, which is preferred in many vision applications such as action recognition, activity recognition, object tracking, and 3D reconstruction. It is also useful in 3D human-computer interaction applications, e.g. telepresence and computer games. Many scene flow methods [1][2][3] employ only RGB stereo images. It is only in recent years that, RGB-D cameras (e.g. Kinect), which provide relatively reliable depth information, have driven a trend to estimate the scene flow from RGB-D data. RGB-D image-based methods [4][5][6][7][8][9][10][11] have emerged thanks to the development of RGB-D cameras.

After reviewing these recent state-of-the-art methods, we find that there are still challenging problems which are unsolved, or partially unsolved, in RGB-D data-based scene flow estimation. The first challenging problem is occlusion in the captured RGB-D data. When occlusion occurs, the positions inside the occluded regions lose their corresponding counterparts in preceding or succeeding frames. Thus, scene flow estimation of these positions should be carried out differently than for non-occluded positions. The second problem is how to handle large displacement motion in the scene. To achieve robust scene flow estimation in the case of large displacement motion, a larger search region is unavoidably required to find satisfactorily matched positions. This immediately increases the computational complexity. In addition, large displacement motion also increases the complexity of motion modeling, and each individual position will present different rotation and/or translation during the motion period. In a case such as this, a simple translational parameter model is not able to well describe the situation [12]. Large displacement motion will increase the complexity of motion modeling. Moreover, these two challenging problems are not always independently isolated, but instead are tangled together. According to the observations, large displacement motion in a dynamic RGB-D video sequence always causes occlusion around the motion regions.

To address both of these challenging issues, we propose an improved scene flow estimation method in this paper, based on RGB-D data. A solution in [8] made some efforts to handle occlusion by truncating the extreme cost values when measuring the similarity of features between different locations. However, this solution is not able to integrate multiple occlusion cues for robust estimation. In other existing methods, the occluded regions are detected after scene flow estimation [10], based on consistency checking. That is, scene flow estimation cannot identify the occluded regions, and false matched positions are consequently produced. Moreover, when the smoothness assumption is applied, such incorrectly estimated motion may affect the accuracy of the estimated motion in nearby positions . Thus, the common occlusion detection carried out independently following the scene flow estimation is made too late to fix the problem. In our method, described in this paper, explicit occlusion modeling is employed to incorporate multiple occlusion cues (both feature inconsistency and depth order in our formulation) for robust estimation. Meanwhile, occlusion handling is also embedded into the scene flow estimation process. In this way, unnecessary or

incorrect estimation around occlusions can be eliminated more effectively and efficiently.

The modeling complexity of large displacement motion can be tackled by modeling rotational and translational components at each point with over-parameterized scene flow representation [9]. However, large displacement motion also causes computational complexity (large search dimension and range problems) in the optimization. Some scene flow methods [9][11] which use over-parameterized scene flow representation can only obtain local minima when conducting optimization and still fail in large displacement motion estimation. To optimize scene flow with large displacement motion, Hornáček *et al.* [10] presented a two-stage method and extended the 2D PatchMatch [13][14] algorithm to a 3D version for over-parameterized scene flow representation in the first stage. However, their final results rely strongly on the modified 3D PatchMatch method in the first stage. As pointed out by Philipp *et al.* [15], incorrect or unreliable estimates are propagated on the texture-less regions and repeated patterns, due to the feature of implicit smoothing in the PatchMatch. To address these problems caused by large displacement motion, we propose a novel two-stage optimization process for both motion and occlusion. In the first stage, an efficient RGB-D PatchMatch method based on the original 2D PatchMatch [13][14] is proposed. Compared to the 3D PatchMatch in [10], our RGB-D PatchMatch method reduces the time complexity of measuring patch similarity by projecting 3D points into 2D RGB-D image space, and further reduces the searching DoF (Degree of Freedom) from six to two by using a new concept (i.e. dominant directions of rotation) in the random sampling process. This generates abundant motion patterns (especially large displacement motion patterns) *within an acceptable time* for high-dimension over-parameterized scene flow representation. All the motion patterns generated in this way are then considered as candidates for each reference position, which avoids *searching all the possible ranges*. In the second stage, the scene flow of all the reference positions is finalized by integrating the best parts of all the candidate motion patterns using explicit occlusion modeling and smoothness regularization, which is robust on texture-less regions, occluded regions and regions with repeated patterns.

Our key contributions can be summarized as follows: (1) we give a new formulation of the scene flow problem that jointly estimates the scene flow and the occluded regions; (2) we present a novel two-stage optimization procedure for over-parameterized scene flow representation [9], which is robust in the presence of texture-less regions, occluded regions and regions with repeated patterns; (3) we propose an efficient RGB-D PatchMatch method in the first stage of our optimization procedure to generate accurate candidate motion patterns.

The remainder of the paper is organized as follows. Section II provides a brief overview of related motion estimation work on occlusion and large displacement motion handling. Section III mainly defines our problem formulation for the joint estimation of scene flow and occlusion. Section IV gives the details of the proposed two-stage optimization procedure to deal with large displacement motion. Section V comprehensively evaluates our method on a variety of datasets, and Section VI concludes the paper.

II. Related Work

Occlusion and large displacement motion are known to be two challenging problems in motion estimation. In terms of the way occlusion and large displacement motion are handled, existing solutions in RGB-D based scene flow methods can be categorized as shown in Table I.

Occlusion means that some points disappear and their correspondence cannot be found in preceding and succeeding frames. Some scene flow methods [16][4][7][11][6][5] assume that there is no or little occlusion between two neighboring frames, but this assumption fails in the presence of occlusion caused by large displacement motion. The solution in [8] has considered the possible effects of the occlusion by truncating the extreme cost values when measuring the feature similarity of two matched position. However, it only utilize one occlusion cue from the feature inconsistency. Consequently, those non-occluded pixels with wrong matched positions and large cost values may be mistakenly deemed to be occluded in some cases. Other methods like [10] detect occlusion by extending the consistency check technique widely employed in stereo vision [17] and optical flow [18], and refining the motion results in the occluded regions using various techniques as a final post-processing step. This consistency check procedure has two drawbacks. The first drawback is that error may be introduced if the estimated motion used for consistency check is incorrect. The second is that the procedure essentially requires the motion field to be computed bidirectionally, which doubles computational consumption. Unlike earlier scene flow methods, our method integrates more occlusion cues based on both feature inconsistency and depth order, and jointly estimates the scene flow and the occluded regions. Thus, the accuracy of the scene flow estimates is improved, especially around the occluded regions. In the stereo, optical flow and stereo-based scene flow literature, there are also some methods [19][20][1] which jointly estimate motion and occlusion. However, they are not suitable, or cannot be directly applied in the RGB-D data-based scene flow estimation, since they are based on certain strict assumptions (e.g. the occlusion sparsity assumption in [20]) or have different characteristics of input data [19][1][21].

Large displacement motion causes difficulties in the computation process, since the search range for scene flow estimation is large. To handle large displacement motion, most prior scene flow methods [11][6] only employ a coarse-to-fine scheme. However, a coarse-to-fine scheme is known to essentially have no ability to recover the motion of objects when the magnitude of the motion is larger than the size of the objects [22][23]. Following the large displacement optical flow method [22], Quiroga *et al.* [7] dealt with large displacement 3D motion by including a constraint to enforce the consistency of feature points (e.g. SURF [24]). However, feature points could be rather sparse on texture-less regions, or ambiguous on the repeated texture. In such cases, the correspondence of these feature points is limited and unreliable. Inspired by [23] who

TABLE I
CLASSIFICATION FOR RGB-D SCENE FLOW METHODS ACCORDING TO OCCLUSION AND LARGE DISPLACEMENT MOTION HANDLING.

RGB-D scene flow	Regular motion	Large displacement motion
No occlusion handling	Hadfield and Bowden [16][4] Quiroga *et al.* [11] Herbst *et al.* [6] Gottfried *et al.* [5]	Quiroga *et al.* [7]
Robust truncation function	Zhang *et al.* [8]	
Cross checking		Hornáček *et al.* [10]
Joint estimation of occlusion and motion		Our work

estimated optical flow based on the nearest neighbor fields of RGB images, and [25] who improved nearest neighbor fields of RGB images with the help of depth images, we seek to obtain abundant motion candidates from a nearest neighbor field of RGB-D images using a novel RGB-D PatchMatch method which generates the large displacement motion pattern candidates for scene flow estimation.

Note that the closest work to ours was recently presented by Hornáček *et al.* [10]. They also used a two-stage approach [13] for scene flow estimation; however the differences are shown in both stages. In the first stage, Hornáček *et al.*'s modified 3D PatchMatch method used a KD-tree to find the nearest points in the 3D space when measuring the similarity of all the 3D patches. The KD-tree leads to a time complexity of $O(n \log\ n)$, where n is the number of points. Our method measures patch similarity in the RGB-D image space,which projects 3D points on the 2D image plane and directly uses the difference of depth values at the 2D projected positions as the geometrical similarity. The computation complexity drops to $O(n)$. In the second stage, Hornáček *et al.* [10] took all the results of the first stage into account and mainly carried out necessary adjustments to the possible occluded regions. Thus, their final results rely on the precision of the first-stage processing. According to our study, this may fail around texture-less regions, occluded regions or regions with repeated patterns. Our method does not suffer from this problem, because in this stage it uses the initial results obtained in the first stage as the candidates. A selection process is developed to preserves the best motion candidates through an optimization process by minimizing a global energy function with explicit occlusion modeling embedded.

III. PROBLEM FORMULATION

Given two RGB-D images $\{I, D\}$ and $\{I', D'\}$, we seek to estimate the scene flow from the reference image $\{I, D\}$ to the target image $\{I', D'\}$. Each pixel $\mathbf{p}$ in the reference image can be deemed as a 3D point $\mathbf{P}$. The goal of our improved scene flow method is to assign a motion pattern $\mathbf{V_P}$ and an occlusion status $O_\mathbf{P} \in \{0, 1\}$ to each point $\mathbf{P}$ in the point set Ω of the reference RGB-D image.

There are two major principles in occlusion handling for estimating scene flow: (1) feature consistency in the non-occluded regions along the temporal domain - the multi-modal features (descriptors) of a non-occluded point and its corresponding position along the temporal domain should be consistent; (2) spatial smoothness regularization - the spatial distances of neighboring points that are close in the 3D space should be similar before and after the movement. Unlike all the previous feature consistency assumptions, our first principle restricts feature consistency on non-occluded regions only. In this way, the problematic feature consistency assumption along occluded regions, which have been seen in the existing scene flow methods [16][4][7][11][6][5][8][10] can be avoided. Our second principle enforces motion smoothness based on a geodesic mapping assumption, which helps in estimating the motion of these occluded points by their non-occluded neighbors. Compared with other smoothness regularizations, the geodesic mapping assumption is more general for non-rigid deformations [26] in the scene and is thus more robust when estimating scene flow. Combining these two principles therefore helps us to overcome the first challenging problem, i.e. occlusion, in the scene flow estimation. Driven by these two principles, the objective is to find the motion field $\mathbf{V} = \{\mathbf{V_P} | \mathbf{P} \in \Omega\}$ and the occlusion map $O = \{O_\mathbf{P} | \mathbf{P} \in \Omega\}$ that minimize the function:

$$\{\hat{\mathbf{V}}, \hat{O}\} = \underset{\mathbf{V}, O}{\operatorname{argmin}}\ (1-\lambda) \cdot E_{\text{feat}}(\mathbf{V}, O) + \lambda \cdot E_{\text{smooth}}(\mathbf{V}) \quad (1)$$

where E_{feat} is an energy term that encourages feature consistency on the non-occluded regions, E_{smooth} is an energy term that enforces smoothness, and λ is a user-specified parameter to adjust the ratio of the two terms. When the scene is texture-less and ambiguous for correspondence, we should rely more on the smoothness term with larger λ; otherwise, λ should be small. The two energy terms correspond to the two proposed principles, respectively. The definitions of E_{feat} and E_{smooth} are given in Eq. 6 and Eq. 11 individually. More details can be found in the rest of this section.

To model large displacement motion, we employ an over-parameterized scene flow representation [9] $\mathbf{V_P} = \{\mathbf{R_P}, \mathbf{T_P}\} \in \mathbb{SE}(3)$. $\mathbb{SE}(3)$ is the symmetry group of 3D Euclidean space, and is used to describe the rotational and translational motion of a rigid body. However, it is difficult to estimate scene flow with such an over-parameterized representation (large search dimension) in the presence of large displacement motion (a large search range). Accordingly, to handle large displacement motion, we propose a two-stage optimization scheme exclusively for this over-parameterized scene flow representation. The details of our two-stage optimization scheme are introduced in the next section (Section IV).

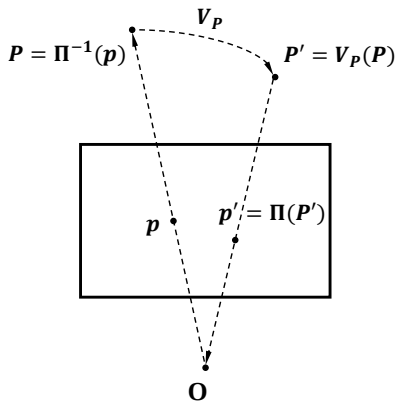

Fig. 1. Notation of the 2D pixels and 3D points.

A. Notation

We briefly give the notation of the 2D pixels and 3D points as shown in Figure 1. We denote $\mathbf{P} = \{X_\mathbf{P}, Y_\mathbf{P}, Z_\mathbf{P}\}$ as the 3D coordinate of the pixel $\mathbf{p}$, and it is computed by back-projecting $\mathbf{p}$ using its depth value $D(\mathbf{p})$ and intrinsic camera parameters $\mathbf{K}$ as follows:

$$\mathbf{P} = \Pi^{-1}(\mathbf{p}) = D(\mathbf{p}) \cdot \mathbf{K}^{-1}\tilde{\mathbf{p}}, \tag{2}$$

where Π^{-1} means back-projection operation, and $\tilde{\mathbf{p}} = [\mathbf{p}^\top, 1]^\top$ is the homogeneous coordinate of the pixel $\mathbf{p}$. In contrast, when the 3D coordinate of a point $\mathbf{p}$ is known, its 2D image projection is

$$\mathbf{p} = \Pi(\mathbf{P}) = \frac{1}{Z_\mathbf{P}} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{K}\mathbf{P}, \tag{3}$$

where Π is the projection operation, and $Z_\mathbf{P}$ is the Z component of the 3D coordinate $\mathbf{P}$. We denote $\mathbf{P}'$ as the predicted 3D position of point $\mathbf{P}$ under the motion $\mathbf{V_P} = \{\mathbf{R_P}, \mathbf{T_P}\}$, and it can be computed by

$$\mathbf{P}' = \mathbf{V_P}(\mathbf{P}) = \mathbf{R_P}\mathbf{P} + \mathbf{T_P}. \tag{4}$$

Similarly, the 2D projection of $\mathbf{P}'$ is then

$$\mathbf{p}' = \Pi^{-1}(\mathbf{P}') = \Pi^{-1}(\mathbf{V_P}(\mathbf{P})) = \Pi^{-1}(\mathbf{R_P}\mathbf{P} + \mathbf{T_P}). \tag{5}$$

B. Feature Consistency in Non-Occluded Regions along the Temporal Domain

In the formulation, we look for the scene flow that best explains the color and depth data; thus, our method combines both color and depth features to describe each 3D point. We assume that the features (descriptor) of a non-occluded point in the reference image should be similar to those of its corresponding position in the target image. The energy term E_{feat} is used for the feature consistency, which defined by

$$E_{\text{feat}}(\mathbf{V}, O) = \sum_{\mathbf{P}\in\Omega} C^{occ}_{patch}(\mathbf{P}, \mathbf{V_P}, O_\mathbf{P}), \tag{6}$$

where Ω is the set of points with valid depth in the reference RGB-D image, and C^{occ}_{patch} is the RGB-D patch-based cost with occlusion modeling which is defined in the following paragraphs.

1) RGB-D Patch-Based Feature: We propose a RGB-D patch-based feature to handle the noise issue that exists in pixel-based features. Compared with pixel-based features, however, RGB color patch-based features (e.g. SAD, NCC and Mutual Information) are not rotation-invariant and are not robust to perspective distortions [27], which are common in the presence of large displacement motion. Many prior methods explicitly or implicitly assume that there is minimal or no rotational motion in the scene [7][6][8]. By leveraging the over-parameterized scene flow representation (both rotation and translation) for each point, we are able to tackle this problem.

We assume local rigidity for points in a small RGB-D neighborhood (RGB-D patch), i.e. all the points in the RGB-D patch are assumed to share the same rigid motion pattern (rotation and translation). The patch-based feature matching cost for each point is obtained by calculating the weighted sum of the pixel-based cost values of its locally neighboring points under a specified motion pattern. To preserve motion boundaries, the weight of each point in a neighborhood is adaptively adjusted according to whether it is on the same surface as the center point of the RGB-D patch. Our initial RGB-D patch-based cost without occlusion modeling is then defined by

$$C_{patch}(\mathbf{P}, \mathbf{V_P}) = \sum_{\mathbf{Q}\in Patch(\mathbf{P})} w(\mathbf{P}, \mathbf{Q}) \cdot C(\mathbf{Q}, \mathbf{V_P}). \tag{7}$$

where $\mathbf{V_P}$ is the specified motion for the patch, $Patch(\mathbf{P})$ is the local neighborhood of the point $\mathbf{P}$ on the RGB-D image plane, $C(\mathbf{Q}, \mathbf{V_P})$ is the point-based cost value of point $\mathbf{Q}$ under the motion $\mathbf{V_P}$, and $w(\mathbf{P}, \mathbf{Q})$ is the weighting function that helps to preserve the motion boundaries according to the observation that $\mathbf{Q}$ is on the same surface as $\mathbf{P}$. We will discuss the occlusion modeling in the following paragraphs.

For the point-based cost $C(\mathbf{Q}, \mathbf{V_P})$, we include a *brightness constancy assumption*, a *gradient magnitude constancy assumption* and a *depth variation constraint* [7], [10], [11] in our formulation. Combining these assumptions and constraints, the matching cost for point-based feature consistency is defined by:

$$C(\mathbf{P}, \mathbf{V_P}) = \Psi(\rho_I) + \lambda_{\nabla I} \cdot \Psi(\rho_{\nabla I}) + \lambda_D \cdot \Psi(\rho_D). \tag{8}$$

where $\Psi(\cdot)$ is the robust L1 norm function, $\rho_I(\mathbf{P}, \mathbf{V_P}) = I'(\mathbf{p}') - I(\mathbf{p})$ is the brightness difference, $\rho_{\nabla I}(\mathbf{P}, \mathbf{V_P}) = |\nabla I'(\mathbf{p}')| - |\nabla I(\mathbf{p})|$ is the gradient magnitude difference, $\rho_D(\mathbf{P}, \mathbf{V_P}) = D'(\mathbf{p}') - Z_{\mathbf{P}'}$ is the depth variation, $D'(\mathbf{p}')$ is the depth value of the pixel $\mathbf{p}'$, $Z_{\mathbf{P}'}$ is the Z component of 3D point $\mathbf{P}'$, and $\lambda_{\nabla I}$ and λ_D control the ratio of different features.

For the weighting function $w(\mathbf{P}, \mathbf{Q})$, previous methods directly use depth variation [6][7][11] or spatial distance [10] for guidance. Spatial distance is also not robust enough, since two points with a small spatial distance may be on different surfaces. Depth variation is only appropriate for 3D fronto-parallel surfaces. To alleviate this problem, we propose a weighting function which further uses normal information $\mathbf{n_P}$ for each point $\mathbf{P}$. Thus, our weighting function for point $\mathbf{P}$ is

defined by

$$w(\mathbf{P},\mathbf{Q}) = \eta \cdot \exp(-\frac{|\mathbf{P}-\mathbf{Q}|^2}{2\sigma_s^2} - \frac{|\mathbf{n_P}\cdot(\mathbf{P}-\mathbf{Q})|^2}{2\sigma_n^2}), \quad (9)$$

where $|\mathbf{P}-\mathbf{Q}|$ is the spatial distance of the points $\mathbf{P}$ and $\mathbf{Q}$, $|\mathbf{n_P}\cdot(\mathbf{P}-\mathbf{Q})|$ is the projection of $(\mathbf{P}-\mathbf{Q})$ along the normal direction $\mathbf{n_P}$. When the surfaces are highly-slanted, depth variation $|Z_\mathbf{P}-Z_\mathbf{Q}|$ is large even if two points are on the same surface. $|n_\mathbf{P}\cdot(\mathbf{P}-\mathbf{Q})|$ can measure how close to each other two points are on a highly-slanted surface by incorporating normal information $n_\mathbf{P}$. Specifically, on the front-parallel surface (i.e. $\mathbf{n_P}=[0,0,-1]$), $|n_\mathbf{P}\cdot(\mathbf{P}-\mathbf{Q})|$ is equivalent to $|Z_\mathbf{P}-Z_\mathbf{Q}|$, since $|n_\mathbf{P}\cdot(\mathbf{P}-\mathbf{Q})| = |[0,0,-1]\cdot[X_\mathbf{P}-X_\mathbf{Q}, Y_\mathbf{P}-Y_\mathbf{Q}, Z_\mathbf{P}-Z_\mathbf{Q}]| = |-1\cdot(Z_\mathbf{P}-Z_\mathbf{Q})| = |Z_\mathbf{P}-Z_\mathbf{Q}|$. Thus, the term $n_\mathbf{P}\cdot(\mathbf{P}-\mathbf{Q})$ is able to handle both a case of highly-slanted surface and also a case of front-parallel surface. σ_s and σ_n are the standard deviations of the Gaussian kernels for the two components respectively and are computed for each 3D point pair $\mathbf{P}$ and $\mathbf{Q}$ using the method given in Sec. V, and η is the normalization factor to ensure that $\sum_{\mathbf{Q}\in Patch(\mathbf{P})} w(\mathbf{P},\mathbf{Q}) = 1$. Here, we adopt PCA [28] to estimate the normal information as in [25].

2) Occlusion Modeling: Different kinds of occlusion often occur in scene flow estimation. For example, points are occluded if other points move in front of them, or they are moving out of the image range. One common consequence is that no corresponding position in the target image $\{I', D'\}$ matches the occluded points in the reference image $\{I, D\}$. Without occlusion modeling, the feature consistency assumption is violated, since feature consistency is enforced on the occluded regions. Scene flow estimation may be affected and there may be an incorrect dragging effect around the occluded regions.

To address this problem, our method explicitly models the occlusion status for each point by considering the constraints of depth order and feature difference [29]. We deem the occluded points to be outliers when searching for correspondences, and use a fixed cost value for these outliers. Finally, our novel RGB-D patch-based matching cost with occlusion modeling is defined by

$$\begin{aligned} C^{occ}_{patch}(\mathbf{P},\mathbf{V_P},O_\mathbf{P}) &= (1-O_\mathbf{P})\cdot C_{patch}(\mathbf{P},\mathbf{V_P}) \\ &+ O_\mathbf{P}\cdot\xi\cdot([Z_{\mathbf{P}'} > D'(\mathbf{p}') + 3\sigma(\mathbf{p}')] \vee [C_{patch}(\mathbf{P},\mathbf{V_P}) > \xi]) \end{aligned} \quad (10)$$

where $[\cdot]$ is the Iverson bracket which denotes a number that is 1 if the condition in square brackets is satisfied, and 0 otherwise, $Z_{\mathbf{P}'}$ is the Z component of point $\mathbf{P}'$, $D'(\mathbf{p}')$ is the depth value of the 2D projection p' of the point $\mathbf{P}'$, and ξ is a specified threshold for matching cost. $[Z_{\mathbf{P}'} > D'(\mathbf{p}')+3\sigma(\mathbf{p})]$ corresponds to the constraint of depth order, which is based on an observation that points in the target image are occluded if other points move in front of them from the camera view. $3\sigma(\mathbf{p}')$ is a margin to handle the noise of the depth measure when judging the occlusion relation, and $\sigma(\mathbf{p}')$ is the standard deviation of Gaussian noise at pixel $\mathbf{p}'$ which can be acquired by calibration techniques such as [30]. $[C_{patch}(\mathbf{P},\mathbf{V_P}) > \xi]$ corresponds to the constraint of feature difference, which suggests that if the feature matching cost of a point is larger than the threshold ξ, the point should be considered as being occluded. If either one of the two conditions is satisfied, the point is deemed as being occluded. Note that C^{occ}_{patch} is the final version of the proposed RGB-D patch-based matching cost with occlusion modeling. Compared with the definition of C_{patch} in Eq. 7, it models the occlusion status for each point and avoids computing the matching cost on the occluded regions without correspondence.

C. Spatial Smoothness Regularization

Previous over-parameterized scene flow methods regularize completed scene flow by enforcing local rigidity smoothness on the motion parameter space of neighboring points [9][11][10]. However, the local rigidity assumption is not robust to large non-rigid deformations. Inspired by the 3D non-rigid surface registration [26] with dramatic geometrical changes on the objects, we propose a general smoothness regularization based on approximate local geodesic consistency, which assumes that the estimated scene flow is an approximate geodesic mapping for local 3D regions on the surfaces from the reference data to the target data. A geodesic mapping is a geodesic-distance-preserving mapping, i.e. the geodesic distance of any two points in the local scene is consistent. It is known that if the geodesic distance between any two local neighboring points does not change, the geodesic distance between any two points in the scene will not change either [26], and the geodesic distance can be then viewed locally as the Euclidean distance [31]. Thus, we can apply our smoothness regularization based on the geodesic mapping assumption by enforcing locally neighboring points that have similar Euclidean distances before and after temporal movement. The smoothness term in the energy function is defined by

$$E_{\text{smooth}}(\mathbf{V}) = \sum_{\mathbf{P}\in\Omega}\sum_{\mathbf{Q}\in N(\mathbf{P})} w(\mathbf{P},\mathbf{Q})\cdot S(\mathbf{P},\mathbf{Q},\mathbf{V_P},\mathbf{V_Q}), \quad (11)$$

where $N(\mathbf{P})$ is the set of 4 (or 8) connected neighboring points of the point $\mathbf{P}$ on the image plane, $w(\mathbf{P},\mathbf{Q})$ is the boundary-preserving weighting function used in *Eq*. 9, and $S(\mathbf{P},\mathbf{Q},\mathbf{V_P},\mathbf{V_Q})$ is used to enforce the local Euclidean distance consistency of two neighboring points, which is defined by

$$S(\mathbf{P},\mathbf{Q},\mathbf{V_P},\mathbf{V_Q}) = \big|||\mathbf{P}-\mathbf{Q}||_2 - ||\mathbf{V_P}(\mathbf{P})-\mathbf{V_Q}(\mathbf{Q})||_2\big|, \quad (12)$$

where $||\mathbf{P}-\mathbf{Q}||_2$ is the Euclidean distance between two points $\mathbf{P}$ and $\mathbf{Q}$, and $||\mathbf{V_P}(\mathbf{P})-\mathbf{V_Q}(\mathbf{Q})||_2$ is the Euclidean distance between the point $\mathbf{P}$ with the motion $\mathbf{V_P}$ and the point $\mathbf{Q}$ with the motion $\mathbf{V_Q}$.

IV. Optimization

We seek to minimize our energy function in Eq. 1 to obtain scene flow estimates; however the energy function is non-convex, and computing the global minimum is known to be NP-hard. Inspired by [32], we can convert this scene flow estimation problem into a pair-wise MRF labeling problem. FusionMoves efficiently combines a set of proposal labels in

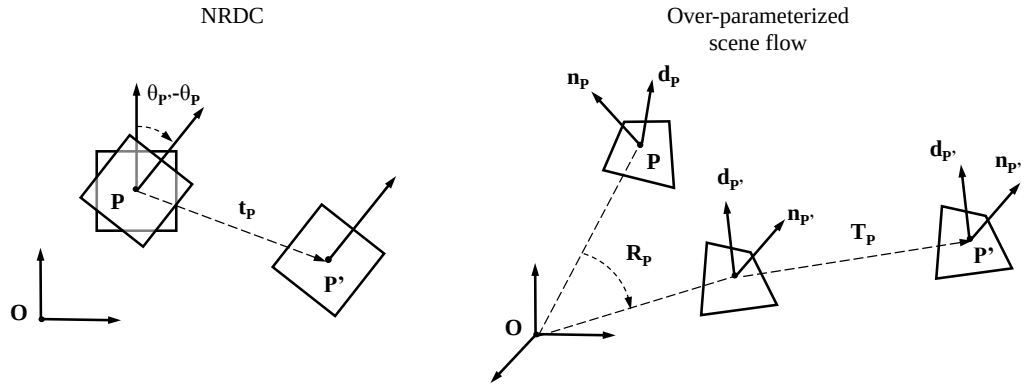

Fig. 2. NRDC and over-parameterized scene flow representation. To clearly see the rotation change for a point, we use a square patch with a 2D/3D dominant direction vector to represent the point.

a theoretically sound way, which is often globally optimal in practice. However, how to generate such a set of proposal labels is an open question for different tasks, and is essential for the accuracy and robustness of scene flow estimation, especially in the presence of large displacement motion.

Thus, we propose a two-stage optimization scheme, namely proposal generation and proposal fusion, to jointly estimate the scene flow and the occlusion of deformable surfaces in the scene. To deal with large displacement motion, we present a novel RGB-D PatchMatch method in the proposal generation stage, which is able to provide abundant large displacement motion proposals by minimizing only the temporal feature consistency term E_{feat}. In the proposal fusion stage, we optimize the scene flow results by further incorporating the spatial smoothness regularization term E_{smooth} to integrate the best parts of these motion proposals. In this way, this two-stage optimization scheme is capable of tackling the second challenge in scene flow estimation, i.e. large displacement motion.

A. Proposal Generation

We generate both motion proposals $\{\mathbf{V}_{prop}\}$ and occlusion proposals $\{O_{prop}\}$ for the later proposal fusion process. The occlusion status has only two discrete values $\{0, 1\}$. For each generated motion proposal $\mathbf{V}_{prop}$, we generate two combined motion and occlusion proposals, $(\mathbf{V}_{prop}, 0)$ and $(\mathbf{V}_{prop}, 1)$. The key to achieving satisfactory results is to generate abundant good-quality motion proposals $\{\mathbf{V}_{prop}\}$, which potentially contain accurate motion patterns, especially accurate large displacement motion patterns. In our method, we generate motion proposals using our feature consistency assumption. The cost function for generating proposals is defined by

$$\{\mathbf{V}_{prop}\} = \underset{\mathbf{V}}{\operatorname{argmin}}\ E_{\text{feat}}(\mathbf{V}, O) \tag{13}$$

The motion patterns calculated for all the points form the set of motion proposals $\{\mathbf{V}_{prop}\}$ for each point, and motion proposals can be generated abundantly.

B. RGB-D PatchMatch

To minimize this cost function and obtain accurate large displacement motion patterns, the ability to leave local minima during optimization is required. PatchMatch [13][14] consists of two steps, i.e. spatial propagation and random sampling. The algorithm performs by iteratively carrying out the two steps until convergence. PatchMatch is able to leave local minima because of the random sampling step, and it can potentially generate large displacement motion patterns while minimizing the energy function. However, the search range and dimension are both large for over-parameterized scene flow representation, and the original PatchMatch algorithm is not able to generate accurate motion results efficiently. To address this problem, we propose a novel RGB-D PatchMatch method. It is known that initial values and search dimension are essential for the accuracy and efficiency of the search algorithms. Our RGB-D PatchMatch algorithm is equipped with a new initialization strategy based on 2D motion and a modified reduced-dimension random sampling strategy for RGB-D data. We give the details of our RGB-D PatchMatch algorithm as follows.

1) Initialization Based on 2D Motion: Some 2D motion methods deal with large displacement motion on the image plane, since the search dimension is smaller than for it is for scene flow situations. Thus, we choose a method called NRDC [33] to generate the initial values efficiently. NRDC can generate a 2D motion field that includes 2-DoF translational vectors $\mathbf{t_p}$ (see Figure 2). However, the required motion parameters for our method are $\mathbf{V_P} = \{\mathbf{R_P}, \mathbf{T_P}\}$. We propose an approach to enable the conversion from a 2D motion field into a 3D over-parameterized scene flow.

To compute the rotation matrix, we assume that each point $\mathbf{P}$ has its corresponding 2D and 3D prominent directions. The 3D prominent direction $\mathbf{d_P}$ is a direction of point $\mathbf{P}$ on the 3D object surface which is orthogonal to its normal $\mathbf{n_P}$ in 3D space, and the 2D prominent direction is the projection of the 3D prominent direction on the 2D image plane. Inspired by [25], we adopt the prominent orientation in SIFT feature detection [34] as the 2D prominent directions, i.e. $[\sin(\theta_{\mathbf{P}}), \cos(\theta_{\mathbf{P}})]$ for the point $\mathbf{P} = \Pi^{-1}(\mathbf{p})$ and $[\sin(\theta_{\mathbf{P}'}), \cos(\theta_{\mathbf{P}'})]$ for the point $\mathbf{P}' = \Pi^{-1}(\mathbf{p} + \mathbf{t_p})$.

According to our definition, 3D prominent direction vectors can be then computed by

$$\mathbf{d_P} = \mathrm{orthonorm}([\sin(\theta_\mathbf{P}), \cos(\theta_\mathbf{P}), 0]^\mathrm{T}, \mathbf{n_P}), \quad (14)$$

$$\mathbf{d_{P'}} = \mathrm{orthonorm}([\sin(\theta_\mathbf{P'}), \cos(\theta_\mathbf{P'}), 0]^\mathrm{T}, \mathbf{n_{P'}}), \quad (15)$$

where $\mathrm{orthonorm}(\cdot,\cdot)$ is the Gram-Schmidt orthonormalization procedure. The rotation variation of a point is reflected by the variations of its normal and prominent 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

$$\mathbf{R_P} = [\mathbf{n_{P'}}, \mathbf{d_{P'}}, \mathbf{n_{P'}} \times \mathbf{d_{P'}}] \cdot [\mathbf{n_P}, \mathbf{d_P}, \mathbf{n_P} \times \mathbf{d_P}]^{-1}. \quad (16)$$

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

$$\mathbf{T_P} = \mathbf{P'} - \mathbf{R_P} \cdot \mathbf{P}. \quad (17)$$

2) Dimension Reduction in Random Sampling: In the random sampling, the search dimension is too large to efficiently obtain good results. We introduce a reduced-DoF random sampling by only generating a random low-dimension 2-DoF translation $\mathbf{t_p}$. The random sampling in the 2D translation space is a uniform sampling in a local window (e.g. 41*41 in all the experiments). Following the original PatchMatch method, we compute the matching cost of each random sampling value and accept it if its matching cost is lower than the current cost. All of these motion patterns are considered as the motion proposal set.

Similar to the situation in the initialization step, we adopt the prominent orientations in SIFT feature detection [34] as the 2D prominent direction vectors of pixel $\mathbf{p}$ in the reference image and pixel $\mathbf{p} + \mathbf{t_p}$ in the target image. The following computation is then the same as in the situation when we convert the 2D motion field to a 3D over-parameterized scene flow in the section *Initialization Based on 2D Motion*. We finally acquire a 6-DoF motion from a 2-DoF random guess using this reduced-DoF random search, and the dimension of random search for the 3D scene flow case is then significantly reduced from six to two.

C. Proposal Fusion

We term the set of existing motion patterns from our RGB-D PatchMatch method as *the motion proposal set* $\{\mathbf{V}_{prop}\}$. In our method, the combined motion and occlusion proposals are the Cartesian product of *the motion proposal set* and of the occlusion value range: $(\mathbf{V}, O)_{prop} \in \{\mathbf{V}_{prop}\} \times \{0, 1\}$. We integrate all the possible combined motion and occlusion labels using the FusionMoves [32] method using QPBO [35] to minimize the energy function in Eq. 1. By fusing the proposals in our energy function framework, we can select and preserve the best parts from each proposal. FusionMoves iteratively integrates the combined motion and occlusion proposals until a stopping criterion is satisfied, and then it outputs the final scene flow result.

V. Experimental Results

To analyze the performance of the proposed method, we comprehensively evaluate our method on the Middlebury dataset and various challenging RGB-D data captured by Kinect cameras. In the first experiment, we focus on the accuracy of the overall method and the efficiency of our RGB-D PatchMatch method. To test the accuracy of the overall method, we apply the method on the Middlebury dataset [36][37] in the same way as prior RGB-D scene flow methods [8][7][4][10]. For the computational efficiency of our RGB-D PatchMatch method, we also compare it with the corresponding methods on the Middlebury dataset. In the second experiment, we focus on the robustness of our method for challenging large displacement motion and occlusion on the RGB-D image pairs proposed by [10]. In the third experiment, we focus on the robustness of our method for challenging deformable surfaces (e.g. tshirt) in the Kinect RGB-D dataset [38]. In the fourth experiment, we focus on the robustness and accuracy of our method for long RGB-D video sequences.

In all the following experiments, we use millimeter as the unit of spatial distance measure, and $[0, 255]$ for the color value range. For the 2D motion method NRDC [33], we use its default parameters. The parameter setting is: $\lambda = 0.8$, $\lambda_{\nabla I} = 10.0$, $\lambda_D = 1.5$, $k = 0.2$, $\sigma_s = 0.5 \cdot k \cdot (Z_\mathbf{P} + Z_\mathbf{Q})$, $\sigma_n = 0.3 \cdot \sigma_s$, and $\xi = 600.0$. As shown in Figure 3, we choose the best value for parameter λ in which there is lowest error under three different evaluation criteria mentioned in Sec. V(A). We use the same heuristic approach to choose the proper values for other parameters. The iteration for our RGB-D PatchMatch is set at 2. Following [8][7][4][10], pixels without depth values are not assigned motion values, and are excluded from comparison. The average running time for one image pair is around 5 minutes using a Matlab and C++/MEX implementation on a desktop PC of Intel Core i5 CPU and 4 GB RAM as shown in Tab. II, which is in the middle level of all the motion estimation methods. Since the proposed RGB-D PatchMatch method and the FusionMoves method employed can be executed in parallel with slight modification, our scene flow algorithm has the potential to be further accelerated on a GPU hardware configuration.

A. Quantitative Evaluation on Accuracy and Efficiency

The RGB-D images in the Middlebury dataset were captured by a set of cameras which are parallel and equally spaced along the X axis at the same time. There are 9 images in the Middlebury dataset for each scene. Following all the existing methods, we took the color images and ground truth disparity maps of frames 2 and 6 of the Middlebury Venus, Cones and Teddy sets as the reference and target RGB-D images. Our approach is compared with three optical flow methods [22][39][33], two stereo-based scene flow methods [3][2] and five RGB-D data-based scene flow methods [8][7][16][4][10][11]. Following [2][3][4], we used end point error ($\mathrm{RMS_O}$), disparity change error ($\mathrm{RMS_Z}$) and average angular error (AAE) as the error measurement criteria. For the stereo-based methods [3][2], the scene flow

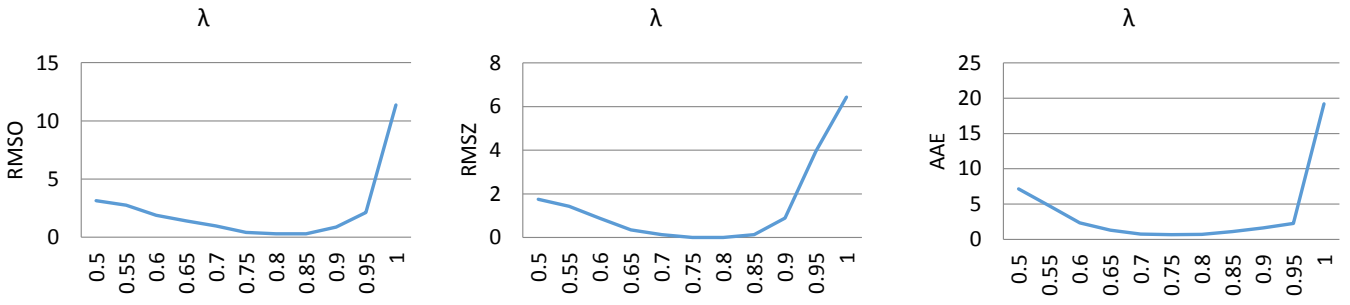

Fig. 3. We select the best value for parameter λ which has the lowest error under three different criteria (RMS_O, RMS_Z and AAE).

TABLE II
THE EVALUATED ERRORS UNDER DIFFERENT EVALUATION CRITERIA AND APPROXIMATE AVERAGE RUNNING TIME OF COMPARED METHODS .

Methods	Venus			Cones			Teddy			Average time
	$\mathrm{RMS_O}$	$\mathrm{RMS_Z}$	AAE	$\mathrm{RMS_O}$	$\mathrm{RMS_Z}$	AAE	$\mathrm{RMS_O}$	$\mathrm{RMS_Z}$	AAE	
Brox *et al.* [22]	0.72	0.14	1.28	2.83	1.75	0.39	3.20	0.47	0.39	30 secs
Xu *et al.* [39]	0.30	0.22	1.43	1.66	1.15	**0.21**	1.70	0.50	0.28	7 mins
Huguet *et al.* [3]	0.31	N/A	0.98	1.10	N/A	0.69	1.25	N/A	0.51	5 hours
Basa *et al.* [2]	0.16	N/A	1.58	0.58	N/A	0.39	0.57	N/A	1.01	N/A
Zhang *et al.* [8]	**0.15**	N/A	1.15	1.04	N/A	0.69	0.73	N/A	0.66	N/A
Quiroga *et al.* [7]	0.31	**0.00**	1.26	0.57	0.05	0.42	0.69	0.04	0.71	20 secs
Hadfield *et al.* [16][4]	0.36	0.02	1.03	1.24	0.06	1.01	0.83	0.03	0.83	10 mins
Hornáček *et al.* [10]	0.26	0.02	**0.53**	0.54	0.02	0.52	**0.35**	0.01	**0.15**	N/A
Quiroga *et al.* [11]	N/A	N/A	N/A	0.45	N/A	0.37	0.49	N/A	0.46	2 mins
NRDC [33]	5.65	N/A	16.2	15.5	N/A	18.3	17.7	N/A	14.3	15 secs
Our method	**0.15**	**0.00**	1.17	**0.33**	**0.00**	0.39	0.40	**0.00**	0.50	5 mins

and disparity were jointly estimated using frames 2, 4, 6 and 8 of Middlebury Cones, Teddy, and Venus in a similar way to the existing methods. For the two optical flow techniques [22], [39], $\mathrm{RMS_Z}$ was computed by estimating 3D translational flow by interpolating the 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 publicly available codes with default parameters.

Table II shows that our method is the top performer among all these optical flow and scene flow algorithms, under most evaluation criteria. Note that, to compare the results across different optical flow methods, here AAE stands for Average Angular Error of the motion on the 2D XY- image plane without considering the angular changes in Z direction. The proposed method should demonstrate better performance in which considering a more comprehensive case where all orientations of X, Y, Z are taken into account. However, if only the XY- image plane is considered, the proposed method may not fully demonstrate its advantages. This is a reason why the proposed method does not always demonstrate the best performance under this weak evaluation criterion. An interesting observation is that although the results of the NRDC method are noisy and non-smooth, our algorithm is still able to extract and preserve potentially correct motion patterns in the final output. Figure 4 gives the estimated scene flow and occlusion results using our method, and it can be seen they are very close to the ground truth.

From Table II, we can also see two trends among the optical flow methods, stereo-based scene flow, and RGB-D data-based scene flow methods. The first trend is that the average performance of the scene flow methods is better than the optical flow methods as a result of the use of various depth cues. The other trend is that the overall performance of the RGB-D based scene flow methods is better than that of the stereo-based scene flow methods. This is because the depth data obtained by the structured-light sensors given in the Middlebury dataset are more precise and more reliable than the estimated depth used in stereo vision-based algorithms. Thus, the overall performance of an algorithm depends on both motion estimation and depth estimation approaches.

We further compare our RGB-D PatchMatch method with the original PatchMatch version [13][14] and the adapted 3D version in [4]. For comparison, all the PatchMatch methods employ the same cost function in Eq. 13. From Figure 5, it can be seen that the original PatchMatch [13][14] for the over-parameterized scene flow representation is extremely slow to converge. Our initialization strategy for RGB-D PatchMatch provides good initial values with low energy on all three RGB-D image pairs: Middlebury Venus, Cones and Teddy. This also reflects that our reduced-dimension random sampling

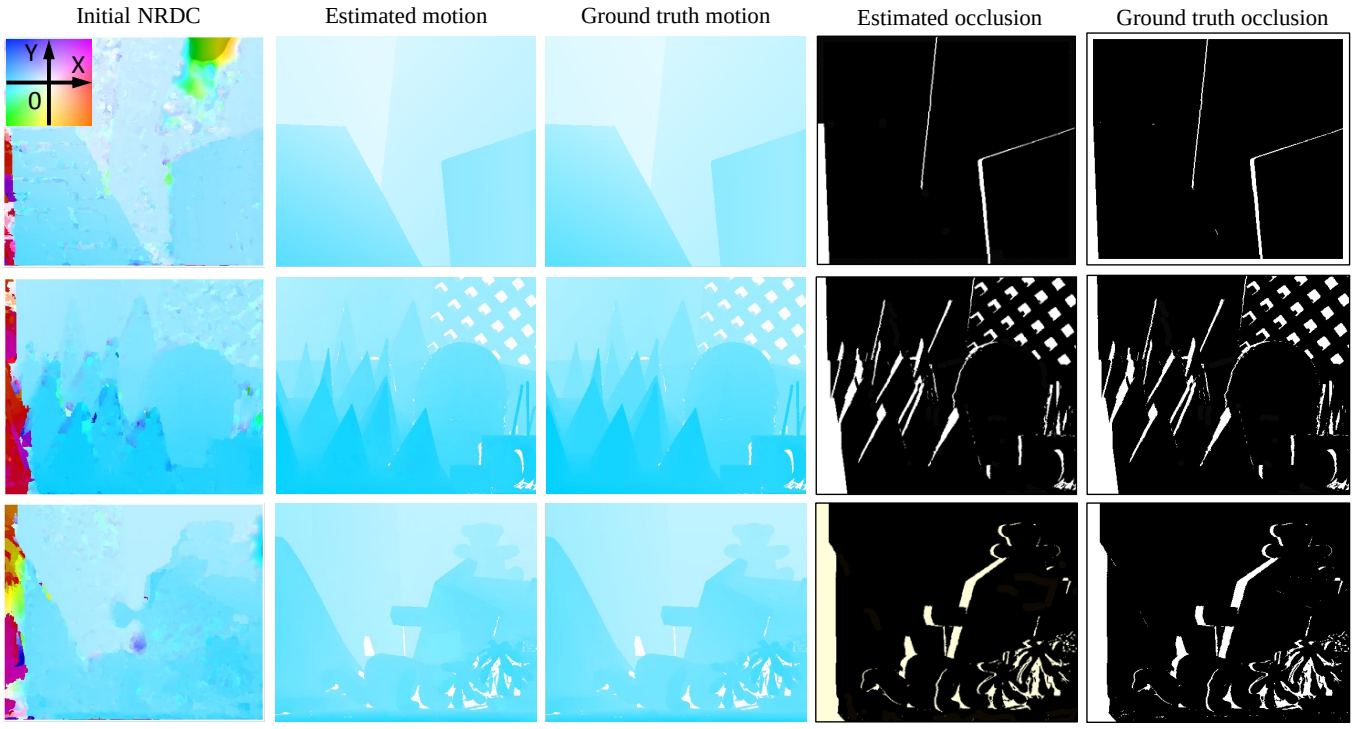

Fig. 4. 2D XY motion by projection of 3D scene flow on the image plane using Middlebury color coding. From left to right, these images are the result of initial 2D motion, estimated scene flow, ground truth motion, estimated occlusion, and ground truth occlusion. From top to bottom, these images are the results of Middlebury Venus, Cones, and Teddy respectively. The XY-motion maps are rendered using the Middlebury coloring method. For scene flow, 3D displacements are projected to image space to obtain 2D XY-motion. Our final scene flow and occlusion results are visually close to ground truth on the Middlebury dataset, although the results of the NRDC method are noisy and non-smooth.

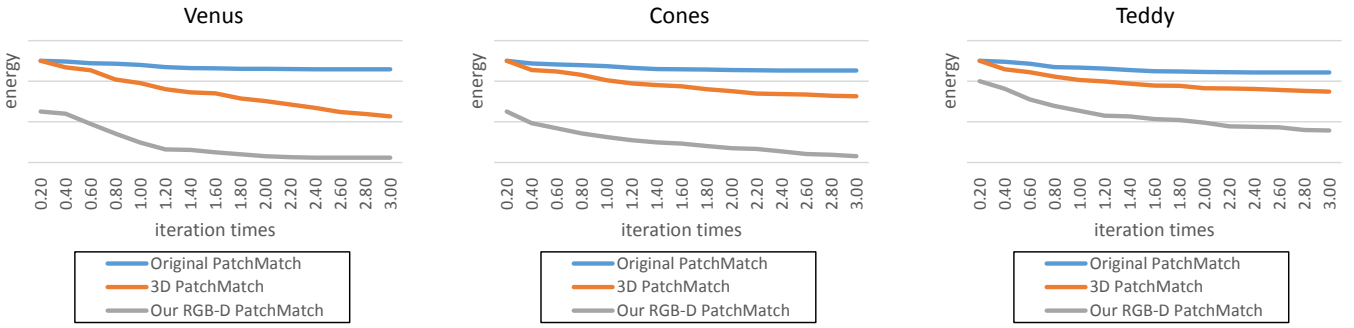

Fig. 5. Performance of the original PatchMatch [13][14], 3D PatchMatch [4] and our RGB-D PatchMatch on Middlebury Venus, Cones and Teddy. The initialization strategy for our RGB-D PatchMatch provides good initial values with low energy on all three RGB-D image pairs, and the reduced-dimension random sampling strategy helps to minimize the cost values efficiently in less than three iterations. Note that, we check the energy value after every 20% pixels is processed in one iteration.

strategy minimizes the cost values more efficiently than the 3D version [4].

Conclusions Our method outperforms other optical flow and scene flow methods on the Middlebury dataset, which is employed as the quantitative evaluation for accuracy by most existing scene flow methods. Compared to the original PatchMatch [13][14] and the adapted 3D version in [4], our RGB-D PatchMatch method is more efficient and effective.

B. Qualitative Evaluation on Robustness for Challenging Large Displacement Motion and Occlusion

It is not enough to fully analyze the performance of our scene flow method, since the scene flow is not complex in the Middlebury dataset. Thus, we tested our method on the Kinect RGB-D image pairs from [10]. The Kinect RGB-D image pairs *Nici*, *Stefan* and *Largehand* from [10] are challenging, since they contain both extremely large displacement motion of human bodies and corresponding occlusion between the foreground objects and the background scene. The motion displacement of some regions is larger than the size of objects (e.g. the moving hand in the RGB-D image pair *Largehand*), which can be easily observed in Figure 6.

Using visual illustration, we qualitatively compare our method with one large displacement optical flow method [22] and four RGB-D data-based scene flow methods [4][6][7][10]. To better illustrate the results, we create XY motion maps

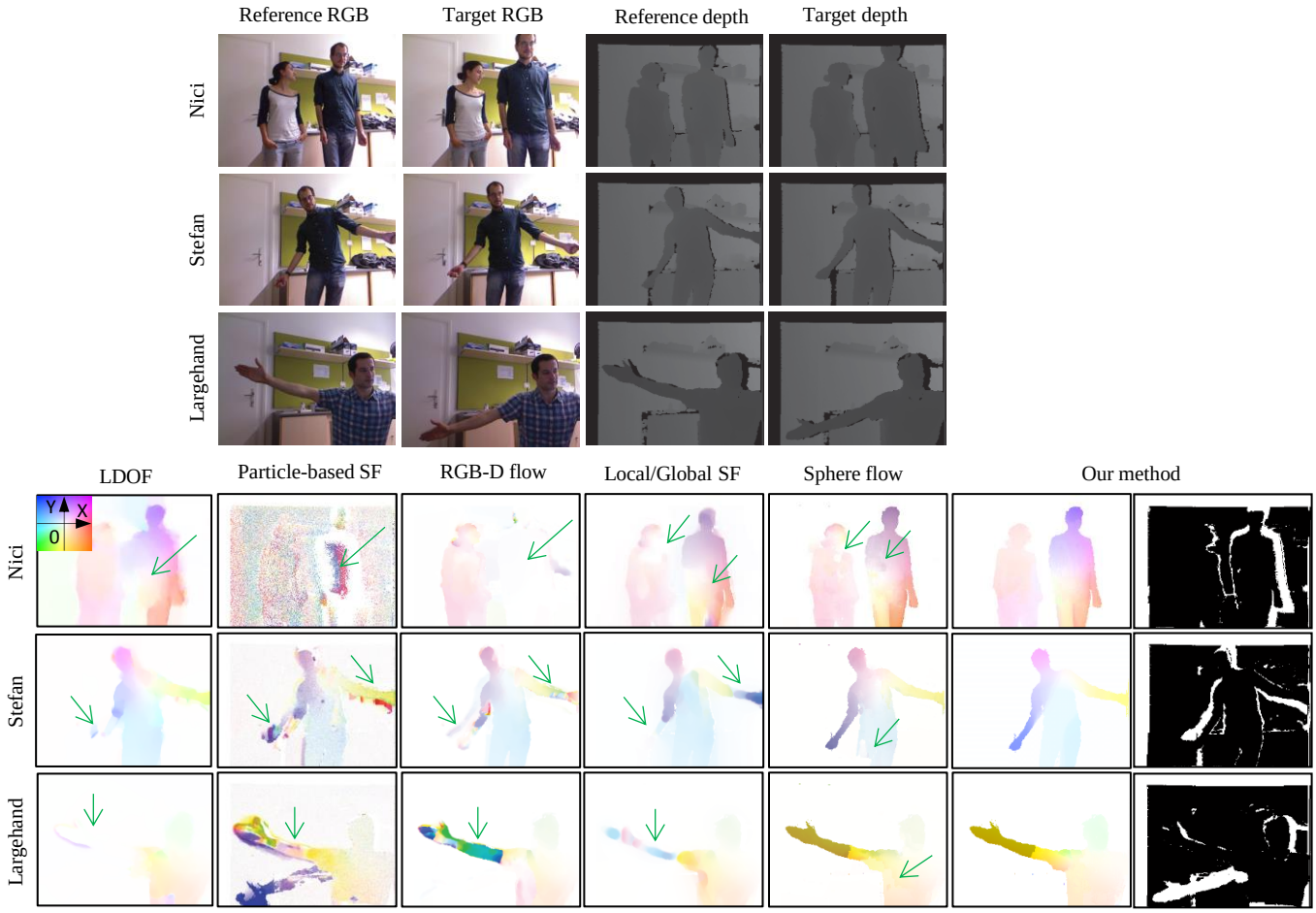

Fig. 6. Scene flow on RGB-D image pairs Nici, Stefan and Largehand from [10]. Across the top three rows, from left to right, they are the reference RGB images, target RGB images, reference depth images, and target depth images, respectively. Across the bottom three rows, they are the XY-motion maps from large displacement optical flow (LDOF) [22], particle-based scene flow [16][4], RGB-D flow [6], local/global scene flow [7], sphere flow [10], and our method (both scene flow and occlusion), respectively. The five competing methods suffer from a variety of problems. For example, the sphere flow fails to estimate the scene flow on some texture-less regions and near the occluded regions marked by green arrows, and generates some non-smooth artifacts. It is clear that the other four competing methods fail to capture the large displacement motion in the scene, and some of them introduce haze artifacts on the occluded regions and motion boundaries. Only our method generates accurate motion results without suffering from haze or non-smooth artifacts. Note that, the yellow error of our result at the right boundary of *Stefan* is mainly caused by the depth sensing error on the original depth images.

to show the motion projection on the 2D image plane. The results of the large displacement optical flow method [22] are reproduced using the publicly available code, while the results of other scene flow methods [4][6][7][10] are based on the results provided in [10].

Figure 6 illustrates the 2D XY-motion maps of all the competing methods. These maps are illustrated based on Middlebury color coding [40]. The optical flow method [22] and the three scene flow methods [4], [6] and [7] clearly fail to capture the large displacement motion of the human bodies and hands, and/or have some haze artifacts on the XY-motion maps. SphereFlow [10] captures the majority of motion patterns in these image pairs. However, it fails to estimate the motion on some texture-less regions and near the occluded regions marked by the green arrows. This is because the method relies heavily on the initial search results of 3D PatchMatch, and the consistency checking in the postprocessing occlusion detection is not able to locate these falsely matched positions. Consequently, it generates some non-smooth artifacts on the regions marked by the green arrows. The proposed method successfully captures the motion of bodies and arms while appropriately estimating the motion of the occluded background.

Conclusions Our method is capable of tackling extremely large displacement motion of objects where the motion magnitude is much larger than the size of the object. Compared with methods [22][4][6][7][10] using postprocessing occlusion detection or no occlusion handling, our method also performs robustly in the presence of occlusion between the foreground human bodies and background scene, and generates accurate final motion results without haze or non-smooth artifacts.

C. Qualitative Evaluation on Robustness for Challenging Deformable Surfaces

We also test our method to assess the robustness of the proposed algorithm in the presence of deformable surfaces. We select the *Tshirt4* sequence from the Kinect RGB-D dataset [38], which is a sequence of a fast moving deformable

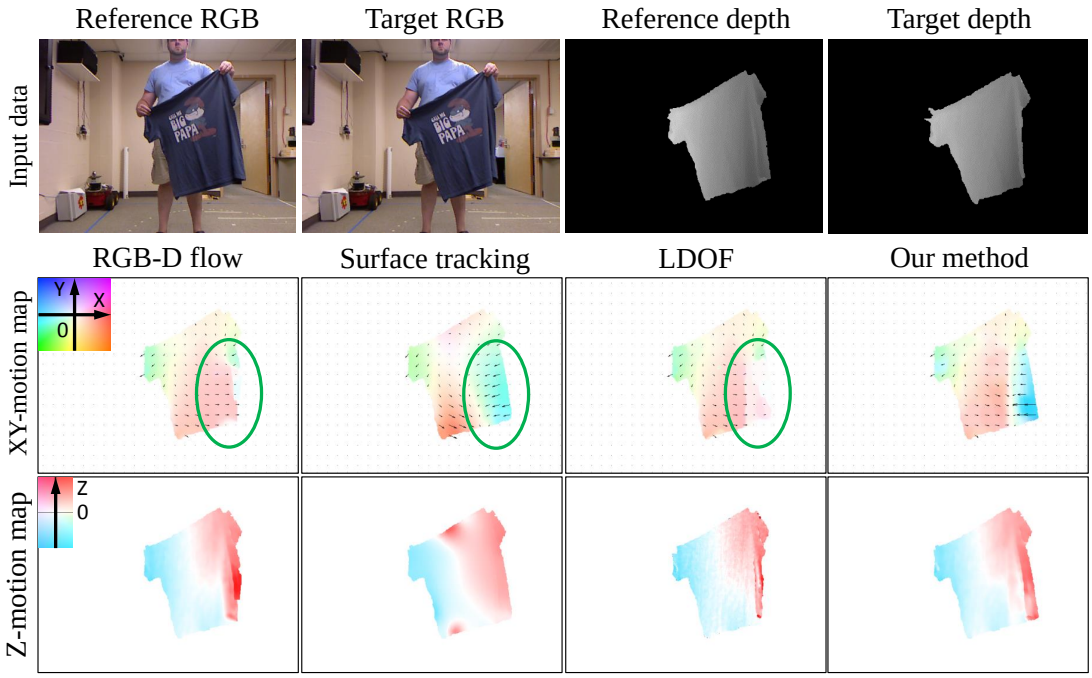

Fig. 7. Scene flow results of two frames in the Tshirt4 sequence. Row 1: Input reference and target RGB-D images. For clarity, we only visualize the depth values of the foreground. Row 2-3: XY-motion (optical flow) maps and Z-motion maps of the RGB-D flow [6], surface tracking [38], large displacement optical flow [22]. The RGB-D flow method and the optical flow method do not capture the motion of the right side of the tshirt marked by the green circles, and the surface tracking method over-smooths the region, while our method works robustly.

tshirt. The dataset [38] contains similar deformable surfaces of clothes (such as tshirts, shorts, trousers) held by hands. We compare the performance of our algorithm with three alternative methods with publicly available code implementations: the optical flow method [22], the *RGB-D flow* [6], and the 3D surface tracking method proposed with the Kinect RGB-D video sequences dataset [38]. Note that the 3D surface tracking method also generates the 3D motion of a deformable surface by estimating the parameters of the surface with an initialization of the surface position at the first frame. The Tshirt4 sequence has 200 frames, and we visually illustrate the scene flow result of frame 58 and frame 61 in this sequence. Here, we create both a 2D XY motion map and a Z motion map to show the motion projection on the 2D image plane and the motion along the depth Z direction for each scene flow result, respectively. These maps are also illustrated according to Middlebury color coding [40]. To visualize the Z motion map, the values along the x-axis of the Middlebury color coding map are employed. For comparison, the scene flow from the optical flow method can be computed by back-projecting 2D motion to the 3D space domain using camera intrinsic parameters and depth values.

Figure 7 shows the results of estimating the scene flow between frame 58 and frame 61 in the Tshirt4 sequence. As shown in rows 2 and 3, the RGB-flow method [6] and the optical flow method fail to capture the distinct motion of the right side of the tshirt marked by the green circle, due to occlusion. The surface tracking method over-smooths the region, while our method robustly estimates the scene flow. This difference reflects the advantages of the occlusion modeling employed by our framework.

Conclusions Our method performs robustly on the deformable surface of the tshirt without over-smoothing the scene flow results.

D. Qualitative and Quantitative Evaluation on Robustness and Accuracy for Long RGB-D Video Sequences

We verify the robustness and accuracy of the proposed algorithm on long RGB-D video sequences. We select the *Tshirt4* sequence from the Kinect RGB-D dataset [38]. The dataset [38] contains similar deformable surfaces of clothes. Our sequence, called *Human Hand Waving*, serves as a complement, because the articulated deformation of the human body is quite different from the deformation of the clothes. The Tshirt4 sequence has 200 frames, and our Human Hand Waving sequence has 400 frames. In detail, we run our method on all the RGB-D image pairs at frame t (reference image) and frame $t + \Delta t$ (target image) for both sequences. We also compare the performance of our algorithm with the three alternative methods [22][6][38].

For the quantitative evaluation of scene flow accuracy, we use a sparse set of hand-tracked points of large displacement motion, since it is prohibitively expensive to label correspondences for every point in the two RGB-D sequences. The displacement of these points serves as a ground truth scene flow. We use two metrics to evaluate these methods. One metric is the mean value and standard deviation of the error under situations with regular or large displacement motion. The other metric is the accumulative RMSE of tracking these landmark points along the sequences.

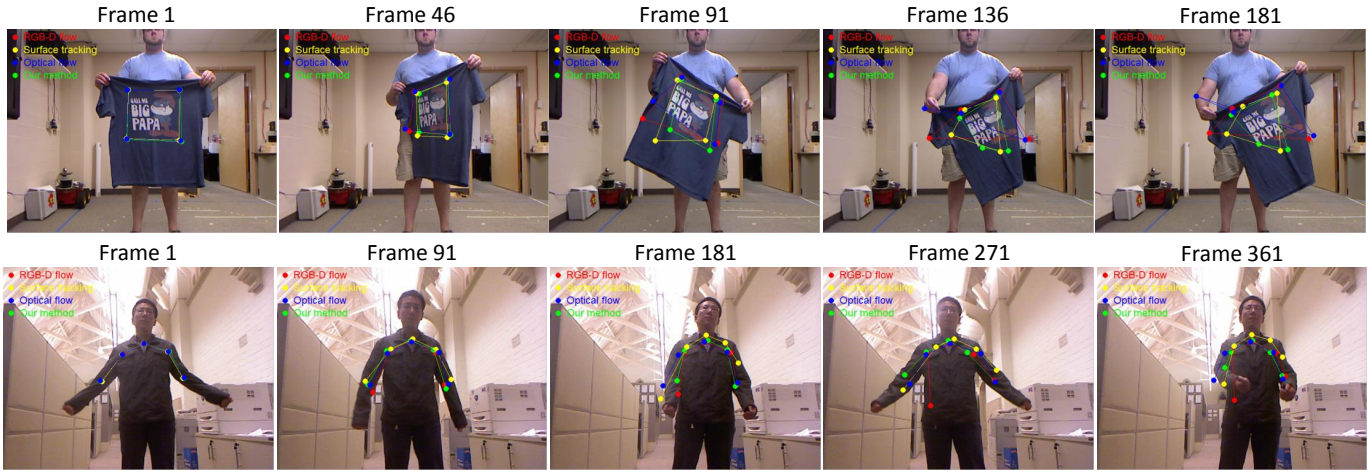

Fig. 8. Tracking results on Tshirt4 sequence and Human Hand Waving sequence. The top row is the Tshirt4 sequence, and the bottom row is the Human Hand Waving sequence. The tracking results of the RGB-D flow method [6], surface tracking method [38], large displacement optical flow method [22] and our method are marked in red, yellow, blue and green, respectively. The three competing methods suffer from serious drift problems. For example, on the Human Hand Waving sequence, points are tracked from the foreground human body to the background scene at frame 181 for the surface tracking method [38] and at frame 361 for the optical flow method [22], and points are tracked from the human hands to the human torso at frame 361 for the RGB-flow method [6]. Similar situations are seen in the Tshirt4 sequence.

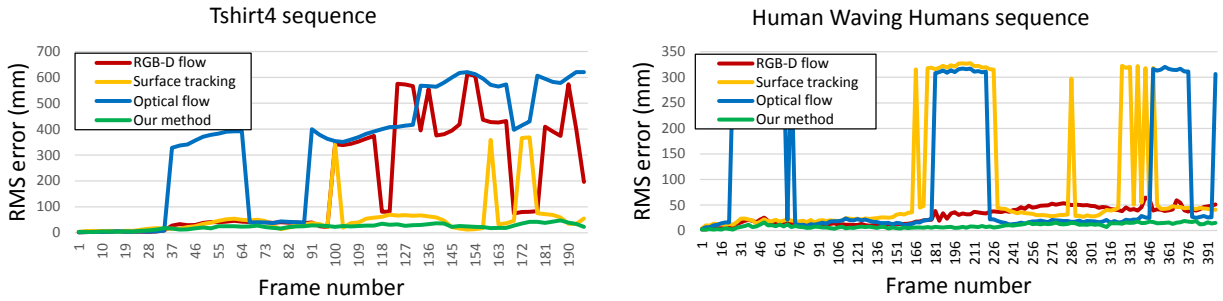

Fig. 9. Accumulated error of landmark points on Tshirt4 sequence and Human Hand Waving sequence under large-displacement motion ($\Delta t = 3$).

TABLE III
THE MEAN VALUE AND STANDARD DEVIATION OF THE TRACKING ERROR (MM) UNDER REGULAR AND LARGE DISPLACEMENT MOTION SITUATION IN *Tshirt4* AND *Human Hand Waving* SEQUENCES.

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 [6]	3.5	2.0	18.8	56.4	6.6	**3.9**	11.3	14.9
Surface tracking [38]	7.4	4.2	71.9	128.8	11.0	15.1	39.0	129.0
Optical flow [22]	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**

We evaluate all the methods on the sequences under two configurations with different time intervals 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 configurations by considering them as the regular displacement ($\Delta t = 1$) and the large displacement ($\Delta t = 3$) motion scenarios, respectively. Table III depicts the mean and standard deviation of error (3D Euclidean distance from the ground truth) with regular and large displacement scenarios in *Tshirt4* and *Human Hand Waving* sequences. From the table, we observe that our method achieves results that comparable to the other three state-of-the-art methods in the regular displacement scenario. In the large displacement situation, our method performs much better and reaches the lowest mean and standard deviation of error. This further proves the robustness of the proposed method in dealing with large displacement 3D motion.

We also compare all the methods in the cases of landmark point tracking. For each sequence, we manually label a number of landmark points in the first frame and their corresponding points in the following frames. These landmark points have

large displacement motion along the video sequences. The positions of landmark points along the sequences serve as the ground truth for measuring the accuracy of the motion estimation methods. We use the landmark point positions of frame t plus the estimated raw motion vectors as the prediction of the point positions in frame $t + \Delta t$. To more clearly show the performance of different methods on large displacement motion, Δt is also set at 3 in all the comparisons. These landmark points are visible in the entire sequence, thus a good motion estimation method should be able to robustly track these points along the sequences using raw motion vectors, and have low cumulative RMS error (3D Euclidean distance from the ground truth).

Figure 8 shows the tracking results of all methods on the two sequences. On both sequences, the three competing methods have serious drift problems. As can be seen from Figure 9, our method has the least accumulated error of all the methods in both sequences.

Conclusions Compared with other methods [22][6][38], our method performs more robustly on the two long RGB-D video sequences. When tracking landmark points with large displacement motion on the moving objects along the RGB-D video sequences, our method has less accumulated error than the other three methods, which proves the robustness and accuracy of our method on long RGB-D video sequences.

VI. Conclusions

In this paper, we have presented a method to address two challenging problems in RGB-D data-based scene flow estimation: occlusion and large displacement motion. We explicitly modeled occlusion, and jointly estimated the scene flow and occlusion, based on the observation of depth order and feature difference. For large displacement motion, we employed an over-parameterized presentation and proposed a novel two-stage optimization procedure for it. In the first stage, an efficient RGB-D PatchMatch method was proposed to provide large displacement motion patterns. In the second stage, our method preserved and integrated the potentially correct motion patterns using FusionMoves. We showed the accuracy, efficiency and robustness of our method on the Middlebury datasets as well as on various challenging Kinect RGB-D image pairs and long video sequences with large displacement motion and occlusion.

References

[1] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013.

[2] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *International journal of computer vision*, vol. 101, no. 1, pp. 6–21, 2013.

[3] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–7.

[4] S. Hadfield and R. Bowden, "Scene particles: Unregularized particle based scene flow estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 3, pp. 564–576, 2014.

[5] J.-M. Gottfried, J. Fehr, and C. S. Garbe, "Computing range flow from multi-modal kinect data," in *Advances in Visual Computing*. Springer, 2011, pp. 758–767.

[6] E. Herbst, X. Ren, and D. Fox, "RGB-D flow: Dense 3-D motion estimation using color and depth," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013.

[7] J. Quiroga, F. Devernay, J. L. Crowley *et al.*, "Local/global scene flow estimation," in *ICIP-IEEE International Conference on Image Processing*, 2013.

[8] X. Zhang, D. Chen, Z. Yuan, and N. Zheng, "Dense scene flow based on depth and multi-channel bilateral filter," in *Computer Vision–ACCV 2012*. Springer, 2013, pp. 140–151.

[9] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, and R. Kimmel, "Group-valued regularization for analysis of articulated motion," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 52–62.

[10] M. Hornacek, A. Fitzgibbon, and R. Carsten, "Sphereflow: 6 DoF scene flow from RGB-D pairs," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

[11] J. Quiroga, T. Brox, F. Devernay, and J. Crowley, "Dense semi-rigid scene flow estimation from RGBD images," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 567–582.

[12] Y. Niu, A. Dick, and M. Brooks, "Compass rose: A rotational robust signature for optical flow computation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 1, pp. 63–73, Jan 2014.

[13] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[14] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 29–43.

[15] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-Huber: PatchMatch with Huber regularization for stereo matching," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2360–2367.

[16] S. Hadfield and R. Bowden, "Kinecting the dots: Particle based scene flow from depth sensors," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2290–2295.

[17] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 492–504, 2009.

[18] M. Mohamed, H. Rashwan, B. Mertsching, M. Garcia, and D. Puig, "Illumination-robust optical flow using a local directional pattern," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 9, pp. 1499–1508, Sept 2014.

[19] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 399–406.

[20] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *International journal of computer vision*, vol. 97, no. 3, pp. 322–338, 2012.

[21] C. Vogel, S. Roth, and K. Schindler, "View-consistent 3D scene flow estimation over multiple frames," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 263–278.

[22] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 3, pp. 500–513, 2011.

[23] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2443–2450.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] Y. Eshet, S. Korman, E. Ofek, and S. Avidan, "DCSH-matching patches in RGBD images," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 89–96.

[26] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis, "The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces," *Advances in neural information processing systems*, vol. 17, pp. 33–40, 2005.

[27] C. Vogel, S. Roth, and K. Schindler, "An evaluation of data costs for optical flow," in *Pattern Recognition*. Springer, 2013, pp. 343–353.

[28] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[29] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 72–91, 2008.

[30] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.
[31] A. B. Hamza and H. Krim, "Geodesic object representation and recognition," in *Discrete Geometry for Computer Imagery*. Springer, 2003, pp. 378–387.
[32] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for markov random field optimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 8, pp. 1392–1405, 2010.
[33] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 70.
[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[35] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete applied mathematics*, vol. 123, no. 1, pp. 155–225, 2002.
[36] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.
[37] ——, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–195.
[38] B. Willimon, S. Hickson, I. Walker, and S. Birchfield, "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, pp. 2711–2717.
[39] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1744–1757, 2012.
[40] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.