Rigid Scene Flow for 3D LiDAR Scans

Ayush Dewan Tim Caselitz Gian Diego Tipaldi Wolfram Burgard

***Abstract*— The perception of the dynamic aspects of the environment is a highly relevant precondition for the realization of autonomous robot system acting in the real world. In this paper, we propose a novel method for estimating dense rigid scene flow in 3D LiDAR scans. We formulate the problem as an energy minimization problem, where we assume local geometric constancy and incorporate regularization for smooth motion fields. Analyzing the dynamics at point level helps in inferring the fine-grained details of motion. We show results on multiple sequences of the KITTI odometry dataset, where we seamlessly estimate multiple motions pertaining to different dynamic objects. Furthermore, we test our approach on a dataset with pedestrians to show how our method adapts to a case with non-rigid motion. For comparison we use the ground truth from KITTI and show how our method outperforms different ICP-based methods.**

I. INTRODUCTION

Perceiving the dynamics of the environment of a robot provides highly relevant information about which aspects change or how the environment might evolve in the future. The dynamics of a scene can be studied either at object or point level. Having pointwise motion or a dense motion field provides major advantages. First, it can help in inferring different motions in the scene without using any prior knowledge. Second, it allows the robot to reason about the underlying cause of the motion. For instance, the rigid motion field for points sampled from a static scene can facilitate the estimation of the sensor motion. Third, it assists in capturing the fine details of dynamics, enabling better semantic understanding. For example, it can help the robot deal with human motion, which is typically non-rigid.

Pointwise motion has garnered a lot of attention mainly in the computer vision community. Various methods have been proposed for estimating 2D pointwise motion in images using color [5]. With the recent advent of affordable depth sensors, different methods exist for scene flow estimation using color and depth images [9], [10], [18]. These methods cannot be applied directly to 3D point cloud data due to inherent differences in the problem structure. First, the constancy assumptions for brightness or gradients are not valid for LiDAR data. The intensity values from LiDAR are often unreliable as they also depend on the angle of inclination. Furthermore, due to the sparse nature of LiDAR data, gradients are not well defined. Second, the concept of the neighborhood is well understood for images (fixed size image patch), whereas a similar well defined structure does not exist for 3D data. Third, most of these methods assume a linear motion (translation), which is justified if data is collected at a high frame rate (30Hz). In cases, in which data is not collected at a sufficiently high frame rate (with a LiDAR scanner at 10Hz), the assumption of linear motion cannot be justified.

All authors are with the Department of Computer Science at the University of Freiburg, Germany. This work has been partially supported by the European Commission under the grant numbers ERC-AG-PE7-267686-LifeNav, FP7-610603-EUROPA2 and FP7-610532-SQUIRREL

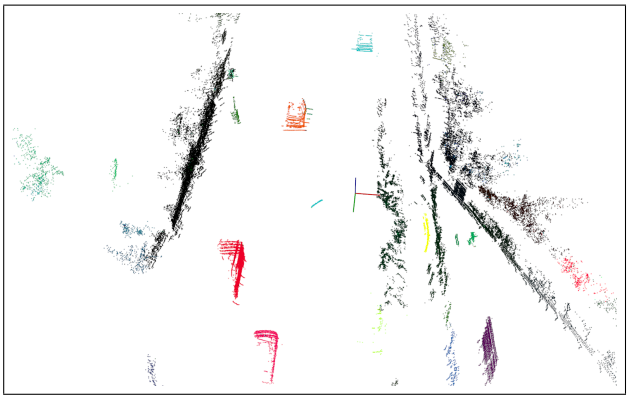


Fig. 1: Pointcloud colorized according to the translational component of the estimated rigid motion flow. Black points represent the static scene. The three red/orange cars have similar hue values since they are moving in same direction, while the green car moves in the opposite direction.

In this paper we propose a novel approach for estimating rigid scene flow that addresses all of the aforementioned challenges. We formulate the problem as an energy minimization problem. Our first contribution is introduction of the concept of *geometric constancy*, i.e., that the local structure is not deformed due to the motion. Next contribution is introduction of a novel neighborhood structure for 3D point clouds. Our approach approximates the scene using triangular meshes and considering two points as neighbors only if they are vertices of the same triangle. Lastly, we estimate the complete 6D rigid motion, instead of linear translational motion only. With these contributions, our proposed method can estimate dense rigid scene flow for 3D LiDAR data.

Figure 1 shows the rigid motion flow estimated by our method. We test our approach on multiple sequences of the KITTI odometry dataset [6] and demonstrate how our method effortlessly estimates arbitrary motion in the scene. For comparison, we use the ground truth and evaluate our approach against ICP-based methods. For the cases, in which ground truth is not available, we quantify our motion estimate by measuring the alignment between scans. Furthermore, we show advantages of having pointwise motion by testing our approach on a dataset with pedestrians. The experiments reveal that our method adapts to the case of non-rigid objects.

II. RELATED WORK

The problem of estimating motion flow has been studied intensively in the past. The different developed methods can be distinguished according to the dimension of the motion field. *Optical flow* [5] describes 2D translation motion in image plane, *sceneflow* [9], [22], [10] describes 3D translation motion, and the *rigid scene flow* [18], [17] describes the rigid motion.

Fortun *et al.* [5] provides extensive literature review for optical flow. Building on optical flow, Vedula *et al.* [22] introduced the term scene flow. They included first order approximations of the depth map to estimate 3D flow. Herbst *et al.* [9] extended the approach presented by Brox *et al.* [1] and include a depth constraint to estimate scene flow. Jaimez *et al.* [10] introduced a real time, primal-dual algorithm based method to estimate scene flow for RGB-D data. However, these methods make assumptions which for reasons discussed in Section I are not valid for our case.

For estimating dense semi-rigid flow for RGB-D data, Quiroga *et al.* [18] solved an energy minimization problem, using TV regularization to estimate piecewise smooth motion. Vogel *et al.* [23] proposed a method to estimate the 3D motion and structure using RGB-D data. The main contribution of their work is using local rigid regularization instead of variational regularization. Newcombe *et al.* [17] proposed a method for dense SLAM using RGB-D scans, where they reconstruct deforming surfaces and simultaneously estimate dense volumetric 6D motion field. These methods show commendable results but they use color and depth images, while our approach relies on sparse depth data, therefore a direct comparison is infeasible.

The methods discussed so far show results mainly for indoor scenes. For outdoor environments, Menze *et al.* [15] propose a method for object scene flow using images. They over segment the scene into super pixels and using CRF jointly estimate rigid motion and an association between super pixels and objects. Even though their assumption about rigid structure of the outdoor environment is not invalid but our assumption of local rigidity enables our method to estimate motion of a non-rigid object (incase of humans), making our method more robust towards change in the environment.

Various methods have been proposed for tracking using 3D LiDAR data in outdoor environments. Kaestner *et al.* [11] proposed a generative Bayesian approach for detecting and tracking dynamic objects. Moosmann *et al.* [16] used a segmentation method based on local convexity for detecting object hypotheses and combined ICP and a Kalman filter for tracking. For 2D LiDAR data Tipaldi *et al.* [19] and Van De Ven *et al.* [21] proposed methods for detecting and estimating motion using CRF. In our previous work [3] we proposed a method for detecting and tracking in 3D LiDAR scans. We estimated multiple rigid motion hypothesis using RANSAC and used a Bayesian approach to associate points in the scene to different motion hypotheses. In this paper, we rather focus on estimating pointwise motion and do not reason about motion at object level. However, our current method can be extended for segmenting dynamic objects on the basis of motion.

Our assumption of local rigidity allows for deformation of surfaces. Addressing this problem, Hähnel *et al.* [7] proposed an approach extending ICP for registering deformable surfaces for sparse LiDAR data. Similar to us they perform pointwise estimation, allowing for smooth deformation of the surface. The main difference between our method and their method is that we estimate feature based correspondences, while they use nearest neighbors for data association. The assumption of nearest neighbors breaks if two surfaces are far from each other, for instance when a dynamic object moves in the direction opposite to the direction of sensor motion. Our feature based method is immune to these cases and can estimate large motion. For registering deformable 3D surfaces, Praveen *et al.* [4] extended the approach by Hähnel *et al.* [7] by introducing the concept of correlated correspondences, where they estimate pointwise deformation and correspondence. Our approach adjusts to the cases of non-rigid objects but in this paper we are not concentrating on registering deformable dense 3D surfaces but instead estimating pointwise motion for sparse LiDAR data, therefore a comparison with the work of Praveen *et al.* [4] and the approach of Cosmo *et al.* [2] is beyond the scope of this paper.

III. PROBLEM FORMULATION

A LiDAR scan P is given by a set of 3D points.

$$P = \{p_k \mid p_k \in \Re^3, k = 1, \ldots, K\}. \tag{1}$$

Given two LiDAR scans P_{t-1} and P_t, the objective is to find a dense rigid motion field that best explains the motion between two scans. A rigid body transformation for a point $p \in \Re^3$ can be written as:

$$T(p) = Rp + t, \tag{2}$$

where $t \in \Re^3$ is the translation and $R \in SO(3)$ is the rotation. Transformation in Equation (2) can be written as:

$$\tau = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \in SE(3) \tag{3}$$

As τ only has 6 degrees of freedom, we also introduce a compact representation $\varsigma = (t^T, q^T) \in \Re^6$, where t is the translation and q is the vector part of a unit quaternion $\tilde{q}$.

The motion of the scene is embedded in a rigid motion field $\mathcal{T}$:

$$\mathcal{T} = \{\tau_k \mid \tau_k \in SE(3), k = 1, \ldots, K\} \tag{4}$$

We represent the problem using a factor graph $G = (\Phi, \mathcal{T}, \mathcal{E})$ with two node types: factor nodes $\phi \in \Phi$ and state variables nodes $\tau_k \in \mathcal{T}$. Here, $\mathcal{E}$ is the set of edges connecting Φ and state variable nodes $\mathcal{T}$. Figure 2 shows the factor graph for our problem.

The factor graph describes the factorization of the function

$$\phi(\mathcal{T}) = \prod_{i \in I_d} \phi_d(\tau_i) \prod_{l \in N_p} \phi_p(\tau_i, \tau_j), \tag{5}$$

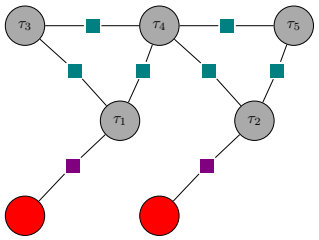


Fig. 2: Factor graph representation of our problem. Gray and red circles represent state variables for the scans P_{t-1} and P_t respectively. Green squares represent factors for the regularization term, while purple squares represent factors for the data term.

where $\{\phi_d, \phi_p\} \in \phi$ are two types of factor nodes describing the energy potentials for the data term (purple squares) and regularization term (green squares) respectively. The term I_d is the set indices corresponding to keypoints in P_{t-1} and $N_p = \{\langle 1,2\rangle, \langle 2,3\rangle, \ldots, \langle i,j\rangle\}$ is the set containing indices of neighboring vertices.

The objective is to find $\mathcal{T}^*$:

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} E(\mathcal{T}), \tag{6}$$

which minimizes the energy $E(\mathcal{T})$:

$$E(\mathcal{T}) = -\ln \phi(\mathcal{T}) \tag{7}$$

A. Data Term

Our approach relies on the assumption of geometric constancy, i.e., that the local geometric structure is not deformed because of the motion. We parametrize the local geometry using a SHOT feature descriptor [20]. Unlike other methods [1], [9], which make similar constancy assumptions, we do not consider the geometric structure explicitly in the optimization, because gradients w.r.t. parametrized geometric structure are not well behaved due to the sparse nature of LiDAR data. Instead, we realize our constancy assumption by matching points in the scan having a similar local geometric description.

To find point correspondences we use the same approach as discussed in our previous work [3]. Since we only have correspondences for a subset of points, the data term is defined only for state variable nodes corresponding to keypoints.

We define the error for data term in the following way:

$$e_d(\tau_i) = \tau_i \tilde{p_i} - \tilde{p_i}' \tag{8}$$

Here, $\tilde{p_i}$ and $\tilde{p_i}'$ are the corresponding points in consecutive scans represented in homogeneous coordinates, τ_i is the unknown rigid motion and $i \in I_d$. The data term is defined as:

$$\phi_d(\tau_i) = \exp(-\|e_d(\tau_i)\|^2) \tag{9}$$

In Figure 2, a purple square connecting state variable nodes of two different scans represents $\phi_d(\tau_i)$.

B. Regularization Term

The data term helps us to estimate the motion, but is defined only for a subset of points. In addition to this, it is known that estimating pointwise motion independently is an ill-posed problem. In case of rigid motion which has 6 unknowns, a minimum of 3 non-collinear point correspondences are required. Therefore, to obtain a well-posed optimization problem, for which a unique solution exists and which has a dense, locally smooth motion field, we include a regularization term in our energy function.

A similar regularization term is often included in energy minimization problems [9], [1], [18]. In case of estimating a motion field in an image domain, the local neighborhood is well understood. A common practice is to consider all pixels in a small image patch as neighbors. In case of 3D point cloud data, the concept of neighborhood is not as straightforward as for images. A naïve way to calculate the neighborhood is to consider all the points in a sphere around a point as neighbors. Since our method relies on the assumption of rigidity of the local structure, it is important that the definition of neighborhood considers the structure of the scene, which spherical neighborhood fails to do. For instance, using spherical neighborhood points on the surface of two different objects can be considered neighbors if objects are close to each other. To circumvent this problem we construct a triangular mesh structure [14] to approximate the surface and consider points as neighbors only if they are vertices of the same triangle.

The error term for the regularization term is defined in the following way:

$$e_p(\tau_i, \tau_j) = \xi(\tau_i^{-1} \tau_j) \tag{10}$$

where, τ_i and τ_j are the motion transformations for neighboring nodes and $\xi(\cdot)$ is the mapping function from $SE(3)$ to a compact representation in $\Re^6$.

The energy potential for the regularization term is defined as:

$$\phi_p(\tau_i, \tau_j) = \exp(-\|e_p(\tau_i, \tau_j)\|^2) \tag{11}$$

In Figure 2, a green square connecting neighboring state variable nodes represents $\phi_p(\tau_i, \tau_j)$.

IV. Optimization

The error for both data and regularization terms is quadratic and the problem is of sparse non-linear least square form. Using Equations (9) and (11), we can simplify Equation (7) as:

$$E(\mathcal{T}) = \sum_{i \in I_d} \|e_d(\tau_i)\|^2 + \sum_{l \in N_p} \|e_p(\tau_i, \tau_j)\|^2 \tag{12}$$

We use the Levenberg-Marquardt algorithm to find an estimate that minimizes the energy. Since the regularization term only connects neighboring points, the problem can be decomposed into multiple independent sub-problems, which can be solved efficiently in parallel.

The point correspondences we estimate contain outliers, which have a strong negative impact on the estimate. To tackle outliers, we use a saturated robust kernel ρ:

$$\rho(x^2) = \left\{ \begin{array}{ll} \frac{x^2}{2}, & \text{if } x^2 \leq c^2 \\ \\ \frac{c^2}{2}, & \text{if } x^2 > c^2 \end{array} \right\} \tag{13}$$

where x^2 is the squared error and c is the kernel size. We use a robust kernel only for the error in the data term and rewrite Equation (12) as:

$$E(\mathcal{T}) = \sum_{i \in I_d} \rho(\|e_d(\tau_i)\|^2) + \sum_{l \in N_p} \|e_p(\tau_i, \tau_j)\|^2 \tag{14}$$

Since, the initial error is large, we perform two steps of optimization. First we optimize without a robust kernel to reduce the error and then perform another optimization run with the robust kernel to minimize the effect of outliers.

As we use sequential data, the motion transformations can also be expected to be temporally smooth. We do not include a term for temporal smoothing explicitly in our energy minimization function, but we initialize the estimate for a vertex in the graph with the estimates from previous scans. Points with known motion are transformed into the frame of reference of the next scan. We perform data association between the transformed points and the points in the next scan on the basis of Euclidean distance and propagate the estimated motion. Since we do not have data association for each point, we still perform two steps of optimization to make sure that state variable nodes without good initial estimates are not treated as outliers due to a large initial error.

V. Results

We perform multiple experiments to evaluate our approach. For the first set of experiments we use five sequences from the KITTI odometry dataset. Estimating the ground truth motion for each point or for every moving object is labor intensive and non-trivial, but if the scene only contains static structure, then the motion of each point is the motion of the sensor. Therefore, to evaluate our method, we choose three sequences (3, 5 and 6) with no moving objects and two sequences (4 and 12) containing multiple dynamic objects. Furthermore, to illustrate the advantages of estimating pointwise motion, we evaluate our method on a new dataset with pedestrians. In this experiment we estimate the motion for a non-rigid structure. We collected this dataset using a Velodyne HDL-32E LiDAR sensor mounted on a robot.

We employ the g2o toolkit [12] for optimization. For all the experiments we remove all the ground points in a preprocessing step. We chose $c^2 = 0.05$ as kernel size (Section IV). This choice depends on the maximum error that can be tolerated.

We compare our method with two variants of ICP. The first one is ICP with no initial estimate, whereas the second one uses RANSAC to calculate an initial estimate before calculating the alignment using ICP. During the application of RANSAC we use the same correspondences as in our approach.

A. KITTI Dataset

We use five sequences from the KITTI odometry dataset to evaluate our approach. For each sequence we calculate dense rigid motion flow. Figure 3 shows results for two sequences. To visualize the motion flow, we colorize the point cloud using the magnitude and the direction of the estimated translational motion. Figure 4 shows the color palette used for colorizing the points.

Figure 3a and 3c show the colorized point clouds for scans in sequences 4 and 5. For better visualization we use the ground truth odometry to compensate for the sensor motion. The static scene in both sequences is represented by black (center of the color palette). Figure 3a also shows two dynamic objects (cars in blue and red) moving in opposite direction. Our method correctly estimates the different motions in the scene, highlighting one of the advantages of our approach.

We observed that the motion estimate for points farther away from the sensor was different to the points closer to the sensor. This can be explained by the lever arm effect, causing the farther away points to have larger rotational motion than points closer to the sensor. This explains the different shades of black that can be observed in both figures. Figures 3b and 3d show the alignment of consecutive scans. In both cases the scans are aligned correctly for the static scene as well as for the dynamic objects, demonstrating that the estimated motion is correct even for the points that are effected by the lever arm effect.

The performance of our approach decreases when a subgraph contains very few points because the optimization problem is ill-posed for these cases. This mainly happens for points on the curb.

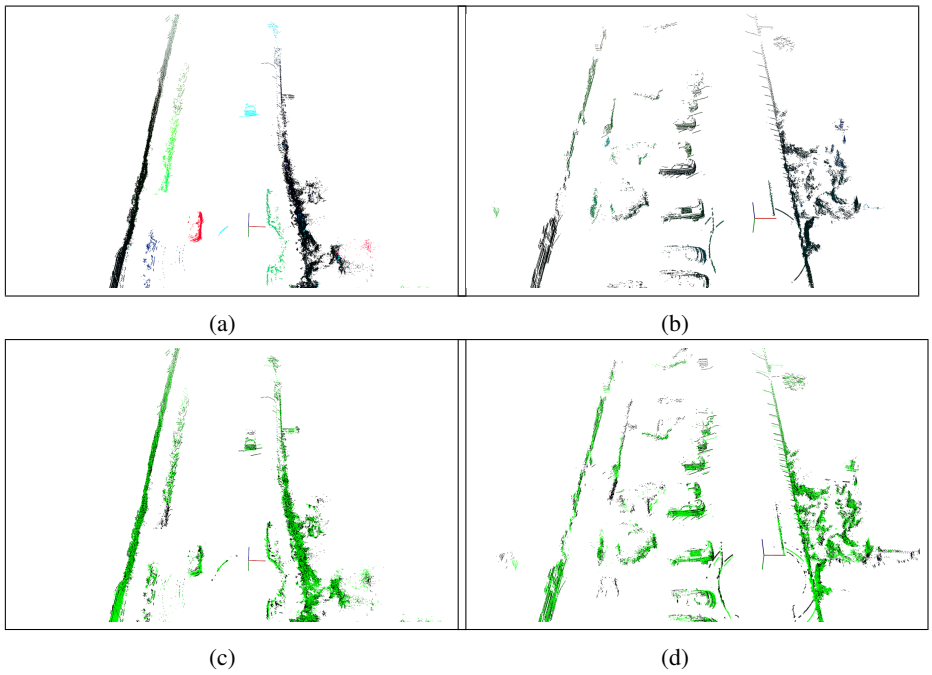
B. Motion Field

Using the ground truth, we calculate an error in translation and rotation for each point. We then calculate average error for each frame, which is further averaged over the entire sequence. Table I contains the translational error t_e (in meters) and the rotation error r_e (in radians) for our method and the ICP methods.

For all the three sequences we calculated the minimum error in translation. Our method outperforms both ICP methods. The ICP method without initialization yields the largest error. This result can be expected as ICP is known to suffer from poor initializations, which is the case here since the sensor is moving. Regarding the rotational error, the results are comparable.

C. Alignment

Ground truth motion is available only for the static scenes. Therefore to quantify the motion estimate for dynamic scenes we measure alignment between two scans by calculating the



(a) (b) (c) (d)

Fig. 3: Estimated motion flow. (a) and (b) show the motion flow for two sequences of the dataset, (c) and (d) are the corresponding aligned scans. Pointcloud in (a) and (b) are colorized according to the color palette in Figure 4. Static structure is represented by black color and the dynamic objects in (a), (blue and red car) are moving in the opposite direction. In (c) and (d) points in black represents the scan P_t and green points represents the scan P_{t-1} transformed into the the next frame.

TABLE I: Error in rigid motion flow

	Sequence 3		Sequence 5		Sequence 6	
	t_e	r_e	t_e	r_e	t_e	r_e
Ours	**0.22±0.07**	**0.02±0.007**	**0.21±0.07**	**0.02±0.008**	**0.50± 0.12**	**0.04±0.01**
ICP	0.77±0.34	0.03±0.02	0.73±0.27	0.05±0.02	1.59±0.51	0.11±0.04
ICP+RANSAC	0.38±0.25	0.02±0.01	0.26±0.12	0.02±0.01	0.65±0.26	0.04±0.02

crispness score C_s, which is [8]

$$C_s = \frac{1}{n_i} \sum_{k=1}^{n_i} G(p'_k - \hat{p_k}, 2\Sigma), \tag{15}$$

where $\hat{p_k}$ is the nearest point in the point set P_t to the transformed point p'_k from the point set P_{t-1} , n_i are the number of points and Σ is the covariance. The crispness score defines the compactness of two point sets: the higher the score, the closer are the two point sets. The scores are scaled between 0 and 1.

TABLE II: Crispness score for outdoor scenarios

	Ours	ICP	ICP+RANSAC
Sequence 3	**0.91±0.01**	0.86±0.03	**0.91±0.01**
Sequence 4	0.87±0.04	0.74±0.07	**0.88±0.04**
Sequence 5	**0.93±0.02**	0.87±0.03	**0.93±0.02**
Sequence 6	**0.89±0.02**	0.79±0.05	**0.89±0.02**
Sequence 12	0.81±0.07	0.63±0.06	**0.83±0.06**

Crispness scores for our method and ICP+RANSAC method are comparable (Table II). This result can be expected since, for outdoor scenarios, objects are rigid and all the points in a subgraph move in a similar way.

Fig. 4: Color wheel representing the magnitude (saturation) and direction (hue) of the translational motion.

D. Non-rigid objects

To test our approach for non-rigid objects, we collected a dataset with four different kinds of human motion: moving arms upwards, moving arms backwards, bending forward and bending sidewards. Figure 5 compares our approach with the ICP+RANSAC method for the cases of the arm moving upwards and the body bending sidewards. In the first case there are multiple motions: the right arm moves upwards and the left arm moves downwards while the rest of the body remains static. Our method estimates the different motions and aligns the scan correctly, whereas for the ICP-based method, the whole body tilts clockwise to align the left arm.

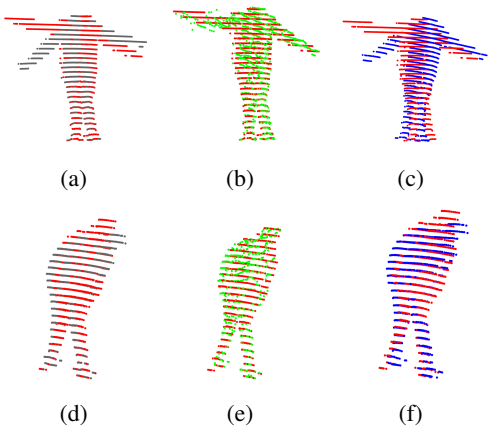


Fig. 5: Alignment of non rigid structure for arm moving upwards ((a)-(c)) and body bending sidewards ((d)-(f)). The first column shows the input scans. The points in scan P_t are shown in red and scan P_{t-1} is shown in gray. The scan aligned by our approach is shown in green (second column), whereas the ICP result is shown in blue (third column).

Similarly for the second case, the alignment proposed by our method better represents the underlying motion.

Table III shows the crispness score for the pedestrian dataset. For all cases our method provided a better score. Our method effortlessly adapts to the non-rigid case and convincingly outperforms the rigid motion based methods. This experiment again demonstrates the advantages of estimating pointwise motion and emphasizes the flexible nature of our approach.

TABLE III: Crispness score for human motion

	Ours	ICP	ICP+RANSAC
Moving arm upwards	**0.98±0.005**	0.91±0.05	0.91±0.05
Moving arm backwards	**0.97±0.02**	0.93±0.04	0.93±0.04
Bending sidewards	**0.97±0.04**	0.94±0.03	0.94±0.04
Bending forward	**0.89±0.04**	0.85±0.03	0.84±0.04

VI. Conclusions

In this paper, we present a novel method for estimating the rigid scene flow for 3D LiDAR data. We introduce the concept of geometric constancy and use spatial smoothing to estimate dense rigid motion flow. Furthermore, we discuss a novel method for estimating neighboring points in 3D point cloud data. Our approach is tested on multiple sequences of the KITTI dataset and on a dataset with pedestrians. For the KITTI dataset we report lower translation error and a comparable alignment score. For the dataset with pedestrians, our method outperforms the rigid motion based methods. Advantages of our method are that it can estimate multiple arbitrary motions in the scene, it performs competitively in case of rigid objects and readily adapts to the cases with non-rigid objects.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*. Springer, 2004.

[2] L. Cosmo, E. Rodola, A. Albarelli, F. Memoli, and D. Cremers. Consistent partial matching of shape collections via sparse modeling. *Computer Graphics Forum*, 2016. to appear.

[3] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3d lidar scans. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[4] P. S. Dragomir Anguelov, H.-C. Pang, D. Koller, and J. D. Sebastian Thrun. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *MIT Press Conference on Neural Information Processing Systems (NIPS)*, 2005.

[5] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21, 2015.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] D. Haehnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI*, 2003.

[8] D. Held, J. Levinson, S. Thrun, and S. Savarese. Combining 3d shape, color, and motion for robust anytime tracking. In *Proceedings of Robotics: Science and Systems*, 2014.

[9] 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)*, 2013.

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

[11] R. Kaestner, J. Maye, Y. Pilat, and R. Siegwart. Generative object detection and tracking in 3d range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[12] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g 2 o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[13] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. A navigation system for robots operating in crowded urban environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[14] Z. C. Marton, R. B. Rusu, and M. Beetz. On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[15] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2015.

[16] F. Moosmann and C. Stiller. Joint self-localization and tracking of generic objects in 3d range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[17] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] J. Quiroga, T. Brox, F. Devernay, and J. Crowley. Dense semi-rigid scene flow estimation from rgbd images. In *European Conference on Computer Vision (ECCV)*. Springer, 2014.

[19] G. D. Tipaldi and F. Ramos. Motion clustering and estimation with conditional random fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[20] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision (ECCV)*. Springer, 2010.

[21] J. Van De Ven, F. Ramos, and G. D. Tipaldi. An integrated probabilistic model for scan-matching, moving object detection and motion estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[22] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *IEEE International Conference on Computer Vision(ICCV)*, 1999.

[23] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a rigid motion prior. In *IEEE International Conference on Computer Vision(ICCV)*, 2011.