

Reconstruction of Rigid Body Models from Motion Distorted Laser Range Data Using Optical Flow

Eddy Ilg

Rainer Kümmerle

Wolfram Burgard

Thomas Brox

Abstract—The setup of tilting a 2D laser range finder up and down is a widespread strategy to acquire 3D point clouds. This setup requires that the scene is static while the robot takes a 3D scan. If an object moves through the scene during the measurement process and one does not take into account these movements, the resulting model will get distorted. This paper presents an approach to reconstruct the 3D model of a moving rigid object from the inconsistent set of 2D measurements by the help of a camera. Our approach utilizes optical flow in the camera images to estimate the motion in the image plane and point-line constraints to compensate the missing information about the motion in depth. We combine multiple sweeps and/or views into to a single consistent model using a point-to-plane ICP approach and optimize single sweeps by smoothing the resulting trajectory. Experiments obtained in real outdoor scenarios with moving cars demonstrate that our approach yields accurate models.

I. INTRODUCTION

Compared to stereo cameras, laser range finders have become popular devices for the acquisition of 3D point clouds in large-scale scenarios because of their ability to obtain accurate long-range measurements [1]. A widely used and cost-effective setup is to mount a 2D range scanner on a tilting actuator. By nodding the laser scanner this approach is able to generate accurate models of static scenes. However, if an object moves during an entire sweep, which with typical configurations takes one second or even more, the individual measurements will not be consistent (see Figure 1a). In this paper, we present an approach that utilizes the optical flow calculated from images grabbed during the point cloud acquisition and combines it with the laser measurements to estimate the movement of the corresponding object. According to the estimated movement, it then calculates an accurate model of this object (see Figures 1b and 1c).

The key idea of our approach is to estimate and perform a compensation for the motion of the moving object. To achieve this, we employ a video camera that we calibrate to the nodding laser range finder. As the camera operates at a higher framerate compared to the 3D point cloud acquisition (10 Hz vs. 1 Hz), our approach estimates the motion of the object in the image plane from frame to frame using optical flow [2] and employs this information to reproject the individual 2D laser range scanlines according to the movement of the object. This, however, requires that we estimate the depth of the individual image pixels after the motion. To achieve this, we assume that the perceived object

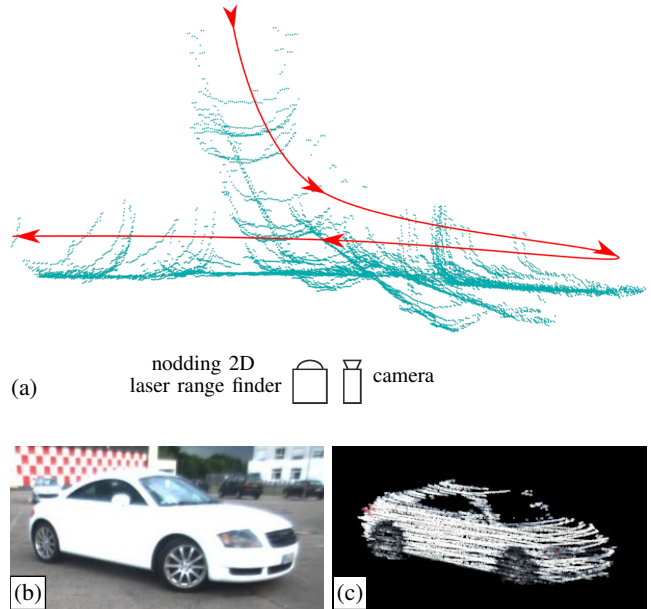


Fig. 1. (a) Top view of a bulk point cloud recorded during 15 sweeps of a nodding 2D laser range finder of a car moving along the trajectory indicated by red arrows. (b) The car used as moving object. (c) The colored point cloud model reconstructed from the distorted point cloud above.

is rigid. If, at one instant of time, we know a partial model of the rigid body, the optical flow provides point-line constraints for the pose of this model at another instant of time. As a minimum model we have a non-degenerated single scanline (recorded in one rotation of the mirror deflecting the laser), which we assume to be recorded instantaneously. Tracking the pose of this model and adding more scanlines over time allows for reconstructing the model of the rigid object.

Pose tracking with optical flow naturally accumulates errors (drift), especially in the viewing direction of the camera. Therefore, as soon as we record a new sweep of the object, we use the model from this sweep as a reference and register the old model to it using an ICP approach. This restricts drift to the time window between two sweeps. The model recorded during one sweep is dense in the direction of the rotation of the mirror of the laser range finder, but sparse in the nodding direction. Due to this sparseness, the alignment of sweeps is not an easy task. We estimate the underlying surface of the existing (possibly denser) model by estimating its normals and register the points of the newly recorded sweep by minimizing their point-to-plane distances to this surface. To reduce drift, we furthermore constrain the motion of the object to the ground plane.

Furthermore, we improve the tracking result by smoothing it with the alignment determined by ICP. This allows us to

reduce the influence of the drift on the quality of the model estimated by our approach. Afterwards, we use the smooth trajectory to recalculate the point cloud of the current sweep before integrating it into the model.

II. RELATED WORK

The typical methods to obtain 3D data with laser range finders can be roughly divided into two categories. The first operates with laser range finders that are rigidly connected to a moving platform, such as a robot [3]. These methods require a precise estimate of the position of the robot to obtain an accurate model. The second category actively actuates the laser range finder either by nodding or rotating it [4], where the laser range finder itself may have a single or multiple beams. Note that there are also exceptions. For example, Zibedee by Bosse *et al.* [5] employs a spring-mounted laser range finder which is passively actuated due to the vibrations induced by driving over non-flat surfaces. Obviously, the data of such a 3D scanning device can be used for various tasks including object reconstruction or mapping an environment.

In addition to the categories mentioned above, dedicated approaches for model acquisition have been proposed. A popular approach to obtain the model is to use a robot for moving the sensor around the object [6], [7]. Krainin *et al.* [8] suggested a method that uses a robotic arm to grasp the object and to move it in front of a depth camera. In contrast to our approach, these methods either obtain a model of a non-moving object or move the object by themselves, whereas our approach considers the data of a continuously nodding 2D laser range finder to obtain a 3D model of a rigid object, which is moving along an unknown trajectory.

Blais *et al.* [9] present an approach that explicitly takes into account that the scanned object moves. Their approach iteratively refines the model but relies on a Lissajous pattern to obtain the range data. Weise *et al.* [10] propose an approach to correct the distortion for active illumination stereo in short time frames. Instead of obtaining a model of an object by scanning a single object from multiple views, Ruhnke *et al.* [11] exploit the presence of multiple instances of the objects in the scene to obtain complete 3D models. Their approach applies spectral clustering to merge the individual instances into one object model.

A related problem in computer vision is 3D reconstruction from a moving camera in the presence of independently moving objects [12]. While a static scene would come down to a structure-from-motion problem, a moving rigid object requires segmentation of the object and estimation of the relative rigid motion.

Our approach combines the data of a laser range finder with vision. This combination has been applied successfully for other applications, such as extending the range for terrain classification [13] or increasing the resolution of the range data [14]. Held *et al.* [15] apply upsampling to the sparse range data of a Velodyne scanner. They show that the denser range data leads to better velocity estimates for tracking a moving object. In contrast to our approach, their method

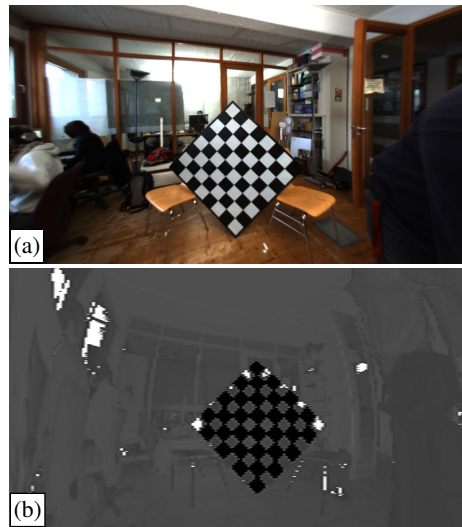


Fig. 2. (a) The calibration pattern in the camera data and (b) in the remission data of the laser range finder.

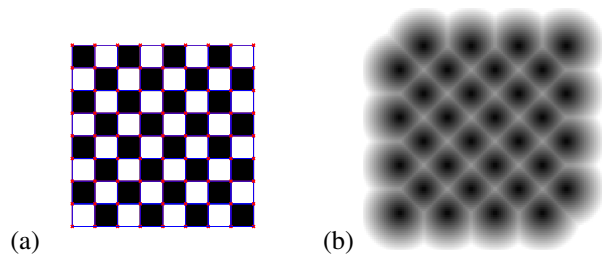


Fig. 3. (a) The ideal model of the calibration pattern. The red dots indicate the corners of the pattern. (b) The error function used to detect the rigid in-plane transformation. Darker areas indicate lower error.

receives a full 3D scan with 10 Hz, whereas we receive such a scan at only 1 Hz and scanline-by-scanline. We exploit that the camera operates at a much higher frame rate than our nodding 2D laser range finder and track the object in the image space to account for the distortion.

III. CALIBRATION AND SYNCHRONIZATION

A. Calibration

For calibrating the tilting laser range scanner and the camera we employ the calibration pattern depicted in Figure 2. To have a distinct pattern that can be well observed by both the laser range finder and the camera, the bright areas consist of highly reflective material conducting a scattering of the laser beam. This results in low remission values (black patches in Figure 2b) and makes it possible to detect the corresponding areas in the 3D scan. For calibration, we nod the laser range finder at a very slow velocity and record multiple sweeps to obtain high resolution scans.

To establish a camera projection matrix \mathbf{P} , which maps the 3D points to image points, we are interested in finding the set of corresponding patch corners in image and 3D space. In the image we compute the intersections of the lines detected from the edges of the patches. In 3D space, we first fit a plane through the points \mathbf{p}_i with low remission values (black patches in Figure 2b) using a RANSAC approach. Given this plane we then interpolate corners of the patches by fitting an

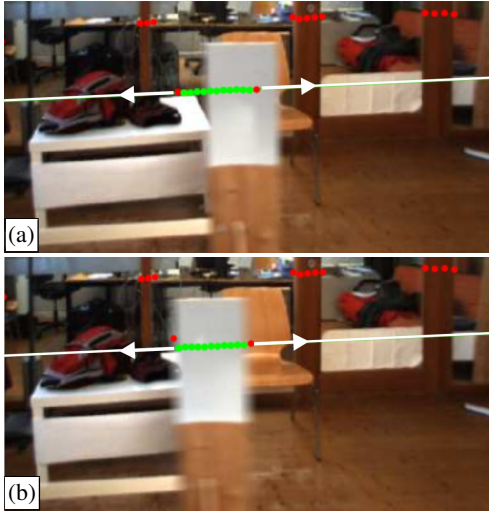


Fig. 4. (a) Calibration object used to determine the time shift between the laser range finder and the camera. Green dots indicate points of the object detected by the laser range finder due to their remission value. The white arrows indicate the direction of movement. (b) Synchronized laser range finder and camera data of a moving object.

ideal model (see Figure 3a) of the calibration pattern to the data. To this end, we define an error function $E(\cdot)$ (see Figure 3b) for the point locations. To determine the pose $(\mathbf{R}_\alpha, \mathbf{t})$ of the pattern on the previously extracted plane we minimize

$$\arg \min_{\alpha, \mathbf{t}} \sum_{i=1}^N E(\mathbf{R}_\alpha \mathbf{p}_i + \mathbf{t}) \quad (1)$$

using gradient descent, where \mathbf{R}_α is a 2D rotation matrix with angle α and \mathbf{t} is a 2D translation.

Once we know the corresponding locations of the corners in 3D space and the image, we establish a homography [16]. Since the calibration object is planar, we take multiple rotated views of the pattern to uniquely determine the internal parameters of the camera and the projection matrix \mathbf{P} .

B. Synchronization

To determine the time shift between the laser range finder and the tilt actuator, we exploit that objects must have the same coordinates within subsequent upward and downward sweeps. Hence, we determine the angle between the ground planes in two subsequent sweeps and iteratively shift the timestamps of the tilt actuator until this angle approaches zero.

By moving another calibration object at high velocities from left to right, we calibrate the time shift between the laser and the camera. The surface of this object (depicted in Figure 4a) consists of the same reflective material as the calibration pattern described before. For the synchronization, we do not nod the laser range finder. We detect the pattern in the laser range finder data by its low remission values and project the points to image space (depicted as green dots in Figure 4). Along the line (shown in white) extrapolated from these points, we search for nearby pixels with the color of the calibration object to determine its location in the image. For a given scanline, we then seek for a camera frame close in time which best matches the position of the

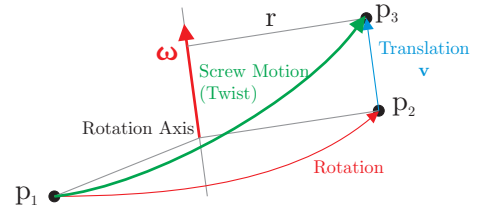


Fig. 5. Illustration of a twist ξ . The motion corresponds to a rotation around the axis ω (red) and a translation along \mathbf{v} (blue) resulting in the screw motion from \mathbf{p}_1 to \mathbf{p}_3 (green).

object. We store the difference in time between the scanline and the image and fit a normal distribution to a histogram over all values. We then use the mean of this distribution to shift the camera timestamps to match the laser range finder timestamps. Figure 4b shows a typical result.

IV. MODEL RECONSTRUCTION

A. Point-Line Correspondences from Optical Flow

To compensate the motion, we assume that a partial model is known at one instant of time. In the beginning, this is the first scanline of the laser range finder. Let \mathbf{x}_i denote the 3D points of this model. With the projection matrix \mathbf{P} from the previous section, these points can be projected into the camera image. We denote the points in the image plane by $\tilde{\mathbf{x}}_i$. We use large displacement optical flow [2] to compute the displacements $\tilde{\Delta}_i$ of $\tilde{\mathbf{x}}_i$ to the next camera frame. The displaced image location $\tilde{\mathbf{x}}'_i = \tilde{\mathbf{x}}_i + \tilde{\Delta}_i$ restricts \mathbf{x}'_i to the projection ray of $\tilde{\mathbf{x}}'_i$.

3D lines can be represented implicitly by so-called *Plücker lines* [17]. Let $\mathbf{L}'_i = (\mathbf{m}'_i, \mathbf{n}'_i)$ be the Plücker line with a unit vector \mathbf{n}' and a moment \mathbf{m}' that corresponds to the projection ray $\tilde{\mathbf{x}}'_i$. The Plücker line representation then yields directly the distance of an arbitrary 3D point \mathbf{x} to \mathbf{L}'_i via [18]:

$$d(\mathbf{L}'_i, \mathbf{x}) = \|\mathbf{x} \times \mathbf{n}'_i - \mathbf{m}'_i\|. \quad (2)$$

Based on these point-line constraints, we estimate the six degrees of freedom of the rigid body motion to move the 3D points to their new position at the time when the current camera image was taken.

B. Pose Estimation

For a given set of non-occluded points \mathbf{x}_i we seek for a rigid body transformation $\mathbf{T} = (\mathbf{R} \mid \mathbf{t})$ with a rotation \mathbf{R} and a translation \mathbf{t} that minimizes $d(\mathbf{L}'_i, \mathbf{x}'_i)$. For the purpose of pose estimation, the twist representation [19] of \mathbf{T} is well suited. As illustrated in Figure 5, a twist is a screw motion around a rotation axis ω and a translation \mathbf{v} . Such a twist can be represented as a vector $\xi = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)$ or in matrix form

$$\hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \quad \hat{\xi} = \begin{pmatrix} \hat{\omega} & \mathbf{v} \\ \mathbf{0}^\top & 0 \end{pmatrix}. \quad (3)$$

Multiplying $\hat{\xi}$ by a factor θ allows for an arbitrary scaling of the motion [20]. We can map a scaled twist to a transformation matrix and back by taking the exponential or logarithm:

$$\mathbf{T} = \exp(\theta \hat{\xi}), \quad \hat{\xi} = \frac{1}{\theta} \log(\mathbf{T}), \quad (4)$$

which can be computed efficiently by the Rodrigues formula [19]. Here, we set $\theta = 1$ and seek for a twist that minimizes the distance $d(\mathbf{L}'_i, \mathbf{x}'_i)$:

$$\arg \min_{\xi} \sum_i \left\| \pi \left(\exp(\hat{\xi}) \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \right) \times \mathbf{n}'_i - \mathbf{m}'_i \right\|^2, \quad (5)$$

where $\pi(\cdot)$ denotes the projection from homogeneous to Euclidean coordinates. To reduce drift, we restrict the motion to the ground plane, which we detect using RANSAC, by aligning the z -axis of the reference frame to the ground normal and by setting

$$\omega_1 = 0, \quad \omega_2 = 0, \quad \text{and} \quad v_3 = 0. \quad (6)$$

Eq. (5) states a non-linear least squares problem, which we solve with the Gauss-Newton method, i.e., we iteratively linearize $\exp(\hat{\xi}) \approx \mathbf{I} + \hat{\xi}$. Hence, Eq. (5) becomes a linear system in each iteration:

$$\arg \min_{\xi} \sum_i \left\| ((\mathbf{I} + \hat{\omega}) \mathbf{x}_i + \mathbf{v}) \times \mathbf{n}'_i - \mathbf{m}'_i \right\|^2. \quad (7)$$

We map the twist corresponding to the solution of this linear system to the corresponding transformation matrix and apply it to the 3D points to perform the next iteration.

C. Treatment of Outliers

Both the estimation of the optical flow and the pose of the object are affected by residual errors. Hence, repeatedly estimating a new pose given the previous result leads to an accumulation of errors, i.e., a drift in the position and the orientation of the object. Close to the object boundary, this can result in model points that need to be projected to a background pixel, where the optical flow does not correspond to the motion of the model. Figure 6a illustrates such a case. Another cause for such outliers can be local errors of the optical flow. To be robust to such outliers, we apply robust statistics and replace the squared error norm in Eq. (5) by a truncated Huber norm [21]. This is equivalent to iteratively reweighted least squares. Using the Gauss-Newton method, we iteratively solve the linear system of Eq. (7). Let us denote this system as

$$\arg \min_{\xi} \sum_i \left\| \mathbf{A}_i \xi - \mathbf{b}_i \right\|^2. \quad (8)$$

If the correspondence $\langle \mathbf{x}_i, \tilde{\mathbf{x}}'_i \rangle$ is an outlier, the residual $\mathbf{r}_i(\xi) = \left\| \mathbf{A}_i \xi - \mathbf{b}_i \right\|$ of this point is large. To reduce its influence on the solution ξ , we introduce the weight

$$w_i(\xi) = \begin{cases} 0 & \text{if } \|\mathbf{r}_i(\xi)\| > \tau \\ \frac{1}{\|\mathbf{r}_i(\xi)\| + \epsilon} & \text{otherwise} \end{cases}, \quad (9)$$

which corresponds to replacing the quadratic norm by the truncated Huber norm with truncation at τ . The parameter ϵ limits the weights of points with low residual and, consequently, the condition number of the system matrix. We start with $w_i = 1$ and then solve Eq. (8) by computing

$$\arg \min_{\xi_k} \sum_i w_i(\xi_{k-1}) \left\| \mathbf{A}_i \xi_k - \mathbf{b}_i \right\|^2 \quad (10)$$

in each iteration k . Figure 6b shows the result of enabling iteratively weighted least squares.

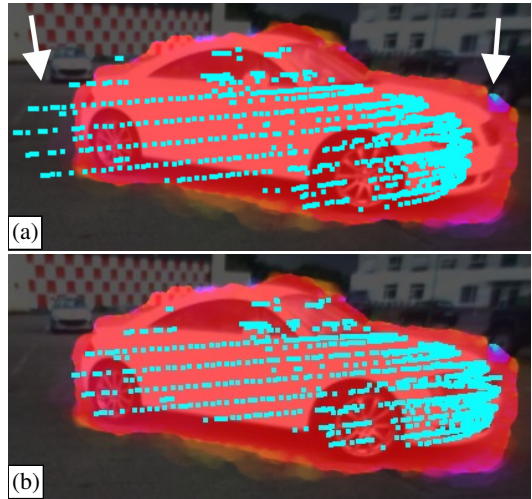


Fig. 6. (a) Illustration of the pose estimation drift during the reconstruction of the first six sweeps of a car moving to the right. The area in red shows the optical flow while the points indicate the projection of the model and its estimated pose. The drift leads to outliers (indicated by the white arrows). (b) Applying Eq. (9) weakens the influence of the outliers.

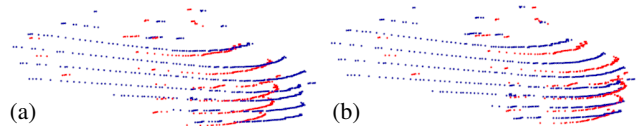


Fig. 7. Illustration of model merging with the first two sweeps of a moving car. (a) Initial misalignment of the first two sweeps. The first sweep is shown in red and the second one in blue. (b) The model after aligning the first two sweeps with our approach. One can also see that the sweeps are dense in the direction of mirror rotation and sparse in the direction of nodding.

D. Model Merging

Applying the pose estimation process described above allows us to track the object and build its model by accumulating points. As already mentioned, this method produces drift. To keep the error of the drift bounded, we build a new model for each sweep. As the pose of the most recently recorded model is most certain, we align the formerly tracked model with the most recent one. Afterwards, we merge the models to an accumulated one, which is then aligned and merged with the next sweep in the same manner.

Let Γ be the accumulated model and let Λ be the model of the most recent sweep. We want to register both models in a common coordinate frame. To this end, we consider the tracking result provided by optical flow as an initial guess for registering both models with an ICP approach (see Figure 7a). We, however, have to take into account that the range data is sparse and that the scanlines hit the object at different height levels. Applying the commonly used point-to-point metric would align the dense scanlines and yield sub-optimal results. Thus, we instead consider a point-to-plane metric. Consequently, we have to estimate the normal vectors. Since the quality of the estimate of a normal vector depends on the density of the underlying point cloud, we compute the normal vectors on the accumulated model Γ , which is in general denser than the newly recorded model Λ .

To estimate the normal vector \mathbf{n}_i of a point $\mathbf{p}_i \in \Gamma$, we

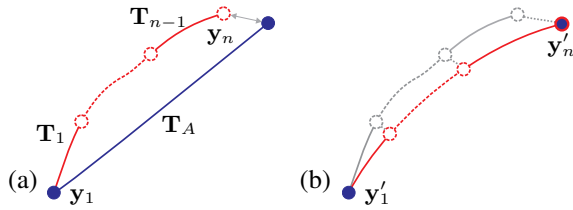


Fig. 8. The trajectory optimization. Image (a) shows the pose estimates between two specific poses \mathbf{y}_1 and \mathbf{y}_n . Blue circles indicate the estimated poses directly after a laser sweep was completed. Red lines indicate pose updates due to optical flow pose estimation while blue lines indicate pose updates due to an ICP alignment. (b) Corresponding optimized trajectory.

first find the neighbors within a certain radius. We compute the normal vector \mathbf{n}_i as the Eigenvector corresponding to the smallest Eigenvalue of the covariance matrix of these neighboring points. Additionally, we determine the sign of the surface normal such that it points towards the camera. We furthermore assume that the surfaces of the captured objects are convex. This allows us to detect occlusions when the object is moving by identifying points whose normals do no longer point in the direction of the camera. Specifically, we mark points as occluded if the angle between the vector from the point to the camera and the normal exceeds 60° . We do not consider occluded points in the pose estimation step. A representation of the tangential plane to a point \mathbf{p}_i with normal \mathbf{n}_i is

$$\mathbf{n}_i \mathbf{x} - d_0 = 0, \quad (11)$$

where $d_0 = \mathbf{n}_i \mathbf{p}_i$. Furthermore, $|\mathbf{n}_i \mathbf{x} - d_0|$ represents the distance of a point \mathbf{x} to this plane. We iteratively seek for the nearest neighbors $\mathbf{q}_i \in \Lambda$ to \mathbf{p}_i bounded by a maximum distance d_{max} and align Λ to Γ by minimizing

$$\arg \min_{\mathbf{T}_A} \sum_i |\mathbf{n}_i (\mathbf{T}_A \mathbf{q}_i) - d_0|^2. \quad (12)$$

Representing \mathbf{T}_A as a twist (see Eq. (4)) allows us to apply the constraint of Eq. (6), namely that the object moves on the ground plane. Again, we find the solution of Eq. (12) using the Gauss-Newton method analogously to the pose estimation problem described before. Figure 7b depicts the outcome of our registration approach.

We found that the robustness of the alignment of the sparse clouds heavily depends on the parameter d_{max} . In contrast to other ICP implementations, our approach starts with $d_{max} = 10$ cm in each iteration and only increases d_{max} if the number of neighbors found is insufficient (less than one-third of the size of Λ).

E. Trajectory Optimization

As mentioned above, the pose of the reconstructed object is more accurate after we registered our existing model Γ and the point cloud Λ of the current sweep. The alignment step corresponds to a correction of the error accumulated by chaining up the pose estimates $(\mathbf{T}_1, \dots, \mathbf{T}_{n-1})$ from optical flow. Figure 8a illustrates such a chain along with the ICP alignment that yields the transformation matrix \mathbf{T}_A . To account for the pose estimation errors made during the construction of Λ , we perform a smoothing of the trajectory to obtain the maximum likelihood estimate for the trajectory

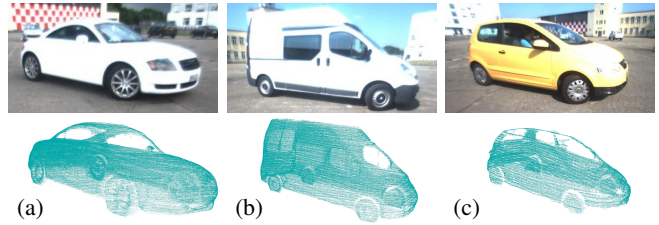


Fig. 9. Pictures and point clouds of the different objects: (a) Audi TT, (b) Van, (c) Polo.

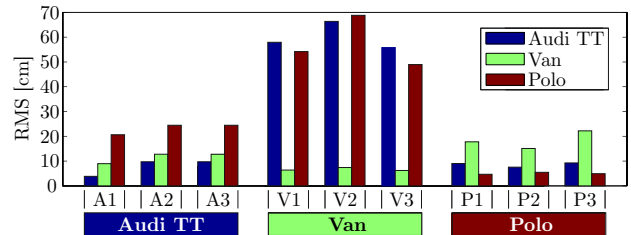


Fig. 10. RMS errors of the different datasets from Figure 11 compared to the different ground truths from Figure 9. As visible, the RMS to the correct ground truth is always the smallest and therefore allows for a classification.

of the object. Let $(\mathbf{y}_1, \dots, \mathbf{y}_n)$ be the poses of the moving object while Λ is acquired. Additionally, let $\mathbf{e}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{T})$ be an error function which computes the difference between a given estimated transformation \mathbf{T} and the expected value given the current state of the poses \mathbf{y}_i and \mathbf{y}_j . We determine $\mathbf{y}'_{1, \dots, n}$ by solving

$$\arg \min_{\mathbf{y}'_{1, \dots, n}} \|\mathbf{e}(\mathbf{y}_1, \mathbf{y}_n, \mathbf{T}_A)\|_{\Psi}^2 + \sum_{i=1}^{n-1} \|\mathbf{e}(\mathbf{y}_i, \mathbf{y}_{i+1}, \mathbf{T}_i)\|_{\Sigma_i}^2, \quad (13)$$

where $\|\cdot\|_{\Sigma}^2$ is the squared Mahalanobis distance weighted by the covariance matrix Σ . Here, Ψ and Σ_i allow us to balance the influence of the measurements. Again, we solve Eq. (13) with Gauss-Newton as implemented in $\mathbf{g}^2\mathbf{o}$ [22]. This leads to a smoothed trajectory $\mathbf{y}'_{1, \dots, n}$ (see Figure 8b) that we use to rebuild the model acquired during the most recent sweep.

V. RESULTS

We evaluate our approach on several real-world data sets in which three different cars are moving through the scene. A Hokuyo UTM-30LX laser rotating with 40 Hz mounted on a servo that is tilting from -50° to $+30^\circ$ provided the 3D range measurements. Additionally, we recorded the image data of one camera of a Point Grey Bumblebee2 stereo camera running at 10 Hz and a resolution of 1024×768 pixels. We use the implementation from Brox and Malik [2] for computing the optical flow.

For a quantitative evaluation, we acquired dense point clouds of the still standing vehicles depicted in Figure 9 by manually steering a robot around the vehicle. We applied sparse surface adjustment [23] to join the views and to obtain ground truth models. To assess the quality of our model reconstruction from a moving vehicle, we constrain the models to the ground plane and align the reconstruction to the ground truth with the previously introduced point-to-plane ICP approach. As an initialization we align the four contact points of the wheels. For each model point we then

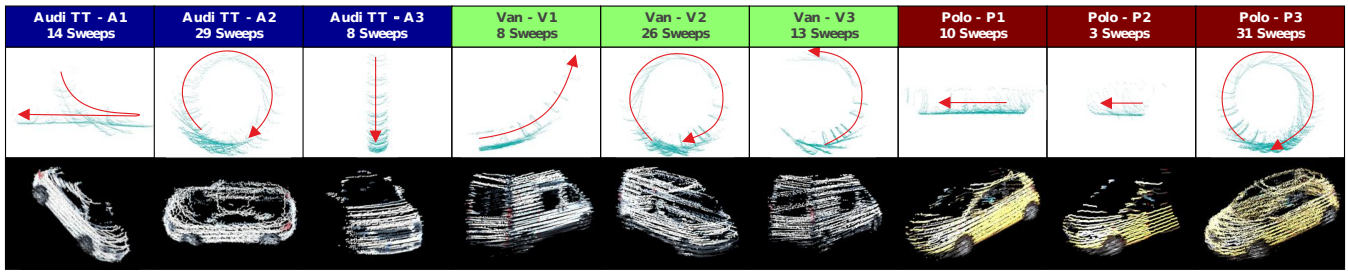


Fig. 11. Illustration of the different datasets. The top row shows the trajectory of the moving object estimated by our approach along with a top view of the bulk point cloud. The bottom row illustrates the models reconstructed by our approach.

find the nearest neighbor in the ground truth and compute the root mean squared error (RMS). We not only compute the RMS of a vehicle compared to itself but also to the other models. The results are shown in Figure 10.

For the evaluation we recorded three different motion scenes of different lengths (from 3 to 30 seconds) for each vehicle, as illustrated in Figure 11. The trajectories include linear motions towards the camera, from left-to-right and right-to-left, stops, curves and circles. For each vehicle, in case of a circle, we are able to reconstruct a 360° model. The RMS with respect to the corresponding ground truth model typically lies between 3 cm and 9 cm. As visible from Figure 10, in all cases (even with only 3 sweeps), the RMS of the reconstructed models suffices to distinguish between different vehicles without the use of further features. This indicates that we can use our approach to classify moving objects based on their 3D shape instead of relying on the visual appearance alone.

VI. CONCLUSION

In this paper, we propose a method to reconstruct models of moving rigid objects captured by a nodding laser range finder and a video camera. To this end, our approach considers optical flow to track the object in the camera frame and to correct the distortion in the range data induced by the motion of the object. The results from different motion scenes of vehicles indicate that our approach allows us to obtain accurate reconstructions. Our evaluation shows that reconstruction errors lie in the range of 3 cm to 9 cm. Furthermore, one could use our approach to classify the moving object based on corresponding, previously recorded 3D models.

There are several options for future work. Potential extensions regard a real-time implementation or the relaxation of the ground plane or rigidity assumption.

REFERENCES

- [1] S. Kumar, D. Gupta, and S. Yadav, "Sensor fusion of laser and stereo vision camera for depth estimation and obstacle avoidance," *International Journal of Computer Applications*, vol. 1, no. 26, 2010.
- [2] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 500–513, 2011.
- [3] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast scale outdoor navigation using adaptive relative bundle adjustment," *Int. Journal of Robotics Research*, vol. 29, no. 8, pp. 958 – 980, July 2010.
- [4] O. Wulf and B. Wagner, "Fast 3D-scanning methods for laser measurement systems," in *Int. Conf. on Control Systems and Computer Science (CSCS)*, 2003.
- [5] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.
- [6] R. Triebel, B. Frank, J. Meyer, and W. Burgard, "First steps towards a robotic system for flexible volumetric mapping of indoor environments," *Proc. of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV)*, 2004.
- [7] T. Foissotte, O. Stasse, A. Escande, P.-B. Wieber, and A. Kheddar, "A two-steps next-best-view algorithm for autonomous 3D object modeling by a humanoid robot," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.
- [8] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.
- [9] F. Blais, M. Picard, and G. Godin, "Accurate 3D acquisition of freely moving objects," in *Proc. of the 2nd Int. Symposium on 3D Data Processing, Visualization and Transmission*, 2004.
- [10] T. Weise, B. Leibe, and L. Van Gool, "Fast 3D scanning with automatic motion compensation," in *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, 2007.
- [11] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard, "Unsupervised learning of 3D object models from partial views," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.
- [12] K. E. Ozden, K. Schindler, and L. Van Gool, "Multibody structure-from-motion in practice," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1134–1141, 2010.
- [13] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Proc. of Robotics: Science and Systems (RSS)*, 2006.
- [14] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, 2010.
- [15] D. Held, J. Levinson, and S. Thrun, "Precision tracking with sparse 3D and dense color 2D data," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.
- [16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [17] F. Shevlin, "Analysis of orientation problems using Plücker lines," in *International Conference on Pattern Recognition (ICPR)*, vol. 1, Brisbane, 1998, pp. 685–689.
- [18] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region- and motion-based 3D tracking of rigid and articulated objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 402–415, 2010.
- [19] R. M. Murray, Z. Li, and S. S. Sastry, *Mathematical Introduction to Robotic Manipulation*. Baton Rouge: CRC Press, 1994.
- [20] B. Rosenhahn, T. Brox, and H.-P. Seidel, "Scaled motion dynamics for markerless motion capture," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [21] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
- [22] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.
- [23] M. Ruhnke, R. Kümmerle, G. Grisetti, and W. Burgard, "Highly accurate 3D surface models by sparse surface adjustment," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.