Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Model Learning and Real-Time Tracking Using Multi-Resolution Surfel Maps

Jörg Stückler and **Sven Behnke**
Autonomous Intelligent Systems, University of Bonn
53113 Bonn, Germany
{*stueckler,behnke*}*@ais.uni-bonn.de*

Abstract

For interaction with its environment, a robot is required to learn models of objects and to perceive these models in the livestreams from its sensors. In this paper, we propose a novel approach to model learning and real-time tracking.

We extract multi-resolution 3D shape and texture representations from RGB-D images at high frame-rates. An efficient variant of the iterative closest points algorithm allows for registering maps in real-time on a CPU. Our approach learns full-view models of objects in a probabilistic optimization framework in which we find the best alignment between multiple views. Finally, we track the pose of the camera with respect to the learned model by registering the current sensor view to the model.

We evaluate our approach on RGB-D benchmarks and demonstrate its accuracy, efficiency, and robustness in model learning and tracking. We also report on the successful public demonstration of our approach in a mobile manipulation task.

Introduction

Object perception is a fundamental capability for an intelligent robot. Many robotic tasks involve the interaction with objects. Either the robot observes and manipulates them for a purpose, or the robot needs to understand the actions on objects by other agents. In this paper, we present a novel approach for learning 3D models of objects using RGB-D cameras. Our method allows to track the camera pose with regard to a model at high frame-rates.

Our approach to 3D modelling and tracking is based on an efficient yet robust probabilistic registration method. From RGB-D measurements, we extract multi-resolution shape and texture representations. We propose a method that is suitable for registering maps generated from single images as well as maps that aggregate multiple views.

For model learning, we fuse several views within a probabilistic optimization framework into a full-view map. We construct a graph of spatial relations between views and optimize the likelihood of the view poses. For this purpose, we assess the uncertainty of the registration estimates. The acquired models can then be used for tracking the camera pose with respect to the models in real-time on a CPU. By the multi-resolution nature of our maps, our method keeps track of the object in a wide range of distances and speeds.

We evaluate the accuracy of our approach to object modelling on publicly available RGB-D benchmarks. We also measure the robustness, accuracy, and efficiency of our tracking approach. Finally, we report on the successful public demonstration of our method in a mobile manipulation task.

Related Work

Modeling the geometry of objects from multiple views has long been investigated in robotics and computer graphics. A diverse set of applications exists for such explicit geometric map representations like, for instance, object recognition or manipulation planning.

One early work (Chen and Medioni 1992) registers several range images using an iterative least squares method. In order to acquire full-view object models, the authors propose to take four to eight views on the object. Each view is then registered to a map that is aggregated from the preceding views. Compared to pair-wise registration of successive views, this procedure reduces accumulated error.

Recently, Newcombe et al. 2011 proposed KinectFusion. They acquire models of scenes and objects by incrementally registering RGB-D images to a map. Although it achieves impressive results, this approach still accumulates drift in the map estimate over long trajectories, since it does not optimize jointly for the view poses. In our approach, we find a best alignment of all views by jointly optimizing spatial relations between views. We determine the relative pose between views and the uncertainty of this estimate using our registration method. Afterwards, we obtain a full-view model given the RGB-D images in the optimized view poses.

Our approach is strongly related to the simultaneous localization and mapping (SLAM) problem in robotics. Over the last decade, some approaches have been proposed that estimate the 6 degree-of-freedom (DoF) trajectory of a robot and a 3D map by means of 3D scan registration (Nuechter et al. 2005; Magnusson, Duckett, and Lilienthal 2007; Segal, Haehnel, and Thrun 2009). However, these approaches

have been designed for mapping measurements of 3D laser-scanners. Weise et al. 2009 match surface patches between range images and align them globally to reconstruct 3D object models. Krainin et al. 2011 extract textured surface patches from RGB-D images, register them using ICP (Besl and McKay 1992) to the model, and apply graph-optimization to obtain accurate maps of indoor environments and objects, respectively. Our approach provides shape-texture information in a compact representation that supports pose tracking from a wide range of distances, since the model contains detail at multiple scales. Engelhard et al. 2011 match SURF features between RGB-D frames and refine the registration estimate using ICP. Our registration method incorporates shape and texture seamlessly and is also applicable to textureless shapes.

In computer vision, much research focusses on the learning of sparse interest point models. In many of these approaches, structure from motion is obtained through bundle adjustment, e.g., (Klein and Murray 2007). Recently, dense structure from motion approaches have been proposed that estimate dense depth from monocular intensity image sequences (Stuehmer, Gumhold, and Cremers 2010; Newcombe, Lovegrove, and Davison 2011). Steinbruecker et al. 2011 propose a method for dense real-time registration of RGB-D images. They model the perspective warp between images through view pose changes and optimize for the best pose that explains the difference in intensity. In our approach, we construct 3D representations of the images and optimize for the relative pose between them. Note that our registration method is more general, since our representation can be easily extended to incorporate 3D data from arbitrary sources. Hence, it can be employed for the registration of images to maps that aggregate multiple views.

Multi-Resolution Surfel Maps

Map Representation

We represent joint color and shape distributions at multiple resolutions in a probabilistic map. We use octrees as a natural data structure to represent spatial data at multiple resolutions. In each node of the tree, we store statistics on the joint spatial and color distribution of the points $\mathcal{P}$ within its volume. We approximate this distribution with sample mean μ and covariance Σ of the data, i. e., we model the data as normally distributed in a node's volume. Instead of directly maintaining mean and covariance in the nodes, we store the sufficient statistics $\mathcal{S}(\mathcal{P}) := \sum_{p\in\mathcal{P}} p$ and $\mathcal{S}^2(\mathcal{P}) := \sum_{p\in\mathcal{P}} pp^T$ of the normal distribution. From these, we obtain sample mean $\mu(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\mathcal{S}(\mathcal{P})$ and covariance $\Sigma(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\mathcal{S}^2(\mathcal{P}) - \mu\mu^T$.

We not only model shape by the distribution of 3D point coordinates in the nodes. We also add the RGB information of a point. By simply maintaining the joint distribution of 3D coordinates and color in a 6D normal distribution, we also model the spatial distribution of color. In order to separate chrominance from luminance information, we choose a variant of the HSL color space. We define the L$\alpha\beta$ color space as $L := \frac{1}{2}(\max\{R,G,B\} + \min\{R,G,B\})$,

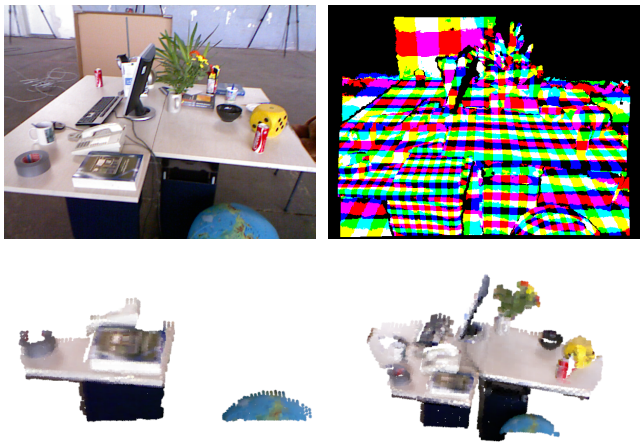

Figure 1: Top left: RGB image of the scene. Top right: Maximum node resolution coding, color codes octant of the leaf in its parent's node (see text for details). Bottom: Color and shape distribution at 0.025 m (left) and at 0.05 m resolution (right).

$\alpha := R - \frac{1}{2}G - \frac{1}{2}B$, and $\beta := \frac{\sqrt{3}}{2}(G - B)$. We obtain the chrominances α and β from the polar hue and saturation representation. Despite the simple and efficient conversion, this color space provides chrominance cues that are almost invariant to illumination changes.

Since we build maps of scenes and objects from all perspectives, multiple distinct surfaces may be contained within a node's volume. We model this by maintaining multiple surfels in a node that are visible from several view directions. We use six orthogonal view directions according to the normals on the six faces of a cube. When adding a new point to the map, we determine the view direction onto the point and associate it with the surfels belonging to the most similar view direction.

Shape-Texture Descriptor

We construct descriptors of shape and texture in the local context of each surfel (at all resolutions). Similar to FPFH features (Rusu, Blodow, and Beetz 2009), we first build histograms of surfel-pair relations between the query surfel and its 26 neighbors in the octree resolution. Each surfel-pair relation is weighted with the number of points in the corresponding voxel. Afterwards, we smooth the histograms to better cope with discretization effects by adding the histogram of neighboring surfels with a factor $\gamma = 0.1$.

Similarly, we extract local histograms of luminance and chrominance contrasts. We bin luminance and chrominance differences between neighoring surfels into positive, negative, or insignificant. Note, that pointers to neighboring voxels can be efficiently precalculated using look-up tables (Zhou et al. 2011).

Real-Time RGB-D Image Aggregation

The use of the sufficient statistics allows for an efficient incremental update of the map. In the simplest implementa-

tion, the sufficient statistics of each point is added individually to the tree. Starting at the root node, the sufficient statistics is recursively added to the nodes that contain the point in their volume.

Adding each point individually is, however, not the most efficient way to generate the map. Instead, we exploit that by the projective nature of the camera, neighboring pixels in the image project to nearby points on the sampled 3D surface — up to occlusion effects. This means that neighbors in the image are likely to belong to the same octree nodes.

We further consider the typical property of RGB-D sensors that noise increases with the distance of the measurement. We thus adapt the maximum octree resolution at a pixel to the pixel's squared distance from the sensor. In effect, the size of the octree is significantly reduced and the leaf nodes subsume local patches in the image (see top-right Fig. 1). We exploit these properties and scan the image to aggregate the sufficient statistics of contiguous image regions that belong to the same octree node. The aggregation of the image allows to construct the map with only several thousand insertions of node aggregates for a 640×480 image in contrast to 307,200 point insertions.

After the image content has been incorporated into the representation, we precompute mean, covariance, surface normals, and shape-texture features for later registration purposes.

Handling of Image and Virtual Borders

Special care must be taken at the borders of the image and at virtual borders where background is occluded. Nodes that receive such border points only partially observe the underlying surface structure. When updated with these points, the surfel distribution is distorted towards the partial distribution. In order to avoid this, we determine such nodes by sweeping through the image and neglect them.

Robust Real-Time Registration of Multi-Resolution Surfel Maps

The registration of multi-resolution surfel maps requires two main steps that need to be addressed efficiently: First, we associate surfels between the maps. For these associations, we then determine a transformation that maximizes their matching likelihood.

Multi-Resolution Surfel Association

Since we model multiple resolutions, we match surfels only in a local neighborhood that scales with the resolution of the surfel. In this way, coarse misalignments are corrected on coarser scales. In order to achieve an accurate registration, our association strategy chooses the finest resolution possible. This also saves redundant calculations on coarser resolutions.

Starting at the finest resolution, we iterate through each node in a resolution and establish associations between the surfels on each resolution. In order to choose the finest resolution possible, we do not associate a node, if one of its children already has been associated. Since we have to iterate our registration method multiple times, we can gain efficiency by bootstrapping the association process from previous iterations. If a surfel has not been associated in the previous iteration, we search for all surfels in twice the resolution distance in the target map. Note, that we use the current pose estimate x for this purpose. If an association from a previous iteration exists, we associate the surfel with the best surfel among the neighbors of the last association. Since we precalculate the 26-neighborhood of each octree node, this look-up needs only constant time.

We accept associations only, if the shape-texture descriptors of the surfels match. We evaluate the compatibility by thresholding on the Euclidean distance of the descriptors. In this way, a surfel may not be associated with the closest surfel in the target map.

Our association strategy not only saves redundant comparisons on coarse resolution. It also allows to match surface elements at coarser scales, when fine-grained shape and texture details cannot be matched on finer resolutions. Finally, since we iterate over all surfels independently in each resolution, we parallelize our association method.

Observation Model

Our goal is to register an RGB-D image z, from which we construct the source map m_s, towards a target map m_m. We formulate our problem as finding the most likely pose x that optimizes the likelihood $p(z|x, m_m)$ of observing the target map in the current image z. We express poses $x = (q, t)$ by a unit quaternion q for rotation and by the translation $t \in \mathbb{R}^3$.

We determine the observation likelihood by the matching likelihood between source and target map,

$$p(m_s|x, m_m) = \prod_{(i,j)\in\mathcal{A}} p(s_{s,i}|x, s_{m,j}), \tag{1}$$

where $\mathcal{A}$ is the set of surfel associations between the maps, and $s_{s,i} = (\mu_{s,i}, \Sigma_{s,i})$, $s_{m,j} = (\mu_{m,j}, \Sigma_{m,j})$ are associated surfels. The observation likelihood of a surfel match is the difference of the surfels under their normal distributions,

$$\begin{aligned} p(s_{s,i}|x, s_{m,j}) &= \mathcal{N}\left(d_{i,j}(x); 0, \Sigma_{i,j}(x)\right), \\ d_{i,j}(x) &:= \mu_{m,j} - T(x)\mu_{s,i}, \\ \Sigma_{i,j}(x) &:= \Sigma_{m,j} + R(x)\Sigma_{s,i}R(x)^T, \end{aligned} \tag{2}$$

where $T(x)$ is the homogeneous transformation matrix for the pose estimate x and $R(x)$ is its rotation matrix. We marginalize the surfel distributions for the spatial dimensions.

Note that due to the difference in view poses between the images, the scene content is differently discretized between the maps. We compensate for inaccuracies due to discretization effects by trilinear interpolation between target surfels.

Pose Optimization

We optimize the observation log likelihood

$$J(x) = \sum_{(i,j)\in\mathcal{A}} \log(|\Sigma_{i,j}(x)|) + d_{i,j}^T(x)\Sigma_{i,j}^{-1}(x)d_{i,j}(x) \tag{3}$$

for the pose x in a multi-stage process combining gradient descent and Newton's method.

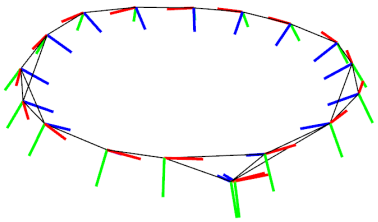

Figure 2: Examplary key view graph.

Since the stepsize for gradient descent is difficult to choose and the method converges only linearly, we use Newton's method to find a pose with high precision. For robust initialization, we first run several iterations of gradient descent to obtain a pose estimate close to a minimum of the log-likelihood.

In each step, we determine new surfel associations in the current pose estimate. We weight each surfel association according to the similarity in the shape-texture descriptors. Our method typically converges within 10-20 iterations of gradient descent and 5-10 iterations of Newton's method to a precise estimate. We parallelize the evaluation of the gradients and the Hessian matrix for each surfel which yields a significant speed-up on multi-core CPUs.

Estimation of Pose Uncertainty

We obtain an estimate of the observation covariance using a closed-form approximation (Censi 2007),

$$\Sigma(x) \approx \left(\frac{\partial^2 J}{\partial x^2}\right)^{-1} \frac{\partial^2 J}{\partial z \partial x} \Sigma(z) \frac{\partial^2 J}{\partial z \partial x}^T \left(\frac{\partial^2 J}{\partial x^2}\right)^{-1}, \quad (4)$$

where x is the pose estimate, z denotes the associated surfels in both maps, and $\Sigma(z)$ is given by the covariance of the surfels. The covariance estimate of the relative pose between the maps captures uncertainty along unobservable dimensions, for instance, if the maps view a planar surface.

Model Learning and Tracking

Model Learning

We aim at the learning of object models from several views. While the camera moves through the scene, we obtain a trajectory estimate using our registration method. Since small registration errors may accumulate in significant pose drift over time, we establish and optimize a graph of probabilistic spatial relations between similar view poses (see Fig. 2). We denote a view pose in the graph as key view.

We register each current frame to the most similar key view in order to keep track of the camera. Similarity is measured by distance in translation and rotation between view poses. At large distances, we add a new key view for the current frame to the graph. This also adds a spatial relation between the new key view and its reference key view. In addition, we check for and establish relations between similar key views.

Our probabilistic registration method provides a mean and covariance estimate for each spatial relation. We obtain the likelihood of the relative pose observation $z = (\hat{x}, \Sigma(\hat{x}))$ of the key view j from view i by

$$p(\hat{x}|x_i, x_j) = \mathcal{N}\left(\hat{x}; \Delta(x_i, x_j), \Sigma(\hat{x})\right), \quad (5)$$

where $\Delta(x_i, x_j)$ denotes the relative pose between the key views under their current estimates x_i and x_j.

From the graph of spatial relations we infer the probability of the trajectory estimate given the relative pose observations

$$p(x_{1,\ldots,N}|\hat{x}_1, \ldots, \hat{x}_M) \propto \prod_k p(\hat{x}_k|x_{i(k)}, x_{j(k)}). \quad (6)$$

We solve this graph optimization problem by sparse Cholesky decomposition using the g^{2}o framework (Kuemmerle et al. 2011). Finally, we fuse the key views in a full-view map using the optimized trajectory estimate. We extract object models from the key views within volumes of interest.

Pose Tracking

We apply our registration method to estimate the pose of the camera with respect to a model. We aggregate the current RGB-D image in a multi-resolution surfel map and register it to the object model. In order to safe unnecessary computations, we process the image in a volume of interest close to the last pose estimate. We only process image points that are likely under the spatial distribution of the model. Mean and covariance of this distribution are readily obtained from the sum of surfel statistics $|\mathcal{P}|$, $\mathcal{S}(\mathcal{P})$, and $\mathcal{S}^2(\mathcal{P})$ over all view directions in the root node of the tree.

Experiments

We evaluate our approach on a public RGB-D dataset (Sturm et al. 2011). The dataset contains RGB-D image sequences with ground truth information for the camera pose. The ground truth has been captured with a motion capture system. In addition, we generated complementary RGB-D datasets for the evaluation of object tracking[1]. The dataset is also annotated with ground truth acquired with a motion capture system. It contains three objects of different sizes (a chair, a textured box, and a small humanoid robot). For modelling, we recorded each object from a 360° trajectory. For evaluating tracking performance, we included three trajectories for each object with small, medium, and fast camera motion, respectively. Each dataset consists of 1000 frames recorded at 30 Hz and VGA (640×480) resolution. We set the maximum resolution of our maps to 0.0125 m throughout the experiments which is a reasonable lower limit in respect of the minimum measurement range of the sensor (ca. 0.4 m). We evaluate timings of our method on an Intel Xeon 5650 2.67 GHz Hexa-Core CPU using full resolution (VGA) images.

Incremental Registration

We first evaluate the properties of our registration method that underlies our object modelling and tracking approach.

[1] http://www.ais.uni-bonn.de/download/objecttracking.html

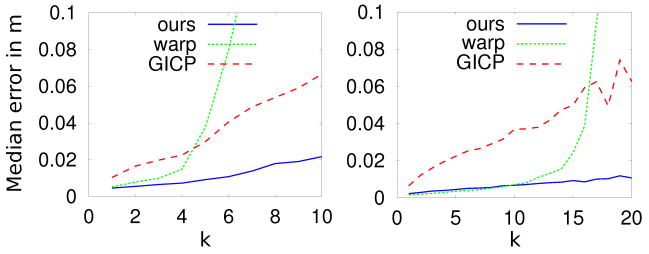

Figure 3: Median translational error of the pose estimate for different frame skips k on the freiburg1_desk (left) and freiburg2_desk (right) dataset.

dataset	ours	warp	GICP
freiburg1_desk	4.62 mm 0.0092 deg	5.3 mm 0.0065 deg	10.3 mm 0.0154 deg
freiburg2_desk	2.27 mm 0.0041 deg	1.5 mm 0.0027 deg	6.3 mm 0.0060 deg

Table 1: Comparison of median pose drift between frames.

We chose the freiburg1_desk and freiburg2_desk datasets as examples of fast and moderate camera motion, respectively, in an office-like setting. The choice also allows for comparison with the registration approach (abbreviated by *warp*) in (Steinbruecker, Sturm, and Cremers 2011).

Our approach achieves a median translational drift of 4.62 mm and 2.27 mm per frame on the freiburg1_desk and freiburg2_desk datasets, respectively (see Table 1). We obtain comparable results to *warp* (5.3 mm and 1.5 mm), while our approach also performs significantly better than GICP (10.3 mm and 6.3 mm (Steinbruecker, Sturm, and Cremers 2011)). When skipping frames (see Fig. 3), however, our approach achieves similar accuracy than *warp* for small displacements, but retains the robustness of ICP methods for larger displacements when *warp* fails. The mean processing time of our approach on the freiburg2_desk dataset is 100,11 msec (ca. 10 Hz).

Model Learning

We evaluate the accuracy of our object modelling approach by comparing trajectory estimates with ground truth. We characterize the trajectory error using the absolute trajectory error (ATE) measure proposed by (Sturm et al. 2011). Correspondences between poses in both trajectories are established by comparing time stamps. The trajectories are then aligned using singular value decomposition, and statistics on the position error between corresponding poses are calculated. It can be seen from Fig. 4 that our approach is well capable of recovering the trajectory of the camera. We provide the minimum, median, and maximum ATE of our trajectory estimates in Table 2. The median accuracy is about 1 cm for all datasets. It can also be seen that graph optimization significantly improves the trajectory estimate. Fig. 5 shows models learned with our approach.

We also evaluate our modelling approach on the freiburg1_desk and freiburg2_desk datasets (see Fig. 6) for comparison with a state-of-the-art RGB-D SLAM

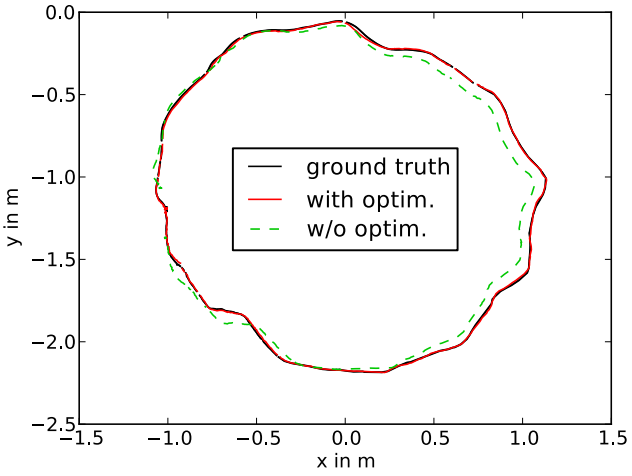

Figure 4: Ground truth (black) and trajectory estimates obtained without graph optimization (dashed green) and with graph optimization (solid red) on the box dataset.

Figure 5: Learned object models at a resolution of 2.5 cm visualized by samples from the color and shape surfel distributions. Left: humanoid, middle: box, right: chair (best viewed in color).

method (Engelhard et al. 2011). Both approaches perform similarly well on the shorter but challenging freiburg1_desk dataset. The accuracy in this dataset is strongly influenced by motion blur and misalignment effects between the RGB and depth images. The freiburg2_desk dataset contains a long trajectory in a loop around a table-top setting. Note that we have not implemented special loop-closing techniques. The drift of our incremental registration method is low enough to detect the loop closure simply through the similarity in the view poses. In addition, our registration method is robust enough to find a correct alignment from large view pose displacement. On this dataset, our method clearly outperforms RGB-D SLAM.

Object Tracking

In the following, we evaluate our object tracking method. The results in Table 3 demonstrate that our approach tracks the learned models with good accuracy in real-time. The tracking performance depends on distance, relative angle, and speed towards the object (see Fig. 7). For far view poses,

dataset	w/o graph optim. min	median	max	with graph optim. min	median	max
humanoid	1.6	61.3	242.7	4.7	14.8	61.8
box	29.0	77.5	169.0	1.2	9.8	38.5
chair	3.8	138.0	328.3	0.9	12.3	55.1

Table 2: Absolute trajectory error in mm obtained by incremental mapping w/o graph optim. and with our object modelling approach (with graph optim.).

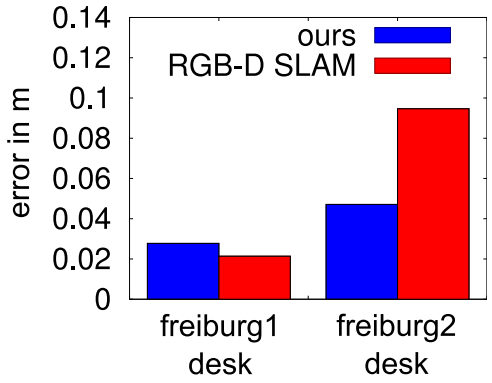

Figure 6: Comparison of the median absolute trajectory error on two datasets.

the measured points on the object map to coarse resolutions in the multi-resolution representation of the image. Thus, our approach registers the image on coarse scales to the object model and the accuracy decreases while the frame-rate increases compared to closer distances. The shape of the object may also influence the accuracy, such that it varies with view angle.

We have demonstrated our real-time tracking method publicly at RoboCup@Home competitions in 2011 and 2012[2]. In the Final at RoboCup 2011 our robot Cosero carried a table with a human (Stückler and Behnke 2011) and baked omelett. For carrying the table, we trained a model of the table. Cosero registered RGB-D images to the model in real-time to approach the table and grasp it. It detected the lifting and lowering of the table by estimating its pitch rotation. Similarly, Cosero approached the pan on a cooking plate by tracking the object with our registration method. At German Open 2012, Cosero moved a chair and watered a plant. It perceived chair and watering can with the proposed method even despite partial occlusions of the objects by the robot itself. The demonstrations have been well received by juries from science and media. Paired with the highest score from the previous stages, we could win both competitions.

[2]Videos of the demonstrations can be found at http://www.nimbro.net/@Home.

dataset	all frames ATE (mm)	all frames time (msec)	real-time ATE (mm)	real-time frames used (%)
humanoid slow	19.5	36.04 ± 6.06	19.01	95.5
humanoid med.	25.28	31.68 ± 5.97	25.52	92.1
humanoid fast	32.35	33.12 ± 7.9	32.47	88.1
box slow	15.09	49.4 ± 9.34	15.25	41.7
box med.	31.78	49.45 ± 23.24	54.38	51.2
box fast	20.48	35.69 ± 27.78	23.95	60.4
chair slow	15.58	48.87 ± 8.34	16.91	47.1
chair med.	15.18	53.10 ± 12.19	15.31	49.3
chair fast	26.87	48.24 ± 13.00	27.77	51.7

Table 3: Median absolute trajectory error, avg. time ± std. deviation, and percentage of frames used in real-time mode for our tracking approach.

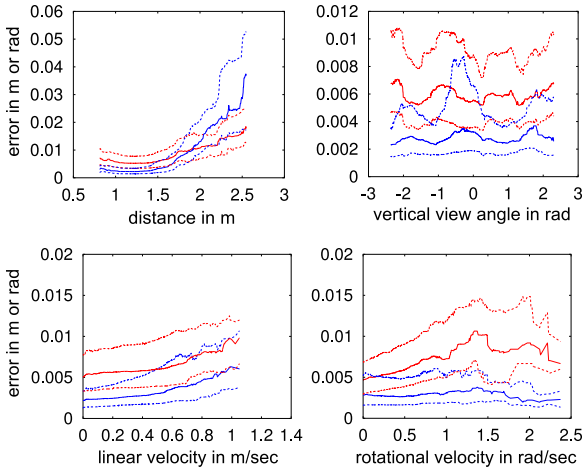

Figure 7: Tracking performance: Median translational (blue) and rotational error (red) and their quartiles (dashed lines) w.r.t. distance, vertical view angle, linear velocity, and rotational velocity on the box tracking datasets.

Conclusion

We proposed a novel approach to model learning and tracking of objects using RGB-D cameras. Central to our approach is the representation of spatial and color measurements in multi-resolution surfel maps. We exploit measurement principles of RGB-D cameras to efficiently acquire maps from images. The transformation between maps is estimated with an efficient yet robust registration method in real-time. Our approach utilizes multiple resolutions to align the maps on coarse scales and to register them accurately on fine resolutions. We demonstrate state-of-the-art registration results w.r.t. accuracy and robustness.

We incorporate our registration method into a probabilistic trajectory optimization framework which allows for learning full-view object models with good precision. Our approach compares well with a recent approach to SLAM using RGB-D cameras. Finally, we use the learned models to track the 6-DoF pose of objects in camera images accurately in real-time. By the multi-resolution nature of our image and object maps, our method inherently adapts its registration scale to the distance-dependent measurement noise.

In future work, we will investigate approaches to further object perception problems such as object detection and initial pose estimation based on multi-resolution surfel maps.

We also will investigate the modelling of larger scenes that requires robust techniques for the detection of loop-closures.

References

Besl, P. J., and McKay, N. D. 1992. A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.

Censi, A. 2007. An accurate closed-form estimate of ICP's covariance. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Chen, Y., and Medioni, G. 1992. Object modelling by registration of multiple range images. *Image Vision Comput.* 10:145–155.

Engelhard, N.; Endres, F.; Hess, J.; Sturm, J.; and Burgard, W. 2011. Real-time 3D visual SLAM with a hand-held camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*.

Klein, G., and Murray, D. 2007. Parallel tracking and mapping for small AR workspaces. In *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*.

Krainin, M.; Henry, P.; Ren, X.; and Fox, D. 2011. Manipulator and object tracking for in-hand 3D object modeling. *Int. Journal of Robotics Research* 30.

Kuemmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; and Burgard, W. 2011. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Magnusson, M.; Duckett, T.; and Lilienthal, A. J. 2007. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*.

Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohli, P.; Shotton, J.; Hodges, S.; and Fitzgibbon, A. 2011. KinectFusion: real-time dense surface mapping and tracking. In *Proc. of the 10th Int. Symposium on Mixed and Augmented Reality (ISMAR)*.

Newcombe, R.; Lovegrove, S.; and Davison, A. 2011. DTAM: Dense tracking and mapping in real-time. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*.

Nuechter, A.; Lingemann, K.; Hertzberg, J.; and Surmann, H. 2005. 6D SLAM with approximate data association. In *Proc. of the 12th Int. Conf. on Advanced Robotics (ICAR)*.

Rusu, R. B.; Blodow, N.; and Beetz, M. 2009. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-ICP. In *Proc. of Robotics: Science and Systems*.

Steinbruecker, F.; Sturm, J.; and Cremers, D. 2011. Real-time visual odometry from dense RGB-D images. In *Workshop on Live Dense Reconstruction with Moving Cameras at the Int. Conf. on Computer Vision (ICCV)*.

Stückler, J., and Behnke, S. 2011. Following human guidance to cooperatively carry a large object. In *Proc. of the 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*.

Stuehmer, J.; Gumhold, S.; and Cremers, D. 2010. Real-time dense geometry from a handheld camera. In *Proc. of the DAGM Conference*.

Sturm, J.; Magnenat, S.; Engelhard, N.; Pomerleau, F.; Colas, F.; Burgard, W.; Cremers, D.; and Siegwart, R. 2011. Towards a benchmark for RGB-D SLAM evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*.

Weise, T.; Wismer, T.; Leibe, B.; and Gool, L. V. 2009. In-hand scanning with online loop closure. In *Proc. of the IEEE International Conference on Computer Vision Workshops*.

Zhou, K.; Gong, M.; Huang, X.; and Guo, B. 2011. Data-parallel octrees for surface reconstruction. *IEEE Trans. on Visualization and Computer Graphics*.