In Proc. of the IEEE Int. Conf. on Multisensor Fusion and Information Integration (MFI), Hamburg, Germany, 2012

Integrating Depth and Color Cues for Dense Multi-Resolution Scene Mapping Using RGB-D Cameras

Jörg Stückler and Sven Behnke

***Abstract*— The mapping of environments is a prerequisite for many navigation and manipulation tasks. We propose a novel method for acquiring 3D maps of indoor scenes from a freely moving RGB-D camera. Our approach integrates color and depth cues seamlessly in a multi-resolution map representation. We consider measurement noise characteristics and exploit dense image neighborhood to rapidly extract maps from RGB-D images. An efficient ICP variant allows maps to be registered in real-time at VGA resolution on a CPU. For simultaneous localization and mapping, we extract key views and optimize the trajectory in a probabilistic framework. Finally, we propose an efficient randomized loop-closure technique that is designed for on-line operation. We benchmark our method on a publicly available RGB-D dataset and compare it with a state-of-the-art approach that uses sparse image features.**

I. INTRODUCTION

A "good" environment model is prerequisite to many applications in computer vision and robotics. We propose a novel approach to acquire explicit geometric models of indoor scenes with a RGB-D camera that include color and depth cues seamlessly. Such types of model can be used to localize a mobile robot precisely in its environment, for motion planning, or it can be enhanced with further semantic information.

We represent our models by local surface elements (surfels) at multiple spatial resolutions. Each surfel consists of the joint spatial and color distribution in its volume of interest. We propose highly efficient means to convert RGB-D images into compact models. Our map representation can be built from only one image or it fuses many images from various view points to build compact local maps of larger volumes. By this, for instance, our mapping framework can be employed in a submap-based SLAM framework.

The use of multiple resolutions has advantages over flat models. By the projective image formation process, sampling density depends on depth and view angle onto observed surfaces. The sampling density defines a maximum resolution at which a surfel is still well observed. Furthermore, we adapt the maximum resolution at a point to its depth. By this, we consider distance-dependent noise in the depth measurements that is intrinsic to the disparity measurement principle.

For scene mapping, we propose an accurate registration method that aligns multi-resolution surfel maps in real-time. Our approach makes use of all available data in the image and registers maps at the finest resolution possible. In order to recover the trajectory of the sensor, we extract key views from the image stream and keep track of the current sensor pose by image registration. We propose a randomized loop-closure technique that establishes constraints between similar key views on-line. Finally, we optimize for the joint likelihood of the pose estimates in a probabilistic graph optimization framework.

All authors are with the Autonomous Intelligent Systems Group, Computer Science Institute VI, University of Bonn, 53113 Bonn, Germany {stueckler, behnke} at ais.uni-bonn.de

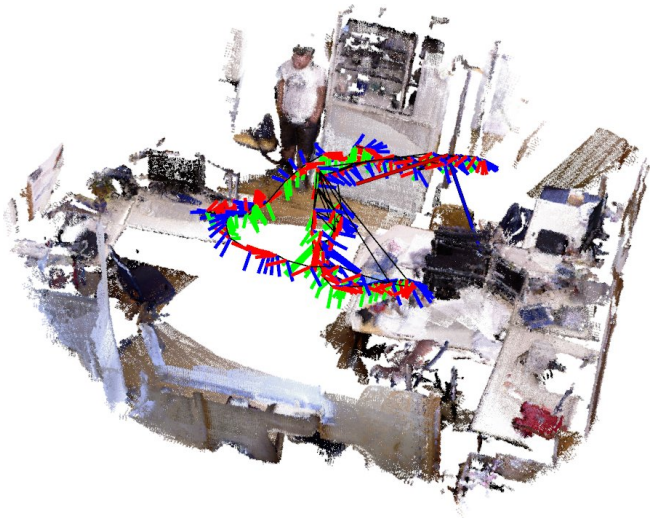

Fig. 1. Map and trajectory acquired with our approach. The map is visualized by samples from the joint color and shape surfel distribution at 5 cm resolution.

Our approach yields accurate 3D texture and shape models of indoor scenes (see Fig. 1). We evaluate our method on a publicly available RGB-D benchmark dataset. This allows for comparison of the performance of our algorithm with state-of-the-art approaches.

II. RELATED WORK

Scene modelling has long been investigated in the computer vision and robotics communities. Early work on simultaneous localization and mapping (SLAM) in robotics has focused on acquiring 2D maps with mobile robots using range sensors such as laser scanners and sonars (e.g., [1]). 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 [2], [3], [4].

In computer vision, many approaches to Structure from Motion (SfM) are based on the extraction and matching of keypoints between images. Stereo vision is frequently used to directly obtain depth measurements for keypoints [5], [6]. Efficient RANSAC methods can then be applied to estimate

the motion of the camera rig. MonoSLAM [7] was one of the first methods that demonstrated SfM in real-time with a single camera. More recently, Klein and Murray [8] proposed a real-time capable bundle-adjustment method within small workspaces.

Current work on SfM in computer vision also includes real-time dense surface reconstruction from monocular videos [9], [10]. Newcombe et al. [10] proposed DTAM, an impressive method for dense tracking and mapping with a monocular camera. DTAM acquires depth maps for individual key images of the RGB camera in real-time on a GPU. These RGB images and their depth maps could be used instead of images from RGB-D cameras in our scene mapping framework.

In recent years, affordable depth cameras have become available such as time-of-flight or structured-light cameras like the Microsoft Kinect. Paired with the developments in computer vision on real-time dense depth estimation from monocular image sequences, exploiting dense depth for robotic perception is now a viable option. However, efficient means have to be developed to utilize the high frame-rate and high resolution images provided by such sensing modalities.

Recently, Newcombe et al. [11] proposed KinectFusion. They incrementally register RGB-D images to a map that is aggregated from previous images using GPUs. While KinectFusion still accumulates minor drift in the map estimate over time as of its incremental nature, the authors demonstrate remarkable performance for the mapping of small workspaces. The approach is applied for augmented reality user interfaces, and supports the tracking of the pose of objects and the camera in real-time. Since KinectFusion is implemented on GPU, it has stronger workspace limitation than CPU-based implementations like ours due to memory restrictions. In order to scale to larger workspaces, local submaps have to built and eventually registered in a submap-based SLAM framework. Our framework supports a compact representation of local submaps. Our registration method is already suited for registering individual RGB-D images as well as entire local submaps that fuse many images. To map environments with loop-closure, we find a best alignment of key views by jointly optimizing spatial relations between views. We determine the relative pose between views using our registration method and assess the uncertainty of the pose estimate.

Some approaches have been proposed that also learn maps from RGB-D images in a trajectory optimization framework [12], [13]. Henry et al. [12] extract textured surface patches, register them using ICP [14] to the model, and apply graph-optimization to obtain an accurate map. Our approach provides shape-texture information in a compact representation. Our maps support registration of further views from a wide range of distances, since the model contains detail at multiple scales. Engelhard et al. [13] 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.

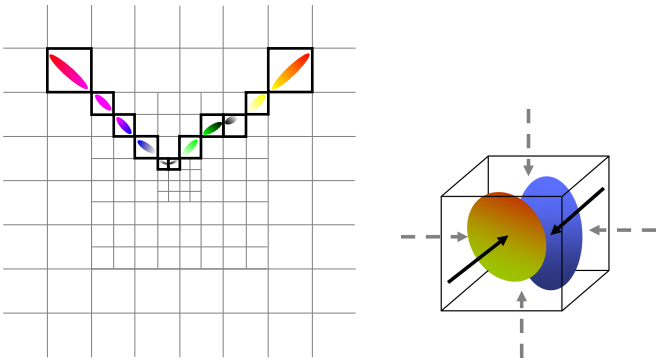

Fig. 2. Multi-Resolution Surfel Maps. Left: We represent RGB-D data by subdividing 3D space into voxels at multiple resolutions within an octree. In each voxel we maintain shape and color distributions of contained points. Considering the depth-dependent noise model in RGB-D images we obtain a local multi-resolution structure by limiting the maximum resolution in the tree with the depth. Right: We accumulate measurements from several view poses in a single map by maintaining surfels for six orthogonal viewing directions in each voxel.

III. MULTI-RESOLUTION SURFEL MAPS

A. Map Representation

We concisely represent RGB-D data in Multi-Resolution Surfel Maps [15]. We use octrees to model textured surfaces at multiple resolutions in a probabilistic way. At each voxel in every resolution of the tree we store the joint distribution of the spatial and color components of the points that fall into the voxel. We approximate this distribution by its first and second moment, i.e., sample mean and covariance (see Fig. 2).

We enhance each surfel in the map with a local shape-texture descriptor to guide data association during registration (see [15] for details). In order to be able to incorporate images from several view poses within one map, we maintain up to six surfels in each voxel from orthogonal viewing directions.

B. Real-Time RGB-D Image Aggregation

Multi-Resolution Surfel Maps can be efficiently aggregated from RGB-D images. We maintain the sufficient statistics of the color and shape distribution in the voxels in order to incrementally update the map. Instead of naively adding each pixel individually to the map, we propose to efficiently accumulate image regions before building up the octree.

This is possible, since points that fall into the same 3D voxel are likely to project to nearby pixels in the image (see Fig. 3). Furthermore, RGB-D sensors often obey a characteristic noise model in which depth measurement noise scales quadratically with the actual depth. By this, the maximum resolution at a pixel can be limited with depth and, hence, pixels at distant positions that belong to the same octree leaf still form larger contiguous regions in the image. The aggregation of leaf statistics within the image allows to construct the map with only several 1,000 insertions of node aggregates for a 640×480 image in contrast to 307,200 point insertions.

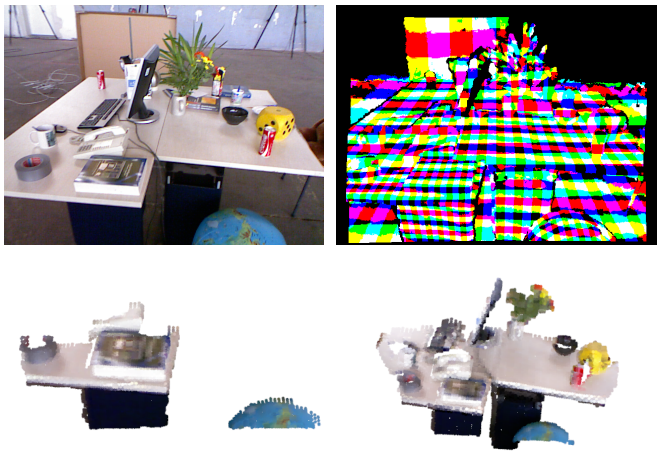

Fig. 3. Top left: RGB image of the scene. Top right: Maximum node resolution coding, color codes octant of the leaf in its parent's node (see text for details). Bottom: Color and shape distribution at 0.025 m (left) and at 0.05 m resolution (right).

IV. ROBUST REAL-TIME REGISTRATION OF MULTI-RESOLUTION SURFEL MAPS

We register Multi-Resolution Surfel Maps in a dual iterative refinement process: In each iteration, we associate surfels between the maps given the current pose estimate. Using these associations we then determine a new pose that maximizes the matching likelihood for the maps. We make use of the multi-resolution nature of our maps for an efficient association strategy. We also handle discretization effects that are introduced into the map by the binning of measurements within the octree to obtain an accurate registration estimate.

A. Multi-Resolution Surfel Association

Since we match maps at multiple resolutions, we associate surfels only in a local neighborhood that scales with the resolution of the surfel (see Fig. 4). 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 amounts to 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.

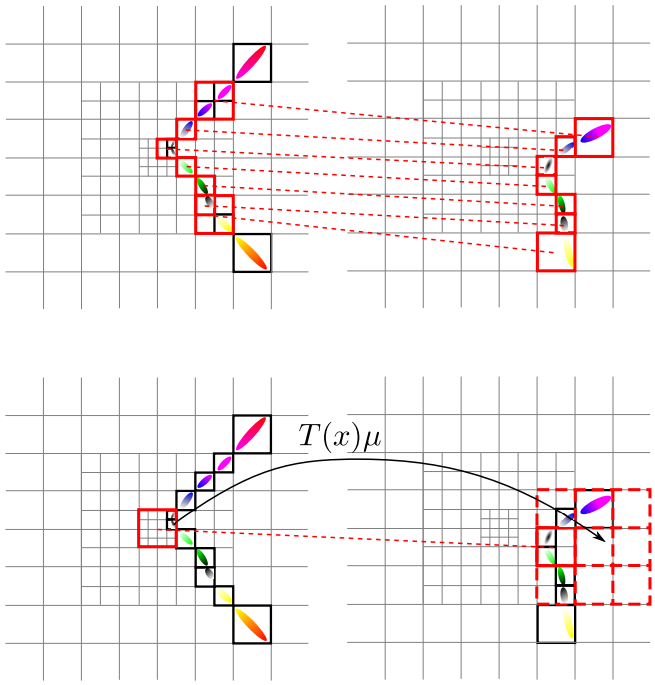

Fig. 4. We register Multi-Resolution Surfel Maps in a multi-resolution data association strategy. Top: We determine the matching on the finest resolution shared by both maps to achieve high accuracy in pose estimation. Bottom: For each surfel in the scene map we search for a closest match (red dashed lines depict associations) in the model map under the pose estimate x. By searching at the projected mean $T(x)\mu$ of the surfel in a local volume (red squares) that scales with the resolution of the surfel, we efficiently correct misalignments from coarse to fine resolutions.

Our association strategy not only saves redundant comparisons on coarse resolution. It also matches 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.

B. 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,

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

where $\mathcal{A}$ is the set of surfel associations between the maps, and $s_{s,i} = (\mu_{s,i}, \Sigma_{s,i})$ and $s_{m,j} = (\mu_{m,j}, \Sigma_{m,j})$ 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. This is possible, when a scene surfel $s_{s,i}$ is directly associated with the model surfel sm;j in the octree node at the projected position of the scene surfel $T(x)\mu_{s,i}$. Instead of directly using the associated model surfel $s_{m,j}$ in the observation likelihood (eq. (2)), we consider the surfel representation in the model map as a Gaussian Mixture model, and determine mean and covariance of the model surfel at the projected position $T(x)\mu_{s,i}$ through trilinear interpolation of neighboring surfels in the model map.

C. 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.

Since gradient descent 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.

D. Estimation of Pose Uncertainty

We obtain an estimate of the observation covariance using a closed-form approximation [16],

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

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.

V. ON-LINE TRAJECTORY OPTIMIZATION

We will now describe our method for simultaneous localization and mapping. 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. We denote a view pose in the graph as key view. We propose a randomized method to add spatial constraints between similar views during on-line operation. By this, we also detect the closure of trajectory loops.

A. Incremental Generation of Key View Graph

We register the current frame to the closest key view (the reference key view) in order to keep track of the camera. We measure 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.

B. Constraint Detection

After each image update, we check for a new constraint for the current reference key view. We determine for all unestablished constraints of the current reference key view v_{ref} to other key views v a probability

$$p_{\text{chk}}(v) = \mathcal{N}\left(d(v_{\text{ref}}, v); 0, \sigma_d^2\right) \cdot \mathcal{N}\left(|\alpha(v_{\text{ref}}, v)|; 0, \sigma_\alpha^2\right) \tag{5}$$

that depends on the linear and rotational distances $d(v_{\text{ref}}, v)$ and $|\alpha(v_{\text{ref}}, v)|$ of the key view poses, respectively. We sample a key view v according to $p_{\text{chk}}(v)$ and determine the relative pose of the key views using our registration method.

In order to validate the matching of the key views, we determine their matching likelihood under the pose estimate. For each surfel in one of the key views, we find the best matching surfel in the second view. We directly take into account the consistency of the surface normals between the surfels and therefore determine the matching likelihood of the surfels as the product of the likelihood under their distributions and under a normal distribution in the angle between their normals. We assign a minimum likelihood to all surfels with a worse match or without a match. In this way, the matching likelihood accounts for the overlap between the views. This likelihood is directional and, hence, we evaluate it in both directions.

Since the matching likelihood depends on the observed scene content, we cannot use a global threshold for deciding if a constraint should be added. Instead we require the matching likelihood of a new constraint to be at least a fraction of the matching likelihood for the initial constraint of the key view. This constraint has been established through tracking from the referred key view and is thus assumed to be consistent.

C. Graph Optimization

Our probabilistic registration method provides a mean and covariance estimate for each spatial relation. We obtain the

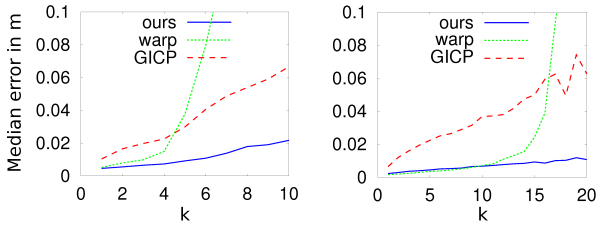

Fig. 5. Median translational error of the pose estimate for different frame skips k on the freiburg1_desk (left) and freiburg2_desk (right) dataset.

likelihood of the relative pose observation $z = (\hat{x}, \Sigma(\hat{x}))$ of the key view j from view i by

$$p(\hat{x}|x_i, x_j) = \mathcal{N}\left(\hat{x}; \Delta(x_i, x_j), \Sigma(\hat{x})\right), \tag{6}$$

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

$$p(x_{1,\ldots,N}|\hat{x}_1, \ldots, \hat{x}_M) \propto \prod_k p(\hat{x}_k|x_{i(k)}, x_{j(k)}). \tag{7}$$

We solve this graph optimization problem by sparse Cholesky decomposition using the g^{2}o framework [17]. At each image update, we optimize the graph for a single iteration.

VI. EXPERIMENTS

We evaluate our approach on a public RGB-D dataset [18]. 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. We measure timings on an Intel Xeon 5650 2,67 GHz Hexa-Core CPU using VGA resolution (640×480) images.

A. Incremental Registration

We first evaluate the properties of our registration method. 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 allows for comparison with the registration approach (abbreviated by *warp*) in [19].

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 I). 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 [19]). However, when skipping frames (see Fig. 5), our approach achieves similar accuracy to *warp* for small displacements, but retains the robustness of ICP methods for larger displacements when *warp* fails.

The mean processing time on the freiburg2_desk dataset is 100,11 msec (ca. 10 Hz).

B. Indoor SLAM

We evaluate our SLAM approach on 11 sequences of the RGB-D benchmark dataset and compare our approach to RGB-D SLAM ([20], [21]). We employ the absolute

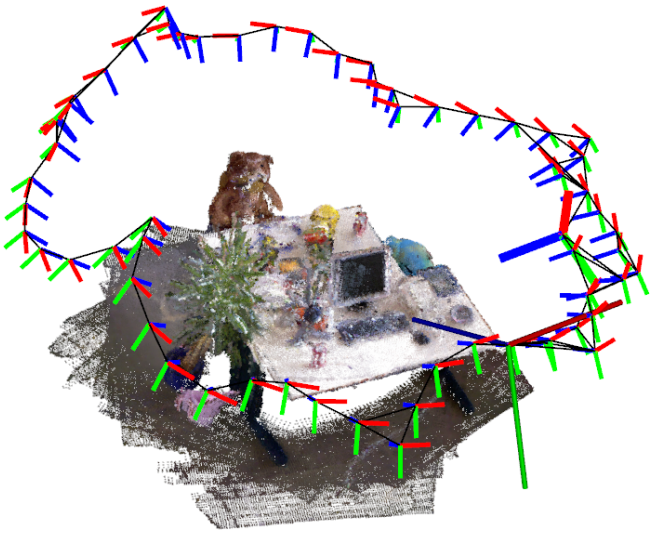

Fig. 6. 3D map (5 cm resolution) and camera trajectory result of our approach on the freiburg2_desk dataset.

trajectory error (ATE) and relative pose error (RPE) metrics as proposed in [18]. For the SLAM experiments, we measure the average RPE over all numbers of frame skips. Fig. 6 shows a typical result obtained with our approach on the freiburg2_desk sequence. This sequence contains moderately fast camera motion in a loop around a table-top setting. The freiburg1_room sequence contains a trajectory loop through an office (see Fig. 1). The camera moves much faster than in the freiburg2_desk sequence. On both datasets, our method clearly outperforms RGB-D SLAM in both error metrics (see Table II). The freiburg1_desk and freiburg1_desk2 sequences do not contain such large trajectory loops. The camera is swept quickly back and forth over a table-top setting. While on freiburg1_desk RGB-D SLAM performs better, both methods achieve similar results on the freiburg1_desk2 sequence. In average, our method yields lower AT and RP errors on the sequences in Table II.

Note, that our method did not succeed on sequences such as freiburg1_floor or freiburg2_large_loop. On freiburg1_floor the camera sweeps over a floor with only little texture that could be captured by the local descriptors of the surfels. Our method also cannot keep track of the camera pose, if large parts of the image contain no valid or highly uncertain depth at large distances.

The processing time for on-line operation is mainly governed by our registration method. At each image update, we

TABLE I

COMPARISON OF MEDIAN POSE DRIFT BETWEEN FRAMES.

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

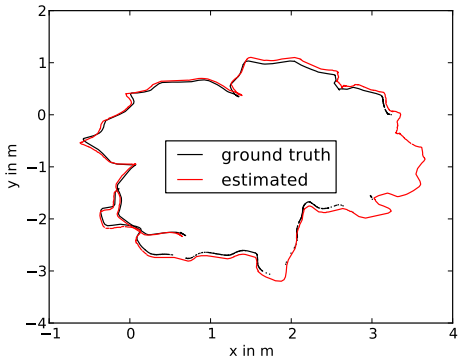

Fig. 7. Projection onto the x-y-plane of ground truth and trajectory estimate of our approach on the freiburg2_desk dataset.

have to register 2 pairs of views. First, we keep track of the current sensor pose by aligning the image to the closest key view in the map. Our randomized constraint detection method invokes a second registration at each image update. In our experiments, one iteration of graph optimization could be performed in the order of 1 ms.

VII. CONCLUSIONS

We proposed a novel approach to SLAM with RGB-D cameras in indoor environments. In our method, we compress the image content efficiently in 3D Multi-Resolution Surfel Maps. This map representation is well suited for accurate real-time registration by directly matching surfels and optimizing their matching likelihood. Our registration method also provides an estimate of pose uncertainty. We use this information to smooth the joint trajectory estimate in a probabilistic optimization framework. We present means to establish spatial relations between similar key views and to detect loop-closures during on-line operation.

Our approach yields accurate estimates of map and trajectory. On a benchmark dataset we could demonstrate, that in most cases the accuracy of our method is similar or even better compared to a state-of-the-art approach that maps sparse image features. Since our method strongly relies on dense depth, it is less accurate, if large parts of the image have no valid or only highly uncertain depth measurements. We will therefore combine our dense depth registration method with the alignment of point or contour features.

In future work, we will further investigate loop-closure detection for long trajectory loops through appearance modalities. We will also incorporate graph management strategies such as sub-graph hierarchies and graph pruning to allow for long-term operation.

TABLE II

COMPARISON OF OUR SLAM APPROACH WITH RGB-D SLAM IN ABSOLUTE TRAJECTORY (ATE) AND RELATIVE POSE ERROR (RPE).

	RMSE ATE in m		RMSE RPE in m	
dataset	ours	RGB-D SLAM	ours	RGB-D SLAM
freiburg1_360	0.069	0.079	0.110	0.103
freiburg1_desk2	0.049	0.043	0.090	0.102
freiburg1_desk	0.043	0.023	0.075	0.049
freiburg1_plant	0.026	0.091	0.044	0.142
freiburg1_room	0.069	0.084	0.139	0.219
freiburg1_rpy	0.027	0.026	0.040	0.042
freiburg1_teddy	0.039	0.076	0.073	0.138
freiburg1_xyz	0.013	0.014	0.020	0.021
freiburg2_desk	0.052	0.095	0.099	0.143
freiburg2_rpy	0.024	0.019	0.034	0.026
freiburg2_xyz	0.020	0.026	0.030	0.037
average	**0.039**	0.052	**0.069**	0.093

REFERENCES

[1] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, 2007.

[2] A. Nuechter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM with approximate data association," in *Proc. of the Int. Conf. on Advanced Robotics*, 2005.

[3] M. Magnusson, T. Duckett, and A. Lilienthal, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. of Field Rob.*, 2007.

[4] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. of Robotics: Science and Systems*, 2009.

[5] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2001.

[6] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *Int. J. of Robotics Research*, 2010.

[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.

[8] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. of IEEE/ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.

[9] J. Stuehmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Proc. of the DAGM Conference*, 2010.

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

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

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *Proc. of Int. Symposium on Experimental Robotics (ISER)*, 2010.

[13] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual SLAM with a hand-held camera," in *Proc. of RGB-D Workshop on 3D Perception in Robotics at Europ. Rob. Forum*, 2011.

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

[15] J. Stückler and S. Behnke, "Robust real-time registration of RGB-D images using multi-resolution surfel representations," in *Proc. of the 7th German Conference on Robotics (ROBOTIK)*, 2012.

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

[17] R. Kuemmerle, 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 and Automation (ICRA)*, 2011.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the IEEE Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.

[19] F. Steinbruecker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Workshop on Live Dense Reconstruction with Moving Cameras at ICCV*, 2011.

[20] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[21] F. Endres, J. Hess, N. Engelhard, J. Sturm, , and W. Burgard, "6D visual SLAM for RGB-D sensors," *at - Automatisierungstechnik*, 2012.