In Proc. of the German Conference on Robotics (ROBOTIK), Munich, Germany, May 2012

Robust Real-Time Registration of RGB-D Images using Multi-Resolution Surfel Representations

Jörg Stückler and Sven Behnke
Autonomous Intelligent Systems, University of Bonn, Germany

Abstract

Fast and robust registration of 3D scans is required in many approaches to perception in robotics such as pose tracking or simultaneous localization and mapping. We propose a novel efficient method to register RGB-D images. We convert the image content into a multi-resolution surfel representation and exploit the dense image neighborhood to construct such views at high frame-rates on a single CPU. Our approach registers views using an efficient and robust variant of the Iterative Closest Points algorithm. We evaluate our method on a recently published benchmark dataset and achieve results beyond the state-of-the-art. We also report on the successful public demonstration of our method at RoboCup 2011.

1 Introduction

Scan registration is an important capability for solving various problems in robotics. Many approaches to simultaneous localization and mapping (SLAM) register 2D or 3D scans to obtain pose measurements. Similarly, scan registration can be used to track the pose of an object. Robustness and efficiency of a registration method determine the magnitude of the sensor motion relative to the object or scene that the method can cope with. In this paper, we propose a robust method that registers RGB-D images at high frame-rates.

In recent years, affordable depth cameras have become available such as time-of-flight or structured-light cameras like the ASUS Xtion Pro. Conversely, in the computer vision community, approaches have been recently proposed that estimate dense depth in real-time from stereo and monocular cameras [9, 4]. Exploiting dense depth for robotic perception is since 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.

In this paper, we propose a fast method to extract multi-resolution surfel views from RGB-D images. Our method represents color and shape distributions at multiple resolutions. We present an approach for high frame-rate incremental registration of views that utilizes color and shape as well as the multi-resolution structure of the views.

2 Related Work

Over the last decades, the robotics, computer vision, and computer graphics communities developed several approaches for incremental registration of color and depth data. Recently, Steinbruecker et al. [7] proposed a method for real-time registration of RGB-D images. Given depth, they model the perspective transformation of an image for changes in view pose. They optimize an energy-function to find the best pose to explain the difference between images. Our formulation determines the best transformation between 3D representations of the images. Note that our registration method is more general, since our representation can be easily extended to incorporate arbitrary 3D data. Hence, it can be readily employed for the registration of images to maps that aggregate multiple views.

In robotics and computer graphics, depth images are frequently registered by derivatives of the Iterative Closest Points (ICP [1]) algorithm. Generalized-ICP [6] unifies the ICP formulation for various error metrics such as point-to-point, point-to-plane, and plane-to-plane. Magnusson et al. [3] propose the 3D normal distribution transform (3D-NDT). They discretize the 3D space in a grid to support efficient nearest neighbor look-ups. Each cell maintains the 3D normal distibution of the points in the model scan. Scans are registered by minimizing the matching likelihood of scene points to the model. In Color-NDT [2], they propose to enrich 3D-NDT with Gaussian mixture distributions of color in each cell. To the best of our knowledge, none of the above ICP methods is reported to support real-time capable scan-matching of RGB-D images.

Our approach bears some similarities to 3D-NDT. However, we propose novel methods to increase robustness and to enable high frame-rate operation on RGB-D images: Our approach exploits measurement principles of RGB-D sensors to extract multi-resolution surfel views at high frame-rates. To register such views efficiently, we propose a multi-resolution strategy to data association. This strategy is supported by the use of shape-texture features to judge the compatibility between surfels. Our highly efficient implementation registers 640×480 RGB-D images at a frame rate of about 10 Hz on a CPU.

3 Efficient Image Aggregation

3.1 Multi-Resolution Surfel Representation

We represent joint color and shape distributions at multiple resolutions using surfels (see Fig. 1).

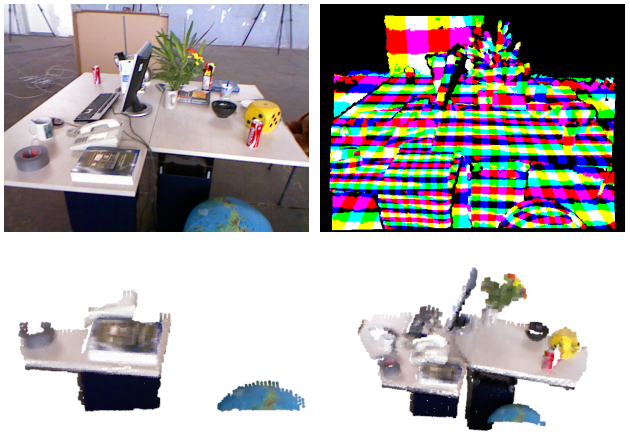

Figure 1: Top left: RGB image of the scene. Top right: Maximum node resolution coding, color codes octant of the leaf in its parent's node (see text for details). Bottom: Color and shape distribution at 0.025 m (left) and at 0.05 m resolution (right).

We use octrees as a natural data structure to represent spatial data at multiple resolutions. Each node in the octree represents a cubic volume, i. e., a voxel. In each node we store statistics on the joint spatial and color distribution of the points $\mathcal{P}$. We approximate this distribution with sample mean μ and covariance Σ of the data, i. e., we model the data normal distributed in a node's volume. Instead of directly maintaining mean and covariance in the nodes, we store the sufficient statistics $\mathcal{S}$ and $\mathcal{S}^2$ of the normal distribution, i. e.,

$$\begin{aligned} \mathcal{S}(\mathcal{P}) &:= \sum_{p \in \mathcal{P}} p, \text{ and} \\ \mathcal{S}^2(\mathcal{P}) &:= \sum_{p \in \mathcal{P}} pp^T, \end{aligned} \tag{1}$$

from which we obtain sample mean $\mu(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\mathcal{S}(\mathcal{P})$ and covariance $\Sigma(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\mathcal{S}^2(\mathcal{P}) - \mu\mu^T$.
Additionally, we maintain an estimate of the local surface normal within the voxel. We obtain surface normals by eigen decomposition of the 3D spatial covariance.
We also represent color in our representation. By 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

$$\begin{aligned} L &:= \frac{1}{2}\left(\max\{R,G,B\} + \min\{R,G,B\}\right), \\ \alpha &:= R - \frac{1}{2}G - \frac{1}{2}B, \text{ and } \beta := \frac{\sqrt{3}}{2}(G-B), \end{aligned} \tag{2}$$

where 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 approximately invariant to illumination changes.

3.2 Shape-Texture Descriptor

We construct descriptors of shape and texture in the local context of each surfel (at all resolutions). Similar to FPFH features [5], 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 [11].

3.3 Real-Time RGB-D Image Aggregation

The use of the sufficient statistics allows for an efficient incremental update of the representation. In the simplest implementation, 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.
However, adding each point individually is not the most efficient way to generate the representation. 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 view with only several 1000 node insertions for a 640×480 image in contrast to 307200 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.

3.4 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.

4 Real-Time Registration of Multi-Resolution Surfel Views

The registration of the images requires two main steps that needs to be addressed efficiently: First, we associate surfels between the views. For these associations, we then determine a transformation that maximizes their matching likelihood.

4.1 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 to match two views. This also saves redundant calculations on coarser resolutions.

Starting at the finest resolution, we iterate through every 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 view. Note, that we use the current pose estimate x for this purpose.

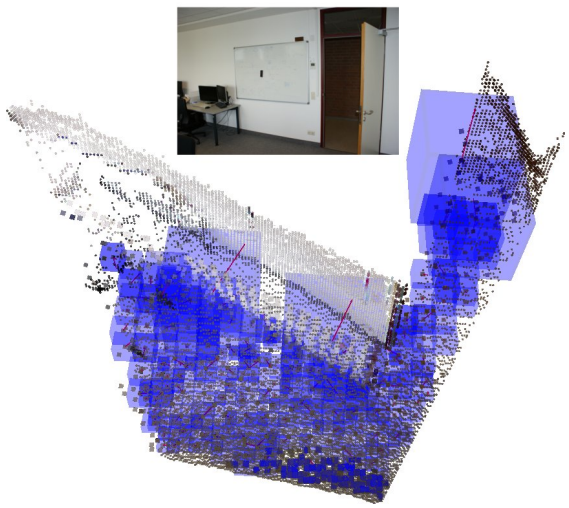

Figure 2: We match surfels at multiple resolutions. Coarse alignments are performed on coarse resolutions, while we choose the finest resolution possible for accuracy.

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 view, but with either none or a compatible one.

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 in each resolution in parallel, our association method can be easily parallelized.

4.2 Observation Model

Our goal is to register an RGB-D image z, from which we construct the source view m_s, towards a target view 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 view 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 view,

$$p(m_s|x, m_m) = \prod_{(i,j)\in\mathcal{A}} p(s_{s,i}|x, s_{m,j}), \tag{3}$$

where $\mathcal{A}$ is the set of surfel associations between the views, 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,

$$\begin{aligned} p(s_{s,i}|x, s_{m,j}) &= \mathcal{N}\left(d_{i,j}(x); 0, \Sigma_{i,j}(x)\right), \\ d_{i,j}(x) &:= \mu_{m,j} - T(x)\mu_{s,i}, \\ \Sigma_{i,j}(x) &:= \Sigma_{m,j} + R(x)\Sigma_{s,i}R(x)^T, \end{aligned} \tag{4}$$

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 views. We compensate for inaccuracies due to discretization effects by trilinear interpolation between target surfels.

4.3 Pose Optimization

We optimize the observation log likelihood

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

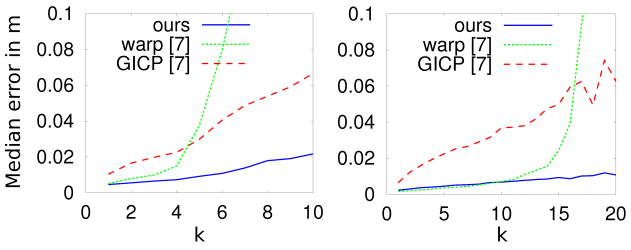

Figure 3: Median translational error of the pose estimate for different frame skips k on the freiburg1_desk (left) and freiburg2_desk (right) dataset.

for the pose x using the Newton-Raphson method. In each step, we determine new surfel associations in the current pose estimate. Our method typically converges within 5-10 iterations to a precise estimate.

5 Experiments

We evaluate our approach on a public RGB-D dataset [10]. 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 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 approach (abbreviated by warp) in [7].

Accuracy and Robustness: 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 [7] (5.3 mm and 1.5 mm), while our approach also performs significantly better than GICP (10.3 mm and 6.3 mm [7]). However, when skipping frames (see Fig. 3), our approach achieves similar accuracy than [7] for small displacements, but retains the robustness of ICP methods for larger displacements when [7] fails.

Efficiency: The mean processing time on the freiburg2_desk dataset is 100,11 msec (ca. 10 Hz) on an Intel Xeon 5650 2,67 GHz CPU using 640x480 VGA images. Since our approach is robust for the skipping of more than 5 frames, we consider our approach real-time capable.

Public Demonstration: We demonstrated our real-time registration method publicly in the RoboCup 2011 @Home league finale. Our robot Cosero carried a table with a human and baked omelet. For carrying the table, we trained a model map of the table [8]. The robot registered RGB-D images to the model in real-time to approach the table and grasp it at predefined grasp points. It detected the lifting and lowering of the table by estimating its pitch rotation.

Dataset	ours	warp [7]	GICP [7]
freiburg1_desk	4.62 mm	5.3 mm	10.3 mm
	0.0092 deg	0.0065 deg	0.0154 deg
freiburg2_desk	2.27 mm	1.5 mm	6.3 mm
	0.0041 deg	0.0027 deg	0.0060 deg

Table 1: Comparison of median pose drift between frames.

Similarly, the robot approached the pan on a cooking plate by tracking the object with our registration method. The demonstration has been well received by the jury. Paired with the highest score from the previous stages, we could win the RoboCup@Home competition 2011.

6 Conclusion

We proposed a novel robust method for real-time registration of RGB-D images. We present efficient means to extract multi-resolution surfel views from RGB-D images. These views are then registered with a robust and efficient variant of the ICP algorithm. Our approach utilizes multiple resolutions to align the views on coarse scales and to register them accurately on fine resolutions.

In experiments, we demonstrate the registration of images at frame-rates of ca. 10 Hz. Our approach yields comparable accuracy to a state-of-the-art algorithm. It yields superior robustness for large differences in view pose.

In future work, we will apply our approach for object reconstruction and 6-DoF SLAM.

References

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

[2] B. Huhle, M. Magnusson, W. Strasser, and A. J. Lilienthal. Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.

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

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

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

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

[7] 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 the Int. Conf. on Computer Vision (ICCV)*, 2011.

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

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

[10] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart. 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)*, 2011.

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