MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Pinpoint SLAM: A Hybrid of 2D and 3D Simultaneous Localization and Mapping for RGB-D Sensors

Ataer-Cansizoglu, E.; Taguchi, Y.; Ramalingam, S.

TR2016-035 May 2016

Abstract

Conventional SLAM systems with an RGB-D sensor use depth measurements only in a limited depth range due to hardware limitation and noise of the sensor, ignoring regions that are too far or too close from the sensor. Such systems introduce registration errors especially in scenes with large depth variations. In this paper, we present a novel RGB-D SLAM system that makes use of both 2D and 3D measurements. Our system first extracts keypoints from RGB images and generates 2D and 3D point features from the keypoints with invalid and valid depth values, respectively. It then establishes 3D-to-3D, 2D-to-3D, and 2D-to-2D point correspondences among frames. For the 2D-to-3D point correspondences, we use the rays defined by the 2D point features to "pinpoint" the corresponding 3D point features, generating longer-range constraints than using only 3D-to-3D correspondences. For the 2D-to-2D point correspondences, we triangulate the rays to generate 3D points that are used as 3D point features in the subsequent process. We use the hybrid correspondences in both online SLAM and offline postprocessing: the online SLAM focuses more on the speed by computing correspondences among consecutive frames for real-time operations, while the offline postprocessing generates more correspondences among all the frames for higher accuracy. The results on RGB-D SLAM benchmarks show that the online SLAM provides higher accuracy than conventional SLAM systems, while the postprocessing further improves the accuracy.

IEEE International Conference on Robotics and Automation (ICRA)

Pinpoint SLAM: A Hybrid of 2D and 3D Simultaneous Localization and Mapping for RGB-D Sensors

Esra Ataer-Cansizoglu, Yuichi Taguchi, and Srikumar Ramalingam

***Abstract*— Conventional SLAM systems with an RGB-D sensor use depth measurements only in a limited depth range due to hardware limitation and noise of the sensor, ignoring regions that are too far or too close from the sensor. Such systems introduce registration errors especially in scenes with large depth variations. In this paper, we present a novel RGB-D SLAM system that makes use of both 2D and 3D measurements. Our system first extracts keypoints from RGB images and generates 2D and 3D point features from the keypoints with invalid and valid depth values, respectively. It then establishes 3D-to-3D, 2D-to-3D, and 2D-to-2D point correspondences among frames. For the 2D-to-3D point correspondences, we use the rays defined by the 2D point features to "pinpoint" the corresponding 3D point features, generating longer-range constraints than using only 3D-to-3D correspondences. For the 2D-to-2D point correspondences, we triangulate the rays to generate 3D points that are used as 3D point features in the subsequent process. We use the hybrid correspondences in both online SLAM and offline postprocessing: the online SLAM focuses more on the speed by computing correspondences among consecutive frames for real-time operations, while the offline postprocessing generates more correspondences among all the frames for higher accuracy. The results on RGB-D SLAM benchmarks show that the online SLAM provides higher accuracy than conventional SLAM systems, while the postprocessing further improves the accuracy.**

I. Introduction

Simultaneous localization and mapping (SLAM) is an important and well-studied problem with many applications in computer vision, robotics, and augmented reality. SLAM using monocular cameras has long been a focus of research. Recently, with the rise of low-cost 3D sensors, many SLAM systems make use of both color and depth data using RGB-D sensors such as Kinect. Although RGB-D sensors provide increased registration accuracy and robustness, they typically provide depth measurements only in a limited depth range (e.g., 0.5 m to 4 m for Kinect) due to the hardware limitations and the noise. Most RGB-D SLAM systems use only the pixels that have valid depth measurements, ignoring the pixels that are too close or too far from the sensor. This yields ineffective use of information provided by sensors and might introduce registration inaccuracy especially for scenes with large depth variations.

In this paper, we introduce pinpoint SLAM, a hybrid of 2D and 3D SLAM systems. Figure 1 illustrates the key idea proposed in this paper. Conventional RGB-D SLAM systems use only 3D-to-3D correspondences between 3D measurements extracted from frames, which generate only short-range constraints between the frames due to the limited depth measurement range. On the other hand, our system extracts both 2D and 3D measurements from frames, where keypoints with valid depth values become 3D measurements and keypoints without valid depth values are treated as 2D measurements. Then it establishes 3D-to-3D, 2D-to-3D, and 2D-to-2D point correspondences. The additional correspondences including 2D measurements provide long-range constraints between the frames: we "pinpoint" 3D points with rays defined by 2D points.

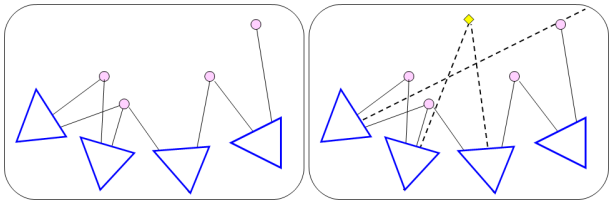

Fig. 1. Illustration of the key "pinpoint" idea proposed in this paper. Dashed lines represent 2D measurements, while circles and solid lines represent 3D measurements. Triangulated 3D points are shown with diamonds. Conventional SLAM systems (left) use only 3D-to-3D correspondences, resulting in short-range constraints between frames due to the limited depth measurement range. Our system (right) additionally uses 2D-to-3D and 2D-to-2D point correspondences to generate long-range constraints. The rays defined by 2D points (keypoints with invalid depth values) "pinpoint" the 3D points (keypoints with valid depth values, or 3D points triangulated by intersecting the rays).

We use the hybrid correspondences in both online SLAM and offline postprocessing. For online SLAM, we limit the correspondence search among neighboring frames for real-time processing. This approach still generates longer-range constraints than conventional approaches using only 3D-to-3D correspondences, but does not exploit correspondences that might exist between distant frames. Our postprocessing establishes such correspondences by iteratively finding the hybrid correspondences between each frame and the rest of the frames. In each iteration, the correspondences are first updated based on the current poses of the frames, which are then used to update the poses in a bundle adjustment procedure.

The experiments are carried out on indoor sequences and two RGB-D benchmark datasets: TUM [1] and ICL-NUIM [2] benchmarks. We compare the performance of SLAM systems using only 3D-to-3D correspondences and using the hybrid correspondences. In addition, we apply the postprocessing to the output of the pinpoint SLAM. The results show that the pinpoint SLAM provides better performance than conventional SLAM systems, and the postprocessing improves the accuracy even more.

Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139, USA
{cansizoglu,taguchi,ramalingam}@merl.com

A. Contributions

The contributions of this paper are summarized as follows.

- We present a method for hybrid use of 2D and 3D measurements to register multiple RGB-D frames.
- We apply our method to both online SLAM and offline postprocessing.
- We show that our system provides higher accuracy than conventional RGB-D SLAM systems that use only 3D-to-3D correspondences by using standard RGB-D SLAM benchmarks.

B. Related Work

Typical SLAM systems use a single primitive (e.g., 2D points, 3D points) as the measurements. For example, feature-based monocular SLAM systems [3], [4] extract 2D point features, use 2D-to-2D point correspondences to initialize 3D point landmarks by triangulation, and then use 2D-to-3D correspondences between the 2D point measurements and the 3D point landmarks to estimate the camera pose in consecutive frames. This is also a common pipeline for structure from motion systems [5]. On the other hand, feature-based RGB-D SLAM systems [6], [7] extract 3D point features and estimate the camera pose using 3D-to-3D point correspondences. Plane features are also used as measurements in some SLAM systems [8], [9]. Recent dense SLAM systems, both monocular [10], [11] and 3D/RGB-D [12], [13], [14], [15], [16], do not rely on feature extraction but rather exploit all the 2D or 3D points in frames and minimize photometric errors or iterative closest point (ICP) costs for direct registration. Those systems still use a single primitive, either 2D points or 3D points.

Some SLAM systems use a hybrid of 3D measurements. Trevor et al. [17] used both plane-to-plane and line-to-plane correspondences. Taguchi et al. [18] used both point-to-point and plane-to-plane correspondences. However, all the measurements used in their systems are 3D primitives.

Our system is most closely related to [19], [20], which addressed the same problem of the lack of depth measurements in some regions of RGB-D images and used both 2D and 3D measurements. Hu et al. [19] heuristically switched between 2D-to-2D and 3D-to-3D correspondences according to the available depth measurements, and thus they did not use both correspondences together. In contrast, our system uses both correspondences in a single registration framework. Zhang et al. [20] used both 2D and 3D measurements to register two frames for visual odometry, but the 3D measurements were assumed only in one of the two frames and thus 2D-to-3D correspondences were used. On the other hand, our system exploits 3D measurements as well as 2D measurements in all the frames. We also propose a postprocessing system for finding long-range constraints between all the frames and generating globally consistent 3D models.

II. Pinpoint SLAM and Postprocessing

This section describes our registration method that makes use of both 2D and 3D measurements. Our method can be incorporated into any point-feature-based RGB-D SLAM systems, but in this paper, we build our system on the point-plane SLAM system [18]. The system uses both 3D points and 3D planes as primitives and is a keyframe-based SLAM system, which stores representative frames in a map.

An overview of our method can be seen in Figure 2. We use the method for both online SLAM and offline postprocessing. In the online SLAM, we process each input frame once and register it to the map consisting of previous keyframes. On the other hand, in the offline postprocessing, we process each keyframe in the map to re-register it with the rest of keyframes and iterate the process as long as the poses of the keyframes are updated. In both cases, the method consists mainly of five steps: (i) feature extraction, (ii) correspondence search, (iii) RANSAC registration, (iv) map update, and (v) bundle adjustment. In the following subsections, we first describe our notations and then detail each of the above steps.

A. Notations

We use the standard terminology of measurements and landmarks: the system extracts measurements from each RGB-D frame and generates landmarks in a global map.

A 3D point measurement is represented by $(\mathbf{p}_m, \mathbf{D}_m)$, where $\mathbf{p}_m \in \mathbb{R}^3$ denotes its 3D position and $\mathbf{D}_m$ denotes its descriptor. A 2D point measurement is denoted by $(\mathbf{q}_m, \nu_m, \mathbf{D}_m)$, where $\mathbf{q}_m = (q_x, q_y) \in \mathbb{R}^2$ is the pixel coordinate, $\mathbf{D}_m$ is its descriptor, and $\nu_m = (\mathbf{c}_m, \mathbf{u}_m)$ represents the ray passing through the camera center and the 2D point measurement such that $\{\mathbf{x} | \mathbf{x} = \mathbf{c}_m + t\mathbf{u}_m, t \in \mathbb{R}\}$, $\mathbf{c}_m = [0,0,0]^T$ and $\mathbf{u}_m = [(q_x - c_x)/f_x, (q_y - c_y)/f_y, 1]^T$ based on the camera intrinsic parameters: the focal lengths (f_x, f_y) and the principal point (c_x, c_y). A 3D plane measurement is represented by (π_m, I_m), denoting plane parameters and the set of 3D inlier points associated to the plane.

A landmark is a collection of measurements. A 3D point landmark is represented by $(\mathbf{p}_l, D_l)$, where $\mathbf{p}_l \in \mathbb{R}^3$ denotes its 3D position and D_l denotes the set of descriptors associated to this landmark. A 2D point landmark is $(\mathbf{q}_l, v_l, \mathbf{D}_l)$, where $\mathbf{q}_l \in \mathbb{R}^2$ is the pixel coordinate, $\mathbf{D}_l$ is its descriptor, and v_l is the line passing through the camera center and the associated 2D point measurement in the map coordinate system. Note that a 2D point landmark is associated to only a single 2D point measurement; as soon as it matches with another 2D/3D point measurement, it is converted to a 3D point landmark. A 3D plane landmark is denoted by (π_l, I_l) with plane parameters π_l and the set of 3D inlier points from all the associated frames I_l.

B. Feature Extraction

Our system extracts 2D keypoints from each RGB image using SURF features [21]. If the corresponding depth value is within a predefined range, then it is considered valid. The keypoints with valid depth values are back-projected and used as 3D point measurements. Keypoints with invalid depth values are considered 2D point measurements and represented as the rays passing through the camera center and the 2D keypoints. 3D plane measurements are extracted

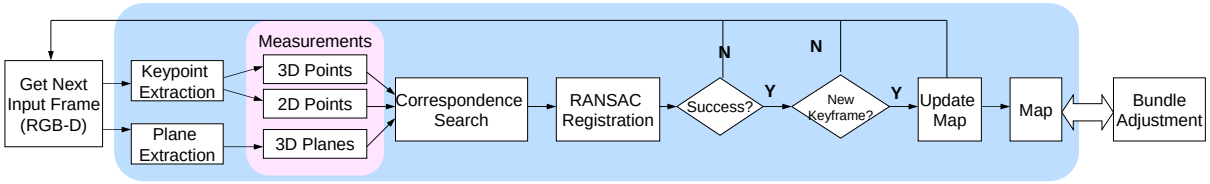

Fig. 2. Overview of our method. For each RGB-D frame, we first extract keypoints and planes. The keypoints with and without valid depth measurements generate 3D and 2D point measurements, respectively. We then perform keypoint matching between the frame and the map to obtain point correspondences, and consider all possible plane correspondences between the frame and the map. Next, we perform RANSAC registration using all correspondences: 3D-to-3D point, 2D-to-3D point, 2D-to-2D point, and 3D-to-3D plane correspondences. If it succeeds, the map is updated with the features and correspondences found in the frame. We run bundle adjustment and loop closing asynchronously to optimize 3D point landmarks, 3D plane landmarks, and keyframe poses.

using [22]. Note that this step is omitted in the postprocessing, since it is applied to an existing map where the features are already extracted.

C. Correspondence Search

After feature extraction, we look for the correspondences between the keypoints extracted from the current frame and the keypoints in the map. Plane correspondences are not searched since the number of planes is fairly small compared to the number of points; instead our RANSAC registration considers all possible plane correspondences.

Online SLAM: We perform all-to-all descriptor matching between the keypoints of the current frame and the keypoints of the last k keyframes of the map. Considering k keyframes instead of all keypoints of the map helps us to improve speed and to narrow down our search since the last keyframes more likely observe the same region as the current frame assuming a continuous camera motion. The matching returns three types of point correspondences, 3D-to-3D, 3D-to-2D, and 2D-to-2D point correspondences, each of which are specially considered in the RANSAC registration phase. 3D-to-2D correspondences exhibit two cases: from 3D landmark to 2D measurement or from 2D landmark to 3D measurement.

Offline Postprocessing: Instead of using the all-to-all descriptor matches, we restrict the candidate matches by using the current pose of the keyframe as a predicted pose. We project all 3D point landmarks of the map to the frame based on the predicted pose. A point measurement in the frame is considered a candidate match with a 3D point landmark, if its projected point falls within a neighborhood of r pixels. This will generate either 3D-to-3D correspondences or correspondences from 3D landmark to 2D measurement. For 3D-to-2D correspondences from 2D landmark to 3D measurement, the search is done in a similar way with a change in the direction of the projection (i.e., the 3D point measurement of the frame is projected to the keyframe that initiates the 2D point landmark). In terms of 2D-to-2D correspondences, we test the distance to the epipolar line to be less than r pixels in order to match a 2D point measurement with a 2D point landmark.

D. RANSAC Registration

As in [18], our RANSAC registration procedure tries different types of hypotheses in the order of (i) three planes, (ii) two planes + one 3D point, (iii) one plane + two 3D points, and (iv) three 3D points. Since we also use 2D point measurements, we add a last hypothesis to this list, which considers three 2D-to-3D correspondences. We apply the P3P algorithm [23] to find the registration parameters for this case. Note that in addition to the 2D-to-3D correspondences, we can treat a 3D-to-3D correspondence as a 2D-to-3D correspondence by ignoring the depth of one of the 3D points.

RANSAC inlier check is carried out based on the type of the correspondence. A 3D-to-3D correspondence is considered an inlier if the distance between the two 3D points is below a threshold. For a 2D-to-3D correspondence, we consider the distance between the 3D point and the line corresponding to the 2D point. The distance between a 3D point landmark $\mathbf{p}_l$ and a 2D point measurement $\mathbf{q}_m$ is computed as $\mathfrak{d}(\mathbf{p}_l, \mathbf{q}_m)$

$$t^* = \frac{< \mathbf{u}_m, \mathbf{T}^{-1}(\mathbf{p}_l) - \mathbf{c}_m >}{< \mathbf{u}_m, \mathbf{u}_m >} \quad (1)$$

$$\mathfrak{d}(\mathbf{p}_l, \mathbf{q}_m) = ||\mathbf{p}_l - \mathbf{T}(\mathbf{c}_m + t^* \mathbf{u}_m)|| \quad (2)$$

where $\mathbf{T}$ is the pose of the keyframe that contains the 2D point measurement, $\mathbf{T}(\cdot)$ denotes application of transformation $\mathbf{T}$ to the point, and $< \cdot, \cdot >$ denotes the dot product. If the distance is below a threshold, then the correspondence is counted as an inlier. For a 2D-to-2D correspondence, we check the distance of the pixel to the corresponding epipolar line of the other point to be less then a threshold.

E. Map Update

If the RANSAC registration succeeds, the registration result is used to update the map.

Online SLAM: A frame is added to the map as a keyframe if its pose is different from the poses of already existing keyframes in the map. If a 2D point measurement is matched with a 3D point landmark, then the set of descriptors of the landmark is enlarged by adding the descriptor of the 2D point measurement. For the case of a 2D-to-2D match, we perform triangulation by finding the middle of the closest points on the two lines, and add it to the map as a 3D point landmark by collecting the descriptors of the 2D measurements. We ignore triangulation for 2D-to-2D matches with a small camera baseline as it introduces noise. If a 3D point measurement is matched with a 2D point landmark, the landmark is updated as a 3D point landmark via the 3D coordinates of the

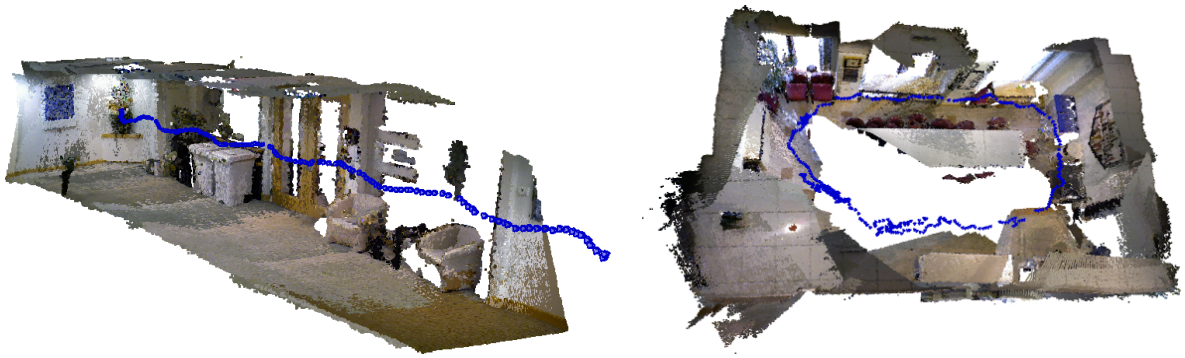

Fig. 3. 3D reconstruction results and camera trajectories for the `lounge` (left) and `lunch room` (right) sequences.

3D point measurement transformed to the map coordinate system. All the unmatched 2D/3D point measurements are added to the map as 2D/3D point landmarks.

Offline Postprocessing: Since the postprocessing is applied to an existing map, all measurements are already associated to the landmarks. Hence, this stage only updates the correspondences between the keyframe and the map. RANSAC inliers are used to refine the correspondences of the keyframe. If a measurement is matched with a different landmark than what it is currently associated to, then it is split from the current landmark and associated to the new inlier landmark. Similarly, if a measurement is not matched with its current landmark, then it is split from that landmark, creating a new landmark on its own.

F. Bundle Adjustment

Let the triplet (k,l,m) denote an association such that a 3D point landmark $\mathbf{p}_l (l = 1,\ldots,L)$ is associated to a 3D point measurement $\mathbf{p}_m^k$ or a 2D point measurement $\mathbf{q}_m^k$ in the keyframe k with the pose $\mathbf{T}_k (k = 1,\ldots,K)$. Similarly, let the triplet (k,l',m') denote an association such that a 3D plane landmark $\pi_{l'} (l' = 1,\ldots,L')$ is associated to a 3D plane measurement $\pi_{m'}^k$ in the keyframe k. Let A_1, A_2, and A_3 contain all the triplets representing the 3D-to-3D point associations, 2D-to-3D point associations, and 3D-to-3D plane associations in the map, respectively. Then the bundle adjustment aims to minimize the following error function with respect to 3D point landmark coordinates, 3D plane landmark parameters, and keyframe poses:

$$E(\mathbf{p}_1,\ldots,\mathbf{p}_L;\pi_1,\ldots,\pi_{L'};\mathbf{T}_1,\ldots,\mathbf{T}_K) = \sum_{(k,l,m)\in A_1} \left\| \mathbf{p}_l - \mathbf{T}_k(\mathbf{p}_m^k) \right\| + \sum_{(k,l,m)\in A_2} \mathfrak{d}(\mathbf{p}_l,\mathbf{q}_m^k) + \sum_{(k,l',m')\in A_3} \sum_a d(\pi_{l'},\mathbf{T}_k(\mathbf{p}_{m',a}^k)). \quad (3)$$

Here $d(\pi_{l'},\mathbf{T}_k\mathbf{p}_{m',a}^k)$ is the distance between the plane landmark $\pi_{l'}$ and a 3D point $\mathbf{p}_{m',a}^k$, which is sampled from the set of inlier points of the plane measurement $(\pi_{m'},I_{m'})$ in the keyframe k.

Online SLAM: The bundle adjustment is performed asynchronously in a separate thread. Our system also performs loop closing in another thread. It first checks the appearance similarity of all pairs of keyframes in the map using the VLAD descriptors [24] computed based on the keypoints. It also checks the pose similarity between the pairs of keyframes (we do not try to close the loop if the current poses of the keyframes are too different). For the pairs of keyframes that pass the similarity tests, our system then performs the RANSAC registration using the hybrid correspondences, and if the RANSAC succeeds, the inlier correspondences are used to update the associations in the map.

Offline Postprocessing: The postprocessing is performed iteratively to update the associations and refine the landmark parameters and keyframe poses. In an iteration of the postprocessing, every keyframe is re-registered with the map including the rest of the keyframes, and its correspondences are updated (i.e., splits and merges of the landmarks are done if necessary). After all correspondences are updated, we run bundle adjustment to refine the landmark parameters and keyframe poses. We repeat the iteration if the average change in the keyframe poses is above a threshold.

III. Experiments and Results

We implemented our system using C++ and built our platform on a Surface Pro 2 tablet with an Asus Xtion sensor. We used the Ceres Solver [25] for the bundle adjustment. The online SLAM system runs about 2 frames per second on the tablet.

We show two sets of experiments where we compare three outputs: (i) SLAM without the use of 2D point measurements which we call 3D SLAM [18], (ii) pinpoint SLAM, and (iii) postprocessing on the pinpoint SLAM result. First, we show qualitative results on two indoor sequences having large depth variations, which were captured with our platform. Second, we report quantitative results on the improved registration accuracy on TUM [1] and ICL-NUIM [2] benchmark datasets.

A. Qualitative Results

We captured two indoor sequences from scenes with large depth variation. 3D reconstruction results along with the camera trajectories after postprocessing are displayed in Figure 3. Both sequences include several regions captured at different distances. Figures 4 and 5 show visual comparisons, where the keyframes are overlayed on the reconstructed

Fig. 4. Visual results on three example keyframes of the `lounge` sequence. Frames are overlayed on 3D reconstruction results with some transparency for better visualization. Rows show original image, 3D SLAM, pinpoint SLAM and postprocessing results from top to bottom. 3D SLAM registers the nearby regions well, but has trouble matching further points (notice the area around the frame on the wall). Pinpoint SLAM and postprocessing improve the results, producing almost perfect alignment.

3D models rendered using the estimated camera poses. 3D SLAM registers nearby regions well, but it cannot match far away points since it does not use any 2D point measurements. Pinpoint SLAM improves the registration accuracy, while postprocessing produces almost perfect alignment at both nearby and distant regions. Please also refer to the supplementary video for better visualization.

Figure 6 shows color map representations of the number of correspondences between keyframe pairs for 3D SLAM, pinpoint SLAM and postprocessing results. 3D SLAM mainly produces large numbers around the diagonal for consecutive frames as well as some off-diagonal entries due to loop closing. As can be seen the number of nonzero entries increases with pinpoint SLAM while postprocessing further improves the interaction between distant keyframes. Table I also verifies this fact, reporting the total number of nonzero entries in the matrices.

TABLE I

NUMBER OF NONZERO ENTRIES IN THE MATRICES SHOWN IN FIGURE 6.

	3D SLAM	Pinpoint	Postprocessing
`lounge`	2581	2865	6846
`lunch room`	11644	17884	27087

B. Quantitative Results Using Benchmark Datasets

We ran 3D SLAM [18] (i.e., without the use of 2D point measurements) and pinpoint SLAM on benchmark datasets. We then applied postprocessing to the output of the online pinpoint SLAM in order to evaluate the improvement. Note that our system does not add all the frames to the map. Hence the evaluation measures are only computed for the keyframes that are added to the map. Also, for some of the sequences our system lost the track and could not relocalize back. Therefore, we report the longest sequence that we were able to process on those datasets. The results are reported using two evaluation measures: root mean square (RMS) of absolute trajectory error (ATE) and RMS of relative positioning error (RPE).

TUM benchmark contains sequences of various scenes. We applied our method on some selected sequences with small and large depth variations. The results are displayed in Table II. The last five rows of the table list the sequences with large depth variations. Pinpoint SLAM performs similar

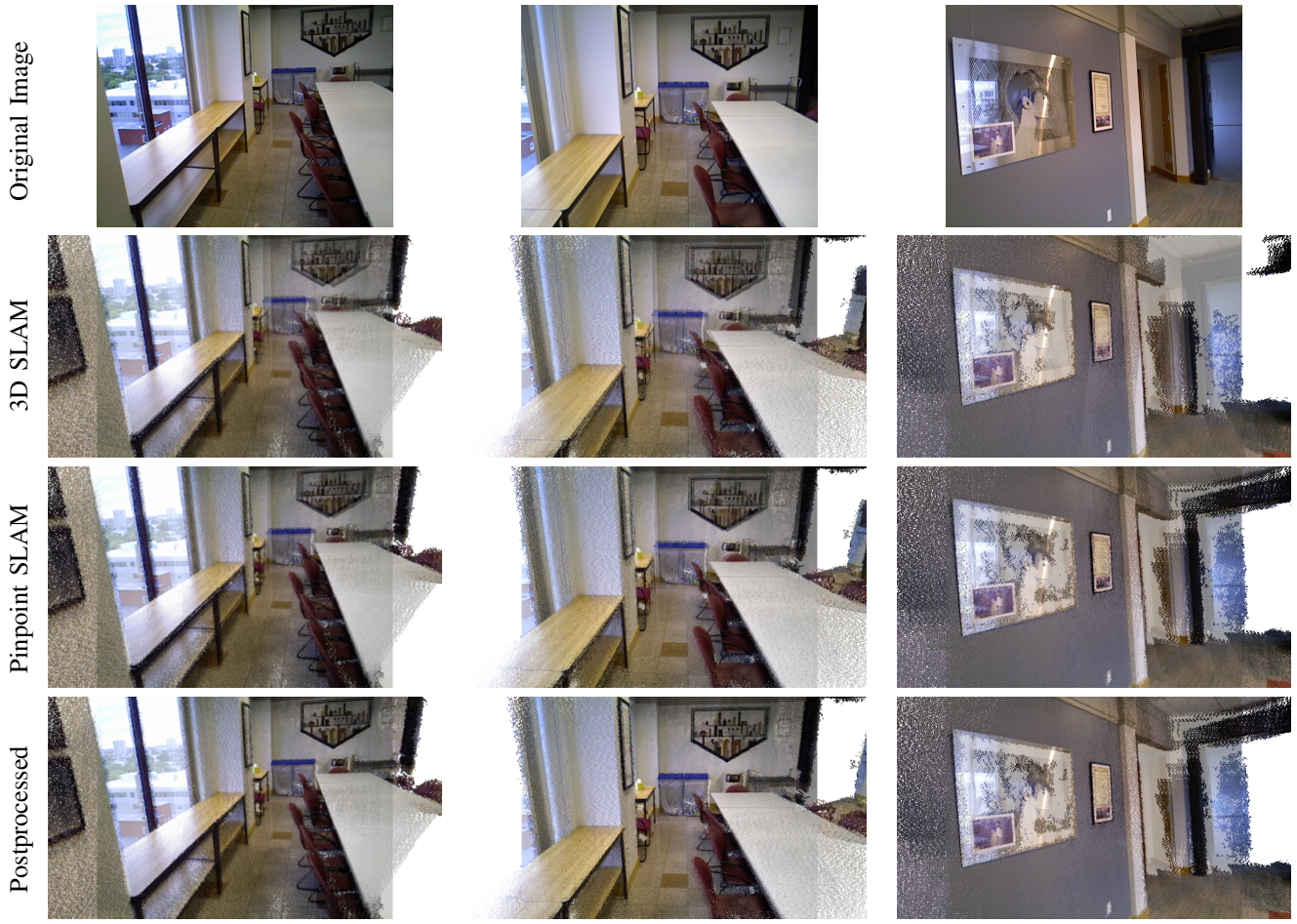

Fig. 5. Visual results on three example keyframes of the `lunch room` sequence. Frames are overlayed on 3D reconstruction with some transparency for better visualization. Rows show original image, 3D SLAM, pinpoint SLAM and postprocessing results from top to bottom. 3D SLAM registers the nearby regions well, but has trouble matching further points. Pinpoint SLAM and postprocessing improve the results, producing almost perfect alignment.

or better in all sequences than 3D SLAM, while postprocessing improves the accuracy in all of them. Moreover, the improvement of our method is larger for the sequences with large depth variations. The last column of the table reports the best performance achieved in the literature. Although our results are reported on the longest sequence that can be processed by online pinpoint SLAM, the results show that the performance of the proposed method is either comparable or better than the state-of-the-art SLAM methods. Figure 7 shows the output trajectories and errors per position for some example sequences. As can be seen, 3D SLAM might have large drifts since it does not use 2D measurements. On the other hand, pinpoint SLAM has smaller drifts and postprocessing refines the trajectory even more. Note also that the improvement is less visible on `fr2/desk` sequence which does not have large depth variations.

ICL-NUIM datasets are generated from relatively smaller regions with small depth variations. Pinpoint SLAM and postprocessing improve the accuracy on these sequences as well, but the difference is less visible compared to the large depth variation sequences of the TUM benchmark. We also report the best results achieved on these sequences at the last column of the table as RMS of ATE.

IV. Conclusion and Discussion

We presented a novel SLAM system with the "pinpoint" approach for effectively using all data obtained with an RGB-D sensor. Our system is a hybrid of 2D and 3D SLAM. The 2D measurements are represented as rays passing through the camera center and the 2D points. Corresponding 3D points are pinned to these rays, generating improved interaction between frames. Two matching 2D measurements are triangulated and added to the map. The better correspondences between frames yield improvement in the registration accuracy. Furthermore, we use the same approach for an offline postprocessing procedure that allows even more refinement of the results. The results on publicly available RGB-D benchmarks show that, for scenes with large depth variations, pinpoint SLAM and postprocessing provide higher accuracy than a conventional 3D SLAM system.

Acknowledgements: We thank Jay Thornton and Chen Feng for their helpful comments and feedback.

References

[1] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 573–580.

TABLE II

QUANTITATIVE EVALUATION RESULTS FOR SOME SEQUENCES OF TUM BENCHMARK [1] REPORTED IN ROOT MEAN SQUARE (RMS) OF ABSOLUTE TRAJECTORY ERROR (ATE) AND RMS OF RELATIVE POSITIONING ERROR (RPE). SEQUENCES WITH LARGE DEPTH VARIATIONS ARE SHOWN WITH A † ON THE NAME. P SLAM STANDS FOR PINPOINT SLAM. RESULT WITH A * SIGN IS MEDIAN OF ATE, WHILE THE REST IS RMSE OF ATE.

Sequence	# Keyframes		RMS of ATE			RMS of RPE			Best RMS of ATE
	3D SLAM	P SLAM	3D SLAM	P SLAM	Postprocessed	3D SLAM	P SLAM	Postprocessed	
fr1/xyz	66	65	16 mm	15 mm	11 mm	24 mm, 1.0°	23 mm, 0.9°	18 mm, 0.8°	**9 mm** [26]
fr2/xyz	102	103	12 mm	12 mm	12 mm	18 mm, 0.9°	19 mm, 0.9°	18 mm, 0.9°	**2 mm** [4]
fr1/floor	167	167	61 mm	61 mm	38 mm	136 mm, 3.2°	142 mm, 3.3°	76 mm, 2.8°	**29.9 mm** [26]
fr2/desk	358	359	61 mm	63 mm	56 mm	110 mm, 2.9°	112 mm, 2.9°	100 mm, 2.6°	**17 mm** [27]
fr3/structure_texture_far	108	106	27 mm	26 mm	25 mm	54 mm, 1.5°	52 mm, 1.4°	48 mm, 1.3°	**24 mm** [28]
fr3/long_office_household	244	245	29 mm	26 mm	23 mm	62 mm, 1.6°	58 mm, 1.6°	48 mm, 1.3°	**7.7 mm** [26]
fr2/large_no_loop†	99	99	165 mm	162 mm	**148 mm**	261 mm, 3.9°	254 mm, 3.7°	244 mm, 3.5°	187 mm* [29]
fr2/large_with_loop†	146	134	191 mm	118 mm	**86 mm**	303 mm, 4.6°	189 mm, 3.2°	155 mm, 2.4°	-
fr2/pioneer_slam†	79	83	30 mm	26 mm	**21mm**	47 mm, 1.2°	41 mm, 1.0°	33 mm, 1.0°	94 mm [30]
fr2/pioneer_slam2†	131	135	148 mm	146 mm	**107 mm**	268 mm, 4.7°	261 mm, 4.1°	205 mm, 2.6°	306 mm [30]
fr2/pioneer_slam3†	218	255	396 mm	132 mm	**88 mm**	1433 mm, 28.1°	356 mm, 8.0°	254 mm, 5.7°	111 mm [30]

TABLE III

QUANTITATIVE EVALUATION RESULTS FOR SOME SEQUENCES OF ICL-NUIM BENCHMARK [2] REPORTED IN RMS OF ATE AND RMS OF RPE. P SLAM STANDS FOR PINPOINT SLAM

Sequence	# Keyframes		RMS of ATE			RMS of RPE			Best RMS of ATE
	3D SLAM	P SLAM	3D SLAM	P SLAM	Postprocessed	3D SLAM	P SLAM	Postprocessed	
office traj 0	116	115	5.1 mm	3.6 mm	3.6 mm	8.0 mm, 0.1°	5.8 mm, 0.1°	5.5 mm, 0.1°	**2.9 mm** [2]
office traj 1	78	78	**1.9 mm**	**1.9 mm**	2.0 mm	4.0 mm, 0.2°	3.9 mm, 0.2°	3.6 mm, 0.1°	38.5 mm [2]
office traj 2	163	163	5.3 mm	6.1 mm	4.6 mm	7.4 mm, 0.1°	8.4 mm, 0.1°	6.8 mm, 0.1°	**1.6 mm** [2]
office traj 3	146	146	3.4 mm	3.4 mm	2.7 mm	5.2 mm, 0.1°	5.1 mm, 0.1°	4.0 mm, 0.1°	**2.1 mm** [2]
living room traj 0	72	72	4.1 mm	4.1 mm	**2.9 mm**	6.7 mm, 0.2°	6.7 mm, 0.2°	4.8 mm, 0.2°	113.8 mm [2]
living room traj1	80	80	19.5 mm	19.3 mm	17.7 mm	47.3 mm, 0.9°	46.5 mm, 0.9°	39.1 mm, 0.8°	**2.3 mm** [2]
living room traj2	158	159	9.9 mm	8.5 mm	6.9 mm	14.1 mm, 0.3°	12.6 mm, 0.3°	10.7 mm, 0.2°	**1.5 mm** [2]
living room traj 3	81	83	27.5 mm	16.0 mm	**14.7 mm**	73.9 mm, 9.6°	34.5 mm, 4.2°	35.5 mm, 4.5°	20.0 mm [2]

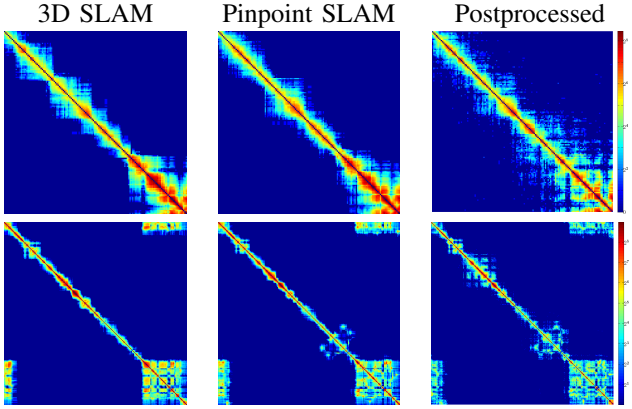

Fig. 6. Color map representations of the number of correspondences between keyframe pairs for the `lounge` (top) and `lunch room` (bottom) sequences. X and Y axes refer to the indices of keyframes in the sequence and the color indicates the number of correspondences (each sequence uses the same color scaling shown on the right).

[2] A. Handa, T. Whelan, J. B. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2014.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[4] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE Int'l Symp. Mixed and Augmented Reality (ISMAR)*, Nov. 2007, pp. 1–10.

[5] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 835–846, July 2006.

[6] 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. Int'l Symp. Experimental Robotics (ISER)*, Dec. 2010.

[7] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proc. Int'l Symp. Robotics Research (ISRR)*, Aug. 2011.

[8] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 3062–3067.

[9] K. Pathak, A. Birk, N. Vaškevičius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-D mapping," *IEEE Trans. Robotics*, vol. 26, no. 3, pp. 424–441, June 2010.

[10] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, Nov. 2011, pp. 2320–2327.

[11] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. European Conf. Computer Vision (ECCV)*, Sept. 2014.

[12] 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. IEEE Int'l Symp. Mixed and Augmented Reality (ISMAR)*, Oct. 2011, pp. 127–136.

[13] J. Chen, D. Bautembach, and S. Izadi, "Scalable real-time volumetric surface reconstruction," *ACM Trans. Graphics*, vol. 32, no. 4, pp. 113:1–113:16, July 2013.

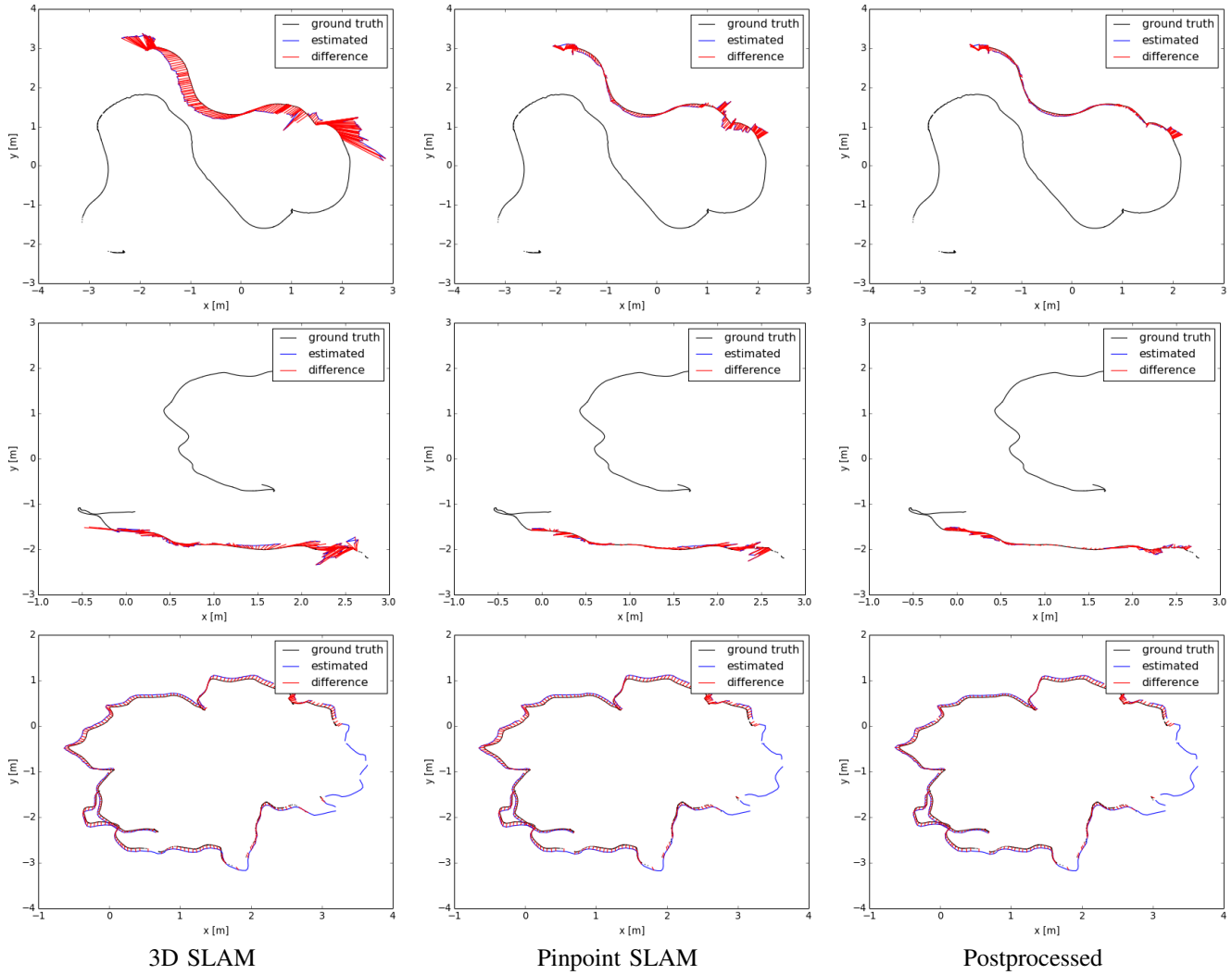

Fig. 7. Resulting trajectories and error per position for `fr2/pioneer_slam3` (top), `fr2/large_with_loop` (middle) and `fr2/desk` (bottom) sequences. The first two sequences contain large depth variations, while the bottom one has small depth variations.

[14] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5724–5731.

[15] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 2100–2106.

[16] R. F. Salas-Moreno, B. Glocker, P. H. J. Kelly, and A. J. Davison, "Dense planar SLAM," in *Proc. IEEE Int'l Symp. Mixed and Augmented Reality (ISMAR)*, Sept. 2014.

[17] A. J. B. Trevor, J. G. Rogers III, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2012, pp. 3041–3048.

[18] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5182–5189.

[19] G. Hu, S. Huang, L. Zhao, A. Alempijevic, and G. Dissanayake, "A robust RGB-D SLAM algorithm," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012.

[20] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Sept. 2014.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[22] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2014.

[23] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int'l J. Computer Vision*, vol. 13, no. 3, pp. 331–356, Dec. 1994.

[24] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sept. 2012.

[25] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[26] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *arXiv preprint arXiv:1502.00956*, 2015.

[27] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2013, pp. 3748–3754.

[28] N. Fioraio and L. D. Stefano, "SlamDunk: Affordable real-time RGB-D SLAM," in *Proc. European Conf. Computer Vision (ECCV)*, 2014, pp. 401–414.

[29] M. Meilland and A. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, 2013, pp. 3677–3683.

[30] J. M. Hess, "Efficient approaches to cleaning with mobile robots," Ph.D. dissertation, University of Freiburg, 2015.