Orthogonal 3D-SLAM for indoor environments using right angle corners

Orthogonal 3D-SLAM for Indoor Environments Using Right Angle Corners

Ahad Harati Roland Siegwart
Autonomous Systems Laboratory
Swiss Federal Institute of Technology (ETH Zürich)
Tannenstrasse 3, CH-8092 Zürich, Switzerland
Email: {ahad.harati, roland.siegwart}@mavf.ethz.ch

***Abstract*— Soon, in many service robotic applications, a real-time localization and 3D-mapping capability will be necessary for autonomous navigation. Toward a light and practical SLAM algorithm for indoor scenarios, we propose a fast SLAM algorithm which benefits from sensor geometry for feature extraction and enhance the mapping process using dominant orthogonality in the engineered structures of man-made environments. Range images obtained using a nodding SICK are segmented into planar patches with polygonal boundaries in linear time. Right corner features are constructed based on the recognized orthogonal planes and used for robot localization. In addition to these corners, the map also contains planar patches with inner and outer boundaries for 3D modeling and recognition of the major building structures. Experiments using a mobile robot in our laboratory hallway prove the effectiveness of our approach. Results of the algorithm are compared with hand-measured ground truth.**

I. Introduction

One of the most fundamental requirements of an intelligent mobile robot is autonomous navigation. This capability is realized by obtaining a suitable map of the environment, which is a compact representation of the robot surroundings. *Simultaneous Localization and Mapping* (SLAM) is a complex problem where the robot is also required to remain localized with respect to the portion of the environment that has already been mapped. Although today there are quite various techniques for solving SLAM, (ex. realtime particle filters, sub-mapping strategies or hierarchical combination of metric-topological representations; see [13] for a survey), still consistency and complexity remain the most challenging problems in the field. This is specially the case when dealing with more dimensions and degrees of freedom, as in the 3D-SLAM.

Regardless of the innate complexity, SLAM algorithms are eventually needed in simple embedded mobile robots with limited processing power. Many of them are targeted for indoor usage such as health care [12] or housekeeping applications [5]. However, less research effort has been done on development of lightweight range-based 3D-SLAM algorithms well adapted for the conditions of indoor environments. As stated in [15], such algorithms are necessary as a basis for many current applications and also a vast number of potential new applications in the future, like autonomous security and surveillance systems or surveying robots for 3D reconstruction.

Considering indoor environments, in this paper we aim to develop a lightweight and realtime consistent incremental SLAM algorithm based on right angle corners and planar surfaces which are the dominant structures. In our previous work we presented *OrthoSLAM* [10] for realtime mapping of office-like environments in 2D. Considering a very common constraint usually present in many indoor environments, the orthogonality, we showed that the uncertainty on the robot orientation can be kept low and bounded. Knowing the robot orientation, SLAM is reduced to a linear estimation problem. The simple assumption of orthogonality on the shape of the environment, comes from the fact that in most indoor engineered environments, major structures, like walls, windows, cupboards etc., can be represented by sets of planes which are either parallel or perpendicular to each other. In fact, ignoring other planes (arbitrary oriented or non-orthogonal) not only does not lead to loss of valuable information during SLAM, but also brings robustness on the robot orientation; and filter out many dynamic objects. This exactly conforms to our goal in mapping and later recognizing the major structures of the buildings.

However, after localizing the robot within the SLAM framework based on mentioned subset of features, clearly it is always possible and straight forward to put the rest of the features in the map in a different layer. Hence, the main idea is to distinguish between two types of features in general: core features (in our case orthogonal features in indoor) which are robust enough to be used for SLAM, and the other features which may be useful to be mapped from the representation perspective, but they are not that much unique, static or well-observed to be used for localization. Arbitrarily oriented planar patches in this work are treated as the second mentioned category, while the SLAM core uses right angle corners based on orthogonal planes.

The assumption of orthogonality as a geometrical constraint has already been used by some other researchers, for example [9, 5]. This geometrical constraint is usually applied as a post-processing step in mapping in order to increase the precision and consistency of the final map. However, in our approach, the orthogonality assumption is not applied as an additional post-processing constraint, rather it is used to select only orthogonal planes for construction of the right angle corners which in turn are used for localization. In the orthogonal framework, lines and planes are represented by just one distance parameter (no orientation is needed). In addition, as far as the robot is able to detect orthogonal features, it is able to determine its orientation with a very low and bounded uncertainty. Since there is no data association at the level of single features in the process of obtaining the orientation, the resulting estimate remains highly consistent. Consequently, robot localization reduces to finding the translation between the robot and the global coordinate system. This is a linear

estimation problem which is solved using high level right angle corner features constructed by the orthogonal planes.

In fact, here the mapping is done based on the orthogonality constraint and in a simplified framework, rather than using the constraint as an extra observation. This is a great difference which leads to the removal of non-linearities in the observation model and a rather precise and consistent mapping. In this sense, our work is similar to Orthogonal Surface Assignment, a recent work of Kohlhepp et al. in [6]. Their approach uses the same constraint in the framework of multi-hypothesis tracking. However, we use right angle corners and the presented algorithm is much simpler.

A nodding SICK provides us with rich point clouds which are segmented into smooth parts using our Bearing Angle based segmentation algorithm introduced in [4]; see Fig. 1. Planar patches are then fitted to each segment and coplanar patches are merged. In addition, boundary polygons are extracted from the range image and treated afterward to capture the shape of the environment. Major orthogonal planes nearby each other are then used to construct corners as robust and well distinguishable high level features. Localization is done in two steps: first robot orientation is inferred using the detected orthogonal planes, then its translation is calculated by iteratively matching the observed and mapped corners. Having the robot position, all the extracted planes are finally mapped along with their boundaries and holes information. This gives the necessary framework for recognition of the major structures in the buildings. Obtaining structure annotated maps is one of our goals towards putting more semantics in robot navigation.

II. Related Work

In order to build 3D maps of the robot environment, apart from stereo cameras which deliver range information just for textured part of the scene, many researchers use a vertical laser scanner that creates registered 3D data by sweeping through the scenes [14, 3, 16]. It is more efficient to move the 2D sensor instead. To do so, usually 3D laser scanners are constructed by rotating a 2D laser range finder. This can be implemented by nodding or rotating the 2D scanner around its lateral or radial axis in a stepwise or continuous manner. Such sensor setups are used in various experiments [17, 11, 18, 1, 7, 8]. In this work a nodding SICK (Fig. 1) is used as the 3D range sensing device and the common stop and move scenario is considered to obtain consistent point clouds.

In most of the approaches to 3D-SLAM, the map directly includes point clouds registered in a global coordinate system with ICP [19]. The problem with this approach or its discretized alternative, occupancy grids [2] is that they don't benefit from the geometrical symmetry present in many target environments like indoor or urban regions; hence the provided map is not compact nor does it have any abstraction.

A well accepted model to capture geometry of the structured environments is the planar representation which can be enhanced by encoding the boundary information (Fig. 2(a)). A major challenge then, is how to segment huge amount of scanned points into homogeneous regions, possibly planar,

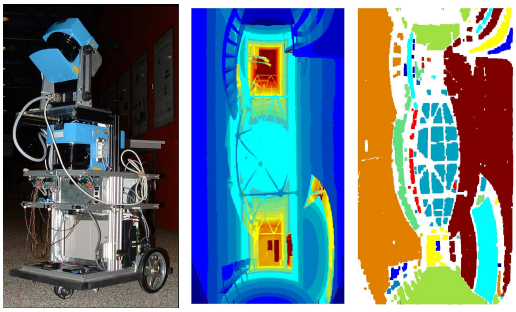


Fig. 1. Right: BIBA robot with nodding SICK attached; note that the other two back to back installed scanners are not used in the experiments reported here. Center: A sample range image saved in the corridor of our lab Left: Segmented range image using BA-based segmentation

with reasonable performance. In [17], RANSAC along with a region growing approach is used. This approach does not take into account non-uniform distribution of the points in the cloud for the initial voxelization which is a dominant fact in constant angular resolution scanners like SICK. An improved Split-and-Merge scenario is utilized in [7]. It mainly lies in edge-based category of segmentation methods with intermediate fittings and iterative splitting which leads to time complexity of $O(n \log n)$. In this paper, BA-based segmentation [4] is used which is $O(n)$ and benefits from geometry of the sensor; Fig. 1 shows a sample scene and result of the segmentation.

Using extracted plane segments, [17] implements an EKF to update the map constructed based on SP-model. Kohlhepp et al [6] propose an elastic view graph which links range views by variable pose transformations as a local tracker underneath of a global layer which closes the loops. A Building Coordinate System is tracked using the proposed orthogonal surface assignment algorithm for man-made work spaces. This is very similar to our usage of the orthogonality constraint. However, they rely on matching single planes and tracking more than one hypothesis to obtain a robust estimate; while we benefit from matching combinations of planar segments related together by means of a 3D right-angle corner. In addition, we enhance the reliability of the data association using iterative corner matching.

III. Feature Extraction

As mentioned, SLAM can be implemented on the same level as the raw data points obtained from the scanner. Single points can be treated as orientation-less features which are less certain and less distinguishable but may be matched in large amount to compensate the uncertainties. A more efficient alternative is to push the mapping one level up where features are less frequent, but more informative, certain and unique. Plane features are a quite reasonable choice for representation of the indoor environment. Here, scanned points are grouped into the smooth regions using Bearing Angle based segmentation [4] and their inner and outer boundaries along with their neighboring regions are obtained. Using Principal Component Analysis (PCA) the least square plane is fitted to each region and coplanar regions are merged. The common Hessian notation is used to represent planar features: $\mathbf{n}.\mathbf{x} - d = 0$, where $\mathbf{n}$ is

the normal vector and d is the signed distance to the origin. the plane normal vector, $\mathbf{n}$, is obtained as the eigenvector corresponding to the smallest eigenvalue of matrix $\mathcal{C}$:

$$\mathcal{C} = \begin{bmatrix} \Sigma\mathcal{X}_i^2 & \Sigma\mathcal{X}_i\mathcal{Y}_i & \Sigma\mathcal{X}_i\mathcal{Z}_i \\ \Sigma\mathcal{Y}_i\mathcal{X}_i & \Sigma\mathcal{Y}_i^2 & \Sigma\mathcal{Y}_i\mathcal{Z}_i \\ \Sigma\mathcal{Z}_i\mathcal{X}_i & \Sigma\mathcal{Z}_i\mathcal{Y}_i & \Sigma\mathcal{Z}_i^2 \end{bmatrix} \quad \text{where} \quad \begin{matrix} \mathcal{X}_i = x_i - \bar{x} \\ \mathcal{Y}_i = y_i - \bar{y} \\ \mathcal{Z}_i = z_i - \bar{z} \end{matrix}$$

which is the covariance matrix of the associated points to the plane. Plane distance parameter is obtained as $d = \mathbf{n}.\mathbf{cp}$, where $\mathbf{cp} = (\bar{x}, \bar{y}, \bar{z})^T$. As a notion of how good the plane estimate is, a certainty factor is considered: $cf = kN\mu^2 S$, where k is a balancing coefficient, N is the number of points assigned to the plane, μ is the ratio of inlier points and S is the area. In the orthogonal framework, $\mathbf{n}$ is known a priori and the planes are represented just with a single parameter d. Hence, cf can simply be considered proportional to the inverse of the variance of d. When $\mu < 80\%$ the extracted region is not accepted as a planar patch.

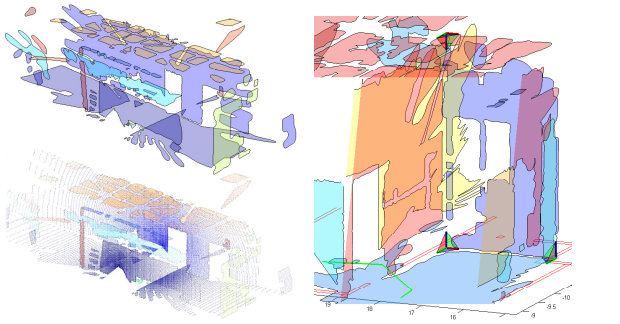
Unfortunately, infinite planes are not that much unique and in cluttered areas data association would be problematic. Adding boundary information and using plane segments may help; specially in localization, where the map is given. However, the shape of segments is not invariant through successive observations in SLAM. The robot may observe unseen parts of the same plane which may not be similar to the previously mapped areas, but still it should be recognized as a re-observed feature. Hence, boundary information may just indirectly help in accepting more deviation in other parameters of the segment when some similarity is detected. The distance measure between plane i and j, when $|d_i - d_j| < \delta_d^{max}$ and $\mathbf{n}_i^T . \mathbf{n}_j > cos(\delta_n^{max})$, is calculated as:

$$d_{ij} = k_d|d_i - d_j| + k_n \cos^{-1}(\mathbf{n}_i^T . \mathbf{n}_j) + k_s(1 - \frac{S_{i\cap j}}{min(S_i, S_j)})$$

where k_d, k_n and k_s are three coefficients balancing the importance of distance, orientation and shape in the final distance measure. The symbol S denotes the surface and $i \cap j$ is the intersection of the boundary polygons of plane i and j. In addition to data association, the boundary information is used to reconstruct the structures present in the scene much more realistically by preserving door openings, windows, etc. (Fig. 2(a)).

However, Much more robust association is obtained when features are matched jointly since the chance of observing similar cases drops dramatically as the number of jointed features increases. To take a first step toward this idea, corners are used here. Corners are intersection points of sets of three planes and encode the whole information of their states. Therefore, in observing and matching each corner feature, in fact a certain combination of the three planar segments is observed and matched with the map. This leads to more unique features and less clutter in the feature space, hence results a more robust data association.

Using the assumption of orthogonality, 3D right angle corners (Fig. 2(b)) are used as high level features for robot localization. It should be noted that these corners are not detected as point features from the raw data, rather they are



(a) (b)

Fig. 2. (a) Top: Planar patches with inner and outer boundaries extracted from the range image of Fig. 1 Bottom: Point representation of the extracted plane segments for the same scene (b) A sample area in the map including 3 corners

constructed using the result of the plane extraction and after filtering out non-orthogonal planes.

Orthogonal planes (segments) are distinguished by considering a global coordinate system which is called Building Coordinate System (BCS) and is aligned to main directions of the building (length, width and height). Such a coordinate system can be automatically assigned in the start of the mapping process using initial observation (as explained later in section V) or manually defined by providing the initial robot orientation. Then, extracted planes are transformed to global frame using odometry readings and planes with normal vector parallel to the main axes of the BCS are tagged as orthogonal planes and treated in each of the three dimensions separately. The planes perpendicular to the x-axis of the BCS are called *i-plane* (examples are planes A, D and G in Fig. 3). Accordingly, *j-plane* and *k-plane* are also categorized:

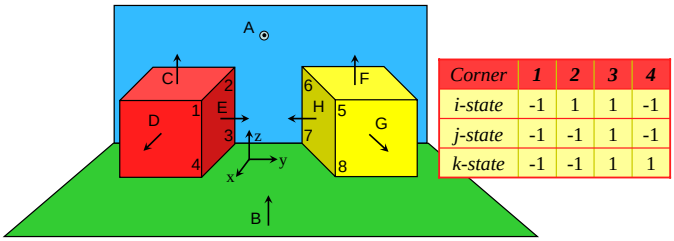
$$\textit{i-planes} = \{i \,|\, |(1,0,0).\mathbf{n}_i| > \cos(\Phi)\} \quad (1)$$

$$\textit{j-planes} = \{i \,|\, |(0,1,0).\mathbf{n}_i| > \cos(\Phi)\} \quad (2)$$

$$\textit{k-planes} = \{i \,|\, |(0,0,1).\mathbf{n}_i| > \cos(\Phi)\} \quad (3)$$

where small and less supported planes[1] are not considered here and Φ is the maximum acceptable deviation which is set to 2^o in the experiments. Selecting one plane from each of these sets, a corner is constructed if they are neighbors or their intersection point lies near (within a $30cm$ range) or inside each of the *i-*, *j-* and *k-plane* segments. If so, the intersection point is considered as corner position with the corresponding certainty factors along each direction. Normal vectors of the three planes are orientation of the corner edges, or its orientation for short. Fig. 2(b) shows three such corners in the map. Using the neighborhood information makes it possible to observe corners which are actually occluded and the corners point itself is not in the sensor view. However, still all the three planes of a corner should be observed and no inference is performed on the shape of the planes to guess existence of the corners; for example in Fig. 3, four corners of each of the cubes which are not numbered, is not observed in the presented scene.

[1]Planes with area smaller than 0.1 m^2 and supported by less than forty measurement points



Corner	1	2	3	4
i-state	-1	1	1	-1
j-state	-1	-1	1	-1
k-state	-1	-1	1	1

Fig. 3. A sample scene containing different types of corners. Small arrows show surface normal vectors aligned with the depicted BCS. The state vector is also given for the first four corners.

In addition, for each intersection line of each corner, a binary state is defined which encodes the relative state of the corresponding planes. Consider the eight numbered corners depicted in Fig. 3. Assume the robot is observing the scene from the same view and surface normal of the segments are always pointing toward the robot as shown with the small arrows. All four corners observed from the red cube (on the left side of the picture) share the same direction; similarly all four corners observed from the yellow cube (on the right side of the picture) are also identical in this regard. However, each corner in these four-corner sets has a different state (is of different type). For example consider corner 1 and 2. The intersection line of plane E and C is in front of corner 2 but behind corner 1. Since this intersection is along the x-axis of the BCS, its state is called *i-state* and is equal to +1 for corner 2 meaning intersection lies in positive side and -1 for corner 1 accordingly. For the other two intersection lines, *j-state* and *k-state* are defined in the same manner. In Fig. 3, state vector is given for the four corners of the red cube.

Adding orientation and edge states to the information encoded in the corners, dramatically increases their uniqueness. On the other hand, since corner information is obtained from major orthogonal planes with large number of supporting measurement points, corners are also robust.

IV. Localization

Although mobile robots in indoor mostly have a 3D configuration space (x, y and robot heading angle), here a full 6D space is considered to support environments with ramps. In addition, this helps compensating some non-systematic errors of the custom built sensing system, like one half a degree change in start angle of the nodding operation, and obtaining more precise 3D maps. Hence, the robot pose is estimated in all 6 dimensions (usually pitch and roll remain below one degree and z vary few millimeters in flat areas). Using the orthogonality constraint, robot localization is achieved in two stages: first its orientation with regard to BCS is found and then its position is obtained.

Relying on this fact that the maximum error of the robot odometry in the robot heading angle between successive observations is bounded, orthogonal planes can be detected by searching orientation intervals around odometry predicted BCS axes; see Eq. 1-3. Using the normal direction of these planes, each weighted by its certainty factor, new BCS orientations are estimated by fitting three orthogonal directions to the observed orthogonal plane set and the whole scene is aligned with the new BCS.

Secondly, translation of the robot is obtained by matching the right angle corner features. Corner features are constructed based on the tagged orthogonal planes as explained in previous section, and matched with the mapped corners in an iterative manner. The process of matching is very similar to ICP, but applied on corners and for translation only. At each iteration corners are binded to their nearest neighbors and hence each matching a set of three planes with the map at the same time. Considering the set of matched planes through the matched corners, a global transformation is calculated by weighted average of the innovations of the matched planes. Certainty factors of the planes are used to obtain the weight. Considering the x direction and ij as a match between *i-planes* of the map and the current observation:

$$\Delta x = \frac{\Sigma[w_{ij}(d_i - d_j)]}{\Sigma w_{ij}} \quad \text{where} \quad w_{ij} = \frac{min(cf_i, cf_j)}{d_{ij} + k_w}$$

where k_w is a constant adjusting the effects of the weaker matches against the stronger ones. Δy and Δz are calculated in the same manner using matches found for *j-planes* and *k-planes*. Then the calculated transformation is applied to the observation set and the process is repeated until bindings remain unchanged in successive steps or a minimum error is reached. Normally when all of the matchings are correct in the first step, no more iterations are needed, otherwise in few steps the jointly compatible match is found. Since, the number of corners is not high and also they are quite distinctive, the matching is very fast.

Iterative matching of the corners, is an additional insuring policy to avoid spurious data association. Although corners are very distinctive features and the chance of a wrong match is generally low, still it is important to compensate for the smaller numbers they are matched in, and also low observation rate of the nodding SICK. It is worth mentioning that matching corners are nothing more than matching the corresponding sets of three planes simultaneously; and in case no corner can be matched, the system reduces to normal plane segment matching but restricted to the orthogonal directions.

V. Mapping

An important issue in robotic mapping is the notion of uncertainty. Dealing with plane features in 3D range measurement, considering only systematic errors modeled using covariance matrices sometimes can be misleading. Usually standard deviation of around 1-2 centimeters is considered for single range measurements of SICK. Then planes with thousands of measurement points turns out to be precise within fractions of millimeter which is clearly unrealistic. As the number of measurement points on a plane increases, non-systematic error dominates the real uncertainty, so heuristic terms should be added to the covariance matrix to compensate. In this paper, instead of calculating the covariance matrix of the planes, and then adding the mentioned heuristic values; a rough estimation of the uncertainty is taken into account based on the introduced certainty factor (cf). Within the framework of orthogonality constraint, orthogonal planes are

represented by a single distance parameter. In addition, given the robot orientation in each step, the estimation of the feature parameters in the map is linear. Therefore, this simple and efficient approximation for the uncertainty fits well.

On the other hand, each set of orthogonal planes is independent of the others. In other words, *i-planes* are independent of the *j-* and *k-planes*, since they are perpendicular to each other. Therefore, three independent maps are used to enhance the efficiency in searching for a match in data association stage. Hence, the system estimates state $\mathbf{X}$ and its covariance $\mathbf{P}$ for *i-*, *j-* and *k-planes* separately:

$$\mathbf{X} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 1/cf_1 & 0 & \cdots & 0 \\ 0 & 1/cf_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1/cf_n \end{bmatrix}$$

where n is the number of planes in each direction and grows with time. To estimate $\mathbf{X}$, a simple Kalman Filter is implemented for each direction which neglects the dependency of distance parameters within each orthogonal set, as $\mathbf{P}$ is assumed to be diagonal.

To initialize the map, first directions of the Building Coordinate System, the global coordinate system which is aligned with the structure of the indoor environment, should be assigned and then as the robot moves, it should be tracked in successive steps. As mentioned, the initialization step can be done by manually providing the robot with its heading angle at the beginning. In addition assuming the robot is standing on a flat ground, one just needs to determine a single rotation which is the robot heading angle. Taking the first observation in a proper condition which lets covering reasonable parts of the surrounding walls, it is possible to infer the BCS. The orientation of the 2D projection of the surface normals on the horizontal plane, calculated for each point after the plane extraction, is used to create an angle histogram. The two largest peaks 90 degrees apart belong to the most observed orthogonal directions which can be assigned to x and y axis of the BCS; then z direction is obtained from their cross product.

After the initialization, in each step, the observed scene is aligned with the BCS and the robot is localized using the corner features. Since the observations are one to two meters apart each other, the odometry uncertainty assumed much bigger than the uncertainty of observations from the nodding SICK; hence the robot position is overwritten with the result of corner matching. As an outcome of matching the corners, corresponding segments in the map are already known for the planes linked to a corner. The rest of the planes are also matched and then the map is updated and the new features (planes and corners) are added.

VI. Experimental Results

As a proof of concept for the proposed algorithm, BIBA robot (Fig. 1) equipped with a nodding SICK is used. The robots is running a real-time operating system (RTAI Linux) with an embedded odometry, obstacle avoidance and remote control module via wireless network. The two fixed SICK scanners are not used in the SLAM process presented here. In this experiment, the robot starts in a crowded office, explore it for some time, then go out to the corridor and finally enters another office. Totally, 27 3D range scans are taken, 216,961 points each. The whole experiment collects 5,857,947 measurement points. Unfortunately, using the current sensor setup, it takes around a minute to complete such 3D scans. This explains why the observation rate is low and the data association is very critical.

Fig. 4(a) shows the map obtained by registering the point clouds just using the robot odometry. The upward deviation of the right part of the map shows the inconsistency caused by unbounded and growing error. In contrast, in Fig. 4(b) the map obtained by the proposed algorithm seems quite consistent and specially quite precise in orientation, which is directly the benefit of using orthogonality constraint. In this figure, the green polyline (light) shows the obtained robot path against

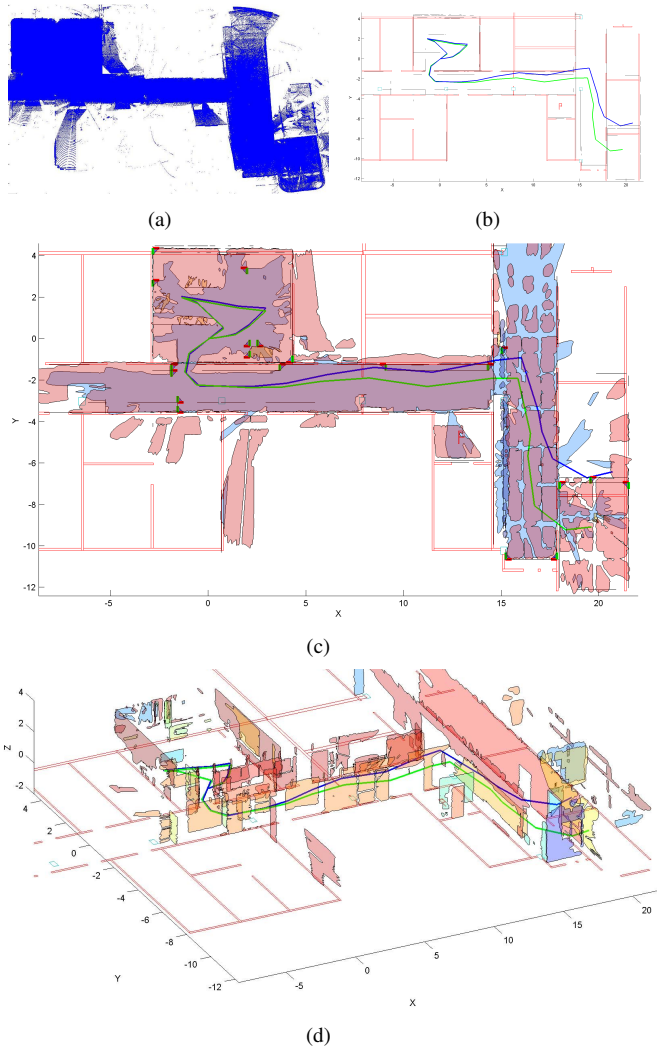


Fig. 4. (a) Point map constructed just using the robot odometry. As it can be seen, the map is inconsistent, but dense cloud makes it less apparent. (b) Top view of the constructed map using corner features (the horizontal planes are omitted). It is clearly consistent and near the plotted ground truth. The blue (dark) polyline is the robot path obtained from the odometry and the green polyline shows the path calculated with SLAM. (c) Same as (b) but including horizontal planes. (d) A different view of the map in (b).

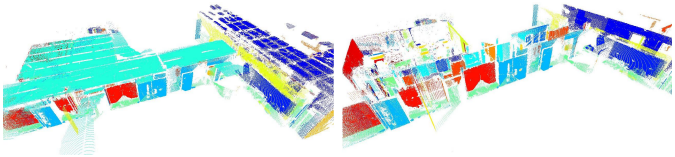


Fig. 5. Point cloud representation of the whole map. For clarity, points are colored based on the plane they belong to. At right the ceilings are omitted to reveal more details.

the odometry measured path which is plotted in blue (dark). In addition, red lines show the hand measured ground truth. For clarity, in this figure horizontal planes (mainly the floor and the ceiling) and non-orthogonal planes are not plotted. Hence, i- and j-planes look like black lines from the presented top view. It can be observed that the orientation is perfectly precise. However, at the right side, small error can be noticed in the x-direction. This is mainly the result of the distance traveled in the horizontal corridor. In the corridor, there are very well observed walls perpendicular to y-axis (j-planes) which are supported by many measurement points and help the robot to be very certain in corners lateral position but in x-direction the situation is vice-versa. As a direct consequence, robot uncertainty is much higher in x-direction and some offset remains in the right part of the map. Ceiling and floor are depicted in Fig. 4(c) along with the detected corner features. Fig. 4(d) shows a different view point. At the top and at the right there are big windows, and since the glass is not visible with laser, most part of them are missing in the map. Colored registered point clouds depicted in Fig. 5 shows that there are not that much measurement points corresponding to those areas, hence, lake of the windows are certainly the problem of sensing system and have nothing to do with the algorithm. However, it also creates hard cases for the algorithm. Since missing a major wall because of the big windows, prevents observing the corresponding corners which can be very valuable features.

VII. Conclusion and Future Work

Using the orthogonality constraint present in many man made environments, a 3D-SLAM algorithm was presented for indoor environments. Corners are proposed as high level features which can reliably solve data association problem within an iterative matching scenario. Experimental results validated the proposed approach and showed the benefit of the used constraint in the map precision and consistency. However, in the experiments, less corners are observed than what was expected. Studying the registered point clouds, it turns out to be because of the blind spots of the sensing system at right and left sides of the robot. Since the robot mostly should see the corners at sides of it, the current installation does not seem suitable. In the future work, the 3D sensing system will be rotated 90 degrees with respect to the robot such that the SICK rotates from right to left. This should increase the observed corners and enhance the precision of the map. In addition, automatic recognition of building structural elements like walls, floor and ceiling is among our future work.

Acknowledgement

This work has been supported by the Swiss National Science Foundation No. 200021-101886 and the EC projects FP6-IST-002020-COGNIRON and FP6-IST-027140-BACS. The authors also wish to thank Dr. Jan Wiengarten for his support in building the sensor and collection of the raw data of the experiment presented in this paper.

References

[1] D. Cole, A. Harrison, and P. Newman. Using naturally salient regions for SLAM with 3D laser data. In *Robotics and Automation, Workshop, 2005. Proceedings. ICRA '05. 2005 IEEE International Conference on*, 2005.

[2] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989.

[3] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. In *Advanced Mobile Robots, The Fourth European Workshop on*, 2001.

[4] A. Harati, S. Gächter, and R. Siegwart. Fast range image segmentation for indoor 3d-slam. Technical Report 166200-2006-01, Swiss Federal Institute of Technology (ETH Zürich), Switzerland, 2006.

[5] P. Jensfelt, H. Christensen, and G. Zunino. Integrated systems for mapping and localization. In *ICRA-02 SLAM Workshop (J. Leonard and H. Durrant-Whyte, eds.)*, 2002.

[6] P. Kohlhepp, G. Bretthauer, M. Walther, and R. Dillmann. Using orthogonal surface directions for autonomous 3d-exploration of indoor environments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3086–3092, Oct. 2006.

[7] P. Kohlhepp, P. Pozzo, M. Walther, and R. Dillmann. Sequential 3D-SLAM for mobile action planning. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 722–729, 28 Sept.-2 Oct. 2004.

[8] M. Montemerlo and S. Thrun. A multi-resolution pyramid for outdoor robot terrain perception. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, San Jose, CA, 2004. AAAI.

[9] P. Newman, J. Leonard, J.D. Tardos, and J. Neira. Explore and return: Exprimental validation of real-time concurrent mapping and localization. In *Proceedings of the International Conference on Robotics and Automation - ICRA*, 2002.

[10] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis. Orthogonal slam: a step toward lightweight indoor autonomous navigation. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5007–5012, Oct. 2006.

[11] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-based laser scan matching for outdoor 6D SLAM. In *Advances in Artificial Intelligence. KI 2005. Proceedings Springer LNAI. 28^{th} Annual German Conference on*, volume 3698, pages 304–319, September 2005.

[12] D. Rodriguez-Losada, F. Matia, A. Jimenez, R. Galan, and G. Lacey. Implementing map based navigation in guido, the robotic samrtwalker. In *Proceedings of the International Conference on Robotics and Automation - ICRA*, 2005.

[13] S. Thrun. *Exploring Artificial Intelligence in the New Millenium*, chapter Robotic mapping: A survey. Morgan Kaufmann, 2002.

[14] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 321–328vol.1, 24-28 April 2000.

[15] J. Weingarten. *Feature-based 3D SLAM*. PhD thesis, Swiss Federal Institute of Technology in Lausanne, September 2006.

[16] J. Weingarten, G. Gruener, and R. Siegwart. A fast and robust 3D feature extraction algorithm for structured environment reconstruction.

[17] J. Weingarten and R. Siegwart. 3D SLAM using planar segments. In *Robotics and Automation, 2006. Proceedings. ICRA '06. 2006 IEEE International Conference on*, volume in press, 2006.

[18] O. Wulf, K.O. Arras, H.I. Christensen, and B. Wagner. 2D mapping of cluttered indoor environments by means of 3D perception. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pages 4204–4209, Apr 26-May 1, 2004.

[19] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.