Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Orthogonal SLAM: a Step toward Lightweight Indoor Autonomous Navigation

Viet Nguyen, Ahad Harati, Agostino Martinelli, Roland Siegwart
Autonomous Systems Laboratory
Swiss Federal Institute of Technology (ETH Zurich)
Rämistrasse 101, CH-8092 Zürich, Switzerland
Email: {viet.nguyen, ahad.harati, agostino.martinelli, roland.siegwart}@epfl.ch

Nicola Tomatis
BlueBotics SA
PSE-C, CH-1015 Lausanne, Switzerland
Email: nicola.tomatis@bluebotics.com

***Abstract*— Today, lightweight SLAM algorithms are needed in many embedded robotic systems. In this paper the Orthogonal SLAM (*OrthoSLAM*) algorithm is presented and empirically validated. The algorithm has constant time complexity in the state estimation and is capable to run real-time. The main contribution resides in the idea of reducing the complexity by means of an assumption on the environment. This is done by mapping only lines that are parallel or perpendicular to each other which represent the main structure of most indoor environments. The combination of this assumption with a Kalman Filter and a Relative Map approach is able to map our laboratory hallway with the size of $80m \times 50m$ and a trajectory of more than $500m$. The precision of the resulting map is similar to the measurements done by hand which are used as the ground-truth.**

Keywords: Lightweight SLAM, Orthogonality, Indoor Environment, Mobile Robotics.

I. Introduction

Autonomous navigation is one of the most basic behaviors needed in many applications and especially in mobile robotics. It is today agreed in the mobile robotic research community that mapping is an important requirement. The research effort in this field has thus increased during the last years. The problem is effectively highly complex especially for *Simultaneous Localization and Mapping* (SLAM), where the robot is required to remain localized with respect to the portion of the environment that has already been mapped.

Regardless of the innate complexity, SLAM algorithms are eventually needed in simple mobile robots. Many of them are targeted for indoor usage as medical or housekeeping applications (see for example a vacuum cleaner platform [1] or a health care robot [2]). Thanks to the fast achievements in sensor developments, normally sensing is not the major problem. Newer sensors are more precise, cheaper, smaller and even more intelligent. However, development of methods and algorithms for proper use of extracted data, exploiting limited processing power available, is usually a greater challenge.

Today there are some techniques for solving SLAM with reasonable complexity [3] (ex. realtime particle filters, submapping strategies or hierarchical combination of metric-topological representations). However, these techniques are developed having powerful computational resources in mind. Less research effort has already been spent on development of lightweight SLAM algorithms which are of vital importance in many applications of mobile robotics.

Considering indoor environments, in this paper we aim to develop a lightweight and real-time consistent SLAM algorithm. The target of this work especially is minimum systems embedded on simple robots. Hence, using simple assumptions, usually present in many indoor environments, and relative map state (to decouple localization and mapping), the process is simplified a lot. This simplification even comes with other benefits also; like robustness, precision and implicit dynamic object filtering.

We will show that using known approaches it is possible to robustly and precisely perform SLAM in indoor, office-like environments by using line features and taking a simple assumption about the shape of the environment: the Orthogonality. It comes from the fact that in most indoor engineered environments, major structures, like walls, windows, cupboards, etc., can be represented by sets of lines which are parallel or perpendicular to each other. For reconstruction of the desired map, it is sufficient to extract and maintain those major lines. In fact, ignoring other lines (arbitrary oriented or non-orthogonal lines) not only does not lead to loss of valuable information, but also brings us amazing robustness on the robot orientation and filter out many dynamic objects.

The assumption of orthogonality as a geometrical constraint has already been used by others, for example [4], [1]. This is normally a post-processing step in mapping in order to increase the precision and consistency of the final map. However, in our approach this assumption is not applied as an additional constraint, rather it is used to select only orthogonal lines as observations, which let us to represent the line features using just one parameter. In fact we do the mapping based on this constraint and in a simplified framework, rather than applying it as extra observation afterward. This is a great difference which leads to the removal of non-linearities in the observation model and rather precise and consistent mapping.

II. Related work

In recent years several techniques have been suggested to reduce the computational requirements of SLAM. Most of them are mainly constructed based on the creation of a hierarchy in the stochastic map framework. By using this

1-4244-0259-X/06/$20.00 ©2006 IEEE

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.

idea, it will be possible to postpone some global updates (for parts of the state covariance matrix) and only process local area of the robot in regular steps. This leads to a reduction of computational requirement while preserving the optimality of the filter. Another related approach is to neglect the correlation between elements of different sub-maps. To preserve consistency, usually ad-hoc strategies (like inflation of the uncertainty within sub-maps) are taken. Few examples of applying some of these ideas are *Decoupled Stochastic Mapping* (DSM) [5], *Compressed EKF Filter* (CEKF) [6], *Constrained Local Sub-map Filter* (CLSF) [7] and *Network Coupled Feature Maps* (NCFM) [8].

FastSLAM [9] is another attempt toward reducing complexity of SLAM. It benefits from the conditional independency of landmark positions given the robot pose. It uses a particle filter to localize the robot in order to overcome the problems caused by the accumulation of non-linearity errors. Furthermore, the complexity of updating the landmarks is reduced by implementing a tree data structure for landmark manipulation. Realtime particle filter [10] extends FastSLAM by using mixture of densities each represented with a set of particles. In this manner in normal observation steps, only a subset of particles is updated and less computational resources are used. At the global update stages, particle population is unified via resampling.

However, all of these methods are not specially optimized for indoor environments where there are dynamic objects (like humans), short and large loops and a more precise map is needed. In such conditions using only point landmarks is inefficient [11]. Having precise range sensors and many geometric features handy, suggests using more abstract features. Among many geometric primitives, 2D lines or line segments are the first and simplest option while very efficient in describing static structures in office-like environments. They were already used by many authors. Vandorpe et al. [12] introduce a dynamic map building algorithm based on geometric lines and circles using laser scanner. Tardos et al. [13] use *Hough Transform* to extract and represent point and line features from 2D data. A very recent work [14] proposes a mapping algorithm based on FastSLAM [9] using segments. *Probabilistic Principle Component Analysis* and a clustering method called G-means are used to extract the segments.

The main difference between the work presented here and the other approaches cited in this section relies in the way the complexity is reduced. While the other approaches modify the basic algorithm by some assumptions and approximations, we propose to use a constraint of the environment. This allows keeping a complete coherent approach that now is actually applied to a simplified problem. This results in a better performance for all cases that do not violate our orthogonality assumption.

In this section, first we give an overview of the current major approaches toward reduction of complexity in solving SLAM, in order to show how far we can go and to justify our approach. Then the history of using line features is briefly reviewed.

III. The *OrthoSLAM* Algorithm

In this section, we describe the *OrthoSLAM* algorithm to perform 2D-SLAM in a typical indoor, structured environment. The goal of this approach is to build a map containing line features representing the walls, cupboards, doors, windows, etc. in the environment.

The main assumption we want to exploit is that most of the major lines (from walls, cupboards, doors, etc) in the environment are either parallel or perpendicular to each other. We select from the environment only the line features verifying this hypothesis. Therefore, lines in our map can be grouped into two distinct classes: horizontal and vertical lines (H/V lines). Hereafter, the terms lines, segments and line segments are used interchangeably. We employ the term "orthogonal lines" to refer to the lines which are either parallel or perpendicular to each other. Finally, the horizontal (vertical) orientation is designated to be aligned with the x-axis (y-axis) of the global frame.

In Fig. 1 we show a scheme of the implemented *OrthoSLAM* algorithm. It consists of several phases which will be discussed in the following subsections.

Line Extraction

Line segments are extracted online from raw 2D laser range scans. We select the popular line extraction algorithm, *Split-and-Merge*, among the existing ones as it has been shown to provide the best performance [15]. In this implementation, in order to have good and reliable extracted lines, we only consider line segments with length greater than $30cm$.

Data Association

Data association (DA) decision is made for each new observed feature, i.e. extracted line segment, to determine if: 1) it is the same as the one currently in the map, 2) it is a new orthogonal line, 3) it is not an orthogonal line or it is spurious. Using the predicted robot pose each step, a nearest-neighbor-filtering (NNF) algorithm is implemented for the DA. This simple method works fine when the robot traverses at a speed of about $0.3m/s$. The other lines are safely discarded as non-orthogonal lines. To make the DA more robust, end points of the segments are used in order to distinguish rather collinear lines or parallel lines near each other. If a line segment is erroneously extracted as several broken segments which usually happens in noisy and dynamic environments, the most overlapped segment will be matched to the one already in the map. The other collinear segments are considered as new observations and added to the map. These broken line segments will be afterward merged (see below). Since the sensor range is bounded, the number of lines in each observation is bounded. Hence the DA has linear time complexity on the number of lines in the map. Limiting the search to the *local views* (described below), the DA can be done in constant time complexity.

Orthogonalization

Once the DA task is done, we know which lines are horizontal and vertical. Due to noise and imperfect orthogonality of the environment, extracted lines are not perfectly orthogonal and

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.

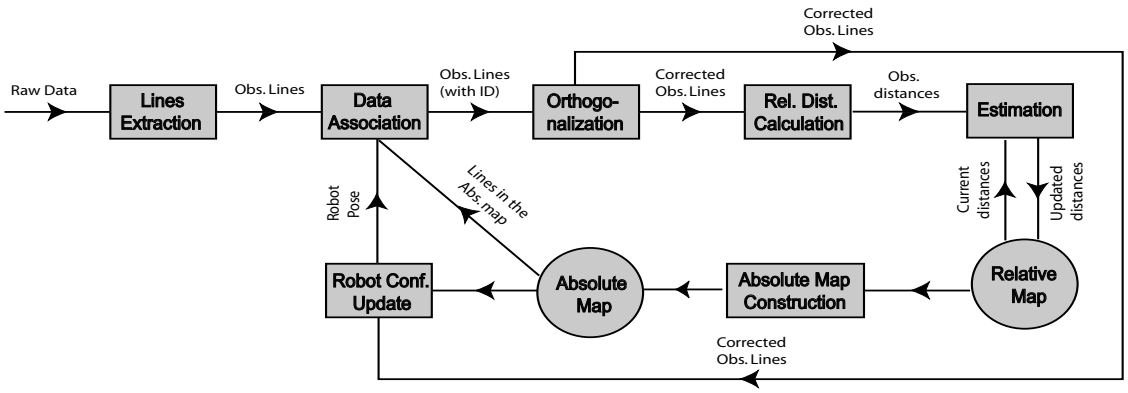

Fig. 1. Simple scheme of the *OrthoSLAM* algorithm.

may be slightly deviated. The aim of this step is to correct the parameters of the extracted lines in each observation using the assumption of orthogonality.

To do this, first, a reference of orientation is defined in each observation for horizontal and vertical directions. The orientation of the horizontal reference is obtained by a calculation of the weighted average of angular parameters of lines (a_i) while subtracting $\pi/2$ in the case of vertical lines. The orientation of the vertical reference is the one perpendicular to that of the horizontal reference. In this calculation, the lengths of the segments are used as the weights. Ideally, the orientation reference should align with the orientation of the global coordinate, but in practice it is not the case because of the noise and the imperfection of the environment. Next, the line segments in the observation are rotated around their midpoints to align with the reference orientation. After the rotation, the lines are perfectly orthogonal to each other.

Figure 2 demonstrates the orthogonalization process, where the robot observes two vertical lines $L1$, $L3$ and one horizontal line $L2$. The observation reference is determined from the line angle parameters $a1$, $a2$, $a3$. The segments are then rotated

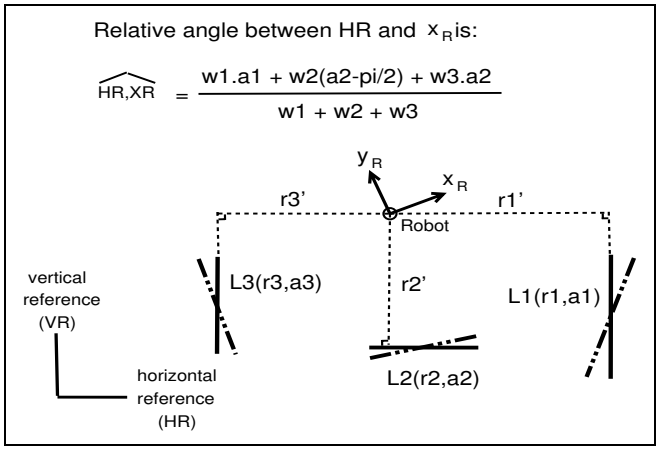

Fig. 2. The Orthogonalization process: The orientation reference of the observation is determined. The observed lines (dashed lines) are then rotated to align with the reference orientation, thus being orthogonal to each other (solid lines).

and new line parameters are computed as $r1'$, $r2'$, $r3'$ (the corrected angles are now shown). The corrected lines are now orthogonal.

Choosing midpoints of the segments as the rotating origins is because usually the line extraction algorithm mistakenly includes a few noisy raw scan points at the ends of line segments, causing the fitted lines being erroneously deviated by a small angle. This is what we are trying to correct in this step. Note that the orthogonalization is independent of the robot pose. It is obvious that the corrected lines are mutually correlated after the orthogonalization, however we do not consider the correlation as an approximation. One might think that asserting lines close to orthogonal as orthogonal would cause a problem. However, the fact that in indoor environments, orthogonal lines (e.g. walls) are usually much longer than lines from non-orthogonalized objects, taking the weighted mean in the Orthogonalization will minimize the error coming from spurious lines.

Calculation of Relative Distances between Lines

This simple procedure takes the lines in the current observation (after being processed through the orthogonalization) as input and returns the distances between vertical and horizontal lines as relative observations, separately. In each observation, relative distances between horizontal lines are computed from a horizontal *base line* which is selected among the observed lines, usually the one has the longest length, thus most reliable one. This is better than using all relative distances because if a line segment is erroneously extracted, it does not affect other distances but only the distance between the base line and itself. For the next observations, if a base line is re-observed, it is selected again, otherwise a longest one is selected as a new base line. If several base lines are observed simultaneously, the one has the smallest ID is selected. Similarly for vertical lines. Notice that the relative distances are independent of the robot motion since the base lines are computed/updated recursively using previously computed relative distances (the parameters of the base lines are obtained from the current Absolute Map).

Figure 3 gives an example. Observation 1 contains 5 vertical lines v_1, v_2, v_3, v_4, v_5. Line v_1 is selected as the base line. The distances $d_{12}, d_{13}, d_{14}, d_{15}$ are computed. In Observation 2,

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.

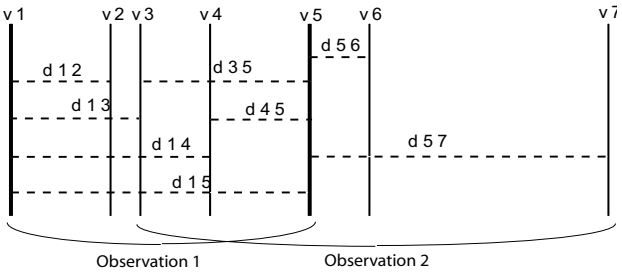

Fig. 3. The estimated state (relative distances) after 2 observations. The base lines are v_1 and v_5.

line v_5 is selected as the base line and the distances d_{35}, d_{45}, d_{56}, d_{57} are computed.

Estimation

As mentioned before, the lines in the map belong to two distinct and independent classes. Therefore, the estimation process is split into two separate procedures: the first one involves the estimation of horizontal lines, the second is the same implementation for vertical lines. Here the case of vertical lines is explained.

To make the estimation process independent of the robot motion, we estimate a state containing quantities which are invariant to the robot pose, namely invariant with respect to shift and rotation (as in [16]). This is an important issue since it makes the estimation process independent of the systematic error in the odometry and/or undetected events (like collisions) which can occur during the robot navigation.

An example of the relative map is shown in Fig. 3. After 2 observations, the state to be estimated contains the distances among the observed vertical lines:

$$X_d = [d_{12},\ d_{13},\ d_{14},\ d_{15},\ d_{35},\ d_{45},\ d_{56},\ d_{57}]^T$$

Clearly, the map state does not contain all the distances among the lines. In particular, as described above, only distances between the selected based lines and other lines are considered.

The estimation of the state is carried out using a Kalman filter. In this paper, we consider a constant uncertainty for all the observed lines. Furthermore, as an approximation we consider the elements in X_d uncorrelated. This makes the implementation very simple and in particular the computational requirement does not increase with the size of the environment, thus has constant time complexity.

In general, the map elements (i.e. distances) are correlated however. There is also some information coming from the geometric constraints among the elements of X_d. For example, one can observe from the Fig. 3:

$$d_{13} + d_{35} - d_{15} = 0 \quad ; \quad d_{14} + d_{45} - d_{15} = 0$$

In this specific case, X_d contains 8 elements while the number of independent distances is 6. It is possible to apply the linear projection filter [17] to integrate such information. Because of the linearity, the filter can be applied only once for ever (e.g. after the last observation step) without any loss of information. This holds even when the covariance matrix is not block diagonal. In this paper, the linear projection filter is not implemented, however the algorithm is still able to produce very good results.

Absolute Map Construction

It is straightforward to reconstruct the absolute map of the environment from the given x-coordinate of one vertical line and the given y-coordinate of one horizontal line together with the estimates of the relative distances (the relative map). The time complexity of the construction is linear on the number of lines.

To update the end points of the line segments, the union of the matched segments are calculated. This is not the best one can do but it performs well enough since end points are only used for the DA.

Updating Robot Configuration

The robot configuration is only necessary to accomplish the DA task. Therefore, it does not affect the estimation process unless there is an error in the data association. We apply the following simple heuristic method: The input of the procedure are the current observed lines (already associated with the lines in the map) and the updated absolute map. By using the updated absolute position of the observed vertical lines and their line parameter r relative to the robot, the absolute x-coordinate of the robot is obtained. In the case that no vertical lines are observed, the odometry information is used. Similarly, the absolute y-coordinate is obtained by using the observed horizontal lines. Finally, by using the observed orientations of both vertical and horizontal lines in the robot frame, it is possible to compute the robot absolute orientation.

Map Management

In order to reduce the computational complexity, a *local view* which contains a small part of the current Absolute Map around the robot is maintained. The diameter of this map region can be chosen as few meters larger than the range of the SICKs. The local view is constructed by selecting the features (line segments) currently in the map within the defined region. When a new line is observed, it is added to both the local view and the global map. As a result, the procedures Data Association and Absolute Map Construction performing on this local view both have constant time complexity. Constructing a local view requires a search in the global map, thus has linear time complexity on the number of lines estimated. However it can be performed periodically depending on the speed of the robot.

It usually happens that a line is extracted as several broken line segments because of noise, occlusions, etc. To overcome this problem, mapped lines in the local view are merged if they are collinear, proximate (determined by their line parameters r) *and* also overlapped (determined by their end points). The parameter r of the newly merged line is determined by computing the weighted average of the r's of the original lines. The weighting coefficients are proportional to the lengths of the original lines and the numbers of times that they are observed. This procedure is performed periodically every ten observation steps or when the local view is renewed.

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.

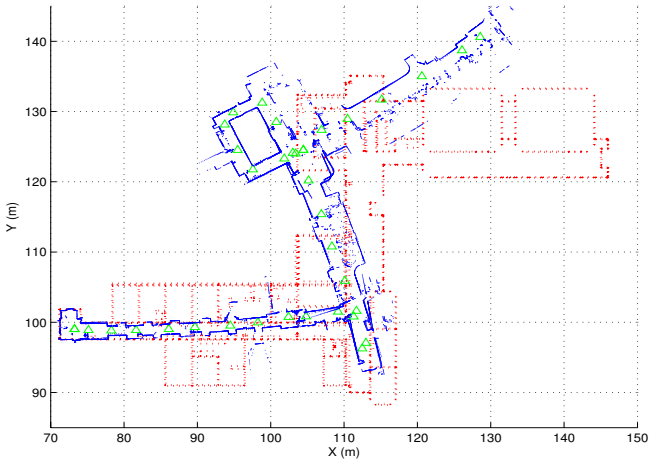

Fig. 4. The map built (blue dots) by using raw scan points and pure odometry information is corrupted by the odometry drift. The ground truth (including office rooms) is shown in red dotted lines.

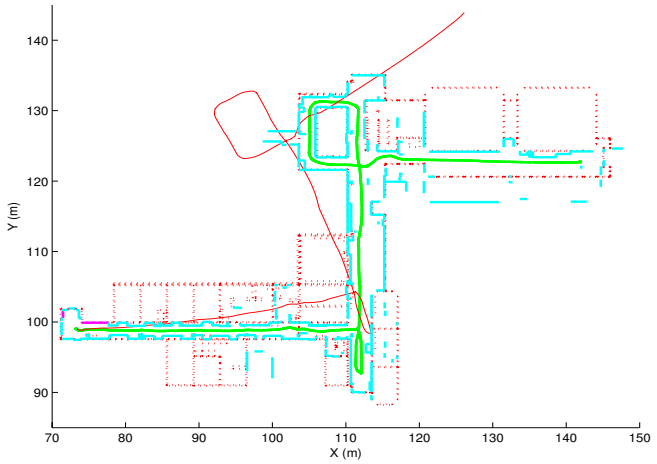

Fig. 5. The map built by using the *OrthoSLAM* algorithm on the Biba Forward run: the estimated lines (blue solid lines), the estimated robot trajectory (green curve in middle of the hallway) and the ground truth (red dotted lines).

IV. Experimental Results

For the experiments, we use 2 robots: the mobile base of the robot RoboX and the robot Biba (*http://www.asl.ethz.ch/*), both are equipped with 2 CPUs, 2 SICK laser range finders. The robots are running a real-time operating system (RTAI Linux) with an embedded obstacle avoidance system and a remote control module via wireless network. Combination of 2 laser scanners enables the robots to scan a full $360°$. In our experiments, we use a maximum scan range of $7.0m$, an angular resolution of $0.5°$ and a sampling rate of 4Hz. We choose our laboratory hallway which is a typical office environment as the testing zone with a map size of $80m \times 50m$. There are walls, doors, cupboards and also glass windows, table, chair and objects lying around. The experiments are carried out during the working hours so that there are lots of people moving around. The average speed of the robots is about $30cm/s$ excluding the time that they are stationary.

We perform three experiments: two *forward* runs (one for each robot) in which the robots navigates from one end of the hallway to the other end; and one *return* run (the way going back) for the Biba robot. Each run contains two small loops and $3000 - 5000$ observation steps. A sample map of raw scan points obtained using pure odometry information is plotted in Fig. 4. One can see that the built map is corrupted by the odometry drift. Moreover, the odometry error cumulates so that when the robot arrives to the other end, it has been totally lost by approximately $20m$ from the true position.

The *OrthoSLAM* algorithm is tested on the three data sets and it generates almost identical maps. One of the resulting map is shown in Fig. 5 together with the ground truth. It can be seen that the shape and orientation of the estimated map is matched precisely with the ground truth. This is as theoretically expected result. Since by using the fact that the orthogonality assumption holds for this environment, we do not have any orientation error in the estimation. Regarding the metric size/position, the estimated lines are in general

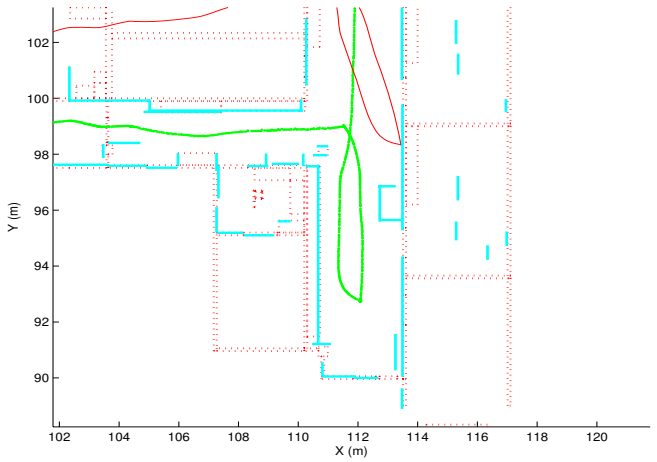

Fig. 6. A zoom-in part of the built map: the estimated lines (blue) and the true lines (red dotted lines) are nearly overlapped. Note that openings and curved, occluded lines are not mapped. Some true lines are doubled showing 2 sides of the walls. There are also new objects placed in the environment which are not included in the ground truth.

overlapped with the true lines. Figure 6 depicts a zoom-in part of the built map in order to show the precision of the map in a larger scale. One can see that the obtained map is very precise in the order of few cm. There are estimated line segments that do not appear in the ground truth representing the new cupboards. Reversely, open doors, glass windows, curved walls and occluded walls do not appear on the estimated map.

The *OrthoSLAM* algorithm is performed on a labtop with a PentiumM-600MHz using the logged data of the three experiments. In term of computational performance, the algorithm is able to run at 40Hz without any special optimization which is 10 times faster than the input data rate. Table I summarizes the results obtained from the three experiments and gives also a comparison with the ground truth. Nearly 300 line segments are estimated. There is a slight difference in the

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.

TABLE I

SUMMARY OF THE EXPERIMENT RESULTS.

Experiment	N. Lines	Obtained Map size	Error	Accuracy
Robox Forward	292	$73.43m \times 45.02m$	$\pm 10cm$	99.8%
Biba Forward	287	$73.62m \times 44.96m$	$\pm 13cm$	99.7%
Biba Return	285	$73.59m \times 44.84m$	$\pm 9cm$	99.8%
Ground-truth	-	$73.52m \times 44.97m$	$\pm 5cm$	-

number of orthogonal lines mapped in the three experiments. This is mainly because of the dynamics of the environment, e.g lines extracted from opening/closing doors, objects inside office rooms, etc. Note that we do not have the correct number of true lines, since for example walls get broken by occlusions of objects lying around and the observability is dependent on the position of the moving robot. In term of metric quality of the estimated maps, the algorithm is able to achieve an accuracy of 99.7% for all the three experiments with an absolute uncertainty of around $10cm$ which is similar to that of the hand measured map. We believe that it is a very good result for such a big indoor environment.

V. CONCLUSION

The paper has presented the *OrthoSLAM* algorithm as a lightweight solution to SLAM in indoor office-like environments. The main contribution of this algorithm resides in the idea of reducing the complexity by means of an assumption on the environment structure. This is done by mapping only lines that are parallel or perpendicular to each other where each of them can be represented by one variable (ρ). Hence, the mapping is reduced to a linear estimation problem and is solved by a Kalman filter. The combination of this approach with a relative map leads to a robust, lightweight SLAM algorithm. The algorithm is capable to run real-time on a normal computer. Only the Management module has linear time complexity and can be performed periodically. All other modules have constant time complexity.

The presented algorithm has been validated by extensive experiments where our laboratory hallway with the size of $80m \times 50m$ and the robot trajectory of more than $500m$ have been mapped consistently. The precision of the resulting map is similar to the measurements done by hand which are used as the ground-truth.

The ultimate goal of this research is to create a simple, robust and lightweight approach to perform real-time SLAM for indoor, office-like environments that has many potential service robotic applications. By making a simple assumption about the structure of the environment, the Orthogonality, we can avoid many problems of linearization and achieve a better estimate of the robot heading angle. Being more certain on the robot orientation, we neglect the correlations between the relative map state and further simplify the computation. We believe that this assumption holds for most of the indoor environments.

The main distinction of this approach relies in the way the complexity is reduced. While the other approaches modify the basic algorithm by some assumptions and approximations, we propose to use a constraint of the environment. This allows keeping a complete coherent approach that now is actually applied to a simplified problem. This results in a better performance for all cases that do not violate our orthogonality assumption. Comparing our approach with a EKF based SLAM algorithm or FastSLAM is one of our future work. We plan also to implement a 3D version of this algorithm for indoor environment.

ACKNOWLEDGMENTS

This work has been supported by the Swiss National Science Foundation No 200021-101886 and the EU project Cogniron FP6-IST-002020. We would like to thank Frédéric Pont for the support in carrying out the experiments.

REFERENCES

[1] 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.

[2] 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.

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

[4] P. Newman, J. Leonard, J. 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.

[5] J. J. Leonard and H. J. S. Feder, "Decoupled Stochastic Mapping," *IEEE Journal of Oceanic Engineering*, pp. 561–571, 2001.

[6] J. Guivant and E. Nebot, "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[7] S. Williams, "Efficient Solutions to Autonomous Mapping and Navigation Problems," Ph.D. dissertation, Uni of Sydney, Australia, 2001.

[8] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments," Ph.D. dissertation, University of Sydney, 2002.

[9] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *AAAI*, 2002.

[10] C. Kwok, D. Fox, and M. Meila, "Real-Time Particle Filters," *Proceedings of the IEEE, Special Issue on Sequential State Estimation*, vol. 92, no. 2, 2004.

[11] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M.Csorba, "A Solution to the Simultaneous Localisation and Map Building (SLAM) Problem," *IEEE Transactions on Robotic and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[12] J. Vandorpe, H. V. Brussel, and H. Xu, "Exact Dynamic Map Building for a Mobile Robot using Geometrical Primitives Produced by a 2D Range Finder," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 1996, pp. 901–908.

[13] J. D. Tardós, J. Neira, P. M. Newmann, and J. J. Leonard, "Robust Mapping and Localization in Indoor Environments Using Sonar Data," *Intenational Journal of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.

[14] E. Brunskill and N. Roy, "SLAM using Incremental Probabilistic PCA and Dimensionality Reduction," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[15] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics," in *Proceedings of IROS*, 2005.

[16] A. Martinelli, N. Tomatis, and R. Siegwart, "Open Challenges in SLAM: An Optimal Solution Based on Shift and Rotation Invariants," in *Proceedings of ICRA'2004*.

[17] P. Newman and H. Durrant-Whyte, "A New Solution to the Simultaneous Localisation and Map Building (SLAM) Problem - the GPF," in *Proceedings of the SPIE conference*, 2001.

Authorized licensed use limited to: HANGZHOU DIANZI UNIVERSITY. Downloaded on September 03,2021 at 06:03:19 UTC from IEEE Xplore. Restrictions apply.