Towards Geometric 3D Mapping of Outdoor Environments Using Mobile Robots

Denis Wolf, Andrew Howard and Gaurav S. Sukhatme

Robotic Embedded Systems Laboratory Center for Robotics and Embedded Systems Department of Computer Science University of Southern California, Los Angeles, California, USA {denis, ahoward, gaurav}@robotics.usc.edu

Abstract— This paper presents an approach to generating compact 3D maps of urban environments using mobile robots and laser range finders. Our algorithm extracts planar information from 3D point cloud maps. The planar representation is very efficient for representing building structures in urban environments when a high level of detail is not required. We also present preliminary results on 3D geometric mapping with incomplete data. Based on previously known models and incomplete data, our system is able to estimate parts of buildings which have never been seen before. As validation we present experimental results using a Segway RMP vehicle in two environments, both approximately the size of a city block.

I. INTRODUCTION

Mapping indoor environments using mobile robots is a well known problem which has been studied for the last two decades (see [2] for a summary). However, most approaches used to map indoor environments cannot be directly used in outdoor environments. Three of the most important aspects that make mapping outdoors a more challenging problem than indoor mapping are: environment representation, scale, and rough terrain.

Most algorithms for indoor mapping generate 2D floor plan-like maps, which can fairly represent walls and other similar indoor features and give a good idea of how the environment looks like. This kind of representation turns out to be very poor when we try to model outdoor environments, which usually have much richer features to be represented such as buildings, trees, and cars.

The second important factor to be taken into account outdoors is the scale of the environment. Most approaches for indoor mapping deal with rooms and corridors while outdoor maps need to scale to square kilometers. One of the frequently used indoor mapping representations is the occupancy grid [1]. While this method is suitable for representing 2D indoor spaces with good accuracy, it does not scale for outdoor spaces. For a 3D representation, one more dimension is added to the map, creating serious scaling limitations for practical use.

Finally, the terrain is normally flat indoors, this is not always the case outdoors. Irregular terrain, depressions and small rocks make the task of mapping a little bit more

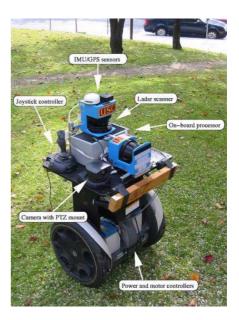


Fig. 1. Segway RMP: 2 laser scanners configuration.

challenging as they make the robot bump and change its direction, inducing errors in proximity sensors and corrupting odometric information.

Outdoor 3D maps have been addressed by the computer vision community for many years [7][6] and also more recently by the robotics community [8] [9]. The approach presented in [6] merges aerial images, airborne laser scans, and images and laser scans taken from the ground. A 2D map is extracted from the aerial images and used along with horizontal laser scans taken from the ground to estimate the robot position during the data acquisition. A Monte Carlo method is used to perform the localization task. Another approach for urban 3D mapping is presented in [5]. In their approach a laser range finder pointing up is attached to the roof of a vehicle. As the vehicle moves forward, 3D information is obtained. Feature matching is used to recover the correct position of the robot during the data acquisition stage. In the approach presented by [10], 3D geometric models with photometric texture mapping are obtained combining range information with 2D images taken from a mobile platform. In [16], a 3D representation of the environment is obtained from images taken from different poses. In [11] 3D maps are built from the range information provided by an helicopter.

Many different methods can be used to represent outdoor environments. A point cloud is one of the most frequently used representation methods. It can describe features in fine detail when a sufficient number of points is used. On the other hand, this method is not memory efficient as it is necessary to store large amounts of data for a detailed representation of large environments. An efficient mapping representation method is to approximate surfaces by planes. This method has been successfully used in indoor environments [12] [13]. Outdoor environments are not always structured, therefore it is somewhat more difficult to extract planes when we have objetcs like trees, cars, and people in the scene.

In our approach, we keep very compact geometric representations for buildings; basically each wall is approximated by one plane. Although these very simple geometric maps do not present the same level of detail as some of the other approaches cited above, they have the advantage of being highly memory efficient. Applications like observability calculation, path planning, and visualization of large spaces from far away vantage points do not require a high level of detail and the efficiency provided by our approach is very convenient.

In the approach presented here, laser range finders are attached to a Segway RMP (Figure 1). The Segway is a two-wheeled, dynamically stabilized vehicle based on the Segway HT. This platform is very convenient for outdoor experiments. It is fast, has good endurance, and can support large payloads. On the other hand, it pitches considerably during acceleration and deceleration, which must be taken into account during the localization and mapping tasks. Our mapping algorithm has 3 steps: (1) generating a point cloud map based on odometry, inertial measurement unit (IMU), GPS, and range information, (2) extracting planes from the point cloud map, and (3) associating planes and geometrically represent buildings. We also present some preliminary results on geometric modeling of buildings based on incomplete information. Based on previously known models, our algorithm is capable of estimating parts of buildings that have not been seen by the robot.

II. POINT CLOUD MAPPING

The point cloud generation algorithm used in this paper is described in [3]. This is the first step in our mapping approach. It is the base for the plane segmentation and building modeling.

Point cloud maps can be generated fairly easily when good pose estimation and range information are available. Accurate pose estimation is not a trivial task in outdoor urban environments. The position information provided by the robot's odometer is subject to drift. Over long runs, the error grows unbounded causing the information provided by the odometer to be completely useless. The mobile platform

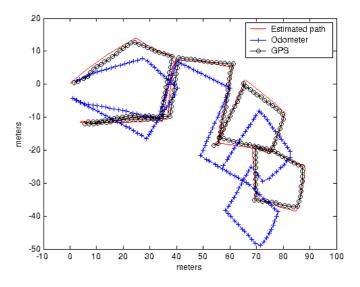


Fig. 2. Localization on the Ft. Benning site.

used during the experiments, a Segway RMP, has some particularities that make pose estimation a little harder. The RMP is a two-wheeled, dynamically stabilized vehicle. Small differences in the tire pressures can produce poor odometry and small rocks and irregularities on the terrain cause the robot to bump and change its orientation. We have used two pose estimation methods in our experiments.

The robot's pose estimation used during our experiments on the USC campus [3] is based on the combination of fine and coarse scale localization. Fine localization is provided by the robot's internal odometer, IMU and scan matching. Coarsescale localization is provided by GPS information and Monte Carlo Localization (MCL) [14] [15]. GPS is a convenient localization method for outdoor experiments, but is at times unavailable due to satellite occlusions by tall buildings. MCL has been used as an alternative coarse-scale pose estimation method. MCL requires a previous 2D map of the environment though.

During our experiments at Ft. Benning, we did not use any a priori information about the site. The pose estimation method used in this case consists of approximatING the information provided by the odometer and IMU to GPS points using a particle filter technique. The action model for the particles is based on the odometer and IMU (plus some random gaussian error). The observation model is based on how distant the particles are from the GPS points. Particles with distance from GPS above a certain threshold are not resampled. The complete history of every particle is stored in memory. At convergence, only particles within a reasonable distance to the GPS points survive and the trajectory of any of those particles is a good approximation of the path followed by the robot.

Figure 2 shows GPS data, odometer, and robot's estimated path for the Ft. Benning site. Although the estimated path for the robot approximates very well to the GPS information, there are some errors in the pose estimation because the precision of the GPS unit we have been using is about 2m.

Both USC and Ft. Benning maps were generated offline but in real-time (the time taken to generate the map is significantly less than the time taken to tour the environment).

III. PLANAR SEGMENTATION

Point cloud maps are relatively easy to build using range sensors once reliable, accurate robot pose estimation is available. These maps are detailed and can capture small features in the environment. However, this type of representation is not memory efficient once a large number of points is required to represent the environment. Depending on the application, more efficient data representations methods can be used. Representing flat surfaces by planes is an efficient way to represent walls for example. Applications like observability calculation, path planning, and visualization of large spaces from far away points do not require a high level of detail and the efficiency provided by our approach is very convenient.

Extracting planar information from a set of 3D points is an optimization problem that consists of finding a set of planes that best fits a given set of points. This problem has been studied by the computer vision community for decades with many different approaches [19] [20]. More recently, this research topic has also been studied by the robotics community [17].

Our approach is based on the Hough transform [18]. The Hough transform is a technique, which can be used to extract features from a set of points. The classical application for the Hough transform has been detecting geometric features like lines and circles in sets of 2D points. This algorithm can be also be extended to work in 3D spaces and with more complex features like planes. The Hough transform algorithm consists of examining each point and finding all the possible features that fit that point. Finally, it is possible to recover the features that fit the larger number of points. Differently from other fitting techniques, which just approximate features to points, the Hough transform can handle cases in which multiples features can fit the points and some features cannot fit at all.

A plane in 3D Cartesian space can be expressed as:

$d = x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta$

where the triplet (d, θ, ϕ) defines a vector perpendicular to the plane. The distance d represents the size of this vector and the angles θ and ϕ the orientation of the vector from the origin. The Hough transform converts a plane in 3D space to a $d - \theta - \phi$ point.

Supposing we have a co-planar set of points in 3D and we are interested in finding the plane that fits all of the points in the set. For a specific point P_0 (x_0, y_0, z_0) in the set, it is possible to plot a curved surface in the $d - \theta - \phi$ space that corresponds to all planes that fit P_0 . If we repeat this procedure for all points in the set, the curves in the $d - \theta - \phi$ space will intersect in one point. That is because there is one

plane that fits all the points in the set. That plane is defined by the value of d, θ, ϕ on the intersection point.

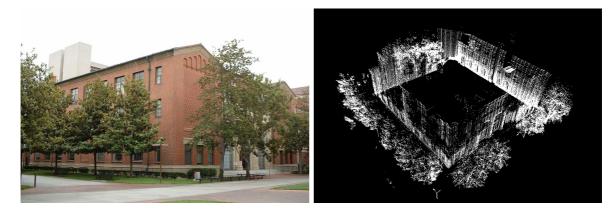
However, there are some small modifications to the algorithm that make the implementation much easier and faster, and as a tradeoff, the results obtained are less accurate. The $d - \theta - \phi$ space can be represented as a 3D array of cells and the curves in that space are discretized. Each cell value is increased by 1 for every curve that passes through that cell. The process is repeated with all curves in the $d - \theta - \phi$ space. At the end, the cell that accumulated the highest value represents the space that fits more points. The size of the grid cells corresponds to the rounding of the value of the $d - \theta - \phi$ parameters to represent a plane. The smaller the grid cells, the more accurate the parameters that describe the plane.

In the case one wants to extract more than one plane from a set of points, every plane for which the corresponding cell value is above a determined threshold is considered a valid plane. This is the case in our particular implementation. Since we are fitting planes to point cloud maps, there are cases where there are one or more planes in a set of points. There are also cases where there are no planes at all, when the points correspond to non-planar parts of the environment. Figure 3c shows planes extracted from the accounting building on the USC campus, using this technique.

IV. GEOMETRIC REPRESENTATION OF BUILDINGS

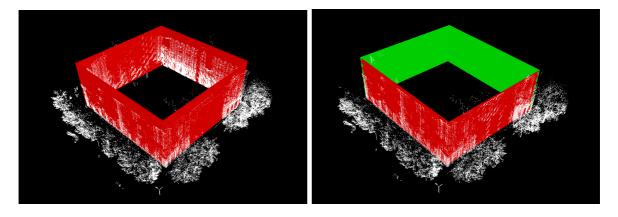
As described in section 2, point clouds are a very detailed representation for 3D environments. The level of detail is proportional to the number of points used the represent the features in the environment. Thus, this representation method can be memory inefficient when we are trying to map large areas. In our case, we are more interested in the efficiency and compactness of the representation rather than a high level of detail. We are interested in representing buildings in the environment, and since buildings are usually composed of large walls they can be efficiently approximated by planes. For example, a rectangular building can be approximated by 8 points in 3D space. It is also important to mention that this approximation implies a considerable lose of details.

Extracting planes from indoor environment point clouds is a relatively easy task once most of the internal parts of built structures are flat surfaces. In outdoor urban environments it can be much harder due to the presence of various elements that are not part of buildings like trees, bushes, people, and vehicles. On many occasions, those obstacles are close to the robot blocking its sensors and causing occlusion, which makes the plane extraction harder. For example, when an obstacle is positioned between the robot and a wall there will be a shadow (absence of points) on the representation of the wall. The closer the obstacle is to the robot, the larger is the shadow on the wall. The presence of non-building obstacles and the effect of occlusion make the task of extracting planar information from point cloud maps more difficult. Another issue in extracting planes from 3D points is that far away points originated by completely different features may align, inducing the presence of large planar structures. In order to



(a) Actual building

(b) Point cloud



(c) Complete data model

(d) Incomplete data model

Fig. 3. Geometric Modeling of the USC Accounting Building.

handle these situations, our approach divides the point cloud into small sections of 3D points before using the technique presented in the previous section of this paper. These sections overlap each other to guarantee that plane surfaces are not broken into two pieces. As a result we have a large set of small planes.

After extracting planes from building structures, it is necessary to combine them in order to represent building structures. On our implementation we make the assumption that valid building walls are vertical or horizontal with a small tolerance (horizontal planes allow us to represent roofs when the robot is underneath them). This assumption simplifies not only the extraction of planes but also their combination. With few exceptions, this assumption holds for most cases in our experiments. As a result of this assumption, the search space on the Hough transform will be decreased making the plane extraction process faster and the association of planes easier.

The algorithm proposed by [4] has been used to combine planes. This algorithm efficiently merges line segments (in our case planes) that are close of each other and have similar directions. It handles both overlapping and non-overlapping segments and weighs (calculate the relative importance) the segments according to their size.

The situation where the robot is not able to visit all the sides of a building is frequent. It may happen because some places are not accessible to the robot or there are considerable occlusions. The information acquired by the robot's sensors is significantly incomplete. For example, only two sides of a rectangular building structure have been measured. In these cases, our approach is capable of generating representations for incomplete data. It handles cases where occlusions make useful data unavailable for the sensors (part of walls cannot be seen) and cases where large part of building information is not available. In the first case, a small virtual segment can be added to compensate the part of the building that has been missed by the sensors. In the second case, our system is capable of guessing how the entire building would look based on the incomplete information provided and some previously known models. Virtual walls are created to complete the building representation. Of course, there is no guarantee that estimated parts of buildings in fact match the real (unob-



(a) Aerial view (courtesy of UPenn GRASP Laboratory)

(b) Point cloud plus planar model

Fig. 4. Geometric Modeling of the Ft. Benning site.

served) parts. As more information is acquired by the sensors the estimate can be improved. So far we have worked on models for rectangle, L and T-shaped buildings, our algorithm tries to fit the partial data on these models in that order respectively. This capability allow us to generate an estimate of how a building would look like if seen from a place never previously visited by the robot. Figure 3d shows the accounting building model based on partial data (two walls). Our algorithm assumes that the building is symmetrical and estimates the unseen parts of the building.

V. EXPERIMENTAL RESULTS

In order to validate our approach, experiments have been performed on the USC campus and Ft. Benning/GA. The maps were plotted using a standard VRML tool, which allows us to virtually navigate on the map. It is possible to virtually go on campus streets and get very close to features like cars and traffic signs and it is also possible to view the entire map from the top.

At the USC campus the point cloud was generated using the approach discussed in section 2. The planar segmentation algorithm presented in section 3 was applied to the 3D point cloud map. After that, we applied our building modeling approach presented in section 4. During the mapping task, the robot made a complete loop around the USC accounting building. This example is particularly challenging because there were many trees and bushes between the robot and the building walls. The actual building can be seen in Figure 3(a) and the point cloud representation is shown in Figure 3(b). The planar model of the accounting building is shown in Figure 3(c).

We manually removed some information from the point cloud map in order to test how our algorithm could handle incomplete data. We performed the experiment with different levels of data incompleteness removing parts of walls, an entire wall, and two entire walls. Our system was able to create virtual walls that complemented the real data provided. In this example we previously had a model for a rectangular building (that is the shape of the USC accounting building). Therefore, given two or three walls with end points close enough to each other, our approach considered these walls part of a building and estimated the planes that would make a rectangular shaped building. Given the partial data available our algorithm fitted the partial information available to a rectangular shape. It is important to mention that this approximation may not be correct when the available information is poor, and in this case the estimated parts would not match the real unobserved parts of the building structure. The virtual planes for the USC accounting building can be seen in Figure 3(d) on lighter color (real planes are colored darker). It is also important to mention that the plane corresponding to the roof of the building has been removed in order to facilitate the visualization of the walls.

During our experiments in Ft. Benning, the robot mapped an area of 50m X 90m (350m tour with an average speed of 1.2m/sec). An aerial view of a section of Ft. Benning, taken by an UAV developed at the University of Pennsylvania, GRASP Laboratory (under the MARS 2020 project collaboration) is shown in Figure 4a [21]. A GPS unit (with accuracy 2m) was used as reference for the robot's pose estimation. The pose estimation error can be noticed in the walls of some buildings, which appear bent in the point cloud map (Figure 4b). These misaligned points make the planar segmentation task harder, but our approach was efficient enough to extract planes from those points. Although the buildings are rectangular shaped, some of the building models are not perfectly rectangular due to the misalignment of the points.

The robot made complete loops around some buildings, but not for all them. For the buildings that only partial information was available, the unseen walls have been estimated and completed the model. They are shown in lighter color in the Figure 4b. Some buildings have their planar model much smaller than their actual size due to the lack of information. In those cases, the robot did not colleted enough data to build complete models for those buildings.

Unfortunately, ground truth was not provided during these experiments, but visual comparisons between the aerial image and the planar model suggest errors around 2m, which is compatible with the GPS error. A better reference for pose estimation would certainly lead our algorithm to generate more accurate models of the environemnt.

VI. CONCLUSION AND FUTURE WORK

This paper describes a method to build representation of 3D urban environments. We addressed different problems that are part of the task of mapping outdoors. Our algorithm scales efficiently in large environments and our localization approaches manage situations when previous information about the environment is provided and also when it is not.

Our approach is capable of creating point cloud maps based on sensor information provided by the robot. This type of representation proves to be very efficient in capturing details of the environment, but on the other had it is also very memory inefficient once a considerably large number of points are required. As an alternative we represented flat surfaces found on point cloud map by planes. These planes do not possess the same level of details of the point clouds but they are memory efficient. When the application of the urban maps does not require fine level of details, planar information may be a convenient alternative.

We also presented initial results on building modeling with incomplete data. Our approach assumes that the planar information available fits into a previously known model and in this case it is possible to estimate the unobserved parts of building structures. As more information is available for our algorithm, better the chances of the right building model be chosen and better the estimated parts (virtual parts) match the unobserved parts of building structures.

Experimental results at USC campus and Ft. Benning were shown to validate our algorithm. Models from both complete and incomplete data have been presented. The experiments in Ft. Benning were particularly challenging due to roughness of the terrain.

As future work we plan to investigate different methods to map urban environments that can be efficient in representing large environments and have better level of details. Combining range information with images is one of the possibilities we are investigating. We also plan to combine range information provided by not only ground robots but also by helicopters.

VII. ACKNOWLEDGEMENTS

The authors thank Boyoon Jung and Marin Kobilarov for their valuable support during the experiments. This work was supported in part by the DARPA MARS program under grants DABT63-99-1-0015 and 5-39509-A (via UPenn), NSF CAREER award IIS-013394, and by the grant 1072/01-3 from CAPES-BRAZIL.

REFERENCES

- Elfes, A "Sonar-based Real-World Mapping and Navigation," *IEEE Transactions on Robotics and Automation*, 3(3):249-265, 1986.
- [2] Thrun, S., "Robotic Mapping: A Survey," In G. Lakemeyer and B. Neberl, editors, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.
- [3] Howard, A., Wolf, D. F., Sukhatme, G. S. "Towards Autonomous 3D Mapping in Urban Environments" In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 419-424, Sendai, Japan, Sep 2004.
- [4] Tavares, J. M. R. S., Padilha, A. J. 'A New Approach for Merging Edge Line Segments," *In Proceedings of 7th. Portuguese Conference of Pattern Recognition, Aveiro, Portugal, 1995.*
- [5] Zhao H., Shibasaki, R. 'Reconstructing Urban 3D Model using Vehicleborne Laser Range Scanners," In Proceedings of the 3rd. International Conference on 3D Digital Imaging and Modeling, pp. 349-356, Quebec, Canada, 2001.
- [6] Fruh, C., Zakhor, A. 'Constructing 3D City Models by Merging Aerial and Ground Views," In IEEE Computer Graphics and Applications ,23-6, 2003.
- [7] Hu, J., You, S., Neumann, U. 'Approaches to Large-Scale Urban Modeling," In IEEE Computer Graphics and Applications, pp 62-69, 2003.
- [8] Thrun, S., Burgard, W., Fox, D. 'A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping.' In Proceedings of the IEEE International Conference on Robotics and Automation, pp. 321-328, 2000.
- [9] Brenneke, C., Wulf, Oliver, Wagner, B. 'Using 3D Laser Range Data for SLAM in Outdoor Environments.' In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 188-193, Las Vegas, USA, 2003
- [10] Reed, M. K, Allen, P. K. '3-D Modeling from Range Imagery: an Incremental Method with a Planning Component.' In Image and Vision Computing 17, pp 99-111, 1999
- [11] Thrun S., Diel, M., Haehnel, D. 'Scan Alignment and 3D Surface Modeling with a Helicopter Platform," In Proceedings of the International Conference on Field and Service Robotics, Lake Yamanaka, Japan, 2003.
- [12] Thrun, S., Martin, C., Liu, Y., Haehnel, D., Montemerlo, R. E., Chakrabarti, D., Burgard, W. 'A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots," *In Transactions on Robotics and Automation, PP* 433-442, 2004.
- [13] Weingarten, J. W., Gruener G., Siegwart, R. 'Probabilistic Plane Fitting in 3D and an Application to robotics Mapping," *In Proceedings of the IEEE International Conference on Robotics and Automation, pp* 927-932, New Orleans, USA, 2004.
- [14] Dellaert F., Fox, D., Burgard, W., Thrun, S. 'Monte Carlo Localization for Mobile Robots," *In Proceedings of the IEEE International Conference on Robotics and Automation, pp 99-141, 1999.*
- [15] Thrun, S., Fox, D., Burgard, W., Dellaert, F. 'Robust Monte Carlo Localization for Mobile Robots," *In Artificial Intelligence, vol. 128 no.* 1-2, pp. 99-141, 2001.
- [16] Bosse, M., Newman, P., John, L., Soika, M., Feiten, W., Teller, S. 'An Atlas Framework for Scalable Mapping," In Proceedings of the International Conference on Robotics and Automation, pp 1899-1906 , 2004.
- [17] Haehnel, D., Burgard, W., Thrun, S. 'Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot," *In Robots and Autonomous Systems 44, pp. 15-27, 2003.*
- [18] Guil, N., Villaba, J., Zapata, E. 'A Fast Hough Transform for Segment Detection," In IEEE Transactions on Image Processing, vol. 4, no.11, 1995.
- [19] Bauer, J., Karner, K., Schindler, k., Klaus, A., Zach, C. 'Segmentation of Building Models from Dense 3D Point-Clouds," *In Workshop of the Austrian Association for Pattern Recognition*, pp.253-259, 2003.
- [20] Bartoli, A. 'Picewise Planar Segmentation for Automatic Scene Modeling," In IEEE International Conference on Computer Vision and Pattern Recognition, pp.283-289, 2001.
- [21] UPenn GRASP Laboratory 'Aerial view of Ft. Benning, taken by an UAV" *Personal Communication*.