6D-Slam with Navigable Space Discovering

Boutine Rachid
Department of computer science 20 august 1955 university, Skikda, Algeria
Boutine_rachid@yahoo.fr

Benmohammed Mohamed
LIRE Laboratory Mentouri university, Constantine, Algeria
Ben_moh123@yahoo.com

***Abstract*—For all mobiles robots that navigate in unknown outdoor environments two basic tasks are required: discovering navigable space, and obstacles detection. In this work, we have proposed a new variant of the 6d-slam framework, wherein planar patches were used for representing navigable area, and 3d mesh for representing obstacles. A decomposition of the environment in smaller voxels is made by an octree structure. Adjacent voxels with no empty intersection of their best fitting plans should be piece together in the same navigable space; otherwise adjacent non planar voxels would form the seed of potential obstacles. To create global map we take obstacles as natural landmarks, it is however necessary to find the correspondences between landmarks by use of a Bhatacharayya distance. Our experiments demonstrate the functionality of estimating the exact pose of the robot and the consistence of the global map of the environment.**

***Index Terms*—6d-Slam, Bhatacharayya distance, Dot operation, Octree**

I. Introduction

Major efforts and publications of robotic community in last decade are oriented toward the development of new generation of robots able to navigate in outdoor environment, although the GPS signal is poor, or missing.

The apparition of new kinds of 3d sensors as 3d laser telemeters, time of flight, and kinect cameras; have contribute in the development of more sophisticated robotic applications.

Our contribution in this work is the proposition of new variant of 6d-slam framework, wherein, we split the task of mapping in two sub tasks: obstacles detection, and navigable space discovering.

We will demonstrate that this separation contributes in the outcome of other robotic sub tasks like: localization, path planning, and obstacle avoidance.

Lot of domains can benefit from the results of this work, such as mapping underground mines, industrial automation, unmanned transportation, disaster rescue mission, etc.

This paper is organized as follow: in first section we start by an introduction, in the second section, we make a survey on the most important related works, in the third section, we present a new obstacle detection algorithm, in the fourth section, we present a solution for navigable space growth problem, and in the fifth section, we propose a novel association method, in the sixth section, we present the proposed 6d-slam variant, finally in the last section, we give the experimental results.

Manuscript received October 20, 2012; revised December 25, 2012.

II. Related Work

In the last decade, a great deal of latest works are oriented towered 6D-Slam, like the work of Paloma de la Puente and all [1], [2], where they present a segmentation and fitting algorithm of 3d points clouds, based on computer vision techniques, wherein a least-squares fitting, and a maximum Incremental probability algorithm formulated upon the Extended Kalman Filter, were used to outcome a position in 6D, and a map of planar patches with a convex-hull.

In literature there exist two kinds of map: The first kind is topological maps, which consists of a set of scans connected by a network of edges in a tree fashion style, like the work of D. Borrman [3], where a graphslam method is presented, a sparse network to represent the relation between several overlapping 3D scans. The second kind is metric maps, which are more reliable to represent geometrical features of the environment, either free, or occupied space.

More specifically in metric maps there are two main approaches: firstly, using feature extracted map, which allows the reduction of the amount of data stored in robot memory, nevertheless it suffers from information loss, secondly, the use 3D raw data directly, although the huge storage space required like the work of R. Schnabel in [4]. Elevation map was proposed by W. Miao and all, [5] to represent the environment as digital elevation. Also, the particle filter on multilevel space was presented by M, V. Prieto, and all in [6], or the work of K. Rainer, which is cited in [7].

Furthermore, M, P. Imtyas have present a new representing model called GP (Gaussian process model), where the robot uses this model to localize while traversing each environment [8]. On the other hand, textured mesh is proposed by C. Beall in [9] where a technique for large scale sparse reconstruction of underwater structures is presented; this technique estimates the trajectory of camera, and 3D feature poses.

doi: 10.12720/joace.1.2.78-81

An alternative way was the split of the 3d raw data, in clusters of low resolution, such as voxel that divide the space in cubic units, like the work of T. Suzuki and others in [10]. Other researchers have proposed a recursive space decomposition method, similar to tree based representation, like the work of N. Fairfield, and all where an octree is used [11], to represent 3d environment, therefore a well distinction between occupied, free, and unknown area is possible.

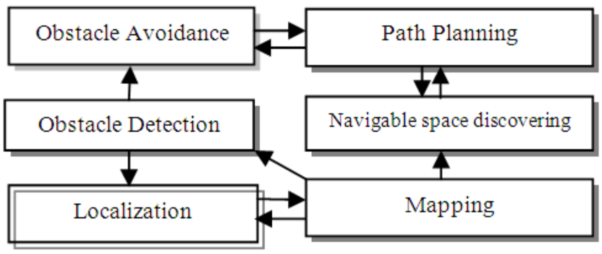


Figure 1. Sub tasks required for autonomous mobile robot in outdoor environment

III. Obstacles Detection

The main difficulty, when working with 3D maps arises from the cost of processing, and the storage requirements, that increase with map size and resolution. We propose the use of least squared plan fitting algorithm, for discovering navigable space which will be used later in path planning task, however non planar patches will be considered as obstacle features and can be used in localization or avoidance tasks, as mentioned in the figure N°01.

For each non planar leaf voxel in the octree, we apply an algorithm of Delaunay triangulation [12] to create a 3d mesh of this obstacle, using the following formula:

$$Del\ P = \{\sigma = Conv\ T| \bigcap_{p \in T \subseteq P} V_p \neq \varnothing\} \tag{1}$$

In other words, k+1 points are in Del P form a Delaunay k-simplex, if their Voronoi cells have non empty intersection.

On the center figure below, we can see a simple mesh of the seed map opposite to the robot, and on the right figure the same mesh from the side face.

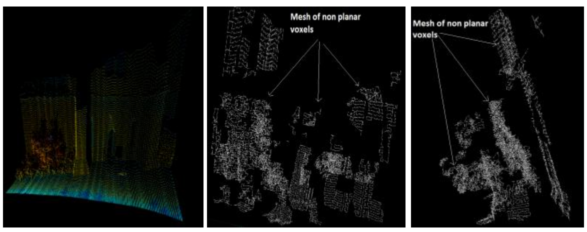


Figure 2. Left: Ground truth scan; Center :mesh of obstacle list opposite to the robot; right: the same mesh from side face

IV. Navigable Space Discovering

Two different strategies exist to reconstruct navigable space models from the detected planar patches. The first strategy tries to detect intersection lines and height jump edges; the second one assumes that all detected planar patches should model some part of the environment [13].

In our proposed method, we consider a navigable space for a mobile robot as a set of contiguous coplanar patches. A navigable space is created by merging a subset of adjacent planar leaf voxels, on which there exists at least an intersection boundary's line of their best fitting plans, according to the following algorithm:

Algorithm 1: NSD(octree)

Inputs: Octree of the scanned environment
For all children
If (children is leaf node) then { NSD(children) }
Else
{(p1, p2, p3, p4) =find the intersection vertices between the best fitting plan and the cubic voxel (see step 1 below).
For all neighbors voxels {If (MSE (neighbor) <σ and (neighbor ∩ $\overline{p1p2}$ or ∩ $\overline{p2p3}$ or ∩ $\overline{p3p4}$ or ∩ $\overline{p4p1}$ ≠∅)
then {Merge (children, neighbor) with elimination of the intersection boundary's.}}}

For discovering the entire navigable space we should execute the following four steps for all planar voxel:

1) Computing the component of the normal of the best fitting plan on which lies the most sub points cloud of this voxel[14]. To find the components of the plan normal's, we should minimize the distance di for the entire points pi belonging to the voxel; an appropriate minimization of di, is the least squares approximation, such that:

$$\sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} (Ax_i + By_i + Cz_i + D)^2 \tag{2}$$

2) Testing the planarity of this leaf node: using the mean squared error formula cited below to see if the major part of sub points clouds of this leaf node lies on the best fitting plan or not

$$MSE = \frac{1}{N} * \sum_{i=0}^{n} \left(\frac{|Ax_i + By_i + Cz_i + D|}{\sqrt{A^2 + B^2 + C^2}}\right)^2 < \sigma \tag{3}$$

where σ is experiment threshold, if MSE is lower than σ the leaf node is flat, otherwise it is not flat [2].

3) Fetching the intersection between the best fitting plan of the leaf node and the cubic voxel boundaries as depicted in the left figure below:

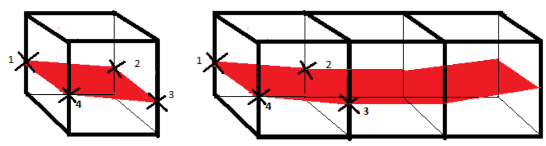


Figure 3. Left: Intersection between the best fitting plan and vertical boundaries of the voxel; Right Merging of best fitting plan of three adjacent voxels

4) Merge adjacent voxels if the intersection of their best fitting plans is not empty as illustrated on the right figure above.

We replace each sub points cloud in that voxels by a square patches superimposing the correspondent best fitting plan, contagious planar patches can be considred as the seed of free navigable space as on the right figure below:

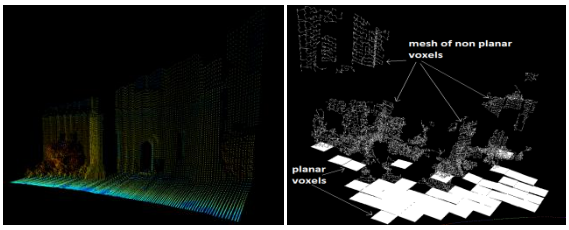


Figure 4. Left: ground truth; Right: Replacement of planar leaf nodes by square patches.

V. FEATURES MATCHING

The most algorithms of matching applied on 3D raw data mapping, are generally the ICP algorithm and its variants [15], where, the matching is done by approximating the solution to the best overlap between two scans [3], the Euclidian distance is used to estimate the correspondences between two scans, unfortunately the small rotations of the sensors are not considered in this distance; which yields large displacements as the distance increased.

Accepting a spurious or an incorrect matching will cause the process of slam to diverge [16]. In our work we have proposed the use bhatacharayya distance [17], as a measure of similarity between two histograms Q, and V of two voxels, we define this distance by following formula:

$$Dist_{bha} = 1 - \sum_{i=0}^{n} \sqrt{Q_i} . \sqrt{V_i} \tag{4}$$

where, n is the number of bins in each histogram, experimentally n=1000;.For each point in the sub points cloud of non planar leaf node, we calculate the local normal to this point, thereafter an histogram will be computed, on dot operation between each couple of normal vectors.

VI. 6D-SLAM

In order to create a large map, we align successive scans [11], non planar voxels are used to find correct matching by the minimization of a bhatacharayya distance between two voxels.

We should find a least squared transformation (T), from scan points cloud to map points cloud, so that the error (E) will be minimized

$$E = \sum_{i=0}^{n} \| T.scan_i - map_i \| \tag{5}$$

In our proposed method, two stages are executed iteratively:

1) Propagation Stage

In this stage

- New measurements are acquired to form the scan points cloud.
- Decomposition of the points cloud by an octree .
- Non planar voxels are extracted and matched with the global map.
- Search of possible matching couples

2) Correction Stage

- Computing the rigid transform, in order, to find the optimal rotation, and translation
- Using the rigid transformation, to correct the position of the robot, and the alignment of the environment's map, by the following two formulas :

$$X_v^{t+1} = X_v^t . T \tag{6}$$

$$Scan^{t+1} = Scan^t . T \tag{7}$$

VII. EXPERIMENTAL RESULTS

To demonstrate the robustness of the proposed method, we have applied it on the newcollege dataset. We have decide to decompose the whole points clouds into a set of smaller sub points cloud; Iteratively, we align the seed and the scan; thereafter the robot position can be corrected. We repeat this process until the exhaustion of the dataset, the following figure illustrates the decomposition of initial point clouds, and the next scan.

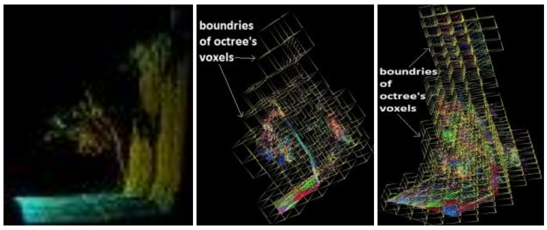


Figure 5. Left: ground truth; Decomposition by an octree center : seed map, right next scan

In the left figure below, the association couples between the seed map and the next scan are mentioned by straight yellow lines. Association couples of voxels are used to merge the two point clouds as in the middle figure, at right figure a global merged map of successive scans:

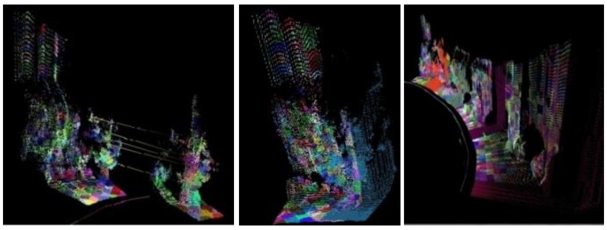


Figure 6. Left :matching of the the two points clouds , middle : the two points clouds merged, right: global map after many iterations

VIII. CONCLUSION

In this article, we have used a 3d modeling structure called octree, in order to simplify the mapping of the environment, and in the same time we have proposed the bhatacharayya distance for doing the matching, between two points clouds.

Our contribution is the proposition of new variant of 6D-SLAM for mobile robot in outdoor environment.

The correction of the environment's map and the position of the robot are realized in the same time by the operation of alignment between the map and the new acquired scan.

The resulting map can be separated on two sub maps: the first sub map contains obstacles of the environment, which can be used either for localization or for obstacle avoidance, and the second sub map represents free navigable space, within the robot can navigates, and executes path planning algorithm.

To validate our method, we have tested it on real 3D dataset; the first results are very encouraging.

In perspective, we intend to use matching with sliding window, to decrease the time of matching, and we can use the RRT algorithm to find a possible path in the free navigable space.

REFERENCES

[1] P. de la Puente, D. R. Losada, A. Valero, and F. Matia, "3D Feature Based Mapping Towards Mobile Robots' Enhanced Performance in Rescue Missions," in *Proc. International Conf. Intelligent Robots and Systems*, 2009, pp. 1138-1143.

[2] P. de la Puente, D. R. Losada, R. López , and F. Matía, "Extraction of Geometrical Features in 3D Environments for Service Robotic Applications," in *Proc. The 3rd International Workshop. Hybrid Artificial Intelligence Systems*, 2008, pp. 441-450.

[3] D. Borrmann, J. Elseberg, K. Lingemann, A. Nuchter, and J. Hertzberg, "The efficient extension of globally consistent scan matching to 6 Dof," in *Proc. the 4th International Symposium. 3D Data Processing, Visualization and Transmission*, 2008, pp. 29-36.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point cloud shape detection," in *Proc. Computer Graphics Forum*, 2007, pp. 214-226.

[5] M. Wang, and Y. H. Tseng, "Automatic 3D feature extraction from structurlized LIDAR data," in *Proc. 26th Asian Conference on Remote Sensing*, 2005, pp. 7-11.

[6] V. Prieto-Marañón, J. Cabrera-Gámez, A. C. Domínguez-Brito, D. H. Sosa, J. I. González, and E. F. Perdom, "Efficient plane detection in multi level surface maps," *Journal of Physical Agents*, vol. 5, no.1, 2011.

[7] R. Kummerle, R. Triebel, P. Pfaff, and W. Burgard, "Monte carlo localization in outdoor terrains using multi-level surface maps," *Journal of Field Robotics*, pp. 346-359, 2008.

[8] M, P Imtyas, "Indoor environment mobile robot localization," *International Journal on Computer Science and Engineering*, vol. 2, no. 03, pp 714-719, 2010.

[9] C. Beall, B. Lawrence, V. Ila, and F. Dellaert, "3D reconstruction of underwater structures," in *Proc International Conf. Intelligent Robots and Systems*, 2010, pp. 4418-4423.

[10] T. Suzuki, Y. Amano, and T. Hashizume, "6-DOF Localization for a Mobile Robot Using Outdoor 3D Point Clouds," *Journal of Robotics and Mechatronics*, vol. 22, no. 2, pp. 158-166, 2010.

[11] N. Fairfield, G. A. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels," *Journal of Field Robotics*, 2007.

[12] B. Sood and K. Kaur, "A Monitoring Approach for Surface Reconstruction from 3D Point Cloud," *Internationl Journal of Computer Technology & Applications*, pp. 277-282, 2012.

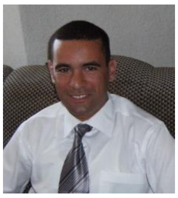
[13] G. Vosselman and S. Dijkman, "3D Building model reconstruction from point clouds, and ground plans," *International Archives of Photogrammetry and Remote Sensing*, 2001.

[14] J. E. Deschaud and F. Goulett, "A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing," in *Proc. International Symposium. 3D Data Processing, Visualization and Transmission*, 2010.

[15] E. Hernandez, P. Ridao, D. Ribas, and J. Batlle, "MSISPIC: A probabilistic scan matching algorithm using mechanical scanned imaging sonar," *Journal of Physical Agents*, vol. 3, no. 1, 2009.

[16] J. A. Castellanos, J. Neira, and J. D. Tardós, "Map Building and Slam Algorithms," *Control Engineering Series Autonomous Mobile Robots*, pp.6-40, 2006.

[17] S. Dubuisson, "The computation of the Bhattacharyya distance between histograms without histograms," in *Proc. 2nd International Conf. on Image Processing Theory Tools and Applications*, 2010, pp. 373-378.



Boutine Rachid was born September 24, 1975 in El Harrouch Wilaya de Skikda, Algeria. Phd Student, and member in LIRE laboratory of Mentouri university Algeria, since 2006. Master degree in computer science, since 2005, from Larbi ben M'hidi University Algeria. Currently, he is an ASSISTANT TEACHER at 20 Août 1955 university Skikda Algeria, since 2006. Its domain of research is autonomous navigation of mobile robots in partially known environments. He has two published papers, the first paper entitled "Global localization and concurrent mapping for mobile robot on the robotic simulator SIMBAD" in CISA'09 Tunisia, the second paper is "Using radius to improve robot localization and mapping" in ICMWI'10, Algiers. Mr. Rachid was a member in the organization committees, of JNEDI'12, and CTAACS'12.