Maciej PRZYBYLSKI, Barbara PUTZ
Warsaw University of Technology, Warsaw

DUAL REPRESENTATION OF 3D MAP FOR MOBILE ROBOTS

Key words

Laser range finder, 3D map building, planes extraction.

Summary

The paper presents several basic issues that were solved in order to create a complete system of 3D map building with the use of a mobile robot equipped with a two-dimensional range finder SICK LMS 200 and a tilt unit. The concept of a dual representation based on the extraction of planes and an occupancy map is the basic assumption of the system. The robust method of the extraction of planes from single 3D scans is presented, together with fast algorithms of generating and simplifying the triangle mesh. For the occupation map building, an efficient dynamic data structure based on the octree has been designed. It provides fast constant access time to each cell. The resulting dual 3D map is generated in the time period shorter than the overall time of the measurement.

Introduction

The 2D laser range finders are very popular in the mobile robotics, where they have proved their usefulness in the map building and obstacle avoidance tasks. However, regarding the constant progress in the area of mobile robotics, 2D maps have become an insufficient representation of the surroundings of the robot. Research on the use of laser range finders in the creation of 3D maps has been carried on for several years. Usually, systems consisting of two (or more)

2D range finders [2–5] or one 2D range finder with a tilt unit are used for the 3D-measurement acquisition [1–6].

The aim of the conducted research is to create a complete system of 3D map-building with the use of mobile robots equipped with a 2D laser range finder and a tilt unit. The above mentioned problem includes several basic issues. This paper covers most of them and presents achievements to date [3].

1. The concept of the 3D map building system

The discussed approach employs dual representation, based on the extraction of planes and the use of an occupancy map. The feature-based map gives basic information about the type of surroundings, and it also may become a step to create a cognitive map. However, it is not very useful in a route planning task, where an occupancy map is usually used. On the other hand, the occupancy map does not give any information about the type of the obstacle. Therefore, dual representation has been proposed [3]. The other hybrid map can be found in [4].

The additional advantage of the proposed approach is that extracted plane segments can be represented as a triangle mesh and other non-plane obstacles can be drawn with voxels, where one voxel corresponds to one occupancy map cell.

The process of 3D map building can be divided into several basic stages, such as, data acquisition, data analysis, 3D measurements matching, and model refinement, including measurements merging and model simplification. The concept of the discussed 3D map-building system is shown in Fig. 1.

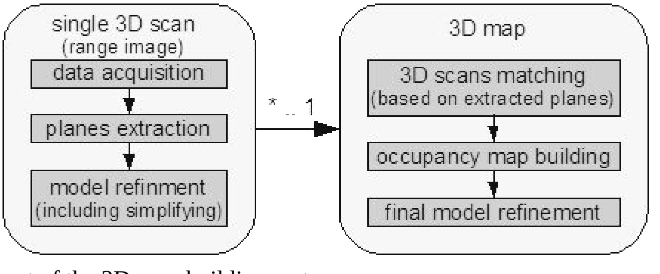

Fig. 1. The concept of the 3D map-building system

Most of the presented tasks have been resolved and are discussed in the following sections, except the 3D scans matching and final model refinement.

2. The 3D measurement acquisition

In the presented solution, a single 3D measurement is a set of 2D measurements acquired with the 2D-laser range finder SICK LMS 200 mounted on a tilt unit [1]. The 3D measurement covers a piece of a sphere on which measured points are evenly distributed. The resolution of the distribution of the points depends on the range finder's angle resolution in one direction and the tilt unit step's value in a second direction. In this respect, it is possible to represent such measurements as a range image of the following size: *(number of points in a 2D scan) x (number of 2D scans)*, where pixel brightness depends on a measured distance from an obstacle. Measured points are specified in a spherical coordinate system and have to be transformed into the Euclidean space.

3. The planes extraction algorithm

The triangle mesh generation

Measured points are evenly distributed in the spherical coordinate system, which is a good base to generate a triangle mesh (Fig. 2a). Consecutive points are simply connected to form neighbouring triangles. For example, points *(i,j)*, *(i,j+1)* and *(i+1,j+1)*, where i,j mean rows and columns of a range image, create one triangle and points *(i,j)*, *(i+1,j)* and *(i+1,j+1)* create the second triangle.

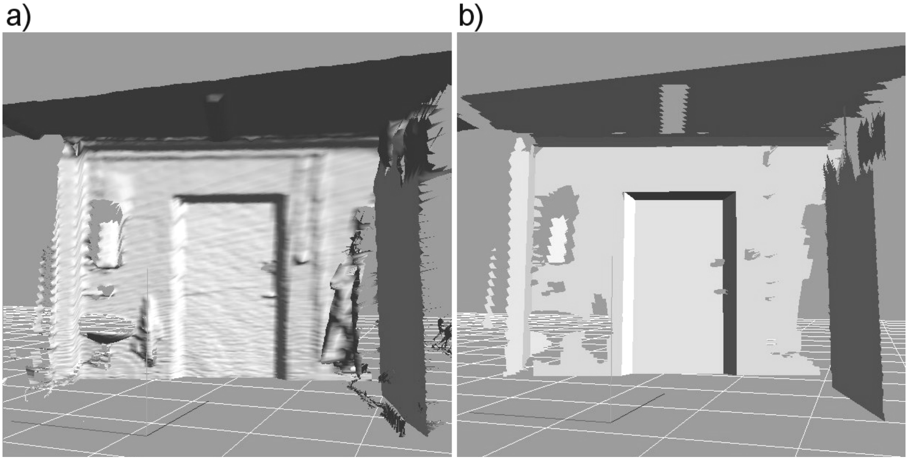

Fig. 2. The single 3D scan: a) after the triangle mesh generation; b) after the 3D model refinement

Additionally, there are two other conditions under which triangles are created. The first one forbids creating an edge that is longer than a given value. The second condition concerns the absolute value of a scalar product of a *(i,j)*-point's position vector and the normal vector of a created triangle. This value should be greater than the assumed threshold. The second condition helps to avoid creating a triangle whose vertexes lie on two odd edges of an offset.

It is important to generate the triangle mesh correctly, because in the succeeding step surface normals in vertices are calculated and the plane extraction algorithm is based on the surface normals analysis.

The primary segmentation

The primary segmentation algorithm is based on the recursive region-growing method. It analyses the surface normals of the neighbouring points. The analysed point will be added to the segment if it fulfils the four following conditions:

- The point has not been added to any segment yet.
- The distance between the current point and the neighbouring point in the Cartesian space is smaller than the assumed threshold.
- The absolute value of the scalar product of the surface normal unit vectors in the current point and the neighbouring point to be added is greater than the assumed threshold.
- The absolute value of the scalar product of surface normal unit vector in the current point and a unit vector between the current point and the point to be added is below the assumed threshold.

If the point is added to the segment, the next step of the recursive segmentation procedure will be called for this point. As a result of this procedure, a set of segments is made, where a segment consists of a set of points. The plane surface equation is calculated for each segment with the commonly known least-square method and regression equation, described for example in [4].

After the plane equation is evaluated, an algorithm computes the average point-plane distance. Then, points whose distance from the plane is greater than the assumed threshold are removed from the segment. The plane equation is evaluated again. This step is not superfluous. It helps to remove from segments the points that have similar surface normal vectors but are situated further than other points and lie on a different surfaces.

The set of points will be accepted as a correct plane segment if it fulfils following conditions:

- The number of points in the segment is greater than the assumed threshold.
- The mean value D of the absolute deviation of the points' distance from the estimated plane is below the assumed threshold D_{max}:

$$D = \frac{\sum_{i=1}^{n} \left| d_i - \hat{d} \right|}{n} < D_{max} \tag{1}$$

where: d_i – point-plane distance (with sign retained),
$\hat{d}$ – average point-plane distance.

Condition (1) has been adjusted experimentally to remove the segments that are not planes. It works better than using standard deviation or the mean deviation of the points' distance from the estimated plane.

The secondary segmentation – primary segments' growth

The primary segmentation procedure creates several plane segments with estimated plane equations as well as an inconsistent set of points that have not been added to any segment. The secondary segmentation procedure is very similar to the primary one, but the difference is in the point addition condition, which is based on the point-plane distance.

The segments analysis order has an influence on the final result, because some of the non-added points fit more than one segment. In order to avoid such a situation, all neighbouring points can be added to the current segment, even if they already belong to another segment.

This approach has a spin-effect, since segments become inconsistent sets. As a result, it was necessary to create a resegmentation procedure. It is also a growing-based method, but in this case, only segment indexes of each point are taken into consideration.

The similar segments merging

The result of the secondary segmentation and the resegmentation is a set of smaller neighbouring segments that can be merged. Therefore, another algorithm has been developed.

The first step is to create an adjacency list of segments. Afterwards, each segment is checked for its capability of being merged with a neighbouring one. Two neighbouring segments will be merged if the value of the dot product of those normal vectors of the surface of the segments is greater than the assumed threshold. There is also a cross test, where the average distance between the first segment points and the second segment plane is calculated, and vice versa. The segment for which this average distance is greater will be merged with the second one. The adjacency list of the first segment will also be added to the second segment list, but the second segment plane equation will stay unchanged.

4. The 3D model refinement and triangle mesh simplifying

If plane surfaces are recognised, it is possible to make refinements in the 3D model, especially to make plane surfaces smooth. The smoothing algorithm is very simple. It calculates the point-plane distance with the retained sign (d_i in Equation 1) for each point that belongs to some segment, and it translates this point by the calculated distance in the surface normal direction. At the end, the process of improving edges between neighbouring segments is performed.

When the model consists of plane segments, the number of triangles becomes much greater than is necessary to present the 3D model correctly. Because the segment is a plane, only points that lie on its boundary are necessary to create a correct mesh. Therefore, most of the vertexes inside the segment can be removed. The algorithm that has been developed does not remove all vertexes from the inside but searches maximum square regions inside the segment. Inside these square regions, the old triangle mesh is changed into a triangle fan with one point in the centre of the square region and few points on the boundary. It is not an optimal solution, but it is fast and reduces the number of triangles by about two-thirds. Additionally, the plane segment representation is changed from the set of points to the set of triangles.

5. The occupancy map

In order to build the occupancy map, a dynamic data structure based on an octree is used. Each cell of this map holds indexes of plane segments and points that lie on other non-plane surfaces. Every element has eight pointers to its children, and it corresponds to a cubic fragment of the 3D workspace. The first element, also called root, includes the whole workspace.

The root is the zero level of an octree structure. Only the last level of an octree is important, because the aim was to divide the whole workspace into regular cells. The memory is reserved only for non-empty cells. As a matter of fact, an efficient dynamic 3D table has been developed, whose dimensions in all directions are the same and have to be the powers of two. The example of the cell access method (Fig. 3) is simplified and presents only one dimension.

The position of the child cell relative to the parent cell is described by indexes *(i,j,k)*. Apart from this, the last level cells also have indexes that describe the position of each cell in the root coordinate system. The first cell has index *(0,0,0)* and the last $(2^L, 2^L, 2^L)$, where L is the number of the octree levels. The access time to each cell is constant, and it requires only L steps. The access is also possible by giving any point from the 3D space.

The example of the resulting dual map is presented in Fig. 4.

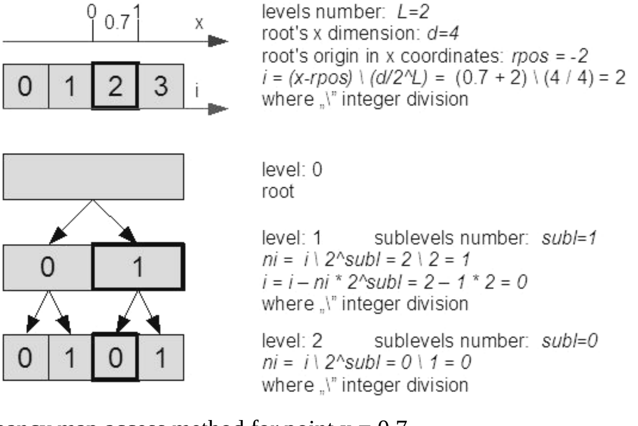

Fig. 3. The occupancy map access method for point x = 0.7

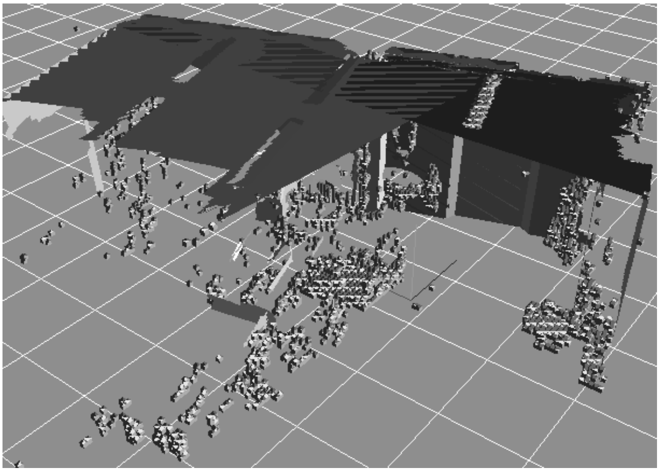

Fig. 4. The dual map: extracted planes and voxels that represent non-plane obstacles

Conclusions

In order to create the 3D map-building system, the dual map representation, based on extracted planes and an occupancy map, has been proposed. In this respect, several basic algorithms have been developed, such as, the special addressing method of the cells of the map, as well as the planes extraction algorithm, where surface normals calculations and region-growing-based methods are used.

To complete necessary algorithms for the system, the studies on 3D scan matching and final model refinement are being continued.

References

1. Chojecki R., Olszewski M., Szynkiewicz W., Trojanek P.: ELEKTRON: a mobile robot for laboratory purposes and its applications. Automation, 2007, Warsaw, Poland.
2. Montemerlo M., Hähnel D., Ferguson D., Triebel R., Burgard W., Thayer S., Whittaker W., Thrun S.: A system for three-dimensional robotic mapping of underground mines. CMU, Pittsburgh, Tech. Rep. CMU-CS-02-185, 2002.
3. Przybylski M.: Dual representation of 3D map based on planes extraction and occupancy map. Master thesis (adviser: B. Putz), Faculty of Mechatronics, Warsaw University of Technology, Warsaw 2007 (in Polish).
4. Siemiatkowska B., Gnatowski M., Zychewicz A.: Fast method of 3D map building based on laser range data. Journal of Automation Mobile Robotics and Intelligent Systems, 2(1), 2007, 35–39.

5. Thrun S., Martin Ch., Liu Y., Hähnel D., Emery-Montemerlo R., Chakrabarti D., Burgard W.: A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. IEEE Transactions on Robotics and Automation, 3(20), 2004, 433–443.
6. Wulf O., Wagner B.: Fast 3D scanning methods for laser measurement systems. International Conference on Control Systems and Computer Science (CSCS14), Bucharest, 2003.

Reviewer:
Andrzej STEPNOWSKI

Dualna reprezentacja trójwymiarowej mapy otoczenia dla robotów mobilnych

Słowa kluczowe

Dalmierz laserowy, tworzenie map 3D, ekstrakcja powierzchni płaskich.

Streszczenie

W artykule zaprezentowana została koncepcja systemu do tworzenia map 3D przez roboty mobilne. Pomiary wykonywane są w czasie postoju robota za pomocą dalmierza laserowego SICK LMS 200 zamontowanego na głowicy obrotowej. Założono, że analiza i wstępne uproszczenie modelu odbywa się już na poziomie pojedynczego pomiaru 3D. Opracowane algorytmy pozwalają na obróbkę pomiaru 3D w czasie krótszym niż czas trwania pomiaru.

Tworzony system przeznaczony jest dla robota poruszającego się wewnątrz budynku, gdzie większość obiektów składa się z powierzchni płaskich. Dlatego opracowano zestaw algorytmów umożliwiających szybkie wykrycie tego typu powierzchni, połączone z generowaniem i redukcją reprezentujących je siatek trójkątów. Ekstrakcja płaszczyzn pozwala na znaczne uproszczenie modelu otoczenia i będzie użyteczna w przyszłych pracach nad samolokalizacją robota. Mapa złożona z płaszczyzn nie jest wygodna w zadaniu planowania ścieżki, dlatego zaproponowano dualną formę reprezentacji złożoną z płaszczyzn i mapy zajętości opartej na drzewie ósemkowym. Opracowana mapa zajętości umożliwia przedstawienie przeszkód niepłaskich za pomocą voxeli i szybki dostęp w stałym czasie do dowolnej komórki tej mapy.