Towards Map Building and Space Coverage Planning in Robotics

JIŘÍ PAVLÍČEK, ROMAN MÁZL, PETR ŠTĚPÁN and LIBOR PŘEUČIL
Department of Cybernetics, The Gerstner Laboratory for Decision Making and Control
Czech Technical University in Prague, Faculty of Electrical Engineering
Technická 2, 166 27 Prague 6
CZECH REPUBLIC
{ }

Abstract: - This paper describes new method for building occupancy grid from single camera, automatic calibration of this method and approach how to use this internal representation for cleaning unknown environment. The algorithm is designed for indoor application. This assumption is used for calculating inverse perspective transformation of the gathered image. The paper brings an overview of path-planning methods targeted on complete coverage of an operating space. Two algorithms for planning coverage based on occupancy grid are outlined and their fundamental properties are compared. Firstly the existing approach to solution of the coverage problem based on cellular decomposition of the working space is discussed and further the method applies border expansion is introduced. Attached experiments verify and compare the newly designed and the existing methods as well as illustrate the final performance of the methods in real environments.

Key-Words: - Mobile robotics, environment mapping, occupancy grids, space coverage, motion planning

1 Introduction

This paper deals with preprocessing of data from single color camera in indoor environments. The indoor environment means that the floor can be approximately described as a plane in 3-D. The presented technique can be used in outdoor environment if this condition is roughly satisfied (for example on the roads, see [1]).

The robot's environment is often represented in the local sensor-based map by an occupancy grid [6][3]. It is a regular grid, where each cell stores the probability that the particular space is occupied. The probability is updated by accumulated sensor readings.

The fusion of the camera data with the range finder measurements is possible using occupancy grid. On the other hand the range finder data can be used to calibrate the probability profiles for the camera preprocessing and can be used for building very accurate 2-D maps of robot environment.

In order to achieve complete coverage of all the free area the central question stands how to plan a trajectory of the robot. Simple and straightforward solution leads to generation of random motions through the workspace - this achieves complete coverage of the accessible area. The major disadvantage here is high inefficiency by multiple overlapping of the cleaned path, what leads to waste of time and energy. To overcome this, the robot must not move around the environment randomly, but in a systematic manner.

The philosophy of covering free area in a systematic way can be based on storing the regions, that the robot has already visited, what gives the possibility to generate moves only through the areas remaining uncovered. It would be very difficult to cover all the free space without visiting some locations more than once, but we can at least minimize such events. Let's suppose w to be the width of the robot. Then, if we want to cover the free space in an efficient way minimizing the count of the repeated visits to the same locations the robot has to navigate very closely to the already covered area. This defines the maximum distance between the neighboring roads being set about to the width of the robot as shown in the fig. 1.

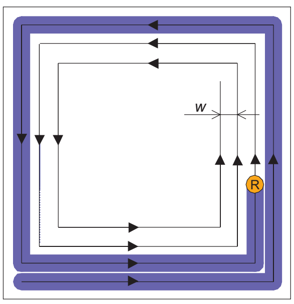

Fig. 1: Coverage process

It is very easy to achieve complete coverage without visiting the same location twice in the case of environment with low structural complexity (without many obstacles). Unfortunately, the coverage solution complexity grows rapidly with raising the environment structure complexity.

2 Building occupancy grid from single camera

In a typical indoor environment or in outdoor environment, where the floor can be approximately described as a plane, all parameters of the robotics system are known. The parameters of the camera system can be calibrated on special images and then used to detect obstacles in the robot surrounding.

Our goal is to get 2-D map of the robot-operating environment. It is necessary to get parameters of the

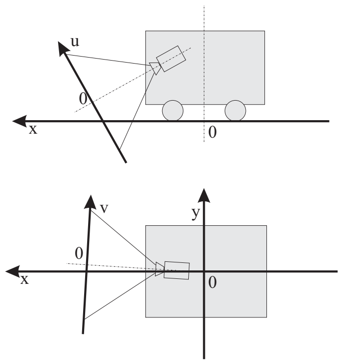

Fig. 2: Coordination systems on mobile robot

robotics system to compute the transformation from the image coordinate system (u, v) to the robot coordinate system (x, y) (see figure 2). It means to determine the projection function $f(u, v) = (x, y)$ [8]. The first step is a reduction of the radial distortion and decentering. The constants u_c, v_c and p_1, p_2, p_3 can be determined from calibration image according to the following equations:

$$\begin{aligned} dist &= (u - u_c)^2 + (v - v_c)^2 \\ u' &= (u - u_c)(1 + p_1 dist + p_2 dist^2 + p_3 dist^3) \\ v' &= (v - v_c)(1 + p_1 dist + p_2 dist^2 + p_3 dist^3) \end{aligned}$$

To set the parameters of radial distortion and decentering the calibration image was used. The image serves to detect the points from lines and the minimization of the square distance of these points to corresponding line was done. The results of this minimization are the parameters u_c, v_c, p_1, p_2, p_3.

The coordinates (u', v') determine the coordinates of the image after correction of the radial distortion. The next step is to detect the shift of the camera from the center of the robot x_{cam}, y_{cam} and parameters of the plane, that describe the floor of the environment $\Delta u, \Delta v, X, Y$. The following equations describe the projection transformation:

$$x_0 = -\frac{(u' - \Delta u)X}{v' - \Delta v} + x_{cam}$$

$$y_0 = -\frac{Y}{v' - \Delta v} + y_{cam}$$

The final transformation is necessary to correct the angular difference of the camera axis and the robot axis α and to get the coordination in the robot world. The final transformation can be described as:

$$\begin{aligned} x &= x_{rob} + x_0 \cos(\alpha + \omega_{rob}) - y_0 \sin(\alpha + \omega_{rob}) \\ y &= y_{rob} + x_0 \sin(\alpha + \omega_{rob}) + y_0 \cos(\alpha + \omega_{rob}), \end{aligned}$$

where the robot position is $x_{rob}, y_{rob}, \omega_{rob}$

The 11 measurements of the box were done by camera and laser range finder to determinate above-mentioned parameters. Each camera measurement represents four points (the borders of the box and two points on the front side) and the laser range finder measures the real position of this box. The minimization of the square distance from the real position to the position computed from camera image in MATALB language produced the parameters $x_{cam}, y_{cam}, \Delta u, \Delta v, X, Y, \alpha$ of the robotics system. The figure 3 shows the result of the calibration process - the position of the box measured by laser range finder and by camera.

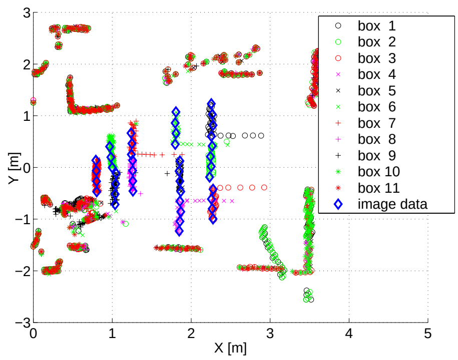

Fig. 3: Result of the camera calibration. The 11 measurement of box by laser range finder and camera

2.1 Definition of the probability profiles

The detection of the free space is done according to color of the floor. Suppose that the color of the floor is different from the color of the obstacle. If the color of the floor is known the probability distribution of a free space can be defined. The occupancy grid [3] is build using Bayes update formula:

$$P(Occ \mid R) = \frac{P(R|Occ)P(Occ)}{P(R|Occ)P(Occ)+P(R|Emp)P(Emp)}$$

$$\begin{aligned} P(Emp) &= 1 - P(Occ) \\ P(R \mid Emp) &= 1 - P(R \mid Occ) \end{aligned}$$

This formula defines the new conditional probability, that the cell is occupied given measured reading $R = (h, s, v)$, is depending on the old value $P(Occ)$ and the probability distribution $P(R \mid Occ)$. The probability distribution $P(R \mid Occ)$ can be represented as difference of the measured color R and the reference color of the floor in the hsv space. When the color of the floor is similar to the color of the obstacles, this method works not very well. Another approach uses the definition of the probability profiles $P(R \mid Occ)$ by additional information about environment.

The additional information can be map of the training environment or sonar and laser data that can correctly measure the training environment. The probability profiles can be calculated directly from the camera images. The free space and the border of the obstacle can be determined from the range finder data or from the map of the environment.

Fig. 4: Image of our office from camera on mobile robot

The matrices $free$ and occ can be computed for discrete hsv space. These matrices define the number of cells with concrete color that are free or occupied respectively. In our example, 10 images with different lighting conditions were used to create the matrixes $free$ and occ with size $20 \times 20 \times 20$. For each pixel of the camera image, the occupancy of this pixel in robot coordination system was calculated. If the pixel with color $(h, s, v) \in < 0, 1 >^3$ represents a free space then the matrix value $free(\lceil 20h \rceil, \lceil 20s \rceil, \lceil 20v \rceil)$ is increased. If the pixel belongs to the occupied space then the matrix value $occ(\lceil 20h \rceil, \lceil 20s \rceil, \lceil 20v \rceil)$ is increased. The probability profile $P(R \mid Occ)$ for reading $R = (h, s, v)$ is defined by the following formula

if $free(h', s', v') + occ(h', s', v') > 0$ then

$$P(R \mid Occ) = \frac{occ(h', s', v')}{free(h', s', v') + occ(h', s', v')},$$

otherwise

$$P(R \mid Occ) = 1,$$

where

$$\begin{aligned} h' &= \lceil 20h \rceil \\ s' &= \lceil 20s \rceil \\ v' &= \lceil 20v \rceil \end{aligned}$$

Figure 4 describes the original data from camera and in the figure 5 the image transformed into the robot coordinate system is depicted. Each pixel determines $5 \times 5cm$ cell in the robot environment.

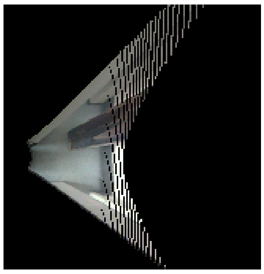

Fig. 5: Image from figure 4 transformed into floor coordinates. Robot position is on the left side of the picture a the robot is heading to the right side.

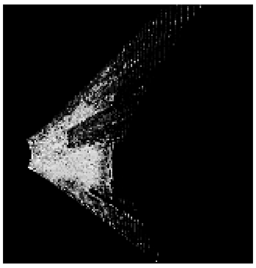

Fig. 6: Occupancy grid created from single camera image from figure 5. The white color represents free space detected by color probability profile of the floor.

Figure 6 depicts the occupancy grid created from image in figure 5 using a probability profile matrix $20 \times 20 \times 20$.

2.2 Building final 2-D map

The final occupancy grid is constructed using information about visibility. It can often happened that the color of the floor can appear at the obstacle too, even when the border of the obstacle is detected very well. The final computation is done from the position of the robot to the border of the grid. The accumulated probability is computed using the following formula

$$P_{new}(Occ) = P(Occ) P_{Acc}(Occ)$$

There are two possibilities how the accumulated probability $P_{Acc}(Occ)$ can be defined. The first one is simply the multiplication of the probabilities of all cells from the viewpoint. The second approach uses the Bayes formula [4] for a recursive definition of this probability.

$$P_{Acc'}(Occ) = \frac{P(Occ)P_{Acc}(Occ)}{P(Occ)P_{Acc}(Occ)+(1-P(Occ))(1-P_{Acc}(Occ))}$$

The result of the Bayes update formula for the visibility occupancy grid is shown at the figure 7.

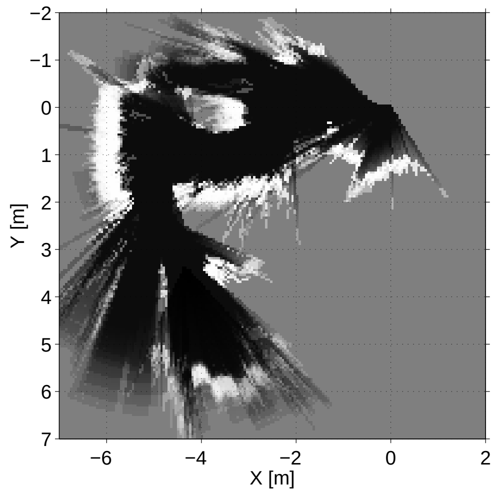

Fig. 7: Occupancy grid created from camera data

3 Coverage Process

The fundamental problem, common to all mobile robots, is the navigation in an environment. The robot requires some prior knowledge about the obstacles in the environment before planning of any action for motion. The chosen abstraction of the world model for the described global path planner was mainly driven by simplicity of the planning algorithm. This led to selection of occupancy grid approach for model representation. Such model assigns a probability measure to each particular space of the environment. Therefore the segmentation of the grid to cell with low and high occupancy values is done first. Such simplified occupancy grid can be obtained by thresholding or via more sophisticated segmentation methods developed in [3], [10]. There is a variety of coverage algorithms using different approaches and operating under certain assumptions about the environment and the sensors. One of the methods for the coverage is based on a geometric structure called cellular decomposition [5], which is the union of non-overlapping sub-regions of the free space called cells. An adjacency graph encodes a topology of the cells in the environment where nodes are cells and edges connect nodes of adjacent cells. Each of the cells is easily covered using back and forth motion pattern and thus complete coverage is reduced to finding an exhaustive walk through the adjacency graph [2].

The suggested approach reaches the complete coverage of a predefined free space via usage of a hierarchical geometric pattern that follows the contours of obstacles (see section 2.4). The adjacency graph encodes the topology of the roads in the environment where nodes are roads and edges define possible transitions between the roads here. When each of the roads is passed, the complete coverage is guaranteed.

4 Boustrophedon Decomposition

The coverage approach described as the ”boustrophedon decomposition” in [2] is an adaptation of the existing complete motion-planning scheme, termed an exact cellular decomposition. Cellular decomposition is a motion planning technique where the free configuration space (set of all admissible robot configurations where the robot does not overlap any obstacle) is decomposed into particular cells. Union of these cells gives the original free space. Each of these cells can be represented as a node in a graph, where the adjacent cells have an edge connecting their corresponding nodes (adjacency graph).

If each cell can be covered by the robot, then the floor coverage problem reduces to determining a walk through the adjacency graph so that each node is visited at least once, i.e., the Traveling Salesman Problem, for which a solution (possibly sub-optimal) always exists. One popular cellular decomposition technique yielding the complete coverage path solution is the trapezoidal decomposition [5] (also known as the slab method [7]). This approach decomposes free space into trapezoidal cells (see fig. 8a). Since each cell is a trapezoid, coverage in each cell can easily be achieved by simple back and forth motions. The exhaustive coverage of the environment is achieved by visiting each cell in the adjacency graph (see fig. 8c). Unfortunately, to achieve completeness of the trapezoidal decomposition approach requires too many redundant back and forth motion. The boustrophedon decomposition slightly reduces such redundant back and forth motions, since it has less number of cells, fig. 8b.

The drawback of both the trapezoidal as well as boustrophedon decompositions approaches is that the both require a polygonal environment. An illustration of the resulting path plan using boustrophedon decomposition is shown in fig. 9.

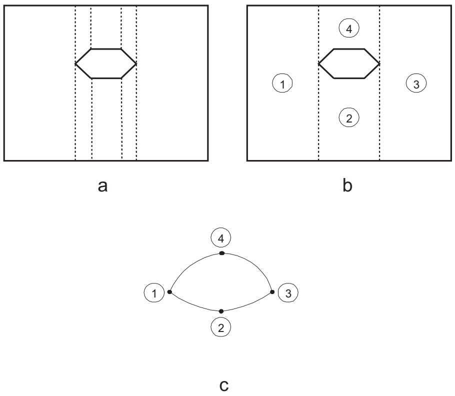

Fig. 8: (a) Exact cellular decomposition. (b) Boustrophedon decomposition with reduced number of cells. (c) Adjacency graph for boustrophedon decomposition.

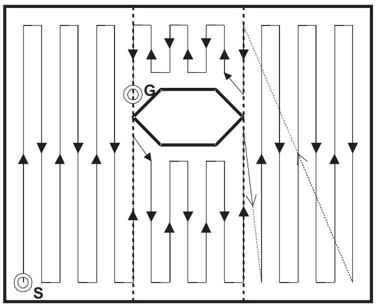

Fig. 9: Resulting coverage path plan using boustrophedon decomposition

5 Border Expansion Method

Our idea of how to reach the complete coverage is to create a set of neighboring roads following the border of obstacles. We suppose that the environment is defined by its border, which is common to the obstacles as well. This means that both the border of the environment and of the obstacles towards each to other.

Once knowing the obstacles with their boundaries, these can be expanded it into free space. The borders are expanded by about the width of the robot *w* and closed road around each expanded obstacle is defined in every step. If the borders of the obstacles or the environment overlap during the growing process, then the overlapping part is removed and the rest is merged into the composite road. The road-generation process is terminated whenever the borders cannot grow any more. Thus, we can generate a set of closed roads that fit the requirement for the complete coverage as illustrated in the fig. 10a. Dashed lines mark the safety zone permitted to the robot.

All the defined roads are categorized into *safety levels* and the adjacency graph is built as described below. The *level 0* defines safety zones δ around obstacles and typically does not integrate main roads. The *level 1* stands for the roads that are closest to obstacle edges up to the level *n*, which denotes the most distant roads from obstacles.

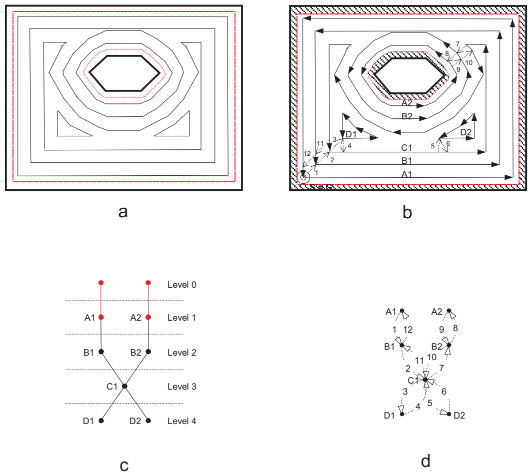

Fig. 10: (a) Generated roads. (b) The description of roads and numbered sequence of transitions between roads. (c) Adjacency graph of roads and assignment of roads to levels. (d) Transition diagram.

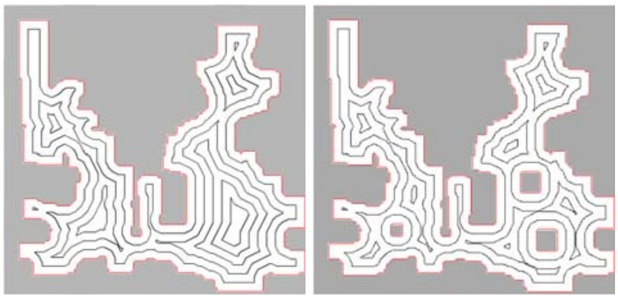

Fig. 11: Experimental results of space coverage

Single roads are defined as: *Level 1* contains the main roads in following order. Road A1 is the main road attached to the boundary edge of the environment. The roads A_i for $i \leq 2$ correspond to the borders of obstacles. All roads defined in the higher levels can be derived from main roads on *Level 1*. The fig. 10b shows the defined roads and fig. 10c shows the appropriate adjacency graph, where the nodes represent single roads and the edges represent possible transitions between the roads. Fig. 10b also shows the assignment (membership) of roads to levels.In order to reach a complete coverage, there also has to be defined a sequence of transitions between the roads. This is done through the adjacency graph when searching for an exhaustive path including all the nodes in the graph. The path search is then a constrained search process starting at the road A1 and with process goal allocated in the road point as shown in the fig. 10b and

10d. Passing the mentioned sequence of transitions a complete coverage of the free space is guaranteed. Experimental results are in the fig. 11.

6 Conclusion

This paper describes a new approach for creating the occupancy grid from single camera. The experiments were done on a real mobile robot [9] in typical office environment without additional changes.

The second part presents two planning algorithms for space coverage. The boustrophedon decomposition approach tends to skips particular regions of the environment nearby boundaries. This phenomenon is likely to be caused by the discrete sampling of the side step. If the robot has a shorter side step, the uncovered areas are reduced. There is a tradeoff between time and coverage here. Nevertheless, the use of simple obstacle-tracking algorithm could also alleviate this problem even after the coverage has been completed. Such a solution is consistent with a typical requirement set in practical application of this task (e.g. a cleaning machine control). The problem with the regions excluded from the coverage can be easily overcome by an additional run of the system along obstacle boundaries.

The border expansion method solves the given problem providing small skipped regions around the boundaries of obstacles while the suggested method has the fundamental attribute to trace borders of all the obstacles in the environment.

The boustrophedon decomposition is based on the idea of reducing a number of cells produced by the exact cellular decomposition through merging less important adjacent cells. Larger cells require a smaller number of motions to achieve the complete coverage. On the other hand, a higher number of obstacles in the environment raises the number of small cells after the decomposition process and the number of motions as well.

The border expansion method doesn't increase significantly the number of motions needed to achieve the complete coverage of environments with higher complexity. Anyhow, two situations can arise at this point. First, the neighbor roads can be closer than the width of the robot at some locations, what leads to overlapping. Second, the neighbor roads are more distant apart compared to the width of the robot so small stripes can remain uncovered. This can be solved by setting the distance between the neighbor roads smaller than the width of the robot. As this leads to overlapping zones in the coverage path, the complete coverage can be guaranteed.

Another drawback of the boustrophedon decomposition is the dependence on the rotation of the environment before performing the decomposition process. The border expansion method is independent with respect to rotation.

7 Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic within the frame of projects ”Decision making and Control for Manufacturing” no. MSM 212300013, by the grant agency FRVŠ no. 2156/2002. and by the CTU internal grant agency no. CTU 0209713, no. CTU 0209413

References:

[1] M. Bertozzi, A. Broggi, and A. Fascioli. An extension to the inverse perspective mapping to handle non-float roads. In *In Proceedings of the IEEE International Conference on Intelligent Vehicles*, pages 305–309, Stuttgart, Germany, October 1998. IEEE.

[2] V. Choset and P. Pignon. Coverage path planning: The baustrophedon decomposition. In *Proc. of the International Conference on Field and Service Robotics*, Australia, December 1997.

[3] A. Elfes. *Occupancy Grids: A Probabilistic Framework fo Robot Perception and Navigation.* PhD thesis, Electrical and Computer Engineering Department/Robotics Institute, Carnegie-Mellon University, May 1989.

[4] P. S. James. *Bayesian Statistics*. John Wiley & Sons, Inc., 1989.

[5] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[6] Dias J. Menezes P., Araújo H. and Ribeiro M. I. Obstacle detection in mobile robots using sonar data. In *Associação Portuguesa de Controlo Automático Int. Conf. - Controlo 94*, Lisabon, September 1994. I.S.T.

[7] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[8] M. Šonka, V. Hlaváč, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing Co., USA, 1999.

[9] P. Štěpán, L. Král, M. Kulich, and L. Přeučil. Open control architecture for mobile robot. In *In Proceedings of the 14th World Congress of IFAC, Volume Q*, pages 163–169, Beijing, China, July 1999. IFAC.

[10] P. Štěpán and M. Kulich. Building maps for autonomous mobile robots. In *Proc. of the 3-rd International Conference on Field and Service Robotics*, pages 321–326, Finland, June 2001.