Exploration and Traversable Area Marking for the VisualSLAM

Jaroslav Rozman*
Department of Intelligent Systems
Faculty of Information Technology
Brno University of Technology
Božetechova 2, 612 66 Brno, Czech Republic
rozmanj@fit.vutbr.cz

Abstract

This paper deals with the exploration strategy for a robot during exploration and 3D map building in an unknown environment. The mapping algorithm detects the significant features, finds correspondences in both images, matches them together and computes their 3D coordinates. This way the robot incrementally creates a 3D map of its surroundings and it tries to explore as much area as possible. The first part of this paper describes the way of marking the traversable path in the map created so far. In the second part the goal point selection and the path planning to this goal point are described. As the map uses points and triangles to represent the shape of the surface, the traversable area marking and the path planning is also done in the map which is represented in the same way.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: RoboticsVision and Scene Understanding

Keywords

robotics, path planning, stereocamera, target selection

1. Introduction

The creation of a map is an important step for the robots to become truly autonomous. The 2D map creation using the laser scaner or the sonars has been widely investigated in the last two or three decades. The investigation begun in the eighties with the works of Elfes and Moravec [9], [5] who used the occupancy grids for map creating. This was later improved using the SLAM algorithm (Simultaneous Localization and Mapping) [4] with Kalman filtering [12] and even later with the particle filter [8]. The creation of the 2D maps was upgraded to 3D maps by using two 2D lasers [6], [21], [22] or by using of the 3D lasers. The task of the 3D map building is now shifted to the 3D map creation with cameras. Important works in this area include those by Davidson who created the algorithm using single camera and called it monoSLAM [3]. Other works are those by Sim and Elinas which are concerned with Rao-Blackwellised particle filters in the visual SLAM [16], [17], [18]. The more recent works are by Morisset and Rusu [11] with grid based 3D mapping. They label the grids in this work according to the probable type - ground, wall, stairs. There is also a paper by Moreno [10], where the probabilistic models for 3D visual SLAM with Speeded Up Robust Features descriptors are described.

Visual Odometry is an important area for the visual SLAM with the stereocamera as the only sensor. There are works by Nister [13] where the stereocamera is used for the travelled distance detecting with an error about 4.1m in 184m travelled path. Another similar work is the one by Agraval and Konolidge [1] where they also use the GPS and inertial sensors.

The task of 3D model building is of course also handled in the area of the computer vision. It is natural, that outcomes from this area are also used in the robotics. Examples of the model building in this area are for example the work by [14]. Other examples are the works by Clipp [2] and mainly by Pollefeys et al. [15].

The area that is not described much in these papers is the path planning during the mapping. There are a lot of algorithms for path planing, several are well described in the [20], but they know the whole map in advance. The paper where the mapping is described together with the path planning algorithm is for example the abandoned mine mapping by Thrun [22]. In this work the robot equipped with two 2D lasers explores the abandoned mine and together with the grid-based mapping it seeks the path to the next unexplored goal point. Another example is the work by Joho [7] where the goal points of the robot equipped with a laser are chosen with respect to the best view of the robot. The approach for occupancy grids that is widely used is by Yamauchi [23] or by Sim [19]. More sophisticated approach is decribed in [20], where the entropy is used.

---

*Recommended by thesis supervisor: doc. František Zbořil. Defended at Faculty of Information Technology, Brno University of Technology on nove,ber 11, 2011.

Rozman, J. Exploration and Traversable Area Marking for the VisualSLAM. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 4, No. 1 (2012) 29-38

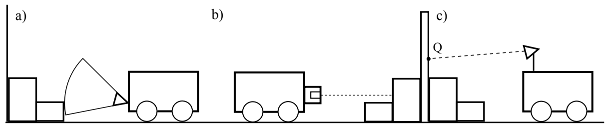

Figure 1: Three examples of obstacle detection. We can notice, that there are more obstacles between robot with cameras and the detected point Q.

The difference between mapping and path planning when using the stereocamera and laser or sonars is the fact that with the sonars or the laser scanner we can assume that the space between the robot and the detected obstacle is empty and therefore traversable, while in the case of using the stereocamera we can not assume that the space between robot and the detected point is traversable. The space between the camera and the point is free, of course, but it does not mean that the robot can freely move to that point (see Fig. 1). There can be other obstacles that robot did not detect during the first sensing. This means that the robot has to mark the traversable path to this map explicitly.

The aim of this work is to propose the suitable representation of the traversable area for the path planning. The important property has to be the integration of this representation into the created map and thus no need for any auxiliary structures or maps.

2. Traversable Area Marking

The method used for the path planning heavily depends on the representation of the map. If the map is created by points and triangles, we have to use different path planning than in the case of the occupancy grid cells. Still, we can use the algorithms for the path planning in the graph like the A* algorithm for example. With regard to the chosen representation of the map in this paper, the points and triangles, the only possible solution is probably to add special points to the map and connect them to the points of the significant features in the map. These special points then will be used for the path planning and as the border points of the traversable area. There are several possibilities for generating such points. We can either generate them in the map randomly or generate them regularly to make a regular grid. In the both cases we should test the points to see if they are in free space. Another approach is to generate the points in the places where the robot has already moved. In this case we do not need to test the points or the path between them as we know it surely, because the robot already was on that place.

The only way to mark the space that is surely traversable is to mark the positions of the robot and connect them as the robot moves. We call the proposed mark the footprint of the robot and it will consist of five points connected to triangles. One of the points will be in the center of the robot and the other four will be on its corners (here we assume, that the robot has a rectangular shape). The edges between corner points will be labeled as constrained and as the robot moves, they will be relabeled as non-constrained and thus can be deleted. As the points and the edges will be used for the path planning, the points will be divided into three parts. One part will consist of the points inside the traversable area and the second part will be the points on the boundaries of the traversable area. The third part will consist of the points on the boundaries which can not be used for path planning. The edge connecting two boundary points will mark the borders of the traversable area found so far.

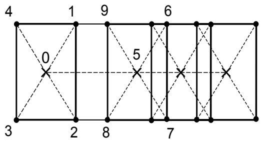

Figure 2: Three consecutive motions of the robot are shown. In the first motion the robot moved more than his length, in the next two moves the robot moved less than its length. The indices of the first two positions are also shown.

After each position update the Delaunay triangulation will be performed to connect new points to the current map. We can divide the map into two parts and perform this task solely on the part with border points and then connect it to the rest of the map. We have to keep the border edges so the triangulation used has to be the Constrained Delaunay Triangulation (CDT).

2.1 The Rules for Traversable Area Marking

The aim of the rules for the marking of the traversable area is to set the constrained edges only in the borders between the traversable and non-traversable areas. This means that we must not label any of the edge inside the traversable area as constrained. The borders will consist of the closed loops, which means that the border points will have exactly two border edges connected to them.

The proposed general rules are as follows:

- The distance between individual footprints has to be smaller than the width of the robot and bigger than the length of the robot. It is preferable to connect center points of the robot for the better path planning even in the case of the CDT.
- The rotation of the robot has to be smaller than the angle between its side corners and the center.
- All five points have to be added to the map in the following order: center, left front, right front, right rear and left rear (Fig. 2). That helps us identify which corner the point is represented.
- New point will be connected to the points added before. First, it will be connected to the points of the previous position, then to the points of the current position.
- If the constrained edges of the new position intersect with the current border edges, the intersection points will be computed and the lines among them will be labeled as non-constrained end erased.

The border marking has to take into account various movements of the robot and various possibilities of movements inside the already created traversable area. These are:

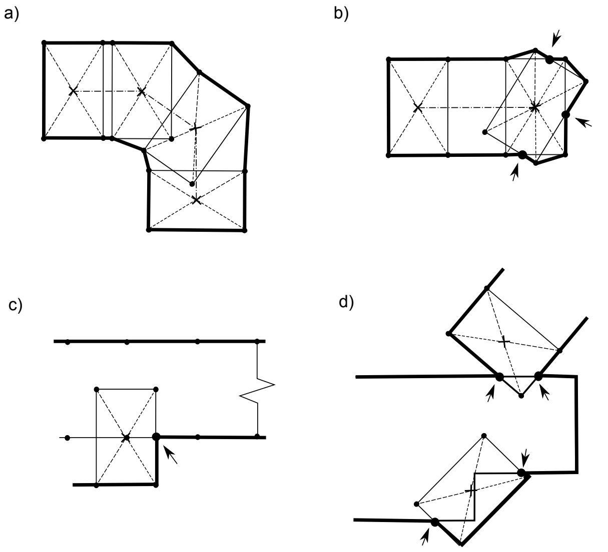

Figure 3: Example of the traversable path created by the robot turning right (a), rotating on the spot after straightforward motion (b), following the border edge (c) and entering and leaving the traversable area (d). The dashed line is for the better idea of the position of the robot. The arrows show the intersections of the new robot position with the old border edges.

a) straightforward movement with turning
b) turning on the spot
c) following the border edge
d) entering the traversable area
e) leaving the traversable area
f) moving inside the traversable area

The rules for the connection of the points of the robot:

1. The central point will be added first, so it can not be connected to any current points. It will be connected to the central point of the old position.
2. Left front point will not be connected to any other point.
3. Right front point will be connected to the left point.
4. Right rear point will be connected to the right front and in the case the robot turns left or goes straightforward to the right front of the old position. If the robot turns right, this point will be connected to the right rear of the old position.
5. Left rear point will be connected to the left front and in the case the robot turns right or go straightforward to the left right. If the robot turns left, this point will be connected to the left rear of the old position.
6. The front edge of the old position will be erased.
7. If the robots turns right, the edge on the right side will be erased.
8. Similarly, if the robot turns left, the edge on the left side will be erased.

When the robot is moving along the internal points, no footprint will be added. After the new border edges are made we have to check them for the intersections with the other border edges. New border marks are added on the places of the intersections and the traversable area is extended (Fig. 3).

1) No intersection
It means that the entire robot is in the new area or in the traversable area. If the points on the corners of the previous position were border points, the robot is in the new area. If the points were not border points, the robot is inside the traversable area and it is not extended.
2) One intersection
The robot is travelling along the border edge. New border edge is made between the intersection point and the front border point of the previous position.
3) Two intersections
The robot enters the traversable area or leaves the traversable area. The robot enters the traversable area if its front points of the previous position were the border points. Then we erase old border edge between new two intersections. Otherwise the robot leaves the traversable area and the border edge will be created between two intersections along the new position of the robot.
4) Three intersections
There are three intersections in the case the robot turns on the spot. In this case we connect old left front points with new one and old right points with new one and the rear point which is outside of the traversable area with the old one. We compute mutual intersections of left, front and right side and connect them to the corner points.

2.2 Triangulation

After new border edges are set, the CDT will be performed. Only the internal points and the border points will be used for the CDT inside the traversable area. The points of the footprint that are not connected will be omitted from the triangulation. If the robot moves along the border points, those will be relabeled as internal points. The points of the map with direct connection with the traversable area will also be used for the CDT and then this part of the map will be added to the rest of the map. The example of the robot moving in the map is in the Fig. 6.

3. Goal Point Selection

The selection of goal points is a crucial task in SLAM algorithms. It is important to arrange for a robot to first make a rough map of the neighbourhood and then specify it. The key point in the SLAM algorithms is the loop-closing. This is the only way for a robot to correct the odometry errors that arise during the exploration. The more the robot rides and especially turns, the more uncertain its position is. The goal,therefore, is to prevent the unnecessary wandering among close observed features. In the ideal case, the robot should go to the furthest feature (obstacle), observe its surroundings and continue to the second furthest obstacle. But it has to prefer going to the unknown area and not returning to the previously explored place. To arrange this, it is not sufficient to select the obstacles only acording to their distance. This could cause that the robot goes back and forth between two obstacles. Therefore, the way proposed here is to choose obstacles according to their distance from the explored borders and the travel cost to this border.

As the robot moves, it marks the borders between the visited and therefore traversable area and the unvisited unknown area. The detected features will lie behind this border (or above the traversable area - lights, ceiling, etc.). The distance to the closest point on the border will be computed. The larger this distance will be, the bigger will the attractive power of the point be. Then, the shortest path from the point on the borders to the robot will be computed. This will represent the repulsive force of the detected point. The resulting attractive force is therefore computed as

$$A_P = d(P_p, P_b) - \eta G(v_S, P_b) \qquad (1)$$

where A_P is the attractive power of the point P, $d(P_p, P_b)$ is the distance from the point P_p which is projection of the point P to the traversable area to its closest point on the border P_b. $G(v_S, P_b)$ is the cost of travelling from the current robot position v_S to the border point P_b, η is the weight of this value. It is recommended to set it between 0 and 0.5.

The point with the highest attractive power will be the next goal point for the robot. We can assume, that this point will be the one detected in the last observation. This will push the robot away from the explored area. In the case that no point will have the distance $d(P_p, P_b)$ bigger then chosen constant $\mathcal{K}$, the robot will switch to the second part of the exploration, the map specifing. In this part the mapping will be based on the frontier-based exploration, and the robot will go to the closest obstacle. If during this part of the exploration robot detects some point with bigger distance then $\mathcal{K}$, it will be switched back to the first phase of exploration.

If the robot is not be able to step on any part of the borders because it would mean the colision with the obstacle the exploration and the map making will be completed. However, only the main part of the exploration will be completed. We assume that the robot uses the cameras to detect the points and this means that even a small change of the light can cause discovering of the other points. So the map making in fact will continue as long as the robot will move in the map.

4. Path Planning

Due to the chosen map representation, the path planning is reduced to the searching in the graph. The question is what properties we expect from the planned path. The path can be the shortest one, the fastest one, the safest one etc.

If we are going to mark the traversable area by the previously described way, we have to arrange for the robot to use slightly different path every time it moves. Otherwise, we will end up with the corridors with the same width as the width of the robot. In the ideal case, the two paths would overlap slightly so that they can be connected into one. As we described earlier, there are inner points in the traversable area and the border points which are connected by the constrained edges. Thus we can use both kinds of points for the path planning. If we want to plan the path without checking the traversability, we can use the inner points. If we use the border points, we have to check the traversability, but as the outcome, we will also extend the known area.

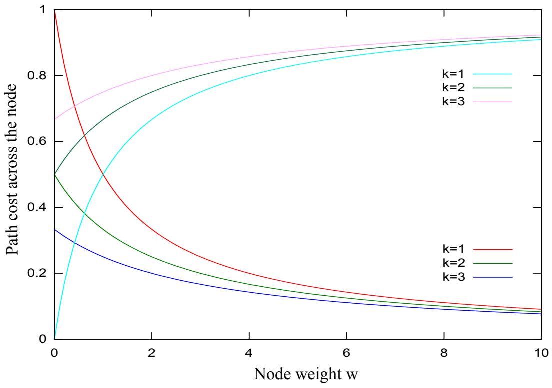

Figure 4: The dependency of the path cost on the node weight w.

To be able to control the properties of the path we will use the analogy with the potential fields. Every vertex, or node, will have the value w that will indicate the number of nodes from the current one to the border one. This allows us to plan shortest or safest path and also the exploratory path.

The cost of the path over the examined node is computed as

$$1 - \frac{1}{w + k} \qquad (2)$$

because in the case of choosing the exploratory path, we want the robot to travel along the border points, but still, we want to minimize the cost of the path. On the other side, the weight of nodes in the case of the safest path will be

$$\frac{1}{w + k} \qquad (3)$$

The $k > 1$ in both equations is added because we do not want the result value for the border edge to be infinite and we can also penalize the nodes with the higher distance from borders by this value (see Fig. 4). The cost of the exploratory path will be computed as

$$G_E(v_S, v_N) = \sum_{n=1:N} \left(1 - \frac{1}{w_n + k}\right) d(v_{n-1}, v_n), \qquad (4)$$

and in the case of the safest path

$$G_s(v_S, v_N) = \sum_{n=1:N} \frac{1}{w_n + k} d(v_{n-1}, v_n). \qquad (5)$$

where v_S is the starting node, v_N is the N node, w_n is the weight of the node n and $d(v_{n-1}, v_n)$ is the distance between nodes v_n and v_{n-1}.

The function that we want to minimize is thus

$$F(v_S, v_G) = G_i(v_S, v_N) + H(v_N, v_G), \qquad (6)$$

where $H(v_N, v_G)$ is the heuristics from the node N to the border node v_G. The $G_i(v_S, v_N)$, where $i \in (E, s)$, is either the exploratory or shortest path.

The shortest path will serve as a measure for the length of the exploratory path. If the ratio

$$p_r = \frac{p_E}{p_s} \qquad (7)$$

of the length p_E of the exploratory path and length p_s of the shortest path will be too high, the shortest path will be used.

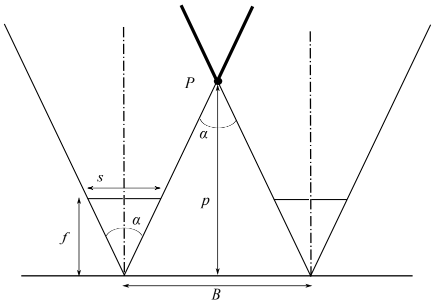

Figure 5: The closest point *P* which is able to be seen by the two cameras in their distance of *d* and with the viewing angle *α*.

4.1 Traversability Testing

After the path is planned we have to test it to see whether it is traversable. First, we have to define some safety area around the robot. This is important because the map is probabilistic, so the robot does not know the exact positions of the points. Here, we created the imaginary rectangle around the robot with length of the sides set for example the double of the width and the length of the robot. In any case, the safety space in front of the robot should be bigger than the distance of the closest point seen by both robot cameras. This is the closest distance to the obstacles the robot should approach. Moving closer than this point is dangerous, because robot would not be able to detect the approaching point. As we can see in the Fig. 5, from the similar triangles we get

$$\tan\frac{\alpha}{2} = \frac{f}{s/2} \tag{8}$$

$$\tan\frac{\alpha}{2} = \frac{p}{d/2} \tag{9}$$

As a result, the equation for the distance p of the closest point $\mathbf{P}$ is

$$p = d\frac{f}{s} \tag{10}$$

where d is the distance of the two cameras, f is the focal length and s is the size of the camera sensor. We will get all three parameters at the calibration of the stereocamera. After the whole path is planned, the space, including the safety area of the robot, is marked as the potential way of the robot. Now we have to test the points of the map for the possible collision with the planned path. The points which will be tested are all points with z coordinate smaller than the height of the robot, again with some safety distance above the robot and those points with the direct triangle connection to the borders of the traversable area.

If no point of the map lies in the planned path, the path is free and the robot can begin its movement. In the opposite case, the central point of the footprint which collides with some point of the map has to be labeled as non-traversable and this part of the path has to be replanned. Labelling of the non- traversable border points is important because it prevents from the repeated attempts to plan the path along this point. One of the goals of the robot exploring the neighbourhood should be to label all border points as non-traversable.

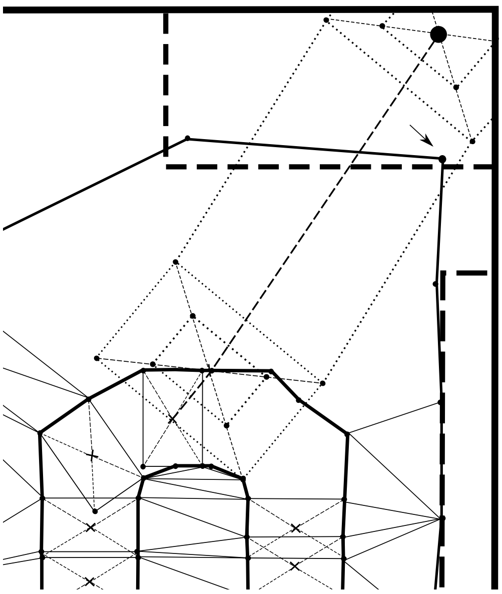

Figure 7: The planned direct path to the point in the upper right corner. The robot and its safety area is shown in dotted lines. The arrow shows the point that is in the collision with the planned path. Triangulation in this part of the figure is ommited.

4.2 Obstacle Avoidance

By the obstacle avoidance we mean avoiding the new points during the direct way to the goal point. The z coordinate and the connection to the borders also have to be tested for the points the robot detects and adds to its map during the way to the goal point. If the new point restricts the planned path, the robot will plan a new path. As the proposed way of path planning is not fine enough for obstacle avoidance, we have to use different algorithm for this task. The suitable algorithm for this is probabilistic path planning, especially the single-query algorithms.

The important property of the probabilistic algorithms is that the length of the path can be shortened and smoothened. This can be used after the path is planned to find even shorter path than for example the A* algorithm. This is because these algorithms search the path only in the known space, while the path shortening probably found a path in the unexplored area. The disadvantage of this is that we have to test this path for the collisions. On the other hand, this is compensated by much shorter path found and larger discovered space.

5. Choosing the Direction of View

We assume that the stereocameras of the robot are on the pan-and-tilt unit that can turn left and right in the angle cca 90° and up and down. This equipment is quite common in the robotics so our assumption is not unqualified. The important thing here is that such a unit allows a robot to go in one direction and look in the other direction. Looking on the sides allows the robot to explore more areas and to discover free spaces behind corners.

For the SLAM algorithm it is important for the robot not to turn often. Every turning of the robot has a negative influence on the accuracy of the localization. Therefore, it is suitable to only turn the stereocamera and not

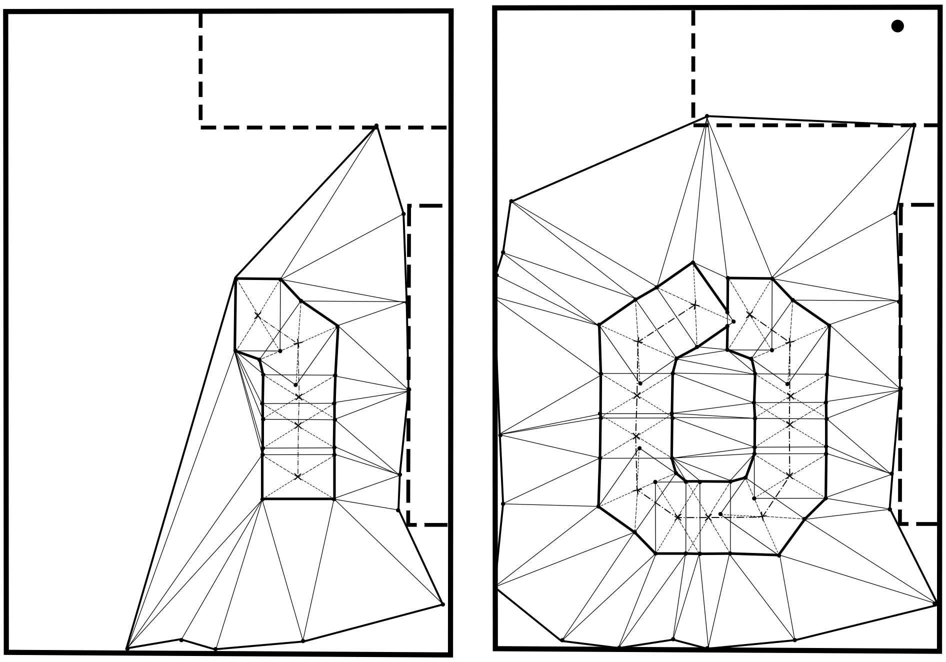

Figure 6: Mapping in the two phases. On the left figure the top view example of the map after four movements of the robot is shown. On the right figure the robot finished the ellipsoidal trajectory. The points with direct connection to the traversable area are shown and triangulated. The circle in the top right corner is the next goal point.

the whole robot. This means that we can use the stereocamera to watch the interest points while approaching the selected goal point. The points connected by the edge that are close to each other but differ a lot in their depth (distance from the camera plane) can be considered the interest points (Fig. 9). We can assume that the edge connecting such two points is wrong and therefore it is neccessary to explore the space between them and seek other points for better approximation of the surface.

The level of interest depends on:

- the angle of the edge to the point of view - the smaller the angle, the bigger the level of interest
- the viewing angle of the both end points of the edge - the smaller it is, the bigger is the level of interest
- the distance of the edge to the point of view - the smaller it is, the bigger is the level of interest
- the length of the edge - the bigger it is, the bigger is the level of interest

The importance of the length of the edge will be ommitted here as it depends on the size of the robot and on the requirements for the map. The angle of the edge can be aproximated by the ratio r of distances to both points. This can be combined with the distance to the edge. The thresholds for the ratio have to be set empirically. For example, if $r \in \langle 0, 0.5\rangle$, the edge will be marked as potentially interesting. If $r \in (0.5, 0.95\rangle$, the edge will be marked as potentially interesting, but it will not be observed by the cameras at this time. If $r \in (0.95, 1\rangle$, the edge will not be marked as interesting, because its distance is too big, or the edge is too short.

The ratio r has to be combined with the viewing angle φ. This will prevent marking the edges that will be observed from point too close to the one of the enpoints as interesting. If the edge satisfies both requirements, it will be marked as interesting and the robot will turn the cameras to observe it. So the conditions that the edge has to fulfill to be marked as interesting and observed are

$$r \in \langle 0, 0.5\rangle \land \varphi \in \langle 0, \pi/6\rangle \tag{11}$$

where $r = p_c/p_f$. The further point is p_f and p_c is the closer point. The value $\pi/6$ is again set empirically.

If the edge fulfills only the weaker condition $r \in (0.5, 0.95\rangle$, the edge will be marked as interesting, but since it is not too close or the edge is not too long, it will be observed next time.

It is important to state here that the level of interest has to be recomputed as the robot moves and sees the points again. This is because the robot can move to the place from where the points will have similar depth without seeing any other point between them. This means that these two points can not be longer considered as interest points. On the other hand, if the edge was once marked as non interesting, there is no need to recompute it again. This is because robot already saw the points from the viewpoint from where their depths were similar without seeing any point between them.

The locus of points that fulfills the first condition will form the circle (more exactly the interior of the circle), that is called Apollonian circle. The second condition also forms the Apollonian circle as seen on the Fig. 8.

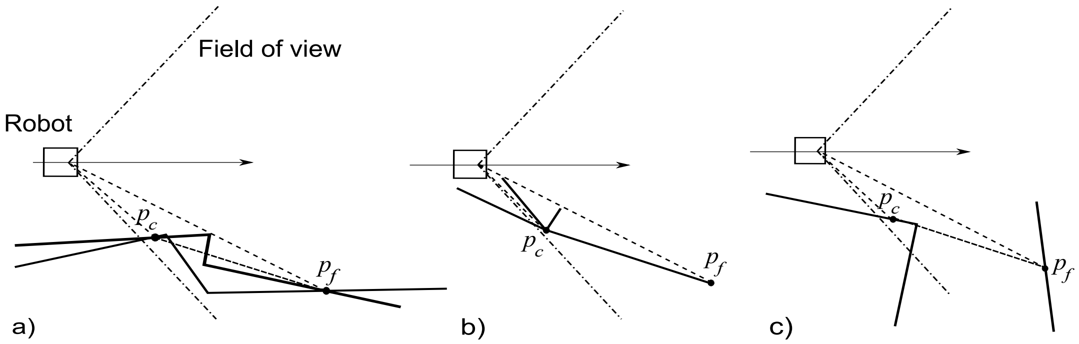

Figure 9: The examples of the various kinds of surface shapes shown in 2D. There is a robot and its field of view (FOV) shown together with the direction of the robot's move. The points p_c and p_f are closer and further interest points connected by edge (bold dashed line) in the map. The bold solid line shows the possible shapes of surfaces. On the Fig. a) there is the corner between the points p_c and p_f. On the Fig. b) there is an example of the four possible configurations of the thin obstacles (e.g. doors). Only the last one corresponds to the surface shape correctly. On the last Fig. c) there is an exapmple of the two points that are not connected directly in the reality, but they were connected by the edge in the map. This mistake can be corrected by the robot finding other points between these two points.

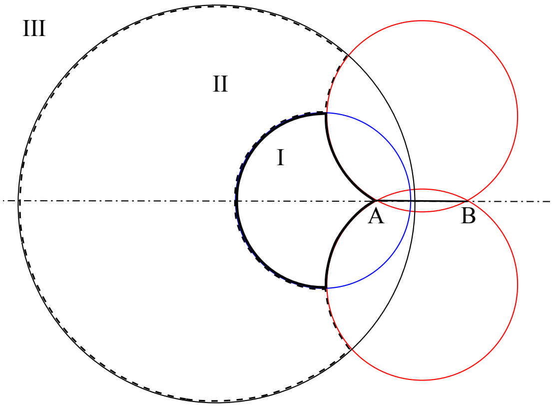

Figure 8: The Apollonian circles of the points A and B. If the robot is inside the area I, the edge $A - B$ is marked as interesting and it will be observed by cameras. From the area II the edge will also be marked as interesting, but it will not be observed. Outside these two areas the edge will not be marked.

5.1 Watching of the Pair of the Interest Points

The behavior of the robot should be similar to the behavior of the human when entering new area. The humans look left and right to see as much space as possible and they even can slow down their walk to observe the explored space better.

When the robot detects a pair of the interest points, it will try to keep the closer point in its field of view. This means that from the point c_s the robot will start the rotation of the pan-and-tilt unit. If the velocity v is be too big and the angular rotation of the cameras is not be able to catch turning the stereocamera to keep the closer point in its field of view, it will slow down its speed v accordingly. The rotation will continue until the cameras reach the maximum angle, which we assume is 90°. After the robot gets to the point c_m, where the closer point leaves its field of view the cameras will start to turn back to the normal position. The angular velocity of the camera ω and the velocity of the robot v have to be synchronized so in the point c_l the border of the field of the view goes through the further point and is perpendicular to the direction of the movement of the robot. The images here show full rotation of the cameras, but if the robot does not detect any other points between p_c and p_f the cameras will return to the normal position after the edge is recomputed as non-interesting.

The best viewing point the observing the pair of the points of interest would probably be from the line perpendicular to the center of the edge between two points. However, this line does not have to intersect the planned path of the robot or the intersection could be far away. This means that this approach is not suitable, so instead of this, we use the one described previously. The other issue that needs to be solved is the point where to turn the cameras back to the normal position. The ideal solution would be to keep the cameras turned in their maximum angle until the farther point disappeares. But this would be dangerous, because the robot would not be able to see the space in front of it. On the Fig. 11 we can see that if we start to turn the cameras back after the closer point disappears, the difference between the areas explored using these two ways is minimal compared to the travelled distance.

The distance a, where the point p_c dissapears from the sight of the robot is

$$a = b\frac{\cos\beta}{\cos\alpha} \tag{12}$$

The angle, that the robot has to turn is then

$$\delta = \frac{\varphi}{2} - \arctan\frac{b}{a} \tag{13}$$

The point c_l can happen to be in front of the point c_m. This would mean, that the robot has to start to turn the cameras before the closer point leaves the field of view. But such behavior is undesirable because it would decrease the explored area, so the starting to turn back at point c_m will have the higher priority than the property that the robot has to see the farther point from the perpendicular angle at the point c_l.

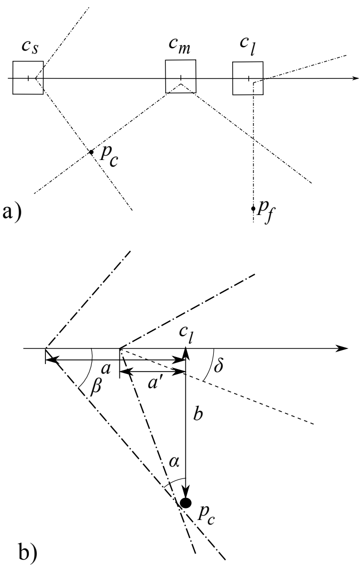

Figure 10: a) The example of the robot moving from left to right as is indicated by an arrow. On the point c_s the robot starts to turn cameras to right to keep the point p_c in its FOV. On the point c_m the robot lost the point p_c from its FOV and started to turn the cameras back. On the c_l the border of its FOV is perpendicular to the direction of the movement and goes through farther point p_f. At this point the robot loses the point p_f from its FOV. b) The robot starts to turn right in the distance a from the point where the line perpendicular to the direction of the robot intersects the point p_c. In the distance a' the robot is turned in the angle α.

5.2 Alternating Between Left and Right Sides

It is easily conceivable that during the exploration the robot has to look on the other side to explore a new pair of points before it finishes the exploration of the pair of points on the first side.

We can set the human behavior as an ideal example of the exploration. If a human during his walk needs to explore areas on his both sides, he starts to explore the closer one and when he thinks that the other would disappear from his field of view, he remembers the explored area so far, quickly turns head to the other side and explores this side. And again, when he thinks the border of the explored area can disappear, he remembers the explored area on the second side and turns back to see the borders of the previously explored area on the first side. If the two areas (points) are on the levels that are too close, the speed of walking can be decreased up to stopping to explore the areas around him better.

Similarly to the humans, the robot will explore the closer point and start to turn the camera to the other side to explore another point. If it finds that its velocity is too big and it does not catch the other point, it will slow down its speed v accordingly or it can even stop if the two points are on the same level.

A robot is not able to remember the explored area in the way the humans do and then to find the borders again. But it can mark the point P on the edge between two points of interest which was seen from the place from where the border of its field of view was perpendicular to the direction of its movement. After the robot finishes the exploration on the other side, it has to turn back to see this point again. This means, that the farthest position from where the robot can see this point is c_f. This point P_A is the replacement for the remembered borders of the area explored by a human.

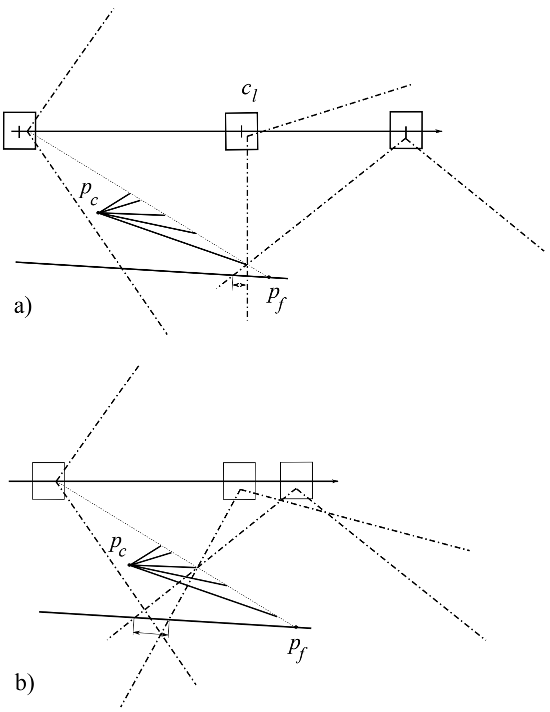

Figure 11: The difference between the explored area between the robot which is turning its cameras back and the robot that keeps the cameras in the perpendicular position.

6. Vertical Movement of the Cameras

During the exploration, there it can happen that one point from the pair of the interesting points is above the field of view (FOV) of the robot. This situation can also occur when the robot approaches the projection of the point to the traversable area. In this situation, the robot should lift the cameras so that the point is in the center of its FOV. The relation between the distance to projection of the point P and the angle of the camera axis can be computed as follows:

$$\alpha = \arctan\left(\frac{h-l}{d}\right) - \frac{\beta}{2} \tag{14}$$

7. Conclusions

This paper presents the proposed method for marking of the traversable area that has been created by a robot during an exploration. This marking should serve for the path planning in the 3D map created by the stereocamera. The significant feature, the detection alone, does not produce the maps that are suitable for the path planning. That is why we need to explicitely mark the traversable area, where the path planning can be performed. The traversable area is created with the help of the Constrained Delaunay Triangulation. The con-

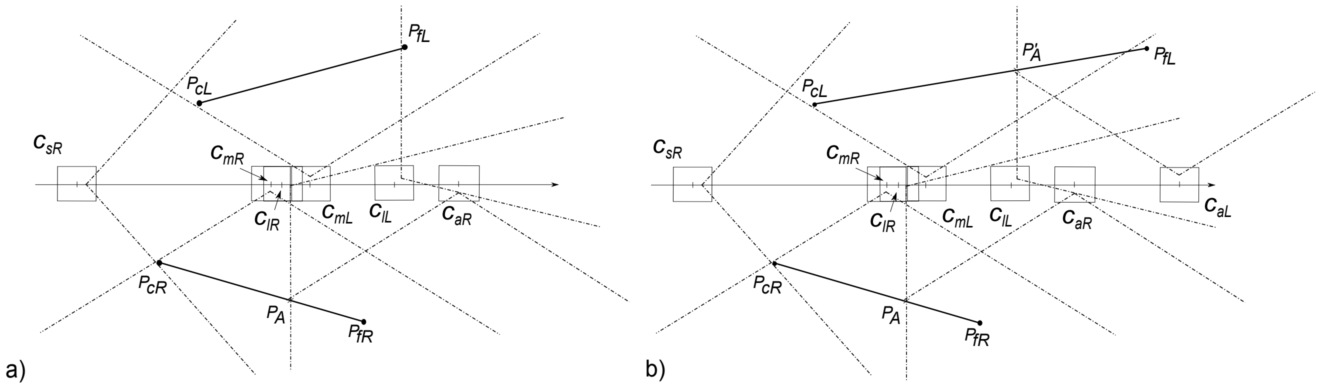

Figure 12: The robot on the Fig. a) turns the cameras right to watch point p_{cR} as in the previous figure. But between points c_{mR} and c_{mL} it has to slow down and turn cameras 180° left to watch point p_{cL} on the oposite side, which is almost on the same level as the point p_{cR}. As the robot did not complete the exploration of the edge $p_{cR} - p_{fR}$ it marked the point p_A where the perpedicular border of FOV intersected the edge and after exploration of the edge $p_{cL} - p_{fL}$ it turned right to get the point p_A to its FOV again and completed the exploration of the edge $p_{cR} - p_{fR}$. There is a similar example on the Fig. b) but the robot had to turn right to complete the exploration of the right edge before it completed the exploration of the left edge. Therefore it marked the point $P_A^{'}$ on the left edge and after completing the exploration of the right edge it turned back to the left edge to finish its exploration.

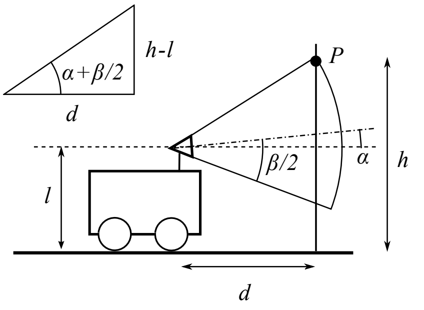

Figure 13: The robot is observing the point P from the distance d. α is the angle the robot has to lift its camera to get the point P to the field of view.

strained edges are made by the robot movements as a set of rules. The vertices on the borders of the traversable area are connected to the detected points and after the CDT is performed, it is incorporated to the 3D map.

The second part of this paper shows the way of choosing the most suitable landmark for the observation to explore the unknown area by a robot with a stereocamera using the SLAM algorithm for map making. We also describe the method for observing the interest points which allows the robot to perform a behavior similar to what the humans do during the exploration of an unknown environment. This method is based on searching for the interest points and its observing by the cameras on the pan-and-tilt unit.

The next work will focus on replacing the safety area around the robot by uncertainty of the particular points. This will help the robot to get closer to the obstacles and also explore the border edges more accurately. Other work will consist in sorting the detected points into more categories such as floor, walls etc., which will help more sophisticated task planning.

References

[1] M. Agrawal and K. Konolige. Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS. In *ICPR*, 2006.

[2] B. Clipp, R. Raguram, J.-M. Frahm, G. Welch, and M. Pollefeys. A Mobile 3D City Reconstruction System.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time Single Camera SLAM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29:1–16, 2007.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *IEEE Robotics and Automation Magazine*, 2:9, 2006.

[5] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, 22(6):46–57, 1989.

[6] D. Ferguson, A. Morris, D. Hähnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, W. Whittaker, W. Whittaker, W. Burgard, and S. Thrun. An Autonomous Robotic System for Mapping Abandoned Mines. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2003.

[7] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard. Autonomous Exploration for 3D Map Learning. In *Autonome Mobile Systeme (AMS)*, pages 22–28. Springer, 2007.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[9] H. Moravec and A. Elfes. High Resolution Maps from Wide Angle Sonar. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pages 116–121, 1985.

[10] F. A. Moreno, J. L. Blanco, and J. Gonzales. Stereo Vision-specific Models for Particle Filter-based SLAM. *Robotics and Autonomous Systems*, 57(9):955–970, 2009.

[11] B. Morisset, R. B. Rusu, A. Sundaresan, K. Hauser, M. Agrawal, J.-C. Latombe, and M. Beetz. Leaving Flatland: Toward Real-time 3D Navigation. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3384–3391, 2009.

[12] P. Moutarliere and R. Chatila. An Experimental System for Incremental Environment Modeling by an Autonomous Mobile Robot. In *The First International Symposium on Experimental Robotics I*, pages 327–346, London, UK, 1990. Springer-Verlag.

[13] D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry. pages 652–659, 2004.

[14] M. Pollefeys, R. Koch, M. Vergauwen, and L. V. Gool. Metric 3D Surface Reconstruction from Uncalibrated Image Sequences. In *3D Structure from Multiple Images of Large Scale Environments. LNCS Series*, pages 138–153. Springer-Verlag, 1998.

[15] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed Real-Time Urban 3D Reconstruction from Video. *International Journal of Computer Vision*, 78:143–167, 2008.

[16] R. Sim, P. Elinas, and M. Griffin. Vision-based SLAM Using the Rao-Blackwellised Particle Filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005.

[17] R. Sim, P. Elinas, M. Griffin, A. Shyr, and J. J. Little. Design and Analysis of a Framework for Real-time Vision-based SLAM Using Rao-Blackwellised Particle Filters. In *Proceedings of CRV*, 2006.

[18] R. Sim, P. Elinas, and J. J. Little. A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM. *International Journal on Computer Vision*, pages 303–318, 2007.

[19] R. Sim and J. L. Little. Autonomous Vision-based Exploration and Mapping Using Hybrid Maps and Rao-Blackwellised Particle Filters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)*, pages 2082–2089. IEEE, 2006.

[20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[21] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, Scott, S. Thayer, and W. Whittaker. A System for Volumetric Robotic Mapping of Abandoned Mines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4270–4275, 2003.

[22] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker. Autonomous Exploration and Mapping of Abandoned Mines. *IEEE Robotics and Automation Magazine*, 11:79–91, 2004.

[23] B. Yamauchi. A frontier-based Approach for Autonomous Eexploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, Washington, DC, USA, 1997. IEEE Computer Society.

Selected Papers by the Author

J. Rozman. Metody plánování cesty robota. In *Workshop Vršov '06*. FEKT VUT, 2006. s. 149–152. ISBN 80-214-3247-0.

J. Rozman. Mobile Autonomous Robot. In *Proceedings of the 13th Conference and Competition STUDENT EEICT*. FIT VUT, 2007. s. 478–482. ISBN 978-80-214-3410-3.

J. Rozman. Grid-Based Map Making using Particle Filters. In *Proceedings of MOSIS '08*. MARQ, 2008. s. 186–192. ISBN 978-80-86840-40-6.

J. Rozman. 2D and 3D Motion Model with Uncertainty. In *Proceedings of XXXIth International Autumn Colloquium Advanced Simulation of Systems*. MARQ, 2009. s. 37–42. ISBN 978-80-86840-47-5.

J. Rozman. Grid-based map making using particle filters. *International Journal of Autonomic Computing*. 2009, roč. 1, č. 2. s. 211–221. ISSN 978-80-86840-47-5.

J. Rozman. Incremental Creation of a 3D Map with a Stereocamera. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA '10)*. Cairo, EG: IEEE, 2010. s. 4. ISBN 978-1-4244-8135-4.

J. Rozman. Sampling-Based Algorithms for the Motion Planning. In *Proceedings of the 16th International Scientific and Practical Conference MTT '10*. Tomsk, RU: IEEE, 2010. s. 110–112. ISBN 0-7803-8226-9.

J. Rozman. Visualization of a 3D Textured Model for the World Modeling. In *Proceedings of CSE 2010 International Scientific Conference on Computer Science and Engineering*. Košice, SK: TU v Košiciach, 2010. s. 314–319. ISBN 978-80-8086-164-3.

J. Rozman, F. V. Zbořil. Path Planning and Traversable Area Marking for Stereo Vision-based 3D Map Building. In *International Conference on Computational Vision and Robotics*. Bhubaneswar, IN, 2010. s. 39–44. ISBN 93-81361-25-8.

J. Rozman, F. V. Zbořil. Potential Fields and their use in Robot Navigation. In *Proceedings of XXVIIth International Autumn Colloquium ASIS 2005*. MARQ, 2005. s. 221–224. ISBN 80-86840-16-6.

J. Rozman, F. V. Zbořil. Trilobot Mobile Robot and its Using in Education. In *Proceedings of MOSIS '05*. MARQ, 2005. s. 188–195. ISBN 80-86840-10-7.

J. Rozman, F. V. Zbořil. A Concept of a Robot for the Robotour Competition. In *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*. Praha, CZ: CVUT, 2010. s. 6. ISBN 0-7803-8226-9.