Vision-Based Path Generation of Mobile Robot for Autonomous Driving in Structured and Unstructured Environments

Pangyu Jeong
Pangyu.Jeong@cs.utcluj.ro

Stefan Sobol
stefans@vision.utcluj.ro

Sergiu Nedevschi
Sergiu.Nedevschi@cs.utcluj.ro

Computer Science Department
Technical University of Cluj-Napoca
Str. Constantin Daicoviciu no. 15, Cluj-Napoca 400020, Romania

Abstract **– The path generation method used by a Mobile robot is dependent on the environment and the type of sensors used. In this paper we present a path generation method that uses a stereovision sensor and combines stereo images and single image processing. The path generation method uses *maximum possible driving region detection based on the local difference probability (LDP)*, *path generation based on the maximum possible driving region* *and path correction based on obstacle collision avoidance*. The presented path generation method can be used to drive a mobile robot in a large variety of structured or unstructured environments. Highway image sequences were used as environment simulation to test our method.**

I. INTRODUCTION

The path generation process is strongly dependent on the application environment. The environments can be divided in two main classes: indoor and outdoor. For each kind of environment there are two possible scenarios: one when the mobile robot knows a start and a goal position and the other when the mobile robot explores an unlimited area without having a goal position.

For the case when the mobile robot knows the start and goal position are many contributors (i.e. [1] [2] [3] [4] [5]). The methods presented there generate many possible paths and the optimal path is chosen, based on the shortest distance to the goal position. These methods use a predefined environment map and also an obstacles map, generated using range sensors. In [6] the mobile robot moving angle is decided in real-time using Fuzzy-Networks based on natural potential field (register networks mapped from the occupancy map). The number of possible moving angles generated from the Fuzzy-Network layer is limited by the user definitions. This means that detailed moving angles cannot be generated. If the method is extended then the operation will require extra time and the path will not be generated in real time. Authors of [7] use a laser scanner to obtain 3D information and the mobile robot is localized by comparing a predefined image map with the 3D reconstructed image map. There is a simple solution for the localization of the mobile robot on a predefined map: using a GPS sensor. However, this solution is not complete in a changing environment.

In the second case, when the mobile robot is exploring an unlimited area, there are not many contributors. This case is more related to autonomous vehicles and especially to mobile robots exploring unknown terrains ([8] [9]).

Our proposed method can be used in both cases. In this paper, we focus our attention on the mobile robot path generation. There is no predefined goal position for the robot. However, the goal position can be given and the process of choosing the new goal position every time a new path is generated can be skipped. The proposed path generation method deals with the optimal path finding regardless of the obstacles existence on the driving direction and with the optimal path finding with respect to the obstacles. The obstacles are detected using stereo reconstruction combined with object grouping.

The optimal path finding process consists of *maximum possible driving region detection* and *dynamic path generation, related to obstacle absence and to obstacle existence*. The maximum possible driving region detection is achieved by using a local difference probability (LDP). The main idea of LDP is to construct the pixel extension based on a difference between two pixels probabilities on gray scale images. If the difference is smaller than a threshold value then the initial seed is extended. The threshold is determined by averaging distances inside the selected sampled area. The sample area is selected from a single image, right in front of the mobile robot. There is a detailed explanation of the sample area selection and the extension of the seed in Section C.

The desired driving angle is obtained after the detection of the maximum possible driving region. The main steps of the process are presented below.

1) The obstacles, detected using stereovision are represented as rectangles in the image space.
2) Using a single image, the maximum possible driving region is detected using LDP. The region represents a map in the 2D image space.
3) Inside the driving region we choose a destination point by imposing some simple safety constrains. A new destination is choused in each image frame. For each new destination a new travel path is generated.
4) In the first stage, a strait path is generated. This is a linear path between the virtual mass center of the robot and the destination point.
5) The rectangles representing objects in the image space are adjust, in order to compensate as much as possible the reconstructions and grouping errors. Also, the rectangles are modified to meet the collision avoidance requirement, by considering the dimensions of the mobile robot projected in the image space at the object distance.
6) The path of the robot is updated with respect to the obstacles size and position. Only the obstacles that are nearby the mobile robot and that have an acceptable size are considered. This condition comes to reject fake objects detection.

In our experiments we evaluate our method in many real situations. The results demonstrate that our proposed method has encouraging characteristics from the dynamic path generation point of view:

i) The method can be applied in structured and unstructured environments.
ii) The path is dynamically generated by considering the current environment state.
iii) The complete 3D reconstruction is not necessary.
iv) High driving angle accuracy.

II. THE PROPOSED METHOD FOR DYMANIC PATH GENERATION

A. Overview of the entire procedure

The dynamic path generation is accomplished by combining two different procedures: one using stereo image pairs to detect obstacles and the other constructs a driving region using a single image.

The new path is generated using the 2D map generated by combining the results of the two driving region detection methods. Figure 1 presents a schematic overview of the entire dynamic path generation procedure. In the next sections we describe the driving region detection and the dynamic path generation process.

At the first step we use stereo image pairs to detect objects. The result of this step is materialized in a 2D map representing the projection of the detected 3D objects in the image.

The map provided by the first step of the algorithm is used by the second step to find a starting region for the image growing region process. This step is used to recheck the image regions marked as drivable (not covered by obstacles) by the previous step.

Using the map completed by the second step, a new path is generated. The path generation process represents a search in the 2D space of the shortest trajectory to the selected destination point.

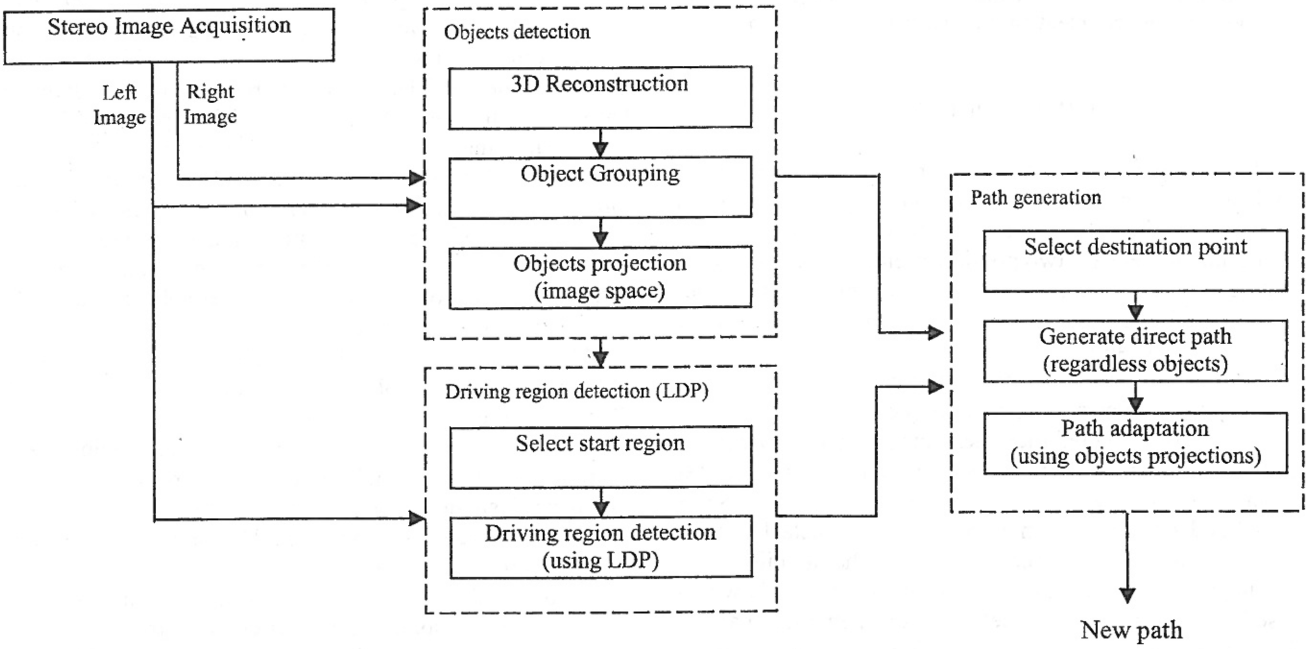


Figure 1. Overview of dynamic path generation process

B. Objects detection using stereovision

The objects are detected using stereo image pairs at 320X240 resolution. We use SVS (Small Vision System) to detect objects. In this paper we don't present a complete description of the stereo system. More details about stereovision can be found in [10].

The detected 3D objects are mapped in the image space as rectangle areas. To minimize the reconstruction and grouping errors we consider only the objects that are nearby the mobile robot and that have an acceptable size. By using this constrains we eliminate the fake detections.

The 2D projection of the objects is used to adapt the path to the destination to avoid obstacles.

C. LDP based maximum possible driving area detection

At this step we perform a possible driving area detection using an image region growing process. The process starts from the initial seed position of the sample selected area. The image selected sample area is chosen in front of the mobile robot by using a 2D map provided by the previous procedure to avoid taking an image sample totally or partial situated on an object. The case when the entire image is covered by an object or there is not enough space to move our robot must be treated like a special case. The decision taken in this situation must consider the robot capabilities (i.e. if the robot can rotate without moving then we can chose another driving direction if the acquired images confirm that is safe to move the robot, or, if the robot is not capable to rotate without moving the robot must stop).

The selected image sample gives us the starting point (initial seed). The growing condition of the initial seed is given by a threshold that is obtained by averaging distances among the probabilities inside selected sample area.
The probability is calculated in every pixel of the sample selected area. Figure 2 presents a graphical explanation of the seed extension.

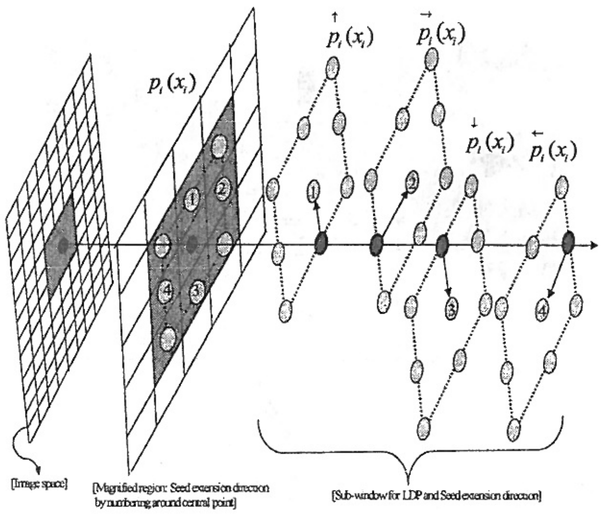


Figure 2. Seed extension direction and Sub-window for LDP

For every pixel included in the sample selected area the Local Difference Probability (LDP) is computed on a N4 type 3x3 neighbourhood.
Let's take a randomly selected point $x_{i(r,c)}$, where r and c are the row and the column of the input image. The four neighbours of the current pixel are $N_i = \{x_{i(r,c-1)}, x_{i(r-1,c)}, x_{i(r,c+1)}, x_{i(r+1,c)}\}$, where i stands for the neighbourhood identifier.
The LDP set in the N4 neighbourhood is:

$$P_i(x) = \{p_j(x) \mid j \in \{1,\cdots,5\}\} \text{ or:}$$

$$P_i(x) = \{p_1(x|x_{i(r,c)}), p_2(x|x_{i(r,c-1)}), p_3(x|x_{i(r-1,c)}), p_4(x|x_{i(r,c+1)}), p_5(x|x_{i(r+1,c)})\} \quad (1)$$

where the probability p_j must satisfy the following two conditions:

$$p_j \geq 0 \quad \text{and} \quad \sum_{k=1}^{m} p_{jk} = 1 \quad (2)$$

where m is the number of neighbours and p_{jk} is the probability of neighbours around p_j.

The components of equation (1) are described below.

$p_1(x|x_{i(r,c)}) = e^{\frac{-(x_{i(r,c)}-\mu_{(r,c)})^2}{\sigma^2_{(r,c)}}}$, where $\mu_{(r,c)}$ the mean of the 3x3 sub-window around (r,c) and $\sigma_{(r,c)}$ is the standard deviation of the 3x3 sub-window around (r,c).
The first order derivatives of the probabilities inside the 3x3 sub-window around (r,c) are:

$$dp_{(r,c)} = p_1'(x|x_{i(r,c)}) = (x_{i(r,c)} - \mu_{(r,c)})^2 p_1(x|x_{i(r,c)}) \quad (3)$$

$$dp_{(r,c-1)} = p_2'(x|x_{i(r,c-1)}) = (x_{i(r,c-1)} - \mu_{(r,c-1)})^2 p_2(x|x_{i(r,c-1)}) \quad (4)$$

$$dp_{(r-1,c)} = p_3'(x|x_{i(r-1,c)}) = (x_{i(r-1,c)} - \mu_{(r-1,c)})^2 p_3(x|x_{i(r-1,c)}) \quad (5)$$

$$dp_{(r,c+1)} = p_4'(x|x_{i(r,c+1)}) = (x_{i(r,c+1)} - \mu_{(r,c+1)})^2 p_4(x|x_{i(r,c+1)}) \quad (6)$$

$$dp_{(r+1,c)} = p_5'(x|x_{i(r+1,c)}) = (x_{i(r+1,c)} - \mu_{(r+1,c)})^2 p_5(x|x_{i(r+1,c)}) \quad (7)$$

We emphasize the distances between the central points and the mean around the central points using their square. The reason why we have to square the distance is that the initial seed is extended based on the difference between two Gaussian properties, therefore our attention is concentrated on the difference between them.
Each pixel has the Gaussian property. It means that if there is a great difference between two Gaussian properties obtained from two pixels we can assume that each pixel is included in two different regions; if there is a small difference between two Gaussian properties we can assume that each pixel is included in the same region. This is the main idea of LDP.
In order to arbitrate the relation between two Gaussian properties we need a threshold. The threshold value is given by the average of distances.
Using equations (3)~(7) we can compute the four distances around (r,c).

$$d_i = \{d_{(r,c)\to(r,c-1)}, d_{(r,c)\to(r-1,c)}, d_{(r,c)\to(r,c+1)}, d_{(r,c)\to(r+1,c)}\} \quad (8)$$

where

$$d_{(r,c)\to(r,c-1)} = \left|dp_{(r,c)} - dp_{(r,c-1)}\right|,$$
$$d_{(r,c)\to(r-1,c)} = \left|dp_{(r,c)} - dp_{(r-1,c)}\right|,$$
$$d_{(r,c)\to(r,c+1)} = \left|dp_{(r,c)} - dp_{(r,c+1)}\right|,$$
$$d_{(r,c)\to(r+1,c)} = \left|dp_{(r,c)} - dp_{(r+1,c)}\right|.$$

We discard the smallest and the largest distance values from the distance sets corresponding to the road sampled area. The distances average value is:

$$d_{th} = \frac{\sum_{i=1}^{M-r}\sum_{k=1}^{4} d_i(k)}{4*M-r} \quad (9)$$

where M is the number of sets and r is the number of discarded distances.
"d_{th} " value is used as the threshold value of the region growing algorithm.
Sometimes the initial seed is not extended completely to the entire driving region. This is because the average distance is obtained by random selection. It means that the threshold doesn't satisfy all distance variances between two local pixel probabilities in the selected sample area. Therefore we need an initial seed acceptance/rejection procedure.
The extended contour has to meet the following condition: the number of expanded points in the contour has to be greater than the number of pixels of the selected sample area. The seed position that satisfies equation (10) becomes the starting position of the seed.

$$\oint_{i \in M} x_i > X \quad (10)$$

If the initial seed satisfies Equation (10) then the extension procedure is performed, otherwise a new randomly selected sample area is required.

The evolution of the seed is decided by comparing the distance obtained from Equation (8) to the discriminator distance. The evolution of the seed position has to satisfy the following condition:

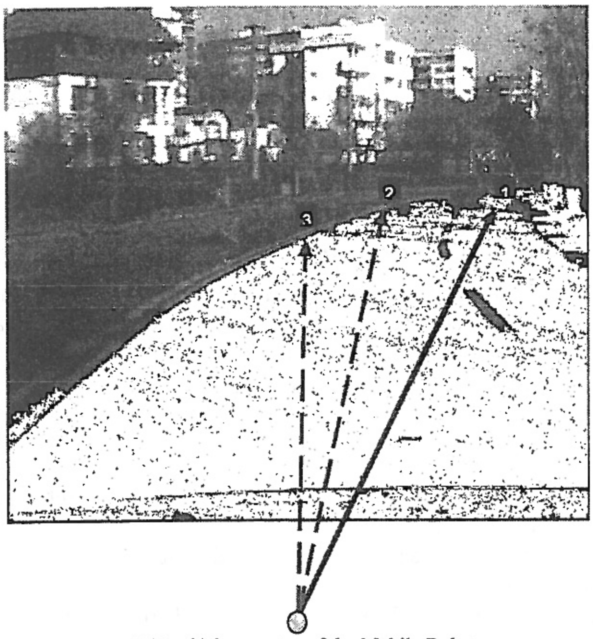
$$E(x)=\begin{cases} 1 & if \quad (\overleftarrow{d}_{(r,c)\to(r,c-1)} \le d_{th}) \| (\overset{\uparrow}{d}_{(r,c)\to(r-1,c)} \le d_{th}) \| (\overrightarrow{d}_{(r,c)\to(r,c+1)} \le d_{th}) \| (\overset{\downarrow}{d}_{(r,c)\to(r+1,c)} \le d_{th}) \\ 0 & if \quad (\overleftarrow{d}_{(r,c)\to(r,c-1)} > d_{th}) \| (\overset{\uparrow}{d}_{(r,c)\to(r-1,c)} > d_{th}) \| (\overrightarrow{d}_{(r,c)\to(r,c+1)} > d_{th}) \| (\overset{\downarrow}{d}_{(r,c)\to(r+1,c)} > d_{th}) \end{cases} \quad (11)$$

The result "1" represents that the seed can extend toward that direction, otherwise, the seed stops evolving.
The process is repeated recursively for each four neighbours of the extended seeds. The process continues until all the seeds stops extending (result "0" for condition (11)).

D. Detection of the maximum seed extension position and shortest path generation (regardless obstacles)

This is an intermediary step of the path generation process. Here we decide the new destination of the robot and its temporary linear trajectory.

The possible driving region is given by the LDP based seed extension algorithm. Each location of the driving region represents a possible moving point for the robot. The robot destination position must be a reachable one; therefore the area that the robot has to pass over when driving to the destination must have an acceptable number of driving region points inside.



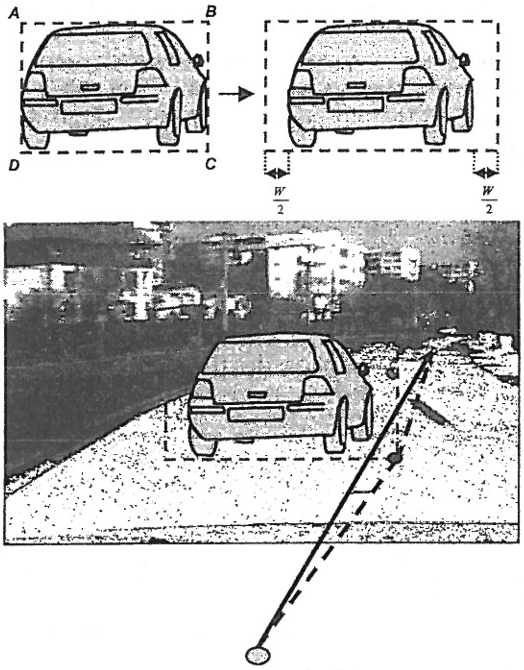
Virtual Mass center of the Mobile Robot

Figure 3. Desired driving angle and Desired moving direction based on the maximum extended seed position (1). 2 and 3 represents two other destination candidates.

The new temporary destination position for the robot is chosen in the image highest area covered by the extended seed. In Figure 3 are presented the chosen position (1) and other two position competitors for the robot destination (2 and 3).

Having the destination position we can describe the desired robot trajectory by connecting the destination position and the virtual mass center (Figure 3 - yellow dot).

The virtual mass center stands for the robot position representation within the image space. The position of the virtual mass center in the image is computed by knowing the camera position regarding the robot and the camera focal distance.



Original path
New generated path
Virtual Mass Center
Object area
Critical Point
W Robot width at the object distance

Figure 4. Example of dynamic path generation

E. Dynamic path generation regarding obstacles shown situation

When there are no obstacles in front of the mobile robot the path is generated very easy. The mobile robot can follow a linear trajectory to the destination point. In this case the trajectory is defined as a straight line between the virtual mass center of the mobile robot and the temporary destination. However, when there are obstacles in the front of the mobile robot that appear in the viewing range, the path has to be changed to avoid collisions.

When we adjust the path of the mobile robot we have to consider the errors that may appear in the objects detection process. Therefore, we consider only the nearest obstacles that have an acceptable size, to reduce as much as possible the fake obstacle detections, generated by noise facts.

The projections of the objects in the image space are represented as rectangles. To compensate the objects detection errors, we resize the objects contours by using information about the driving region.

Lets consider the points ***A(x,y)***, ***B(x,y)***, ***C(x,y)*** and ***D(x,y)*** as the corners of the rectangle. Where, ***(x, y)*** represents positions in the image space. The rectangle segments are: ***AB***, ***BC***, ***CD*** and ***DA*** (Figure 4). Only the ***BC***, ***CD*** and ***DA*** segments are directly related to the path generation process. Therefore, we check only these segments when we resize an object mapped area.
A vertical or horizontal segment ***S*** can be defined as:

$$S = \{I(x,y) \mid n \le x \le m, p \le y \le k\} \quad (12)$$

Where, n and m are the starting and the ending x position of the segment, p and k are the starting and the ending y position of the segment. The number of pixels covered by the segment is $m - n + k - p$. By using (12) we can define the ***BC***, ***CD*** and ***DA*** segments.

Each of these three segments is moved to the inner or outer direction until they pass the occupancy condition. This condition verifies the number of pixels covered by the segment that represents locations of the driving region. The segment that is covered over 80% by the extended seed points has to move to the central direction of the object rectangle area to reduce the occupancy. The segment that is under 80% occupied has to move to the outer direction of the rectangle. The 80% comes from the acceptable Gaussian property.

We also have to consider the mobile robot size. When we represent the robot by his virtual mass center, we assume that the robot has no spatial volume. This can be dangerous when the obstacles are very close to the robot. To prevent dangerous situations we resize the rectangles representing objects to avoid collisions when the robot is passing very close to the obstacle. We inflate the rectangle by moving ***AD*** and ***BC*** segments to the outer direction with half of the mobile width, projected in the image space at the object distance (Figure 4).

After resizing the objects projections we update the path of the mobile robot by searching a safe path to the destination.

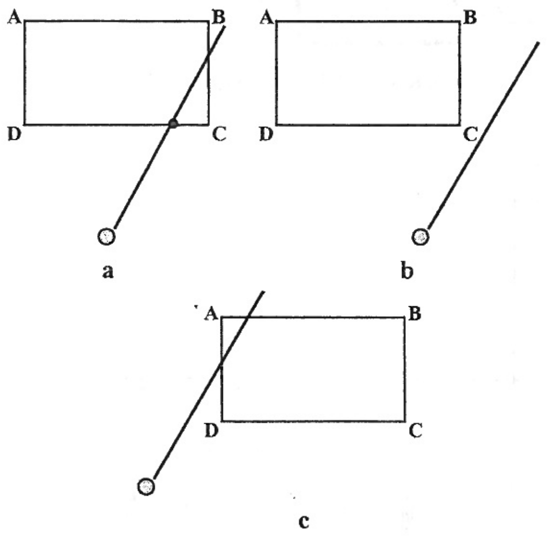


Figure 5. Critical point detection
a) The path is intersecting the **CD** segment
b) The path is not intersecting the **CD** segment
c) The path is intersecting the object area, but is not intersecting the **CD** segment

First we need to find which segments are intersected by the line constructed from the virtual mass center of the mobile robot and the temporary destination. In Figure 4, ***BC*** and ***CD*** are intersected by the original path that is generated regardless of the obstacles existence. However, only the intersection with the ***CD*** segment is relevant. If the path is intersecting the rectangle, but is not intersecting the ***CD*** segment then we consider that the mobile robot can safely pass. Figure 5 shows three possible situations that may appear: **a)** the path is intersecting the rectangle by passing over the ***CD*** segment, **b)** the path is not intersecting the object area and **c)** the path is intersecting the object area, but is not intersecting the ***CD*** segment. Therefore, we can test if the current path is safe by finding the intersection point between the path and the ***CD*** segment for each object. If the intersection is between ***C(x,y)*** and ***D(x,y)*** points then we have to find the critical point, otherwise the path remains unchanged. The critical point represents the closest object position that can be reached by the mobile robot.

If the path is crossing over the ***CD*** segment then the critical point will be ***C(x,y)*** or ***D(x,y)***. We choose the point that is closest to the intersection point. This point is used to update the current path.

If the temporary destination is placed inside an object area then the current destination is considered to be the detected critical point.

The path is updated step by step for each considered object. We start updating the path by computing the critical point for the nearest object and we continue checking the intersections with the remaining objects. At each step we consider the intersection of the current object with the latest path. Objects are taken from the nearest to the farthest.

The final path is generated by connecting the robot virtual mass center, critical points and the destination.

III. EXPERIMENT

The method presented in this paper was tested using an outdoor image sequence (1000 images). The main aim of the experiment was to check the path adaptation in the obstacles shown situation. We used a highway scenario because is a very dynamic environment.

For each frame a new temporary destination was chosen and the path was updated with respect to the detected obstacles. We consider only the obstacles nearby the mobile robot. The maximum range for detecting obstacles is strongly dependent on the stereo system.

Figure 6 presents a short image sequence (only the left image) with the generated path for each image. When the path is not intersecting an obstacle then the path remains unchanged (strait line to the destination).

In Figure 7 first row we depict the road curvature for the image sequence presented in Figure 6. The road curvature is very important for a long range path generation. If the road is not flat and we have no knowledge about the road curvature then we cannot directly measure the distance to a trajectory point. But if we generate a short distance path we can assume that the road is flat (for the highway scenario).

The second row in Figure 7 presents the driving direction for the current frame. The driving direction is given by the first segment of the detected path. The result looks like a noise carrying signal because the position of the destination is chosen using the extended seed and also the size of the detected objects is not constant. The destination is placed in an irregular position due to the characteristic of the seed extension probability. To achieve a softer variation of the driving direction we applied a Kalman filter on the generated driving angle and on the destination position. The value of the Kalman filter measurement error for the driving angle is directly related to the detection errors of the first object avoided by the current path. If the path is a strait one to the destination point then the measurement errors are directly related to the process of destination choosing. The final result is presented in the third row of Figure 7.

IV. CONCLUSION

We proposed a simple approach for mobile robot path generation. Our approach combines stereovision with single image processing. The path generation process has two main phases: direct path generation and path updating to avoid collisions. The accuracy of the generated paths can be improved by using high quality stereovision equipment.

The method was tested using an outdoor image sequence and the results are encouraging. Tests were shown that our approach can be used in structured and unstructured environments.

V. REFERENCES

[1] Goel, A.K.; Ail, K.S.; Donnellan, M.W.; Gomez de Silva Garza, A.; Callantine, T.J., " Multistrategy adaptive path planning", *IEEE Intelligent System*, vol. 9, no. 6, Dec. 1994, pp. 57-65.

[2] Laubach, S.L.; Burdick, J.W., " An autonomous sensor-based path-planner for planetary microrovers", *IEEE International conference on Robotics and Automation*, vol. 1, May. 1999, pp. 347-354.

[3] Tsourveloudis, N.C.; Valavanis, K.P.; Hebert, T., " Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic", *IEEE Trans. Robotics and Automation*, vol. 17, no.4, 2001, pp. 490-497.

[4] Aurelio Piazzi, Corrado Guarino Lo Bianco, Massimo Bertozzi, Alessandra Fascioli, and Alberto Broggi, " Quintic G2-splines for the Iterative Steering of Vision-based Autonomous Vehicles", *IEEE Trans. Intelligent Transportation Systems*, vol. 3, no.2, March, 2002, pp. 27-36.

[5] Pedrosa, Diogo P.F.; Medeiros, Adelardo A.D.; Alsina, Pablo J.," Point-to-Point Paths Generation for Wheeled Mobile Robots", *IEEE Int'l Conf. Robotics and Automation*, Sept, 2003, pp. 3752-3757.

[6] Nikos C. Tsourveloudis, Kimon P. Valavanis, and Timothy Hebert," Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic", *IEEE Trans. on Robotics and Automation*, Aug, 2001, vol. 17, no.4, pp. 490-497.

[7] Fawzi Nashashibi, Michel Devy, Philippe Fillatreau," Indoor scene terrain modeling using multiple range images for autonomous mobile robots", *International conference on Robotics and Automation*, May, 1992, vol. 1, pp. 40-46.

[8] Davids, A.," Urban search and rescue robots: from tragedy to technology", *"IEEE International conference on Intelligent System*, April, 2002, vol. 17, no.2, pp. 81-83.

[9] Aboshosha, A.," Adaptation of rescue robot behaviour in unknown terrains based on stochastic and fuzzy logic approaches", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct, 2003, vol. 3, pp. 2859-2864.

[10] http://www.ai.sri.com/~konolige

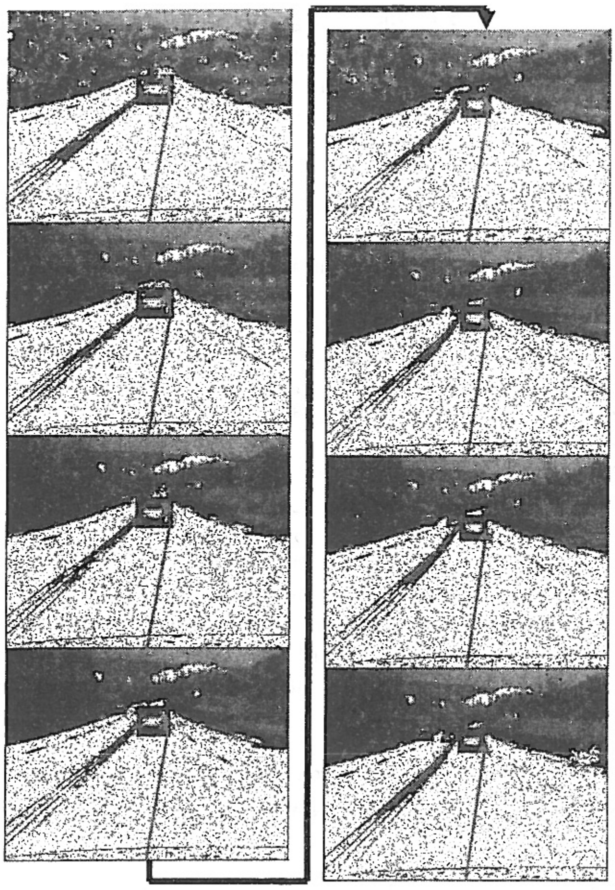


Figure 6. Results of the dynamic path generation

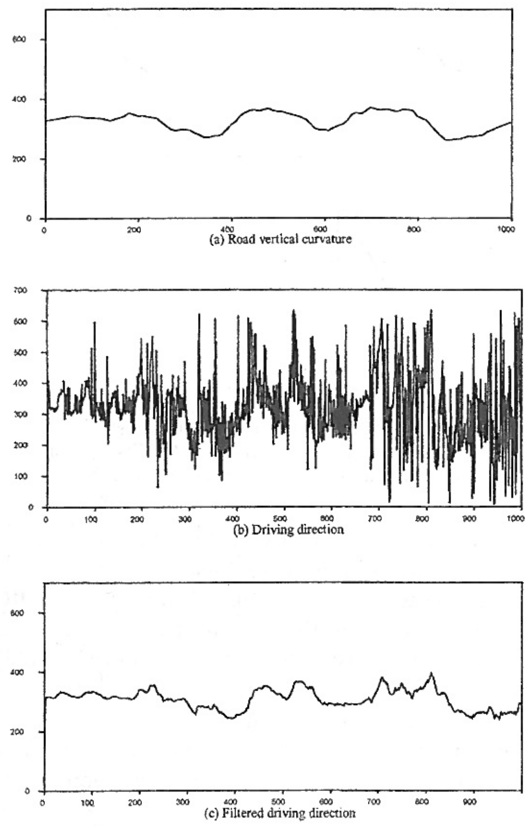


Figure 7. Simulated path generation for mobile robot autonomous driving