2005 IEEE International Conference on Robotics and Biomimetics

Real-Time Region-Based Obstacle Detection with Monocular Vision

Hui Wang Kui Yuan Wei Zou Yizhun Peng
Institute of Automation
The Chinese Academy of Science
Bei Jing, China
{hwang ,yuankui, wzou, yzpeng}@hitic.ia.ac.cn

Abstract - **In this paper, a region-based obstacle detection algorithm for indoor navigation is presented. Firstly, the image is segmented into regions. Then each individual region is classified as belonging either to an obstacle or the ground based on its average 'disparity'. The proposed method only uses a single passive color camera, and can provides a local obstacle map at high resolution in real-time. Experimental results are given for real indoor scene images to validate the efficiency of this method.**

Index Terms - obstacle detection; monocular vision; stereo vision; inverse perspective mapping; region segmentation.

I. INTRODUCTION

Obstacle detection is one of the fundamental problems for the navigation of mobile robots. In order to navigate in the real world, it is necessary to detect those portions of the world that are dangerous or impossible to traverse. For a large class of navigation problems, the world in front of the robot can be modeled as a flat plane, and any point that deviates from the planar model can be considered to be obstacles. Some examples of these problems are indoor mobile robots, and practically all of the Automated Guided Vehicle (AGV) [1] .

Some mobile robots rely on range data for obstacle detection, such as ultrasonic sensors, laser rangefinders, and radar, etc. These sensors are inherently suited for the task of obstacle detection and can be used for many cases because they can directly measure the distances from obstacles to the robot. However, none of these sensors can perfectly accomplish the task of obstacle detection. Ultrasonic sensors are cheap but suffer from specular reflections and usually from poor angular resolution. Laser rangefinders and radar provide better resolution but are more complex and more expensive [2].

A lot of works has been done in which visual sensors are used to carry out the obstacle detection tasks [3,6]. These vision-based works can be classified into two categories, the stereo vision approach and the motion-based approach[8]. It has been well known that the stereo vision approach is very time-consuming and may be confronted with the matching problem in many cases. Although it has become possible to execute a stereo algorithm in real-time [4], the matching problem still remains unsolved. The pessimistic conclusion about this problem is that it is not solvable in the general case at all[7]. In the motion-based approaches, motion field is computed from the consecutive images obtained from the same camera, and other static or moving objects are then detected when their motions are dominant from the scene [10,11]. These methods work well in some types of scenes having dominant motions but their deficiency is that they can not provide 3d information of the environment and can not give correct results when both the robot and the environment keeps still.

Assuming that all obstacles are on the ground, obstacle detection problem can be considered to be a ground region search task from the image. The most common approach for region search problem is appearance-based methods, which directly search the ground using visual clues such as color, texture and edge [2]. These methods need prior knowledge about the environment and are subject to the lighting conditions. The method presented in Reference [9] uses multiple visual clues such as corner points, color and texture to estimate the homography and plane normal to detect the ground, but this approach is iterative and very time-consuming.

In this paper a new region-based obstacle detection method is presented which uses the sensor fusion techniques and has the following advantages:

1) The information from the odometer of the robot is used to obtain the locomotion between image sequences to avoid the time-consuming computing of motion estimation.

2) Only one camera is needed to get the image sequences, which assures the electrical and photometric identities between the image pairs.

3) Image is segmented into several regions and only the border pixels need to be matched between the image pairs, which mean that the computing expense can be reduced significantly.

4) Obstacle detection is simply done by threshold comparison, which means that expensive computation of 3D environmental information is avoided.

This paper is organized as follows: Section 2 describes the assumptions and the definition of the coordinate systems. Section 3 introduces the proposed algorithm in more detail. Experimental results from the application of the method on real images are presented in Section 4. The paper is concluded with a brief discussion in Section 5.

II. ASSUMPTIONS AND MODELS

This work is partially supported by the National High-Tech Research and Development Plan of China under Grant No. 2001AA422200 and the National Natural Science Foundation of China under Grant No. 60375037.

The purpose of obstacle detection is to avoid collisions between robot and obstacles. Therefore it is not necessary to reconstruct the 3D model of environment precisely[14]. Only the information helpful to obstacle avoidance is needed, such as the direction, distance and width of obstacles, etc. In this paper, we first make the following assumptions:

1) The robot is moving on a locally planar ground. This assumption is normally satisfied in indoor environment.

2) The camera is precisely calibrated beforehand and is mounted on the robot fixedly, as shown in Fig. 1. H and Θ are constant and Θ is not subject to large vibrations when the robot moves on the ground.

3) The odometric data corresponding to the image sequences in a small time interval are accurate enough in spite of the accumulative error during the long run. This assumption is reasonable because the accumulative error could be ignored in a short time period, e.g. one second.

Define three coordinate systems as in Fig. 1. C is the camera. The camera coordinate has the focal point O_c as its origin. The world coordinate system has origin at point O_w which is the intersection of ground plane and the perpendicular of the O_c to the ground plane. We can align the world coordinate system to the camera coordinate by performing a transformation of a translation vector and a rotation matrix. The coordinate axis Z_c is aligned with the optical axis and points away from the image plane. The image coordinate has axes aligned with the camera coordinate system, with X_i, Y_i lying in the image plane.

III. REGION BASED OBSTACLE DETECTION

A. Algorithm

P is a 3D scene point as illustrated in Fig. 2, $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are two images grabbed when robot moves from O to O'. P is projected to the image planes $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ as point x and x' respectively. Suppose that point P is on the ground, then there is a linear transformation between point x and x':

$$x' \to x = Hx' \tag{1}$$

Where x and x' are homogeneous coordinates of point x and x', H is the homography matrix induced by the ground plane of the two images. Then it can be determined whether point P is on the ground by comparing the similarity of the two pixels' color:

$$|\pi(x) - \pi'(x')| \begin{cases} \geq \Delta, x' \to obstacle \\ < \Delta, x' \to ground \end{cases} \tag{2}$$

where Δ is the threshold which represents the maximum allowed dissimilarity between the correspondence pixels. $\boldsymbol{\pi}$(x)=$\boldsymbol{\pi}$(Hx') and $\boldsymbol{\pi}'$(x') are the colour values of the pixels. This transformation results in the image of the ground being registered in the two views, leaving all obstacles extruding from the ground plane unregistered. Subtracting the first image from the warped one, we can declare points where the absolute value of the computed difference is above a threshold as belonging to obstacles. This method is the so-called inverse perspective mapping [5,12]. It brings the ground surface to zero disparity so that the feature extraction and matching procedures of traditional stereo vision methods are completely avoided in the obstacle

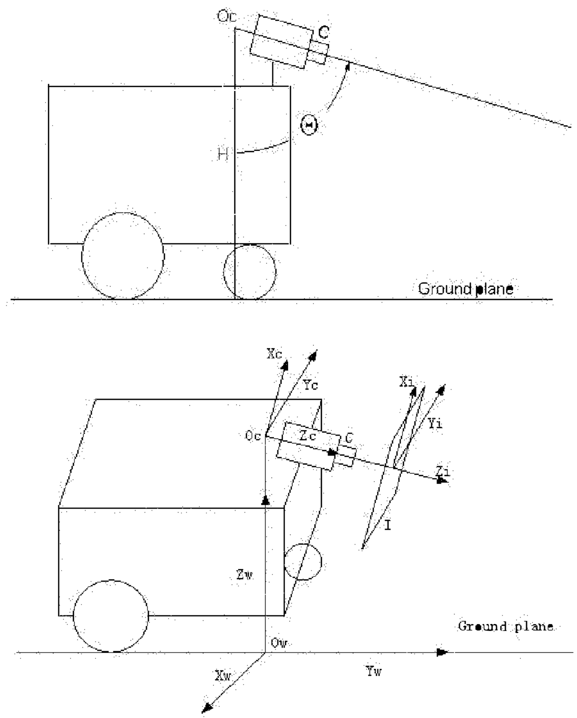


Fig.1 Geometrical model

detection algorithms. Although very efficient, it needs sufficient texture in the image to perform properly. This requirement can not be satisfied in a lot of cases for an indoor environment. In addition, this method suffers from the trivial vibrations which may occur when the robot moves around.

To overcome the above difficulties, we combine the inverse perspective mapping with stereo vision and apply the region segmentation technique to solve the matching problem in indoor environment.

Consider pixel x'' as the correspondence of x' in image $\boldsymbol{\pi}$ derived by some matching algorithm. It may deviate from its true position, x, due to mismatch. Then we can estimate whether the point P is on the ground by comparing the Euclidean distance between x' and x'' with a threshold D:

$$\|x'' - Hx'\| \begin{cases} \geq D, x' \to obstacle \\ < D, x' \to ground \end{cases} \tag{3}$$

Equation (3) is more robust than equation (2). Even H is not estimated accurately it still works quite well. Now the following problem is how to find the precise correspondent pixels between the two images. Unfortunately, it is very difficult for the indoor environment because it usually consists of many non-textured, flat object, e.g. walls and ground.

As we have stated in the beginning of this section, it is not necessary to perform a complete 3D reconstruction of the environment for the task of obstacle detection. This task can also be done in the following way: segmenting the obtained image into regions and then estimating whether these regions belong to ground or belong to an obstacle.

The region estimation is done according to the following criterion.

$$\frac{\sum_{x_i' \in R_j} \left\| x_i'' - Hx_i' \right\|}{k} \begin{cases} > D, R_j \rightarrow obstacle \\ \leq D, R_j \rightarrow ground \end{cases} \tag{4}$$

Where R_j is a region in image $\boldsymbol{\pi}$, x_i is the border pixel of R_j, x_i'' is the correspondent pixel of x_i' in image $\boldsymbol{\pi}$, k is the sum of border pixels of R_j. This algorithm can be executed following the steps given below:

1) Capture image sequences and save the last 20 images into the computer's memory. Read the odometric data and save the data to a file.

2) S*elect* a corresponding image $\boldsymbol{\pi}$' for the current image $\boldsymbol{\pi}$ from the image in memory according to the following criteria. As shown in Fig. 2, the distance d between the two images should greater than 50mm and less than 100mm, $|\theta|<5^{\circ}$.

3) Calculate the locomotion between the two images with the odometer data so as to get the homography of the selected two images.

4) Segment the current image into regions.

5) Classify every region as obstacle or ground according to (4).

6) Project the ground region to the world coordinate to get the passable map within the robot's view field.

B. Computing Homography from Odometric Data

After two images are selected from the image sequences, as shown in Fig. 2, the homography of these images could be derived by the odometric data, d and θ.

Let $[X, Y, Z, 1]^T$ be the homogeneous coordinates of point P in world frame, and $[u, v, 1]^T$ be the homogeneous coordinates of the same point's projection to the image plane $\boldsymbol{\pi}$. Assume that point P is on the ground plane, i.e. Z=0. Point P maps to the image plane as follows:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5}$$

Where s is an arbitrary scale, M is a 3×3 matrix containing information about the intrinsic and extrinsic parameters of the camera. Let $[X', Y', Z', 1]^T$ be the homogeneous world coordinates of point P after the robot moves to O' and $[u', v', 1]^T$ be the homogeneous coordinates of P's projection to the current image plane. Since Z'=0, then

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{6}$$

Where:

$$R = [r_1 \quad r_2 \quad r_3] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} d\cos\theta \\ d\sin\theta \\ \bullet \end{bmatrix}$$

When robot moves to O' from O, Equation (5) becomes as:

$$s'\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = M \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} \tag{7}$$

Where s' is an arbitrary scale.

By (5), (6) and (7), we can get that (s'' is an arbitrary scalc):

$$s''\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = M[r_1 \quad r_2 \quad T]M^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{8}$$

Then the homography H is:

$$H = M[r_1 \quad r_2 \quad T]M^{-1}$$

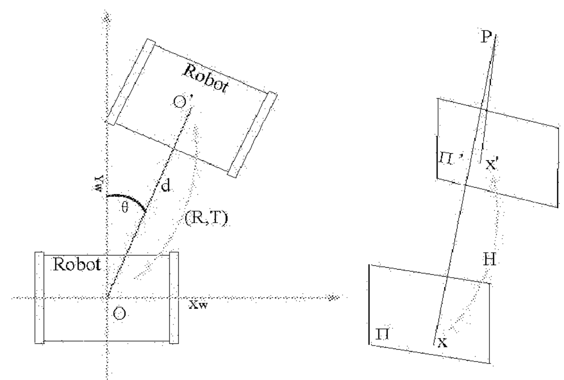


Fig.2 The robot moves from O to O' and captures two images: $\boldsymbol{\pi}$ and $\boldsymbol{\pi}$'

C. Region Segmentation

Region segmentation is the key step of the proposed method. The purpose of segmentation is to get individual regions so as to avoid matching the entire image pixel. The result of segmentation should meet the following two rules:

$$I = \bigcup_{i=1}^{s} R_i, R_i \bigcap R_j = \phi, i \neq j \tag{9}$$

$$R_i(p) = TRUE, i = 1, 2, \ldots, n \tag{10}$$

Where I is the image, R_i represents region i, n is the total number of the regions, p is a pixel in the image and $R_i(p)$ is a homogeneity evaluation of the pixel p. A fast segmentation algorithm based on color information is proposed. This algorithm is very effective and only need to scan the image for two times. This algorithm is executed by the following steps.

1) Every pixel is considered as a single region before processing. Each region's color value is defined as the mean R, G and B value of its pixels.

2) Define a criterion for merging two adjacent regions according to (10). If the color values of two adjacent regions, R_1 and R_2, satisfies

$$\begin{cases} |R_1(R) - R_2(R)| < E \\ |R_1(G) - R_2(G)| < E, \\ |R_1(B) - R_2(B)| < E \end{cases} \quad (11)$$

then merge them into one region and modify the new region's color value. E in (11) is a threshold which may be between 10~30.

3) Scan all pixels by row and column. Check the right and bottom neighbors of the current pixel in turn. If its color value satisfies (11), then it is merged into the current region and the current region's color value is modified.

4*)* The second scan: set the current pixel as border pixel if it's four neighbors do not belong to the same region.

The segmentation results are shown as in Fig. 3. The raw image is shown at first, and the segmentation result images with different E are shown as the following 3 images where border pixels are green. This method is so fast that it only takes 1ms to process a 320×240 image by a common PC.

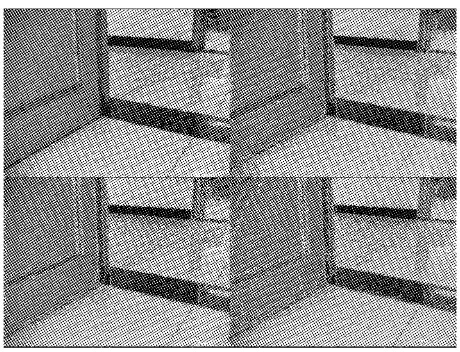


Fig. 3 Region segmentation

D. Stereo matching and Constraints

The good result of the region segmentation reduces the trouble of matching problem. We use a block matching method with epipolar constraint, disparity smoothness constraint and uniqueness constraint [7]. Different from the general block matching, our method adopts the criteria of the minimum mean absolute difference (MAD) with weights. Since the border pixels are much more important than other pixels for the matching problem, the weights corresponding to them are larger than the other pixels. The size of comparing sliding window is 5×5. The matching criterion is as follows:

$$MAD(d_x, d_y) = \frac{1}{25} \sum_{(u,v) \in B} w \left| \pi(u,v) - \pi'(u + d_x, v + d_y) \right|$$

The weight w is determined as follows:

$$\begin{cases} w = 2, if\ (u,v)\ is\ a\ border\ pixel\ in\ image\ \pi \\ w = 1, otherwise \end{cases}$$

Where (d_x d_y) is the disparity along X and Y axes of the image coordinates, B is the comparing sliding window. Then the matching estimate is given by:

$$[\hat{d_x}, \hat{d_y}] = \arg \min_{(d_x, d_y)} MAD(d_x, d_y)$$

Epipolar is the most important constraint in stereo vision which reduces the search space to 1D. Under the condition described in this paper, the equation of epipolar line is [13]:

$$\begin{bmatrix} u & v & 1 \end{bmatrix} * \left(T \times (R * \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}) \right) = 0$$

E. Generating Local Map

After all of the regions of the image have been classified as ground or obstacle region, the image pixels are projected into the 2D ground plane as shown in Fig. 4 to generate the local map. The map is used for the robot to search passable paths that can guide the robot to navigate in the environment. The equation of the projection is as follows:

$$s \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = M^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Where s is an arbitrary scale.

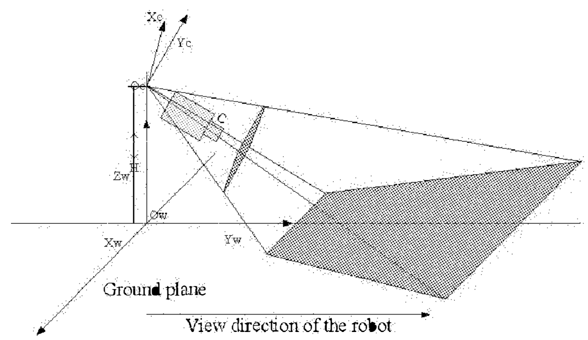


Fig.4 Project the image to the ground plane

IV. EXPERIMENTAL RESULTS

We have developed a mobile robot as shown in Fig. 5 and it serves as the test-bed for our algorithm. The robot is equipped with a color CCD camera (mounted H=700mm above the ground) and a video grabber that captures images at 25 frames per second (FPS). This robot is also equipped with an on-board computer (C3-1G CPU and 256M SDRAM). The experiment is carried out in the corridor of our lab.

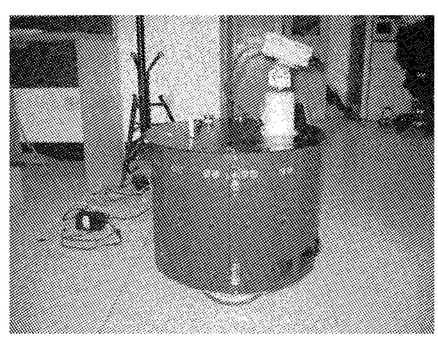
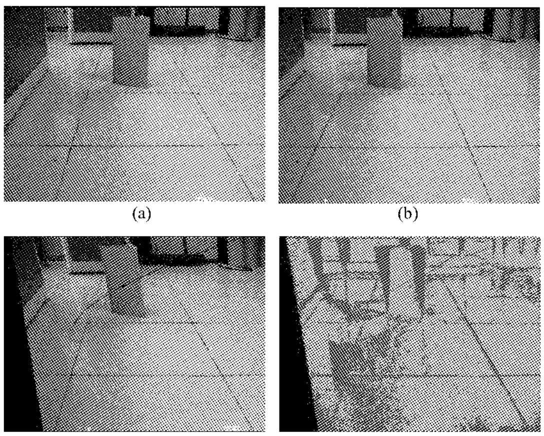


Fig.5 Experimental platform

A. Obstacle Detection Using Inverse Perspective Mapping

Fig. 6 shows an example of the obstacle detection results using inverse perspective mapping method. The top 2 images of Fig. 6 are taken from the image sequences grabbed during the robot moves around. Fig. 6(b) is wrapped to Fig. 6(c) with the homography of the ground between Fig 6(a) and Fig. 6(b). Subtracting Fig. 6(c) from the Fig. 6(a), we can get the obstacle result, Fig. 6(d), where the red pixels mean the detected obstacles and the green pixels mean the detected ground plane. From Fig. 6(d) we can see that the result of this method is very poor.



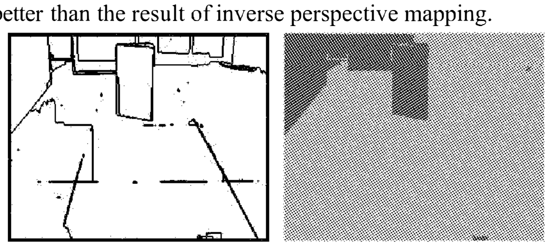
(a) (b)

(c) (d)

Fig.6 Obstacle detection result of inverse perspective mapping

B. Obstacle Detection Result Using Our Method

Fig. 7 shows one of the experimental results using our method. Fig. 7(a) is the region segmentation result of Fig. 6(a) by the proposed method (the border pixels are labeled in black and the other pixels are white) and Fig. 7(b) is the obstacle detection result using the method of this paper. It is better than the result of inverse perspective mapping.



(a) (b)

Fig.7 Obstacle detection result of our method

C. Local Map

Projecting Fig. 7(b) to the ground plane, we get the local map, Fig. 8. The green region is the ground area that the robot can traverse; the red region is the obstacle and the black region the area which is out of the view of the robot. This map can be used in path planning or map-building.

In our experiments, the robot only takes 20-30ms to process one image with a size of 320×240. This means that the proposed method can work in real time and can save extra time for other processes.

V. Conclusion And Future Work

In this paper, a region-based obstacle detection algorithm for indoor robot navigation is presented. It has been proven by our experiments that this method works quite well in a lot of cases.

In the future work, we shall try to use more sensor information, such as ultrasonic sensors, with the proposed algorithm to improve its robustness for the detection of large obstacles and try to use it for the obstacle detection and avoidance in outdoor environment.

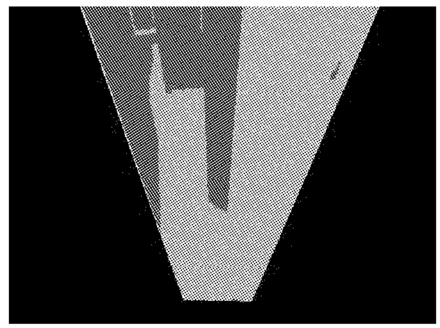


Fig.8 Local map derived by projecting Fig. 7(b) to the ground plane

References

[1] T. Williamson and C. Thorpe, "A Specialized Multibaseline Stereo Technique for Obstacle Detection," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June, 1998.

[2] Iwan Ulrich, Illah Nourbakhsh. Appearance-Based Obstacle Detection with Monocular Color Vision. In: Proceedings of the AAAI National Conference on Artificial Intelligence, 2000

[3] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli, "A Stereo Vision System for Real-Time Automotive Obstacle Detection," In Proceedings ICIP - Third IEEE International Conference on Image Processing, Lausanne, CH, September 16-19 1996. IEEE Signal Processing Society.

[4] Heiko Hirschmüller, "Improvements in Real-Time Correlation-Based Stereo vision", CVPR 2001 Stereo Workshop / IJCV 2002.

[5] Manolis I. A. Lourakis, Stelios C. Orphanoudakis. "Visual Detection of Obstacles Assuming a Locally Planar Ground," ACCV (2) 1998: 527-534.

[6] B. Heisele and W. Ritter. "Obstacle Detection Based on Color Blob Flow," Proceedings of the Intelligent Vehicles '95 Symposium, 1995.

[7] M. Sonka, V. Hlavac, R. Boyle: Image processing, Analysis, and Machine Vision, Brooks/Cole Publishing Comp, 1998.

[8] Y.-G. Kim, H. Kim, "Layered ground floor detection for vision-based mobile robot navigation," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04.

[9] N. E. Pears and B. Liang, "Ground Plane Segmentation for Mobile Robot Visual Navigation," Proc. of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1513-1518, 2001

[10]J. Santos-Victor and G. Sandini, "Uncalibrated obstacle detection using normal flow," Machine Vision and Applications, vol.9, no. 3, pp.130-137, 1996.

[11]N. O. Stoffler, T. Burkert, and G. Farber, "Real-Time Obstacle Avoidance Using an MPEG-Processor-based Optic Flow Sensor," Proc. 15th Int. Conf. on Pattern Recognition, vol.4, pp.161-166, Barcelona, E, September, 2000. IEEE Computer Society Press.

[12]Zhu zhi-gang, Lin Xue-yin. Real-Time algorithms for obstacle avoidance by using reprojection transformation. In:Proceedings of the IAPR Workshop on Machine Vision and Applications. Tokyo, Japan, 1990. 393-396.

[13]Ma SD, Zhang ZY. Computer Vision: Computational Theory and Algorithmic Basis. Beijing: Science Press, 1998 (in Chinese).

[14]Darius Burschka, Stephen Lee, Gregory D. Hager: "Stereo-Based Obstacle Avoidance in Indoor Environments with Active Sensor Re-Calibration," ICRA 2002: 2066-2072