Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Stereo-Based Obstacle Avoidance in Indoor Environments with Active Sensor Re-Calibration

Darius Burschka, Stephen Lee and Gregory Hager
Computational Interaction and Robotics Laboratory
Johns Hopkins University
Baltimore, MD 21218
{burschka|slee|hager}@cs.jhu.edu

Abstract

We present a stereo-based obstacle avoidance system for mobile vehicles. The system operates in three steps. First, it models the surface geometry of supporting surface and removes the supporting surface from the scene. Next, it segments the remaining stereo disparities into connected components in image and disparity space. Finally, it projects the resulting connected components onto the supporting surface and plans a path around them. One interesting aspect of this system is that it can detect both positive and "negative" obstacles (e.g. stairways) in its path.

The algorithms we have developed have been implemented on a mobile robot equipped with a real-time stereo system. We present experimental results on indoor environments with planar supporting surfaces that show the algorithms to be both fast and robust.

1 Introduction

Sensor-based obstacle avoidance is a central problem in mobile systems. Obstacle avoidance systems are essential to protect mobile robots from collisions with obstacles or driving towards staircases or gaps (*negative obstacles*) while operating in unknown or partially known dynamic environments. Many obstacle avoidance systems are based on sensors that furnish direct 3D measurements, such as laser range finders and sonar systems [1, 4]. In some cases, e.g. [9], cues from a monocular camera combined with prior knowledge of supporting surface geometry and appearance have been used. In contrast, our system relies completely on the data from a real-time stereo system with relative few prior assumptions.

There are three basic steps to our approach: 1) modeling the geometry and position of the supporting surface from stereo disparities and removing those disparities from the image; 2) segmenting the resulting disparities into connected objects, and 3) projecting those objects onto the supporting surface and planning a path for the robot. We show how these steps can be performed quickly and robustly, with the result that we can operate in real time in unstructured indoor environments. The real-time performance is particularly useful for in cases where the camera is itself being servoed (e.g. with a pan-tilt unit) while in operation. In practice, this system has been used as a basic component in a variety of vision-guided navigation systems [3, 2].

An additional advantage of our system is that it is small and lightweight, and can therefore be carried on most mobile systems, including our mobile robot *Speedy* (Fig. 1). The space, weight, and power footprint of a stereo imager is far less than a SICK laser, yet provides much higher density data than sonar. The complete system can be implemented on a common laptop computer running Linux.

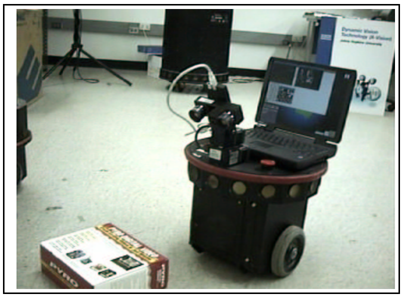


Figure 1: Small mobile robot *Speedy* equipped with a binocular stereo system.

The remainder of this paper is structured as follows. In section 2 we describe the geometry of a mobile stereo system, indicate how we estimate and suppress the supporting plane, and detail the segmentation algorithms used to detect obstacles. We end this section with an example showing how the generated data is used in a real system for obstacle avoidance. In section 3 we present the results of the experiments with our obstacle avoidance system emphasizing its speed and accuracy. We conclude with a few remarks about the advantages and disadvantages of the presented system and we outline our goals for the future work in section 4.

2 Approach

We present a system that is capable of detecting of positive (object B) and *negative* (object D) obstacles, such as gaps and staircases (Fig. 2). We start by assuming a binocular camera system with baseline parallel to the ground plane. We discuss a way to compensate for a possible roll error of the system in section 2.3.

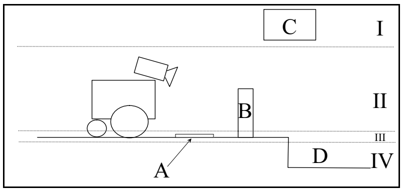


Figure 2: Classification of objects detected by the mobile robot.

The main problem in vision-based obstacle detection is the classification of the detected features into the four regions depicted in Fig. 2. It is obvious that objects of type C entirely contained in the region I are not considered as obstacles and may be neglected in further processing. The same is true for objects of type A entirely contained in region III that represent small bumps and dips in the ground.

In our system, we use dense disparity images from a correlation-based stereo reconstruction algorithm as our data source [7]. The goal is to extract single standing clusters in the image, representing objects, from the continuous disparity image such as the one shown in Fig. 3. This image shows an example where objects at different distances from the ground are correctly classified according to the considerations depicted in Fig. 2. The entire floor area, including the newspaper, is removed, because both belong to region III, which is not considered to be an obstacle.

The floor connects the entire image into one large cluster, making any kind of segmentation to extract obstacles and to allow path planning in-between them impossible. Since the floor is not interesting for further processing, our first step is to remove it from the input data.

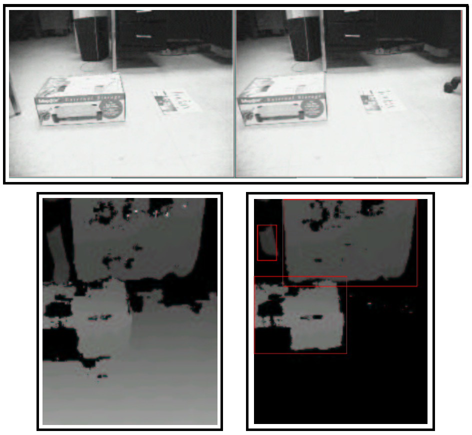


Figure 3: Detected three obstacles (right) from a pair of images (top) in the dense disparity image (left).

2.1 Geometrical constraints

In stereo data processing, the disparity value d_p in the image depends solely on the distance z_p between the imaged point, P, and the image plane of the sensor (Fig. 4). In case of a camera pointing at an empty floor and tilted at the angle Θ against the ground plane, each pixel in a horizontal image line has the same disparity

$$d_p = \frac{B}{z_p} \cdot \frac{f}{p_x}, \tag{1}$$

because z_p is independent of the horizontal coordinate in the image. B is the distance (baseline) between the two cameras, f is the focal length of the camera, which is used as base for the reconstruction (in our case the left camera), and p_x is the horizontal pixel-size of the camera chip.

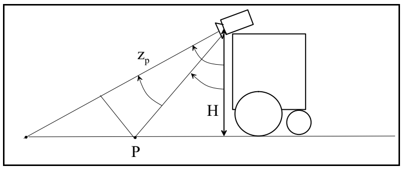


Figure 4: Geometrical consideration for expected disparity in an image.

The value for d_p or z_p in equation (1) needs to be estimated from the information in the image directly.

This is done in an on-line estimation process that is described in the next section.

2.2 Estimation of the image row disparities d_p

2.2.1 Direct learning

A simple approach to suppress the ground plane is to "learn" the distribution of the disparities d_p directly for all image rows in an empty area and to use these values in subsequent steps. However, this approach cannot adapt to changing tilt angles. Furthermore, a typical image during regular operation in indoor environments includes multiple obstacles, which makes complete learning of the disparities for all image rows difficult.

This technique is suitable for scenarios, where at least parts of the ground plane are always visible. It allows a robust removal of arbitrary shaped smooth surfaces. It can be used for detection of hilly surfaces such as roads, because the typical scenario in this case involves mostly empty road segments, which can be used for learning. It cannot be applied to an arbitrary terrain, but roads are smooth in the direction parallel to the image plane, where each image row for an empty space creates a narrow peak in a histogram for the entire row representing the disparity d_p for the ground surface in this row (Fig. 6).

2.2.2 Estimation of the ground plane

In indoor environments the systems operate usually on flat surfaces. Such surfaces can be robustly estimated from the sparse information available in the images.

The internal camera parameters and the orientation of the cameras to each other are assumed to be known from an off-line calibration process [11, 6]. The re-calibration procedure running on-line in each reconstruction step estimates the orientation of the camera system with respect to the ground plane μ (Fig. 5). Basically, the presented calibration process estimates the rotation between the coordinate systems of the camera (u, v, e) and the world (x, y, z).

The calibration values are calculated based on the reconstructed position of two points P_i, P_j, which are part of the ground plane $\mathcal{P}$. The stereo reconstruction process reconstructs the 3D position of each point in the image. Since the system is supposed to be used for collision avoidance, the probability is high that the bottom rows of the image contain, at least partially, the ground plane. In our approach, a histogram over the entire image row is calculated in 10 different rows in the bottom part of the image and the peak values are used to estimate disparity value d_p for this row. A pixel in each row with exactly this disparity disparity value is used to estimate the coordinates of the point P_x in the coordinate system (u, v, e) of the camera system.

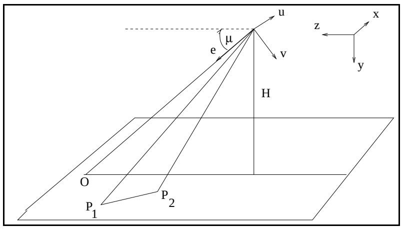


Figure 5: Calibration of Θ and H from two arbitrary points P_1, P_2.

The angle μ_{ij} can be calculated using the condition $P_i \in \mathcal{P} \wedge P_j \in \mathcal{P}$ from the scalar product of the normalized difference vector $\overline{P_iP_j}$ between any two of the ten extracted points and the normalized vector along the z-axis $\overline{z_0}$.

$$\begin{aligned}
n &= |P_j - P_i| \\
&= \sqrt{(u_j - u_i)^j + (v_j - v_i)^j + (e_j - e_i)^j} \\
\mu_{ij} &= \arccos \left[\frac{1}{n} \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ e_j - e_i \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right] \\
&= \arccos \frac{|e_j - e_i|}{|P_j - P_i|} \qquad (2)
\end{aligned}$$

The set $\{\mu_{ij}\}$ is used to vote for the estimated angle μ_{est}. RANSAC [5] method can be used to estimate a valid set $\mathcal{S}$ of points reconstructing μ_{est}. The calibration value Θ can be calculated using μ_{est} to

$$\Theta = \frac{\pi}{2} - \mu_{est} = \frac{\pi}{2} - \arccos \frac{|e_j - e_i|}{|P_j - P_i|}. \qquad (3)$$

The height of the camera system H can be estimated for example from the scalar product of the vector $\overline{P_x}$ with the z-axis expressed in the coordinate system of the camera

$$\begin{aligned}
H_x &= \begin{pmatrix} u_x \\ v_x \\ e_x \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \cos \mu_{est} \\ \sin \mu_{est} \end{pmatrix} \\
&= v_x \cdot \cos \mu_{est} + e_x \cdot \sin \mu_{est} \\
\Rightarrow \quad H &= \frac{1}{|\mathcal{S}|} \cdot \sum_{x \in \mathcal{S}} H_x. \qquad (4)
\end{aligned}$$

We included a “sanity” check in our system that verifies the computed values to catch outliers. If the calculated height changes differs significantly from the initially estimated value then the calibration estimate is rejected.

2.3 Prediction of the disparity for the ground plane

The parameter z_p can be calculated from the geometrical values depicted in Fig. 4. The distance z_p can be estimated to be

$$\begin{aligned} z_p &= \frac{H \cdot \cos\gamma}{\cos\beta}, \qquad (5) \\ \textit{with} \quad & \beta = \Theta + \gamma \ \wedge \ \gamma = \arctan\frac{v_p \cdot p_y}{f}, \end{aligned}$$

where v_p is the vertical pixel coordinate in the image relative to the optical center, pointing downwards and p_y is the vertical pixel-size of the camera. The angle γ is the vertical angle in the camera image between the optical axis and the current line v_p.

Using the equations (1) and (5) we can formulate the equation for the expected disparity value for a given line v_p to

$$\begin{aligned} d_p &= \frac{B \cdot f}{H \cdot p_x} \cdot (\cos\Theta - \sin\Theta \cdot \tan\gamma) \\ &= \frac{B \cdot f}{H \cdot p_x} \cdot \left(\cos\Theta - \sin\Theta \cdot \frac{v_p \cdot p_y}{f}\right). \qquad (6) \end{aligned}$$

Using the equation (6) the initial disparity image is processed to remove the supporting plane and all objects entirely contained in region III (Fig. 2) , which is the floor area in case of the mobile robot.

We make the assumption that the ground plane is parallel to the baseline of the stereo system. In this case the histogram for a single image row in an empty space should be a single value. In the reality, the cameras cannot be aligned so exactly. Fig. 6 shows that in the normal case the ground plane still covers a small disparity range $[185; 187] = 3$ (left histogram). This range extends to $[162; 178] = 17$ (right histogram) when the robot rolls by 30°. The shift in the middle values of the disparity range can be explained in the way, we rolled the robot. In this experiment the left wheel was lifted accordingly to achieve the desired angle, which increased the height H of the sensor.

In the current implementation we use a fixed value for the disparity range that is supposed to cover the floor area (region III). In this approach small objects far from the robot disappear from the planning map. We are currently modifying it to an adaptive approach, which will use the estimated disparity d_p and angle Θ to calculate a disparity range depending on the distance from the sensor system.

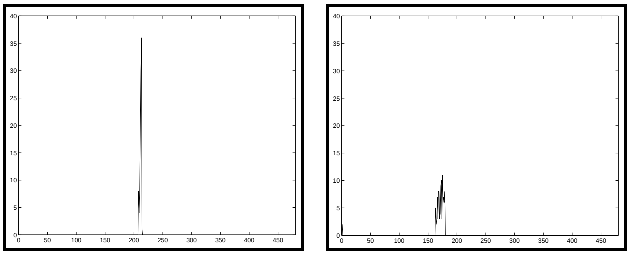


Figure 6: The range covered by the floor in a single row of an image: (left) almost perfect alignment (right) roll angle of 30°

2.4 Region segmentation

In the resulting disparity image the detected objects are represented as pixel clusters that need to be segmented into regions. Imperfections in the stereo reconstruction process and poor texture on the imaged surfaces result in frequent gaps in the reconstructed dense disparity images. Therefore, we lessen the direct neighborhood requirement for two elements of the cluster. We allow gaps between neighboring elements to prevent the reconstructed regions from splitting into too many small parts, which would fall below the spatial resolution of the planning algorithm that is supposed to use this data.

2.4.1 Compactness in the object space

Regions are represented by compact clusters spanning a range in the disparity domain. We require that clusters are continuous in the disparity domain to ensure that regions separate into objects correctly. We define this as the *disparity constraint*. This range can be quite large in case of, e.g., flat objects on the floor. Therefore, the range limits need to be calculated for each cluster separately.

An example of this is shown in Fig. 3. There, two clusters belonging to the desk and the box overlap in the image and would result in a segmentation into a single region without the disparity constraint. However, they need to be separated to allow navigation between them. The box has, in this example, a much larger disparity range, because the top surface is visible, while the desk appears almost as a single vertical surface with a narrow disparity range.

In real images, areas of an object may not be detected correctly due to texture properties of the surface in this area. Therefore, we allow a maximum distance ϵ_d between two object elements in space. The value ϵ_d is chosen to be half the size of the robot to prevent splitting the resulting obstacle map into too many small regions that do not provide enough space between them to navigate. The actual distance between points t_x in the object space of the object $\mathcal{O}_k$ may not exceed a maximum value ϵ_d

2.4.2 Compactness in the image space

In the ideal case, the extraction of a region requires uniformity of the imaged surfaces in this domain. This assumption is often violated in real images due to sensor noise, texture and shape of the surface, and lighting conditions. To compensate for these errors we allow gaps between neighboring pixels.

Therefore, the distance between neighboring image elements (pixels) p_x in the considered cluster should not exceed a given threshold ϵ_c. This value is chosen dependent on the current disparity value of the region and represents half the size of the robot.

$$|p_i - p_j| < \epsilon_c \qquad (7)$$

2.5 Navigation map

The region segmentation step from the previous section generates a list of clusters $\mathcal{C}$, where each cluster can be described by its middle point, extension d_x, d_y and depth range d_z.

2.5.1 Map generation

The elements of the cluster list $\mathcal{C}$ are projected, in our case, on a two-dimensional grid array, which is later used to generate safe paths through the environment (section 2.5.2).

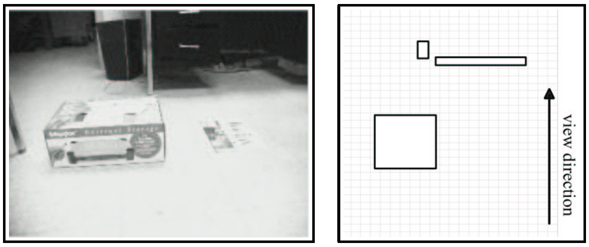


Figure 7: Navigation map for the scene from Fig. 3 with the desk, trashcan and a box reconstructed.

The depth extension does not necessarily reflect the real dimensions of the object because of the occlusions in the scene. This is obvious in the example of the desk, where basically just the vertical front plane is reconstructed. In contrast the box was reconstructed in its full extension.

2.5.2 Path planning

The map constructed in section 2.5.1 is used to generate safe paths through the environment. The algorithm is based on the gas diffusion analogy and is described in [10] in more detail.

The diffusion equation (8) describes the distribution of gas in the area. Once the diffusion process stabilized the optimal path can be found along the steepest increase in the concentration of the gas from *Start* to *Goal* (Fig. 8). In this equation u describes the concentration of the gas in a given point, Ω defines the area of operation, $\delta\Omega$ represents boundaries of objects in this area and a^2 and g are diffusion constants.

$$\begin{aligned}\frac{\partial u}{\partial t} &= a^2 \cdot \nabla^2 u - g \cdot u, && (8)\\ u(t;\underline{x}_G) &= 1, \quad \underline{x}_G \in \Omega \subset R^n &&\\ u(t;\underline{x}_0) &= 0, \quad \underline{x}_0 \in \delta\Omega \subset R^n &&\end{aligned}$$

A discrete form of this equation can be written in a recursive form for $r \in \Omega'$ as

$$u_{k+1;r} = \frac{1}{1+M} \cdot (\sum_{m=1}^{M} u_{k;m} + u_{k;r}) - \tau \cdot g \cdot u_{k;r} \qquad (9)$$

For the boundary conditions from equation (8) the recursive form is simplified to

$$u_{k+1;r} = \begin{cases} 0, & for \quad r \in \delta\Omega' \\ 1, & for \quad r = r_G \end{cases} \qquad (10)$$

The exact derivation is described in [10] in more detail. The number of iterations required to calculate the diffusion field depends on the size of the field and it defines the quality of the generated paths (section 3.3). To reduce the computational effort for path generation the Lee algorithm [8] is used (white line in Fig. 8) to estimate the size of the required map (dark area in Fig. 8) to generate a safe path from start to goal.

3 Results

In our experiments we used a Nomad Scout as a mobile robot with a PentiumIII@850MHz notebook

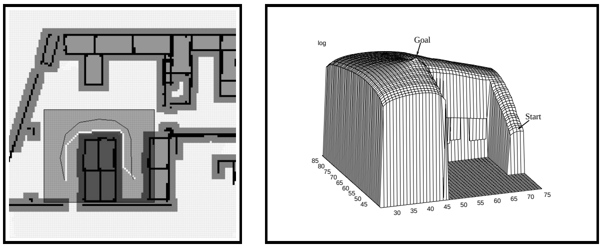


Figure 8: Path generation in a local area calculated from the Lee-algorithm.

running Linux-OS. The system was equipped with SRI's MEGA-D Megapixel Stereo Head. The cameras were equipped with 8mm lenses. The cameras were mounted 8.8cm from each other.

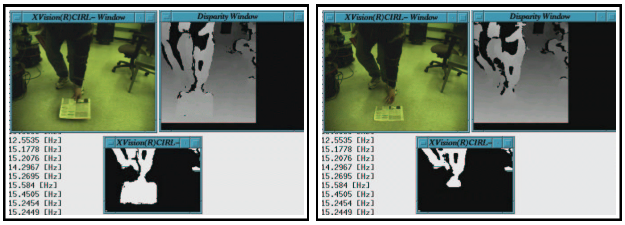
The typical tilt angle of the camera system during the experiments was $\Theta = 53°$. It was mounted $H = 48.26cm$ above the ground. It allowed us to robustly detect the obstacles in front of the robot while still allowing a range of up to 4m in front of the robot at the same time.

In this configuration the system was running with a reconstruction rate of 11.2 Hz for the entire obstacle detection cycle. In this case the path planning module was not enabled since the main focus of this paper is the robust obstacle detection and this extension shows merely, how to use the generated data in a real application.

3.1 Quality of the ground detection

Ground suppression is fundamental for the entire process. An example of a suppression is shown in Fig. 9. It shows the resolution of the system, which is capable distinguishing between the ground plane and objects as low as 1cm above the ground at a distance of up to 3m. The newspaper disappears as an obstacle as soon as it lays flat on the ground. Each image triple shows the real image in the upper left corner, the computed disparity image in the upper right corner and the detected obstacles at the bottom.

Ground suppression was tested on different types of floor. The number of pixels that could not be removed correctly usually lies in the range of 0.6% of the total number of pixel in the image for an empty space (Fig. 10). In case of the white plane in the bottom image, no disparity values were obtained and a warning was generated, because the size of the resulting information gap was larger than a threshold value. This value is defined by the expected size of the robot in the image in this area based on the calculations in the ground suppression module.



```
12.5535 [Hz]
15.1778 [Hz]
15.2076 [Hz]
14.2967 [Hz]
15.2695 [Hz]
15.584 [Hz]
15.4505 [Hz]
15.2454 [Hz]
15.2449 [Hz]
```

12.5535 [Hz]
15.1778 [Hz]
15.2076 [Hz]
14.2967 [Hz]
15.2695 [Hz]
15.584 [Hz]
15.4505 [Hz]
15.2454 [Hz]
15.2449 [Hz]

Figure 9: The newspaper is classified as obstacle left, but it disappears in the right image.

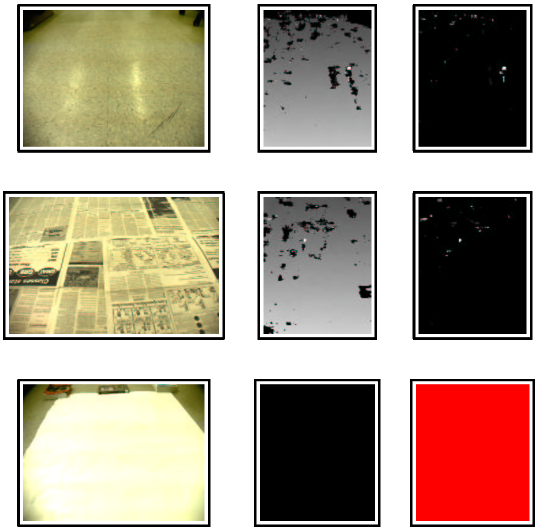


Figure 10: Ground plane suppression (left-to-right: video image, disparity image, floor suppressed): (top) regular lab floor, (middle) newspaper covered ground, (bottom) white plane results in an alarm in the segmentation window

3.2 Quality of the obstacle detection

The algorithm was applied in a variety of situations and generated reliable results in all situations where the scene contained enough structure for the stereo reconstruction algorithm. A few examples are shown in Fig. 11.

3.3 Quality of the generated paths

The best path is found when the diffusion process reaches equilibrium where no changes in the distribution $u_{k;r}$ (equation 10) occur. In real applications a minimum number of iterations K_{min} need to be calculated to achieve a good result. Experiments have

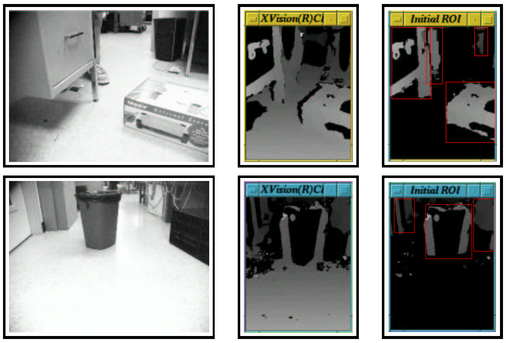


Figure 11: Examples of obstacle detection in different scenes.

shown that a threefold path length (in grid units of the map) estimated from the simple Lee algorithm [8] gives a good assumption for the number of iterations required for the diffusion algorithm. An example for the different quality of the generated paths is shown in Fig. 12.

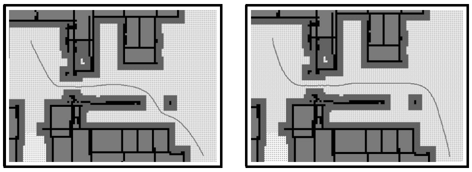


Figure 12: Planned paths after: (left) 100, (right) 400 iterations.

4 Conclusion and Future Work

We have presented a system that performs a robust, real-time obstacle detection and avoidance using on a binocular stereo system. The system performs a robust obstacle detection under changing system parameters by re-calibrating the external sensor parameters on-line as described in this paper. This system provides a vision-based alternative to the range sensors for robots with restricted space and resources.

The main weakness of the it is the stereo reconstruction. In environments with little texture the reconstruction has gaps that can result in poor detection of large obstacles with uniform surfaces. A good example is the detection of the trashcan in the last image set in Fig. 11, where merely the boundaries of the obstacle were detected. An improvement in this area is expected with structured light source added to the system, which should add the necessary texture on such object types.

An interesting additional research field can also be a test of the system on the road to distinguish between lines on the ground and obstacles on the road. The on-line calibration can be extended to deal with non-planar surfaces as mentioned in section 2.2.1. In this case the expected disparity for the road is learned for each line from a histogram and no general rule for the whole image is assumed.

Acknowledgments

This work was supported by the MARS project, and by the NSF RHA program.

References

[1] J. Borenstein and Y. Koren. Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments. In *IEEE International Conference on Robotics and Automation*, pages 572 – 577, May 1990.

[2] D. Burschka and G. Hager. Dynamic composition of tracking primitives for interactive vision-guided navigation. *In Proc. of SPIE 2001, Mobile Robots XVI*, pages 114–125, November 2001.

[3] D. Burschka and G. Hager. Vision-based control of mobile robots. *In Proc. of IEEE International Conference on Robotics and Automation*, pages 1707–1713, May 2001.

[4] G. Dudek, P. Freedman, and I. Rekleitis. Just-in-time sensing: efficiently combining sonar and laser range data for exploring unknown worlds. *In Proc. of ICRA*, pages 667–672, April 1996.

[5] M.A. Fischler and B.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. ACM*, 24(6):381–395, 1981.

[6] L. Iocchi and K. Konolige. A Multiresolution Stereo Vision System for Mobile Robots. *AIIA (Italian AI Association) Workshop, Padova, Italy*, 1998.

[7] T. A. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society Press, 1996.

[8] Lee. An algorithm for path connection and its application. *IRE Trans. Electronic Computer*, EC-10, 1961.

[9] L.M. Lorigo, R.A. Brooks, and W.E.L. Grimson. Visually-Guided Obstacle Avoidance in Unstructured Environments. *IEEE Conference on Intelligent Robots and Systems*, pages 373–379, September 1997.

[10] G. Schmidt and K. Azarm. Mobile Robot Navigation in a Dynamic World Using an Unsteady Diffusion Equation Strategy. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 642–647, 1992.

[11] Roger Y. Tsai. A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Transactions of Robotics and Automation*, RA-3(4):323–344, August 1987.