Single Camera Stereo for Mobile Robot World Exploration

Dmitry O. Gorodnichy and William W. Armstrong
Department of Computing Science,
University of Alberta,
Edmonton, Alberta, Canada T6G 2H1
{dmitri,arms}@cs.ualberta.ca

Abstract

This paper introduces a single-camera-based stereo vision system used for creating local 3D occupancy models. The design of the system is described, the range data error analysis is presented and the sensor model which assigns the values of evidence to the registered data is built. The application of the proposed system for mobile robot world exploration is shown. Data obtained by running a single camera mobile robot are presented.

Keywords: Occupancy grids, visual sensor model, range data fusion, evidence theory.

1 Introduction

In world exploration, the occupancy model of the world is one of the most commonly used [3, 20, 7, 25]. In this model, the evidence that a point in space is occupied is calculated, based on the data registered by a range sensor.

Originally developed for building 2D maps [9], the occupancy-based approach has recently been extended to build 3D models of the world [15, 17], which provide much more information about the environment.

There are two problems however with building 3D occupancy world models. The first problem is the representation of the occupancy information. The conventional way of storing the occupancy data using grids requires a considerable amount of memory and is time consuming. It is also inefficient for map extraction. The second problem concerns range sensors. Sonar sensors are not expensive and their models are known [3, 16]. However, they do not yield accuracy sufficient for 3D modeling [15]. At the same time, laser range sensors and highly calibrated stereo systems, which also have well defined sensor models, are very expensive and cannot be used in many situations.

These were the problems we addressed in the Boticelli project. As a solution, first, we proposed a regression-based technique for fusing range data, which allowed us to build 3D occupancy models represented in a parametric way, and second, we designed a single camera visual sensor, which allowed us to register 3D range data with the aid of an off-the-shelf video-camera. Boticelli is the name of the robot we used for the proof-of-concept demonstrations. It explores an unknown environment by building local occupancy world models based on the visual data captured by a single camera. While the issues of range data fusion, occupancy model representation and vision-based navigation are covered in [4], [5] and [1], respectively, this paper is dedicated to the vision part of the project.

We show how a single camera visual sensor can be designed so as to provide range data needed for building 3D occupancy models of the required quality. This includes designing the stereo rig, presenting the error analysis of the stereo algorithm, building the visual sensor model and showing the advantages of using the proposed visual sensor for mobile robot world exploration.

The paper is organized as follows. The design of the stereo setup is presented in the next section. The sensor model which assigns the values of evidence to registered depth data is built in Section 3. Experimental results and discussions conclude the paper.

2 Visual sensor design

The design of a visual sensor depends on the objective of the project. In our project the sensor is used for building local occupancy models, where we want the models to be fast in calculation and compact in representation. There are many applications where such models can be used and in this paper we concentrate on their application for the mobile robot exploration task. Let us outline this task.

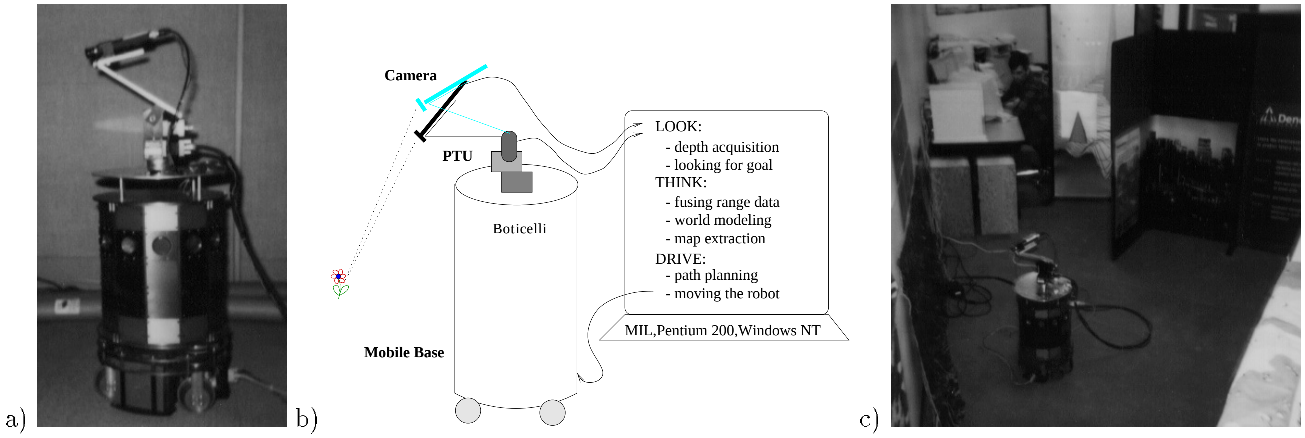

Figure 1: *Boticelli (a), its architecture (b) and an environment to explore (c).*

2.1 The exploration task

We consider the task of exploring the environment for the purpose of locating a hidden target. Figure 1.c shows the room which is used as a testbed environment in the project. Starting from an arbitrary position, the robot has to find a target, which is chosen to be a corner of a green triangle glued to white paper seen on the back wall in the figure.

The decision how to explore is determined by 1) the knowledge of already observed obstacle points, 2) the knowledge of exploration points, i.e. points where no information is obtained yet, and 3) the knowledge of the target location, if available. This determines a three-module architecture of the robot, which we refer to as the "Look-Think-Drive" architecture (see Figure 1.b). The first module is the vision module, during operation of which the robot tries to locate the target and collects range data around itself. The amount of these data should not be very large so as not to impede the mobility of the robot. On the other hand, it should suffice to build a precise enough 3D occupancy model of the world, where the precision is measured by the ability to navigate using the maps extracted from the model.

2.2 Single camera stereo

Many problems in world exploration by mobile robots are attributed to the odometric errors of the robot. Therefore, it is desirable to get as much information around the robot without having the robot move. This is achieved with a camera which has enough degrees of freedom to capture the entire environment.

We use a Sony XC-999 camera mounted on an L-shaped support on a Direct Perception pan-tilt unit (PTU) on the top of a mobile base as shown in Figure 1. The angle and the length of the support are chosen in such a way that the camera can observe completely the part of the world from the floor to the height of the robot, within a range from one decimeter to three meters. A grabber grabs 640x480 colour (RGB) images, which are then preprocessed with an averaging filter to produce 160x120 pixel images[1]. Depth acquisition is done on these lower resolution images. The Matrox Imaging Library (MIL) is used as an image grabbing and processing tool.

The readings from a single camera stereo, which are 3D depth data, are obtained by the following three step procedure (see Figure 2). A set of features is selected in the first frame (step 1). Each feature is then tracked along the epipolar line in the second frame, which is grabbed after the camera has moved, and the best match is obtained (step 2). The depth to those features which are selected and successfully tracked is then calculated on the basis of the disparity of the features in the two frames (stage 3). In the next subsection, we provide more specifics on the procedure and below we address another important issue — the issue of uncertainty of visual sensor data.

Uncertainty of range data

As a result of camera distortion, changing light conditions and incorrect registration of features, the obtained 3D information is not certain. Figure 4 shows the image of monochrome green rectangles as observed by a camera. The warping of the picture and the different intensities of the uniformly green surface can be clearly seen. This results in imperfect tracking and matching of features, which, in turn, results in under- or over-estimating the depth value corresponding to a feature

[1] This size of image has been found optimal not only by us, but other researchers [7].

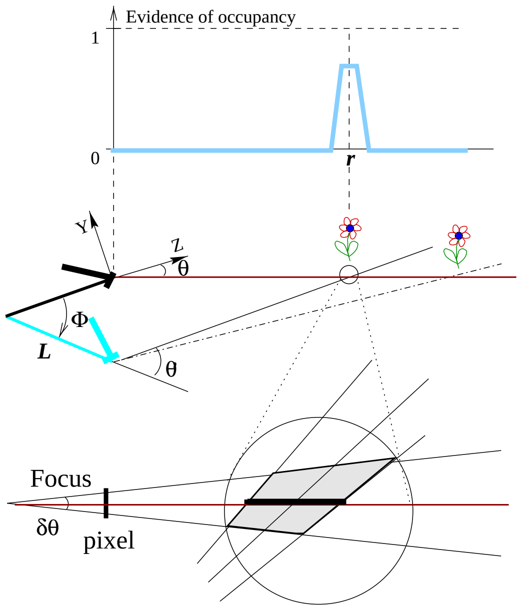

Figure 2: *Depth data obtained using a single camera.*

as illustrated in Figure 2. The figure also shows another major reason for uncertainty in depth estimation — limited resolution of the camera. All these has to be taken into account when building a visual sensor model.

At the same time, in order to decrease the depth estimation error, we resort to the following two techniques. First, we disregard the marginal features (as in [25, 7]), since they introduce high error not only because of the image quantization but also because of the warping of the image. Second, when the robot views the surrounding world, we make sure that each selected feature is observed at least twice, so that it appears at least once in the middle of the image, where its error is low. This is achieved by adjusting the angle of pan rotation.

2.3 3D data registration

According to the objective to build a world model just good enough for exploration and in order to make fusion and world modeling faster, we select only about 500 features per image. In particular, the pixels with a high intensity derivative in the vertical direction are selected as features.

The second frame is grabbed after the camera moves vertically down, which explains our choice of selecting features. The angle of the camera tilt rotation is $\Phi = 7.7^o$ and the lever length $L = 21$ cm, which results in the baseline $h \approx 3$ cm. This produces the disparity of 10 pixels on average, which is the same as in [7]. In the second frame, each feature is tracked along the epipolar line, which is chosen 3 pixels wide to account for warping of the image, using a 5 by 5 scanning window centered on a pixel.

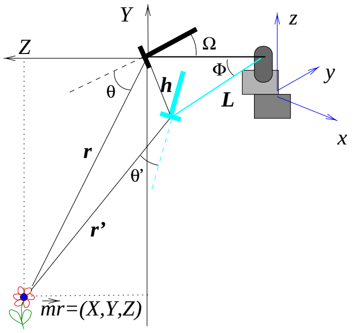

Figure 3: *Depth calculation procedure.*

A feature is considered successfully tracked if the error E between the best match and the original feature is lower than a certain threshold E_{thresh}. By lowering the threshold E_{thresh}, we can reduce the amount of uncertain data. This filters away approximately 60% of features.

The match error E is calculated as the Euclidean distance between the normalized[2] N-dimensional vectors ($N = 25$) obtained by using the scanning window:

$$E = ||\vec{V} - \vec{V}'||^2 = \sum_{n=1}^{N} (V[n] - V'[n])^2, \qquad (1)$$

which is a standard approach in feature tracking [6].

Finally, the depth r to those features which are selected and tracked is calculated, using the triangulation based on the projective camera model [8]:

$$\vec{m}r - \vec{h} = R\vec{m}'r', \qquad (2)$$

where $\vec{m} \doteq [i, j, F]_{unit}$ and $\vec{m}' \doteq [i', j', F]_{unit}$ designate unit vectors determined by the positions of a feature in the first and the second frame respectively. F is the focal length of the camera, which is known from the camera specifications or calculated in advance using the vanishing point technique described in [8]. The camera we use has $F = 150$, $i \in [-53, 53]$ and $j \in [-40, 40]$. R is the rotation matrix and $\vec{h}$ is the translation vector of the camera. Both are known, since only the pan-tilt unit moves and not the robot during depth acquisition.

Since a feature moves vertically only, Eq. 2 can be rewritten, using the coordinate method (see Figure 3), as

$$\left\{ \begin{array}{l} r' \sin(\Phi + \Omega - \theta') = Z \tan(\Omega - \theta) - h \cos \frac{\Phi}{2} \\ r' \cos(\Phi + \Omega - \theta') = Z + h \sin \frac{\Phi}{2} \end{array} \right. , \qquad (3)$$

[2] More exactly, the intensity of a center pixel is subtracted from the intensities of all pixels in the window.

where $\tan(\theta') = \frac{j'}{F}$, $\tan(\theta) = \frac{j}{F}$ and Ω is the angle of the camera support.

Dividing the first equation by the second one yields the following formula for (X, Y, Z) coordinates of a feature in the coordinate system centered on the first location of the camera as shown in Figure 2:

$$\begin{cases} Z = \frac{h\cos\frac{\Phi}{2} + h\sin\frac{\Phi}{2}\tan(\Phi+\Omega-\theta')}{\tan(\Omega-\theta)+\tan(\Phi+\Omega-\theta')} \\ X = Z\tan\theta \\ Y = Z\tan\theta_x, \quad \text{where} \quad \tan\theta_x = \frac{i}{F} \end{cases} \tag{4}$$

To obtain the coordinates (x, y, z) of a feature in the PTU-centered coordinate system, vector $\vec{mr} = (X, Y, Z)$ is multiplied by a homogeneous matrix describing the current position of the camera, which is a function of camera pan and tilt angles.

After depth is calculated for the current position of the camera, the camera is panned on the PTU clockwise and the procedure is repeated for the new angle of view, until finally all parts of the world around the robot are observed.

A thing to be mentioned about the single camera vision system is the parallelism of its operations — the depth is calculated, while the camera is moving. Because of that, the time needed to acquire depth information about the surrounding environment is just equal to the time needed to complete the full rotation of the camera. It takes 15 different pan positions of camera to observe the whole environment and the whole process of a building a sparse depth map of the entire environment takes about one minute.

Figures 6.a and 6.b show the depth information acquired by the robot by looking around from two different locations. Registered 3D features are shown projected on the floor (Oxy plane), the robot is located in the center. The figures also show grabbed images (in left top corners) and pairs of 2D features used in depth calculation (in left bottom corners): in white are the features selected in the first frame, while in black are the features which are tracked in the second frame.

2.4 Searching for the target

As opposed to a stereo setup with a fixed camera configuration, a single camera stereo allows arbitrary motion of the camera. This gives more flexibility not only in tracking the features but also in searching for the target.

In this project, we are not concerned with the issue of target recognition. Instead we choose the target to be invariant to the distance, which explains our choice of a corner of an object as the target. The target is sought by checking each image frame for the existence of a pattern previously stored in memory. MIL has a function which can do this operation efficiently. If the target is found, the same depth calculation routine which is used for features is used again to produce the location of the target with respect to the current position of the robot.

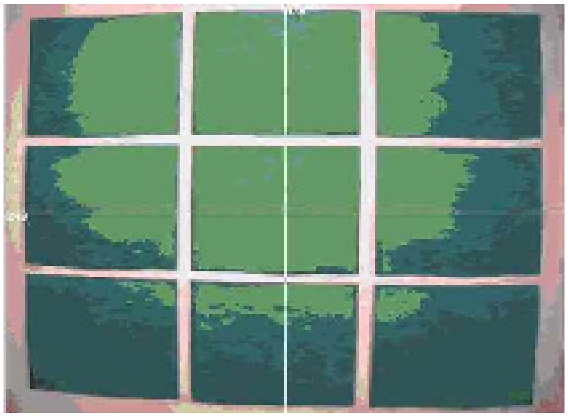

Figure 4: *The corruption of an image by a camera.*

3 The visual sensor model

3.1 Evidential approach

In sensor fusion, the concept of the *sensor model* is of prime importance. Probabilistic approaches [22, 9] define sensor model as the conditional probability $P(\vec{r} = occ|\vec{r}_S)$ that a point $\vec{r}$ in space is occupied, given a range sensor measurement $\vec{r}_S$.

It has been argued however that probabilistic approaches are not valid in building a sensor model when a sensor is not reliable [23, 16]. For example, if a sensor works properly only 3 times out 4 (because of power failures or other problems), then a measurement $\vec{r}_S$, which, we may say, is 75% reliable, provides some information about the occupancy of a point, but it does not give any data about the negation, i.e. about the emptiness of this point.

The evidential approach has been suggested to circumvent this problem. Rather than dealing with probabilities, this approach considers two values of evidence: the evidence $m_{occ}(\vec{r})$ that a point is occupied as well as the evidence $m_{emp}(\vec{r})$ that a point is empty. These values of evidence are the functions of the parameters describing the reliability of the measurement and are obtained using the intrinsic characteristics of the sensor.

The evidential approach is also credited with resolving the "unknown *vs* contradictory" ambiguity, which arises in first moment probabilistic approaches. Probabilistic approaches, which use second moments of the unknown variables, like Kalman filter approaches [13], are too computationally expensive and therefore are not very suitable in mobile robotics.

The main criticism of the evidential approach concerns the Dempster-Shafer rule, which is used to combine the evidence data. This rule assumes that sources of evidence are distinct and independent, so that no evidence is counted repeatedly [24]. In sensor fusion however, the same piece of evidence is often observed more than once. This is why in [4] we proposed a new, regression-based technique for combining range evidence data. This technique is used for fusing range data registered by the single camera stereo and is formulated as follows. Given a set of sample points $\vec{r}$ along with their evidence values m_{occ} and m_{emp} provided by the sensor model, find a smooth piece-wise linear approximation of functions $m_{occ} = m_{occ}(\vec{r})$ and $m_{emp} = m_{emp}(\vec{r})$ on the entire input domain. This determines the design of the single camera stereo sensor model.

3.2 Uncertainty of registered data

Industrially manufactured sonar sensors [3] and laser range finders [17] have well defined sensor models provided by a manufacturer. However, there is no general sensor model of visual range sensors, which is due to the diversity of the visual system setups. Thus, we have to design our own model of the single camera range sensor.

Taking into account the quantization error

As mentioned in Section 2, the depth data obtained by a vision system is not certain for many reasons.

Due to the finite resolution of the image, the angle θ' in the Eq. 4 is known only with the precision $\delta\theta' = \frac{1}{F}$ (see Figure 2.b). This results in the range error δr, which can be estimated by taking a derivative of $r = (X, Y, Z)$ with respect to θ' in Eq. 4.

Another way of estimating the range error is to use the results obtained for non-convergent dual camera stereo systems. The analysis of the uncertainty due to image quantization has been done in [2, 10, 18] and using the result obtained in [18], we get the following estimate of the range error:

$$\delta r = \frac{2r^2}{hF + r}. \qquad (5)$$

Taking into account the match error

Calculation of the evidence values assigned to the registered range data is based on the following idea. If we are 100% confident in the range data, then the range data should get the evidence value one. On the other hand, if the sensor is completely unreliable, then the range data should get the evidence value zero.

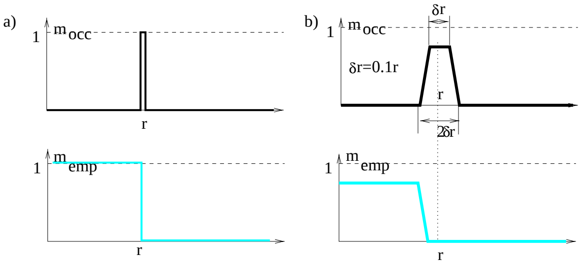

Figure 5: *Visual sensor model for ideal (a) and real (b) sensor.*

In the case of the single camera stereo, the measure of confidence of registered depth data $\vec{r}$ is provided by the match error E obtained during the depth calculation procedure (Eq. 1). In particular, we obtain the evidence of a 3D point $m_{occ}(\vec{r})$, by applying the Tuckey by-weight to the error E:

$$m_{occ}(\vec{r}) = \begin{cases} (1 - (\frac{E}{E_{max}})^2) & \text{if } E < E_{max} \\ 0, & \text{otherwise} \end{cases} \quad , \qquad (6)$$

which is a common approach in robust estimation [11, 19]. E_{max} is a constant which is chosen in agreement with the threshold value E_{thresh} used in filtering the outliers in Section 2.3.

This approach is different from that of [14] and resembles that of [15]. It produces the value of evidence in range [0,1], which is used in fusing the range data.

3.3 Linear representation

In the case of the ideal visual sensor, all points between the camera and the observed point will be given the evidence values $m_{emp}(\vec{r})$ and $m_{occ}(\vec{r})$, as illustrated in Figure 5.a. Figure 5.b shows the visual sensor model for the real visual sensor which is built according to the ideas described above.

The maximum value of evidence is determined by Eq. 6 The width of the range error δr is approximated using the Eq. 5 as $\delta r = 0.1r$. We also make the evidence grow gradually from zero to its maximum value, using the range error δr as a guide in determining the steepness of the slope, so that not to have infinite derivatives of the occupancy function. The evidence behind the observed point is zero for both occupancy and emptiness evidence values.

The piece-wise linear representation of the sensor model is chosen because of two reasons. First, it facilitates the approximation of the occupancy function with linear surfaces. Second, it significantly reduced the amount of sample data used in fusion. In particular, the sensor model can be represented with only

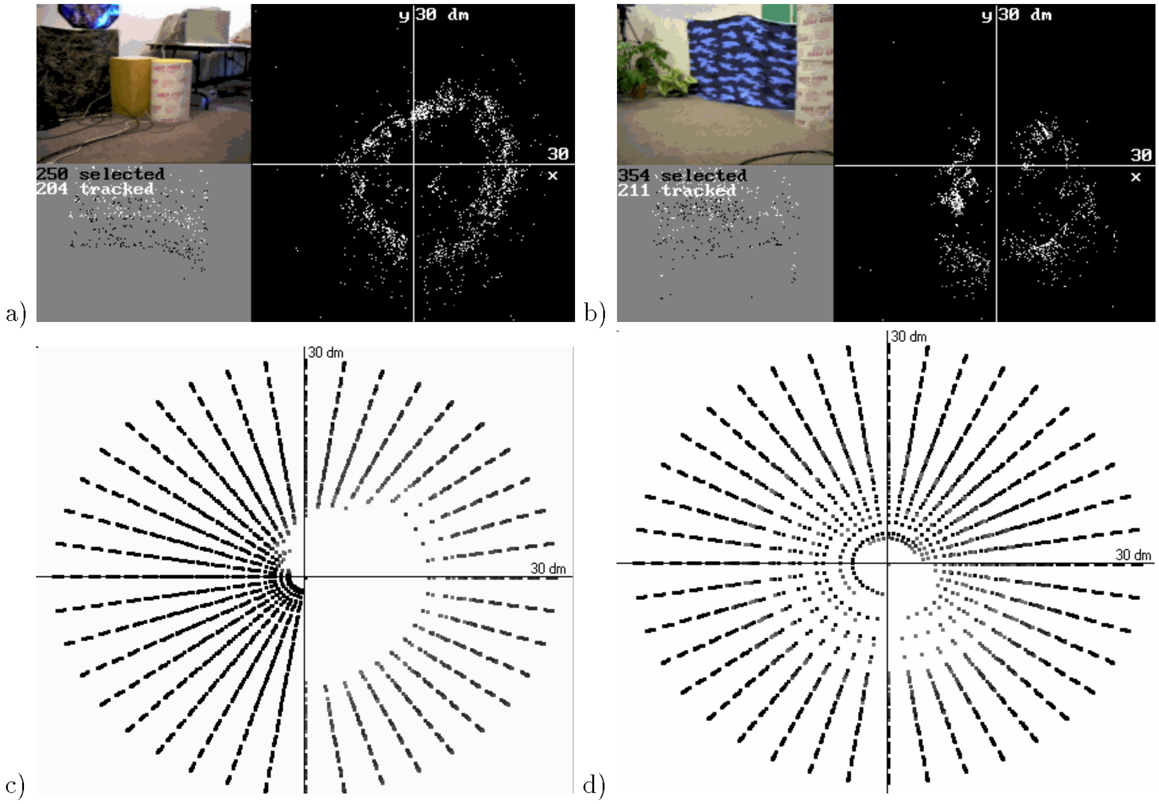

Figure 6: *Depth data obtained by a single camera stereo (a,b) and the area available for navigation acquired from the depth data (c,d) obtained at two different locations of the robot.*

few sample points on the ray of view, providing that there are certain constraints imposed on the function, which is described in more detail in [4, 5]. This provides the solution to the problem of redundancy of processed data, which, as was mentioned in the introduction, is the major problem of the occupancy-based approach.

4 Discussions

The single camera stereo vision system described in the paper is tested using a mobile autonomous robot Boticelli. The robot is placed in an approximately 5 by 6 by 1.5 m room surrounded by walls which it has to explore in order find a target which is hidden behind one of the walls. Figure 1.c shows the room and the target. Starting from an arbitrary location, the robot explores the environment until it finds a target. The exploration policy of the robot is determined by the knowledge of obstacle and navigation points, which are extracted from multiple 3D local occupancy models built on the basis of the range data registered by the single camera visual sensor, and also by the knowledge of the target location, which is acquired by the same sensor.

In order to ensure that there are enough visual features in the environment, we put camouflage clothes on the walls. These can be seen in Figures 1.c and Figure 6. Other objects inside the exploration area include a tree (seen in Figure 6.b), a couple of boxes (seen in Figure 6.a) and extension cords lying on the floor.

Figures 6.c and 6.d show 3D occupancy models obtained from depth data shown in Figures 6.a and 6.b. The points with occupancy values m_{occ} higher then 0.6 are shown projected on the floor. The robot is located in the center of the figures and is surrounded by an unoccupied area. This area is considered to be available for the navigation.

The occupancy models constructed from the registered visual data are found to be sufficient for making the navigation decisions. In our experiments, the robot successfully locates the target while avoiding obstacles and the areas already explored. Thus we conclude that the single camera stereo vision system proposed in the paper, which is able to register efficiently visual features around the robot, is very suitable for mobile robot exploration. For more details on how occupancy models are built from the visual range data see [4, 5].

The technique we use for feature selection and tracking (Section 2), while simple and not time consuming,

suffices for applications like the one described above. Yet, if there is a need for a more precise depth data registration, then the following steps can be undertaken to improve the performance of a single camera stereo:
– using a digital camera instead of an analog one [21];
– rectifying the images [15], if an analog camera is used;
– using an interest operator to select features [15];
– using robust tracking approaches, e.g. like those described in [12, 11].

As for the visual sensor model (Section 3), a better approximation of the range error should be used for large scale environments. In addition, other approaches in assigning the evidence values to registered range data can also be tried. However, since the final map of an area available for navigation is determined by a threshold on an occupancy function, this assignment seems not to affect much the navigation planning process.

Acknowledgments

We gratefully acknowledge the support by the Defense Research Establishment Suffield (contract W7702-6-R594/001) and by the Natural Sciences and Engineering Research Council of Canada (OGP 0172860). We also thank Xiaobo Li for valuable feedback on the project. Help of Monroe Thomas and Ron Kube in designing the PTU software is gratefully acknowledged as well.

References

[1] W.W. Armstrong, B. Coghlan, and D.O. Gorodnichy. Reinforcement learning for robot navigation. In *International Joint Conference on Neural Networks (IJCNN'99) proceedings , Washington DC, July 21-23*, 1999.

[2] S. Blostein and T. Huang. Error analysis in stereo determination of 3-d point postition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):752–772, 1987.

[3] L. Feng, J. Borenstein, and H.R. Everett. Where am i? sensors and methods for autonomous mobile robot. Technical Report UM-MEAM-94-21, The University of Michigan, 1994.

[4] Dmitry O. Gorodnichy. On using regression in range data fusion. In *Canadian Conference on Electrical and Computer Engineering (CCECE'99) proceedings, May 9-12, Edmonton*, 1999.

[5] D.O. Gorodnichy and W.W. Armstrong. A parametric alternative to grids for occupancy-based world modeling. In *Quality Control by Artificial Vision (QCAV'99) conference proceedings, May 18-21*, 1999.

[6] D.O. Gorodnichy, W.W. Armstrong, and X. Li. Adaptive logic networks for facial feature detection. In *Lecture Notes in Computer Science, Vol 1311 (ICIAP'97 Proceedings, Vol. II), pp. 332-339, Springer*, 1997.

[7] C. Jennings and D.Murray. Stereo vision based mapping and navigation for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation, pp. 1694-1699*, 1997.

[8] K. Kanatani. *Geometric Computation for Machine Vision.* Oxford University Press, 1993.

[9] M.C. Martin and H.P. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, CMU RI, 1996.

[10] L. Mathies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics Automation*, 3/3:239–247, 1987.

[11] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[12] C. Menard and A. Leonardis. Stereo matching using m-estimators. In LNCS–*1296, (*Proc. of CAIP'97*)*, pages 305–312, 1997.

[13] A. Mitiche. *Computational Analysis of Visual Motion.* Plenum Press, New York and London, 1994.

[14] J. Miura and Y. Shirai. Vision-motion planning for a mobile robot under uncertainty. *Int. J. of Robotics Research*, 16:806–825, 1997.

[15] Hans P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, CMU RI, 1996.

[16] D. Pagas, E. Nebot, and H. Durrant-Whyte. An evidential approach to probabilistic map-building. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 165–169, 1995.

[17] P. Payeur, P. Hebert, D. Laurendeau, and C.M. Gosselin. Probabilistic octree modeling of a 3d dynamic environment. In *Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1289-96*, 1997.

[18] J. Rodriguez and J. Aggarwal. Stochastic analysis of stereo quantazation error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12/5:467–470, 1990.

[19] P.J. Rousseeuw and A.M. Leroy. *Robust regression and outlier detection.* New York : Wiley, 1987.

[20] S. Thrun. The mobile robot rhino. *AI Magazine*, 15:31–38, 1994.

[21] A. Ude and R. Dillmann. Vision-based robot path planning. In *Advances in Robot Kinematics and Computational Geometry, Kluwer, Dordrecht, pp 505–512,*, 1994.

[22] J. van Dam, B. Krose, and F. Groen. Neural network application in sensor fusion for an autonomous mobile robot. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 263–277, 1995.

[23] Frans Voorbraak. Reasoning with uncertainty in ai. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 52–90, 1995.

[24] Pei Wang. A defect in dempster-shafer theory. In *Uncertainty in Artificial Intelligence Conference Proceedings (http://www.sis.pitt.edu/~dsl/uai.html)*, 1994.

[25] B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map-building with continuous localization. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium*, 1998.