Obstacle Detection Based on Partial 3D Reconstruction*

Zhongfei Zhang **Richard Weiss** **Allen R. Hanson**

Computer Science Department
University of Massachusetts
Amherst, MA 01003
Email: zzhang@cs.umass.edu

Abstract

Three different algorithms for qualitative obstacle detection are presented in this paper. Each one is based on different assumptions. The first two algorithms are aimed at yes/no obstacle detection without indicating which points are obstacles. They have the advantage of fast determination of the existence of obstacles in a scene based on the solvability of a linear system. The first algorithm uses information about the ground plane, while the second algorithm only assumes that the ground is planar. The third algorithm continuously estimates the ground plane, and based on that determines the height of each matched point in the scene. Experimental results are presented for real and simulated data, and performances of the three algorithms under different noise levels are compared in simulation. We conclude that in terms of the robustness of performance, the third one works best.

1 Introduction

Obstacle detection is an important issue in mobile robotics. One of the questions that is examined in this paper is the role of knowledge in detecting obstacles. Three different algorithms have been developed which make use of different knowledge of the visual sensors and the environment; they are compared with respect to robustness to noise. Using knowledge can usually improve the robustness of a system provided the knowledge is accurate. The first algorithm assumes that the ground is planar and its equation is known. Using this information together with the intrinsic and extrinsic parameters of the cameras, a linear system of equations is derived. The existence of a point not on the known ground plane (i.e. a potential obstacle) implies that the linear system is not solvable. One of the drawbacks of this algorithm is that errors in the equation of the ground plane due to camera tilt, grades, or hills in the road as well as noise in image measurements can cause the algorithm to fail.

The second algorithm does not use quantitative knowledge about the environment. It assumes that the ground is planar, but its equation as well as the camera parameters are unknown. This algorithm also reduces the obstacle detection problem to determining the solvability of a linear system. The third algorithm adaptively updates its knowledge of the environment by estimating the ground plane over a sequence of partially calibrated stereo pairs. This algorithm estimates the height above the ground plane for all points in a region of interest in an image. Faugeras [2] and Hartley[6] have independently shown that from uncalibrated stereo one can recover 3D structure up to a family of projective transformations. This paper shows that it is possible to recover *partial metric* information from *partial calibration*, where the depth of the second camera is known in the frame of the first camera. More detailed information on these algorithms appears in [9].

Speed is also an important issue for obstacle detection. The first two algorithms listed above only require one image to be processed at each time step. The tradeoff is that one processes

*This work was supported in part by ARPA and TACOM under contract DAAE07-91-C-R035

fewer images to detect obstacles, but the algorithms only give a yes/no answer. If a yes answer is obtained, the vehicle can slow down and use the third algorithm to localize the obstacles.

Because of its practical interest, obstacle detection has received widespread attention in the research literature[1, 7, 4]. Most existing algorithms for detecting obstacles use active range sensors, stereo, or optical flow. For example, Nelson and Aloimonos[7] used flow field divergence for obstacle detection and avoidance in visual navigation. Enkelmann[1] approached this problem by evaluating the difference between the calculated optical flow and the predicted model-based flow. This method assumes the vehicle motion is pure translation, which is not true in real environments. More recently, Grandjean and Matthies[4] approached the problem of real-time obstacle avoidance by using depth from stereo.

Throughout this paper, the following notation will be used. Let $\boldsymbol{p}$ denote the calibrated image vector for the left camera (in stereo) or for the first camera position (in motion) represented in homogeneous coordinates. Let $\boldsymbol{p}'$ denote the calibrated image vector for the right camera (stereo) or for the second camera position (motion) represented in homogeneous coordinates. $\boldsymbol{P}$ denotes the corresponding 3D point. Upper case is used for 3D coordinates, and lower case is used for the 2D coordinates. Hence, $\boldsymbol{p} = (x, y, 1)^T$, and $\boldsymbol{P} = (X, Y, Z)^T$, etc. Boldface is used to denote vectors or matrices; nonboldface characters denote scalar variables. Thus, $\boldsymbol{n}$ denotes the normal vector of a plane, and n denotes the number of feature points in a set. We use $\boldsymbol{t} = (t_X, t_Y, t_Z)^T$ to denote the translation vector between two cameras, and $\boldsymbol{R}$ to denote the 3×3 rotation matrix between the two camera coordinate systems. This rotation is also sometimes represented by $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^T$, the axis of rotation, scaled by the angle magnitude. $\boldsymbol{M} = (\boldsymbol{\Omega}^T, \boldsymbol{t}^T)^T$ is the motion parameter vector between two cameras. The symbol $\Longleftrightarrow$ is used to represent the "correspondence" relationship.

2 Obstacle Detection with Known / Unknown Ground Plane

In this section, we present two binary decision algorithms for obstacle detection. The first algorithm assumes that the ground plane equation with respect to the first camera is known, the rotation between the first and the second cameras is small, and the intrinsic calibration of the camera is known. We call this algorithm the **known ground plane** algorithm (**KGP**). The second algorithm, however, does not require a priori knowledge of the ground plane equation, or the internal/external calibration. Like the first one, it also assumes the ground plane is locally flat. We call this algorithm the **unknown ground plane** algorithm (**UGP**). Both algorithms can be applied to either a stereo pair or motion images.

Given an arbitrary 3D ground plane point $\boldsymbol{P}$, the velocity/displacement of $\boldsymbol{P}$ is:

$$\dot{\boldsymbol{P}} = \boldsymbol{\Omega} \times \boldsymbol{P} + \boldsymbol{t} \tag{1}$$

With the intrinsic camera parameters, the projection can be modeled as a calibration transformation followed by a pin-hole camera projection. Substituting the relationship between the 3D point $\boldsymbol{P}$ and its corresponding 2D image point $\boldsymbol{p}$, together with the ground plane equation, produces the following matrix equation of the flow of point $\boldsymbol{p}$:

$$\dot{\boldsymbol{p}} = \boldsymbol{H}\boldsymbol{M} \tag{2}$$

where $\boldsymbol{H}$ is a function of ground plane equation coefficients and the image point $\boldsymbol{p}$ (see [9] for the detailed derivation). We regard $\boldsymbol{H}$ as a linear function, which transforms a motion parameter vector $\boldsymbol{M}$ into a flow vector for $\boldsymbol{p}$. If there are n points $\boldsymbol{p}_1, ..., \boldsymbol{p}_n$ in the image, then if they are on the ground plane the following equation is satisfied.

$$\boldsymbol{D}\boldsymbol{M} = \boldsymbol{b} \tag{3}$$

where $\boldsymbol{D} = (\boldsymbol{H}_1, ..., \boldsymbol{H}_n)^T$ and $\boldsymbol{b} = (\dot{\boldsymbol{p}}_1, ..., \dot{\boldsymbol{p}}_n)^T$. It can be shown[9] that the following proposition holds:

Proposition: *The n points are all ground plane points iff the linear system Eq.3 has a solution; that is, iff Rank($\boldsymbol{D}$) = Rank($\boldsymbol{Db}$).*

The rank of a matrix is the same as the number of non-zero singular values in the singular value decomposition (SVD). To test if the two matrices have the same number of non-zero singular values, the ratios of the appropriate corresponding singular values are used. Since the real data are always corrupted with noise and numerical round-off, the computed singular values are always non-zero. Let $\sigma_{min}(\boldsymbol{D})$ be the smallest singular value of matrix $\boldsymbol{D}$, and $\sigma_{min}(\boldsymbol{Db})$ be the smallest singular value of matrix $[\boldsymbol{Db}]$. A feasible criterion for determining if $\text{Rank}(\boldsymbol{D}) = \text{Rank}(\boldsymbol{Db})$ is to check if the ratio $\frac{\sigma_{min}(\boldsymbol{D})}{\sigma_{min}(\boldsymbol{Db})}$ is sufficiently large. Based on our simulation analysis under different noise levels, and our experimental results on real images[9], a threshold δ between 5 to 10 on this ratio value is sufficient to detect obstacle points 1 ft. or more off the ground plane. Since the discretization error is not being modeled, this does not depend on the distance to the obstacle.

UGP works in a similar way. It is easy to show[3] that given an arbitrary ground plane point, $\boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i'$, there is an invariant 3 by 3 matrix:

$$\boldsymbol{A} = h_c\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T \tag{4}$$

such that

$$k_i\boldsymbol{p}_i' = \boldsymbol{A}\boldsymbol{p}_i \tag{5}$$

where h_c is the height of the focal point of the first camera, $\boldsymbol{R}$ is the rotation between the two cameras/instants, $\boldsymbol{t}$ is the translation between the two cameras/instants, $\boldsymbol{n}$ is the normal vector of the ground plane with respect to the first camera coordinate system, and k_i is a scale factor of the point pair $\boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i'$. Thus, we have a similar proposition:

Proposition: *The n points are all ground plane points iff the linear system of Eq. 5 is consistent, i.e. the rank of the coefficient matrix $\boldsymbol{A}$ is equal to that of the matrix augmented by $k_i\boldsymbol{p}_i'$.*

3 Obstacle Detection Based on Ground Plane Estimation

Since roads may have hills and curves, the orientation of the local ground plane may change as the vehicle moves. The algorithm presented in this section is one which can adapt to these changes. Estimation of the ground plane is done in such a way that the 3D heights of points with respect to the plane can be estimated from partially calibrated stereo. The previous two algorithms measure the deviation of the data from a planar configuration (known or unknown). This algorithm examines the heights of all of the points. Since ground plane points will not necessarily have height zero, it is necessary to use a threshold to distinguish between ground plane points and obstacle points. For sake of clarity, we call the estimated unknown ground plane the *reference plane*. The algorithm for computing the heights is based on the following assumptions:

- There is no translation component between the two uncalibrated stereo cameras along the Z direction (focal axis direction) of the first camera coordinate system, i.e. $t_Z = 0$. This is what is meant by *partially calibrated stereo*. Information about the absolute pose of these cameras (e.g. pan/pitch/tilt angles) and the other two translation components (e.g. baseline between the two cameras) is not required.
- The height h_c of the first camera above the reference plane is known.

In addition to the above assumptions, we also assume that if there is a nonzero rotation between the two cameras, then the internal camera parameters are known. In practice, only the special case (which is perhaps the most frequently encountered) for which there is no rotation between the two stereo cameras is implemented. In other words, the two cameras are aligned. Alignment means that the optical axes of the two cameras are parallel, but the baseline may not be necessarily aligned with the horizontal or vertical axes of each of the cameras. In fact, the baseline can be any line in the $X - Y$ plane. We show below that in this special case, there is no need to know the internal camera parameters.

This algorithm is called the **estimated ground plane** algorithm, and is abbreviated **EGP**. Each 3D point $\boldsymbol{P}_i$ is at a height h_i from the reference plane. If h_i is 0, this point is exactly on the plane; if h_i is positive, this point is above the reference plane; if h_i is negative, this point is below the plane. Now we can preset a threshold δ_h such that any point with its height $|h_i| \leq \delta_h$ is regarded as a ground plane point. In this way, we view the ground plane as a plank with thickness $2\delta_h$ instead of a precise plane. If the road actually lies within this plank, then the height of an obstacle that can be detected will depend on δ_h. In the worst case, an obstacle which lies in an indentation will need to have height at least $2\delta_h$ to be outside this volume.

Let $\boldsymbol{p}_i$ and $\boldsymbol{p}_i{}'$ be the two corresponding images of $\boldsymbol{P}_i$ in the two image planes, respectively. Note that $\boldsymbol{P}_i$ has the same motion parameters $(\boldsymbol{R}, \boldsymbol{t})$ with respect to the two cameras. Based on Eq. 5 and the definition of $\boldsymbol{A}$, we have:

$$k_i{}'(h_c - h_i)\boldsymbol{p}_i{}' = [(h_c - h_i)\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T]\boldsymbol{p}_i \qquad (6)$$

where h_c is the height of the left camera above the reference plane, h_i is the distance of the 3D point $\boldsymbol{P}_i$ above the reference plane, and $k_i{}'$ is the scale factor corresponding to $\boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i{}'$[9].

We define the state vector $\boldsymbol{S}$ to be the vector consisting of the eight variable elements of the matrix $\boldsymbol{A}$. Note that this vector combines information about the ground plane and the transformation between cameras. Assuming we know the matrix $\boldsymbol{A}$, then for an arbitrary 3D point $\boldsymbol{P}_i = (X_i, Y_i, Z_i)^T$, with its corresponding image point at the left image plane $\boldsymbol{p}_i$, there must be a corresponding 3D reference plane point $\boldsymbol{Q}_i = (X_{Qi}, Y_{Qi}, Z_{Qi})^T$ which shares the same line of sight with $\boldsymbol{P}_i$ in the left image plane. The image point $\boldsymbol{p}_i{}''$ of this reference plane point $\boldsymbol{Q}_i$ in the right image plane can be obtained by using Eq. 5:

$$k_i{}''\boldsymbol{p}_i{}'' = \boldsymbol{A}\boldsymbol{p}_i \qquad (7)$$

Applying the definition of $\boldsymbol{A}$ produces:

$$k_i{}''h_c\boldsymbol{p}_i{}'' = (h_c\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T)\boldsymbol{p}_i \qquad (8)$$

where $k_i{}''$ is the scale factor corresponding to $\boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i{}''$.

Combining Eq.6 and Eq.8:

$$k_i{}'(h_c - h_i)\boldsymbol{p}_i{}' - k_i{}''h_c\boldsymbol{p}_i{}'' = -h_i\boldsymbol{R}\boldsymbol{p}_i \qquad (9)$$

Let $\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3$ be the row vectors of $\boldsymbol{R}$, i.e. $\boldsymbol{R} = (\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3)^T$, and $\boldsymbol{t} = (t_X, t_Y, t_Z)^T$. Since we have

$$\boldsymbol{P}_i \Longleftrightarrow \boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i{}', \quad \boldsymbol{Q}_i \Longleftrightarrow \boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i{}''$$

by the definition of k_i[9],

$$k_i{}' = \frac{Z_i{}'}{Z_i} = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i + \frac{t_Z}{Z_i} \qquad (10)$$

$$k_i{}'' = \frac{Z_{Qi}{}'}{Z_{Qi}} = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i + \frac{t_Z}{Z_{Qi}} \qquad (11)$$

Based on the matrix $\boldsymbol{A}$, we can solve for the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t}$ [8] . However, with unknown Z_i and Z_{Qi}, if $t_Z \neq 0$, $k_i{}'$ and $k_i{}''$ remain unknown. Thus, Eq. 9 is not sufficient to solve for h_i, since the three equations are not independent. If $t_Z = 0$, which is the case we assume, with known relative rotation $\boldsymbol{R}$, $k_i{}'$ and $k_i{}''$ can be determined from

$$k_i{}' = k_i{}'' = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i \qquad (12)$$

Therefore, using Eq. 9, the solution for h_i is:

$$h_i = \frac{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i{}' - (\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i{}''}{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i{}' - \boldsymbol{R}\boldsymbol{p}_i} h_c \qquad (13)$$

Here we follow the convention that the quotient of two vectors is the vector of quotients of their corresponding elements. Thus, Eq. 13 can be rewritten as follows:

$$\begin{aligned} h_i &= (x_i{}'' - x_i{}')(\frac{\boldsymbol{R}_1 \cdot \boldsymbol{p}_i}{\boldsymbol{R}_3 \cdot \boldsymbol{p}_i} - x_i{}')^{-1}h_c \\ &= (y_i{}'' - y_i{}')(\frac{\boldsymbol{R}_2 \cdot \boldsymbol{p}_i}{\boldsymbol{R}_3 \cdot \boldsymbol{p}_i} - y_i{}')^{-1}h_c \end{aligned} \qquad (14)$$

This solution looks very simple. It does not require the absolute pose information of the cameras such as pan/pitch/tilt angles, nor does it require orientation and distance of the baseline between the two cameras.

A special case occurs when the two cameras are aligned. In this case, $\boldsymbol{R} = \boldsymbol{I}$, where $\boldsymbol{I}$ is an 3×3 identity matrix. Thus, Eq. 14 is simplified as:

$$h_i = \frac{x_i{}' - x_i{}''}{x_i{}' - x_i} h_c = \frac{y_i{}' - y_i{}''}{y_i{}' - y_i} h_c \tag{15}$$

Note in this case, even internal calibration is not necessary, because $x_i, x_i{}', x_i{}''$ and $y_i, y_i{}', y_i{}''$ can be represented in terms of any arbitrary uncalibrated camera coordinate system.

Now the remaining question is how to estimate the state vector $\boldsymbol{S}$, i.e. $\boldsymbol{A}$. Assuming there are no obstacles in the first few image pairs, an initial estimate of the state vector can be obtained by solving an overconstrained linear system using least mean squares. For every subsequent stereo frame, this state vector is updated based on the information obtained from that frame. Hence, the algorithm can be expressed as:

- For current frame i, use the current estimate of the state vector up to the last frame $\hat{\boldsymbol{A}}_{i-1}$ to estimate the height $\hat{h}_{ij}$ of each point j in the current frame i. Then the set of ground plane points can be found:

  $$S_i = \{P_{ij} : |\hat{h}_{ij}| < \delta_h\}$$

  Based on S_i, compute the state vector $\boldsymbol{A}_i$ and covariance matrix $\boldsymbol{\Lambda}_i$ of the current frame; then refine the state vector $\hat{\boldsymbol{A}}_i$ and its covariance matrix $\hat{\boldsymbol{\Lambda}}_i$ using a Kalman Filter.

- Based on $\hat{\boldsymbol{A}}_i$, recalculate the height h_{ij} of each point j at this frame; report obstacles for any point P_{ij} if $|h_{ij}| > \delta_h$

Another issue that needs to be addressed is that the above algorithm works well only if the ground plane is globally flat. In practice, this may not be true, and there may be hills at different scales. Hills at large scales can be accommodated by weighting previous estimates so that the Kalman Filter only accumulates its history from the last n frames. Experimental results[9] show that this modified version works much better for real road scenes.

4 Experiments

A thorough comparison of the three algorithms with respect to robustness in the presence of noise in real and simulated image sequences can be found in [9], where we addressed the question of what is the smallest obstacle that can be reliably detected for a given level of noise. Generally speaking, according to our simulation results, **EGP** is the most robust algorithm of the three with respect to small-scale deviations of the points from a plane.

All the three algorithms have been tested on real image data[9]. Here we only report part of the experimental results using the **EGP** algorithm. Fig.1 is the left image of a sample of a sequence taken by the stereo cameras onboard the UMass Mobile Perception Lab[5]. The height of the cameras, h_c, is 7.8 ft. The vehicle moved about 6 ft. between frames and the fifth frame was taken about 50 ft. away from the obstacles. The 6 points labeled in the image are the obstacle points. The heights of ground points 1 to 5 are 2.35 ft., and point 6 is 1.50 ft. For the 6 obstacle points, Table 1 shows the estimated heights after 4 frames and after the fifth image pair is used.

As more frames were incorporated[9], the errors decreased for all of the points as each new measurement was added. It was observed in [9] that the error decreased monotonically as the vehicle approached the obstacles. The relative error is of the same order of magnitude as in the simulation[9].

5 Conclusions

In terms of robustness with respect to ground plane variation, the experimental results[9] show that the **EGP** algorithm is the best. This algorithm does have many other advantages as well since it is adaptive to changes in the ground plane and the Kalman filter updates are fast to compute. However, this algorithm assumes that partially calibrated stereo cameras are available as resources, and that one of their heights above the ground is known. When this information is not available, other algorithms such as **UGP** or **KGP** may be sufficient. Note that although the **EGP** algorithm only requires partial calibration, it is still able to recover the heights of points above the ground plane. In future experiments, we hope to be able to compare these algorithms on rougher terrain to determine if the a priori information would be useful in those cases.

Acknowledgement

Special thanks go to Prof. Edward Riseman for his enlightening comments, and Prof. John Oliensis, Dr. Bruce Draper, and Dr. Robert Collins for their inspiring discussions.

References

[1] W. Enkelmann. Obstacle detection by evaluation of optical flow fields. In *Proc. 1st ECCV*, 1990.

[2] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd ECCV*, pages 563–578, Santa Margherita Ligure, Italy, May 1992. Springer-verlag.

[3] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:485–508, 1988.

[4] P. Grandjean and L. Matthies. Perception control for obstacle detection by a cross-country rover. In *Proc. ICRA*, pages 20–27, Atlanta, Georgia, May 1993. IEEE.

[5] A.R. Hanson, E.M. Riseman, and C.A. Weems. Progress in computer vision at the university of massachusetts. In *IUW*, pages 39–47. Morgan Haufmann, April 1993.

[6] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *CVPR*. IEEE, 1992.

[7] R.C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *Trans. PAMI*, 11(10), 1989.

[8] Zhongfei Zhang. Euclidean 3D reconstruction based on externally uncalibrated cameras. *CMPSCI TR*, 1994.

[9] Zhongfei Zhang, Richard Weiss, and Allen R. Hanson. Qualitative obstacle detection. *CMPSCI TR94-20*, 1994.

Figure 1: The left image of a stereo sample of a road sequence

Table 1 — Height estimate errors

Note: Absolute error in ft. Relative error in %

Point	4 frames		5 frames	
Labels	Abs	Rel	Abs	Rel
1	0.23	9.8%	0.11	4.7%
2	0.20	8.5%	0.13	5.5%
3	0.21	8.9%	0.15	6.4%
4	-0.01	-0.4%	0.0	0.0
5	-0.08	-3.4%	-0.05	-2.1%
6	-0.11	-7.3%	-0.07	-4.7%