Qualitative Obstacle Detection

Z. Zhang, R. Weiss, A. Hanson

CMPSCI TR94-20

March 1994

This work was supported by the Advanced Research Projects Agency (via TACOM) under contract number DAAE07-91-C-RO35.

QUALITATIVE OBSTACLE DETECTION[1]

Zhongfei Zhang
Richard Weiss
Allen R. Hanson

Computer and Information Science Department
University of Massachusetts
Amherst, MA 01003
Phone : (413)545-0528
Email : zzhang@cs.umass.EDU
March 22, 1994

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.

Key words: motion analysis and stereo, qualitative vision, obstacle detection, 3D reconstruction, partial calibration

[1]This work was supported in part by DARPA and TACOM under contract DAAE07-91-C-R035

1 Introduction

Obstacle detection is an important issue in mobile robotics. Intuitively, anything that obstructs the motion of a vehicle is an obstacle. However, making a precise definition is surprisingly difficult. If the ground were a perfect plane, then an obstacle point could be defined as any point higher than a fixed value. A more complete analysis would involve the slope of the surface of a potential obstacle, which is not considered here. In general, there will be variations in the ground height occuring at different scales. At the smallest scale, there are bumps and indentations which are on the order of a few inches high. At intermediate scales, the road may have a crown, may be banked, or may have large bumps, and at large scales, there are hills and long ramps. Depending on the environment, an obstacle detection algorithm may need to handle all of these variations. Qualitative obstacle detection here is related to qualitative 3D reconstruction, because it determines whether a set of points can be approximated by a plane (known or unknown).

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 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 [5] and Hartley [11] 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.

From the experiments with real and simulated data which are presented in Section 5, it is seen that the adaptive algorithm is the most robust with respect to noise. However, there is still potential value in using the other algorithms. One of the limitations on the speed of a mobile robot is how fast it can detect obstacles given the available resources. These resources can be computational (how long it takes to run the algorithm) and sensor hardware (which cameras or other sensors are devoted to this task). The third algorithm requires stereo processing and computes the heights of each of the feature points, while the first two algorithms use monocular views and compute a single statistic for all of the points combined. In each of the algorithms, we assume that the correspondences have been computed. One could imagine that even if a vehicle had two cameras, resources could be saved by using the first or second algorithm until an obstacle is detected. Then, the vehicle could slow down or stop and use the third algorithm to compute the location of the obstacle and plan a path around it.

Because of its practical interest, obstacle detection has received widespread attention in the research literature[2, 3, 4, 15, 16, 17, 18, 9, 13, 14]. Most existing algorithms for detecting obstacles use active range sensors, stereo, or optical flow. For example, Nelson and Aloimonos[15] used flow field divergence for obstacle detection and avoidance in visual navigation. Daily et al[2, 3] used a laser range sensor to detect obstacles. Enkelmann[4] approached this problem by evaluating the difference between the calculated optical flow and the predicted model-based flow. Young et al[18] again presented another way to detect obstacles based on flow information. Both of these methods assume the vehicle motion is pure translation, which may not be true in real environment. More recently, Matthies and Grandjean[9, 13, 14] attacked the problem for real-time vehicle navigation by using stereo maps.

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 (in motion) represented in homogeneous coordinates. Let $\boldsymbol{p}'$ denote the calibrated image vector for the right camera

(stereo) or for the second camera (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. $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^T$ is the rotation vector between two cameras, whose three components are the rotation angles with respect to the three axes. $\boldsymbol{M} = (\boldsymbol{\Omega}^T, \boldsymbol{t}^T)^T$ is the motion parameter vector between two cameras. The symbol $\Longleftrightarrow$ is used to represent "correspondence" relationship.

2 Obstacle Detection with a Known Ground Plane

In this section, we assume 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. This algorithm can be applied to stereo images, where the first and second images are the left and right images of the stereo pair. It can also be applied to a motion sequence where the two images are taken at different time instants from a monocular camera in motion. We call this algorithm the **known ground plane** algorithm (**KGP**).

Given an arbitrary 3D ground plane point $\boldsymbol{P}$ with respect to camera motion/stereo, we know that the velocity/displacement of $\boldsymbol{P}$ is:

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

Given the intrinsic camera parameters, the projection can be modeled as a calibration transformation followed by a pin-hole camera projection. By projective geometry, the relationship between the 3D point $\boldsymbol{P}$ and its corresponding 2D image point $\boldsymbol{p}$ is:

$$x = \frac{X}{Z}, \quad y = \frac{Y}{Z} \tag{2}$$

Substituting these equations into Eq.1, together with the ground plane equation:

$$k_X X + k_Y Y + k_Z Z = 1, \tag{3}$$

produces the following matrix equation of the flow of point $\boldsymbol{p}$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -xy & 1+x^2 & -y & k_X x + k_Y y + k_Z & 0 & -x(k_X x + k_Y y + k_Z) \\ -(1+y^2) & xy & x & 0 & k_X x + k_Y y + k_Z & -y(k_X x + k_Y y + k_Z) \end{pmatrix} \begin{pmatrix} \Omega_X \\ \Omega_Y \\ \Omega_Z \\ t_X \\ t_Y \\ t_Z \end{pmatrix} \tag{4}$$

The above relation can be abbreviated as:

$$\dot{\boldsymbol{p}} = \boldsymbol{H}\boldsymbol{M} \tag{5}$$

Here we regard $\boldsymbol{H}$ as a linear function, which transforms a motion parameter vector $\boldsymbol{M}$ into a flow vector for $\boldsymbol{p}$, when the image of a ground plane point is given. Thus, if there are n ground plane points $\boldsymbol{p}_1, ..., \boldsymbol{p}_n$ in the image plane, then

$$\begin{pmatrix} \dot{\boldsymbol{p}}_1 \\ \cdot \\ \cdot \\ \cdot \\ \dot{\boldsymbol{p}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}_1 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{H}_n \end{pmatrix} \boldsymbol{M} \tag{6}$$

Let

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{H}_1 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{H}_n \end{pmatrix}$$

$$\boldsymbol{b} = \begin{pmatrix} \dot{\boldsymbol{p}}_1 \\ \cdot \\ \cdot \\ \cdot \\ \dot{\boldsymbol{p}}_n \end{pmatrix}$$

Eq.6 is abbreviated as:

$$\boldsymbol{DM} = \boldsymbol{b} \tag{7}$$

It is assumed throughout the paper that there are at least three ground plane points. An obstacle point is one which is above some fixed distance from the ground plane. For an obstacle point in the first image, there must be a corresponding ground plane point which shares the same line of sight. These two points will have distinct images in the second image unless the motion is a pure rotation around the line of sight of the point in the first image and/or a pure translation along the line of sight. In either case, the flow value of the point under consideration will be zero, which is easily detected (other points will have non-zero flow values). Assuming that all such zero-flow points have been eliminated, then each obstacle point will have a different flow value from the corresponding ground plane point. Now suppose we partition the point set into two subsets: one composed of ground plane points, and the other of obstacle points. Under this assumption, Eq.7 becomes

$$\begin{pmatrix} \boldsymbol{D_1} \\ \boldsymbol{D_2} \end{pmatrix} \boldsymbol{M} = \begin{pmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \end{pmatrix} \tag{8}$$

where the subsystem with subscript 1 denotes ground plane points, and the subsystem with subscript 2 obstacle points. Now if we replace the obstacle points with their corresponding ground plane points which share the same lines of sight with them, another linear system can be obtained:

$$\begin{pmatrix} \boldsymbol{D_1} \\ \boldsymbol{D_2} \end{pmatrix} \boldsymbol{M'} = \begin{pmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2'} \end{pmatrix} \tag{9}$$

If we assume that the ground plane points are non-collinear, then the linear system in Eq. 7 has a unique solution using only the ground plane points. This is a subsystem of both Eq.8 and Eq.9. If they both have solutions, then the linear systems must be the same. However, since $\boldsymbol{b_2} \neq \boldsymbol{b_2'}$, it follows that the two linear systems Eq.8 and Eq.9 cannot be satisfied at the same time.

A linear system like Eq.7 has solution *iff* Rank($\boldsymbol{D}$) = Rank($\boldsymbol{Db}$). Thus, we have proved the following proposition:

Proposition: *The n points are all ground plane points iff the linear system Eq.7 has a solution; that is, iff Rank($\boldsymbol{D}$) = Rank($\boldsymbol{Db}$).*

A practical problem with this proposition is how to calculate the rank of these matrices in the presence of noise. 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. We always assume that the number of feature points in a scene is no less than 3 and they are not collinear. It can be easily shown that if there are at least three non-collinear points, the matrix $\boldsymbol{D}$ will be full rank, i.e. Rank($\boldsymbol{D}$) = 6, so none of the singular values will be zero. If there is at least one obstacle point, Rank($\boldsymbol{Db}$) = 7, which means the linear system Eq.7 is inconsistent; otherwise, Rank($\boldsymbol{Db}$) = 6, which means that the linear system is consistent. Thus, the problem in this case is to decide whether or not the smallest singular value of $\boldsymbol{Db}$ is zero. However, since the real data are always corrupted with noise, together with the problem of numerical processing, 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 Rank($\boldsymbol{D}$) = 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 in Section 5, 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. Note that this criterion is the most conservative one, since a singular value of a matrix is always less than or equal to the corresponding singular value of its augmented matrix [8]. Let $\lambda_1 \geq \lambda_2 \geq ...\lambda_6$ be the six singular values of matrix $\boldsymbol{D}$, and $\eta_1 \geq \eta_2 \geq ...\eta_7$ be the seven singular values of the augmented matrix ($\boldsymbol{Db}$). Hence, we have

$$\eta_1 \geq \lambda_1 \geq \eta_2 \geq \lambda_2 \geq ...\lambda_6 \geq \eta_7$$

Therefore, the value of $\frac{\lambda_6}{\eta_7}$ is the most conservative criterion for detecting if the two matrices, $\boldsymbol{D}$ and ($\boldsymbol{Db}$), have rank 6.

Based on the above analysis, the basic steps of the algorithm are:

For each set of image correspondences (either stereo pair or motion pair),

- build the linear system defined in Eq.7;
- compute the singular values of matrices $\boldsymbol{D}$ and $[\boldsymbol{Db}]$;
- if $\frac{\sigma_{min}(\boldsymbol{D})}{\sigma_{min}(\boldsymbol{Db})} > \delta$, no obstacle is detected; else, report obstacle;

In the very rare case that all the n points are collinear in 3D, the above algorithm could be modified. In this case, matrix $\boldsymbol{D}$ would be singular, and Rank($\boldsymbol{D}$) = Rank($\boldsymbol{Db}$) = $i < 6$ iff the n points lie on a line on the ground plane. If there are obstacle points, then Rank($\boldsymbol{Db}$) = $i+1$. Thus, instead of using the ratio between the smallest singular values of $\boldsymbol{D}$ and $[\boldsymbol{Db}]$, one would first determine i, then determine the consistency of the linear system by checking $\frac{\sigma_i(\boldsymbol{D})}{\sigma_{i+1}(\boldsymbol{Db})} > \delta$.

This algorithm assumes that the internal calibration is known, the ground plane is locally flat, and the plane equation with respect to the first camera is known (in other words, the external calibration of the first camera is known). It only gives a binary decision for the detection of obstacles from a pair of images. The algorithm in the next section removes the requirement of knowing the ground plane equation. In Section 5, an analysis of the choice of the threshold δ is given for simulated data.

3 Obstacle Detection with Unknown Ground Plane

This algorithm uses a similar method to the algorithm in the previous section, although it does not require a priori knowledge of the ground plane equation, or the internal or external camera calibration. Like the above algorithm, it assumes the ground plane is locally flat, it can be applied to either stereo pair or motion images, and it only gives yes/no answer for obstacles. We call this algorithm the **unknown ground plane** algorithm, which is abbreviated as **UGP**. Clearly, this algorithm is more general than the previous one, but **KGP** will perform better if one has good a priori estimates for the parameters. Section 5 shows how the two algorithms compare in practice.

Since we are using a pinhole camera model, the 3D coordinates of a point $\boldsymbol{P_i}$ can also be used to represent the image point $\boldsymbol{p_i}$ in homogeneous coordinates. Using the ground plane

equation, it is easy to show[7] that given an arbitrary ground plane point, $p_i \Longleftrightarrow p_i'$, there is an invariant 3 by 3 matrix:

$$A = HR + tn^T \tag{10}$$

such that

$$k_i p_i' = A p_i \tag{11}$$

where H is the height of the focal point of the first camera, R is the rotation between the two cameras/instants, t is the translation between the two cameras/instants, 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 $p_i \Longleftrightarrow p_i'$, which accounts for the fact that the representation in homogeneous coordinates is not unique.

Since A can be normalized up to a factor that is "absorbed" into k_i, let

$$A = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \\ s_7 & s_8 & 1 \end{pmatrix} \tag{12}$$

Then Eq. 11 can be written as:

$$k_i \begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \\ s_7 & s_8 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{13}$$

Eliminating k_i, we have:

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y_i' & -y_i y_i' \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{pmatrix} = \begin{pmatrix} x_i' \\ y_i' \end{pmatrix} \tag{14}$$

The above equations were derived for the pinhole camera model, but they actually hold in general, if the two images are obtained from cameras with the same internal parameters (e.g. monocular motion or stereo with two identical cameras). Let $\boldsymbol{C}$ be the internal calibration matrix [6], and $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{p}}'$ be the uncalibrated 2D vectors for the first and second cameras, respectively. We have

$$\boldsymbol{p} = \boldsymbol{C}\hat{\boldsymbol{p}} \tag{15}$$

and

$$\boldsymbol{p}' = \boldsymbol{C}\hat{\boldsymbol{p}}' \tag{16}$$

By substituting $\boldsymbol{p}$ and $\boldsymbol{p}'$ in Eq. 11 with $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{p}}'$, respectively, we have

$$k_i\hat{\boldsymbol{p}}'_i = \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{C}\hat{\boldsymbol{p}}_i \tag{17}$$

Obviously, the linear system Eq. 14 is still valid. The only differences are that the definition of the unknown vector changes to the elements of $\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{C}$ instead of elements of $\boldsymbol{A}$, and now x_i, y_i and x_i', y_i' refer to the uncalibrated image vectors. That is why this algorithm does not need internal calibration.

Assuming that there are at least three ground plane points visible, the following proposition is the basis for the algorithm. The proof is similar to that of the proposition in the previous section.

Proposition: *Given n point correspondences $\boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i'$, $i = 1, ..., n$, they are all ground plane points iff the following linear system is consistent:*

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ \cdots & & & & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{pmatrix} = \begin{pmatrix} x_1' \\ y_1' \\ \cdots \\ x_n' \\ y_n' \end{pmatrix} \tag{18}$$

i.e. the rank of the coefficient matrix is equal to that of the augmented matrix of this linear system.

In this case, the equality in rank can be interpreted as a coplanarity constraint. The remainder of the algorithm is identical to the **KGP** algorithm; however, the dimensionality of the system is greater.

Note that both of these algorithms utilize a singular value decomposition to determine if a linear system is consistent, under the assumption that the ground plane is a true plane. In practice, the road surface may have bumps and dents. Thus, the surface may not satisfy the coplanarity constraint. Although we can use a lower threshold to tolerate this kind of variation of the road surface, the simulation results in Section 5 show that when the noise increases, the ratio value decreases dramatically. That means these two algorithms are sensitive to noise. The next section describes an algorithm based on 3D reconstruction that seems to be more robust in the presence of noise.

4 Obstacle Detection Based on Ground Plane Estimation

Since roads may have hills and curves, the orientation of the local ground plane may be changing with time. 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 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, we aligned the two cameras and did not determine the internal parameters.

4.1 Derivation of the Algorithm

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 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 and the same plane orientation $\boldsymbol{n}$ as those points on the reference plane. It can be shown[7] that the relationship between $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$ is:

$$k_i' d_i \boldsymbol{p}_i' = (d_i \boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T)\boldsymbol{p}_i \tag{19}$$

where d_i is the distance from $\boldsymbol{P}_i$ to the origin of the camera coordinate system in the left image in the direction of $\boldsymbol{n}$, and k_i' is a scale factor, which is defined as the ratio of the two depths of the same physical point $\boldsymbol{P}_i$ viewed at the left and right camera coordinate systems, i.e.

$$k_i' = \frac{Z_i'}{Z_i} \tag{20}$$

Clearly, we have

$$d_i = H - h_i \tag{21}$$

where H is the height of the left camera above the reference plane. Thus,

$$k_i'(H - h_i)\boldsymbol{p}_i' = [(H - h_i)\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T]\boldsymbol{p}_i \tag{22}$$

We define the state vector $\boldsymbol{S}$ to be the vector consisting of the eight elements of the matrix $\boldsymbol{A}$ in Eq.12. 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 (see Fig. 1). 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. 11:

$$k_i''\boldsymbol{p}_i'' = \boldsymbol{A}\boldsymbol{p}_i \tag{23}$$

After eliminating k_i'', the right image coordinates are:

$$x_i'' = \frac{s_1 x_i + s_2 y_i + s_3}{s_7 x_i + s_8 y_i + 1} \tag{24}$$

$$y_i'' = \frac{s_4 x_i + s_5 y_i + s_6}{s_7 x_i + s_8 y_i + 1} \tag{25}$$

Thus, we have:

$$k_i''H\boldsymbol{p}_i'' \doteq (H\boldsymbol{R} + \boldsymbol{t}\boldsymbol{n}^T)\boldsymbol{p}_i \tag{26}$$

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

Combining Eq.22 and Eq.26:

$$k_i'(H - h_i)\boldsymbol{p}_i' - k_i''H\boldsymbol{p}_i'' = -h_i\boldsymbol{R}\boldsymbol{p}_i \tag{27}$$

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

$$\boldsymbol{P}_i \Longleftrightarrow \boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i'$$

$$\boldsymbol{Q}_i \Longleftrightarrow \boldsymbol{p}_i \Longleftrightarrow \boldsymbol{p}_i''$$

by the definition of k_i as defined in Eq. 20,

$$k_i' = \frac{Z_i'}{Z_i} = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i + \frac{t_Z}{Z_i} \tag{28}$$

$$k_i'' = \frac{Z_{Qi}'}{Z_{Qi}} = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i + \frac{t_Z}{Z_{Qi}} \tag{29}$$

Based on the matrix $\boldsymbol{A}$, we can solve for the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t}$[2]. However, with unknown Z_i and Z_{Qi}, if $t_Z \neq 0$, k_i' and k_i'' remain unknown. Thus, Eq. 27 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

$$k_i' = k_i'' = \boldsymbol{R}_3 \cdot \boldsymbol{p}_i \tag{30}$$

Therefore, using Eq. 27, the solution for h_i is:

$$h_i = \frac{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i' - (\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i''}{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i' - \boldsymbol{R}\boldsymbol{p}_i} H \tag{31}$$

Here we follow the convention that the quotient of two vectors is the vector of quotients of their corresponding elements. Thus, Eq. 31 can be written as follows:

$$\begin{aligned} h_i &= (x_i'' - x_i')\left(\frac{\boldsymbol{R}_1 \cdot \boldsymbol{p}_i}{\boldsymbol{R}_3 \cdot \boldsymbol{p}_i} - x_i'\right)^{-1} H \\ &= (y_i'' - y_i')\left(\frac{\boldsymbol{R}_2 \cdot \boldsymbol{p}_i}{\boldsymbol{R}_3 \cdot \boldsymbol{p}_i} - y_i'\right)^{-1} H \end{aligned} \tag{32}$$

[2]With given $\boldsymbol{A}$, the rotation $\boldsymbol{R}$ and the translation $\boldsymbol{t}$ can be completely recovered in a closed-form solution[19]. The accuracy of the recovered parameters, however, highly depends on the accuracy of the camera internal parameters, especially the coordinates of the principal point, according to the experimental results in [19]. That is why we usually use known relative rotation, especially the alignment of the two cameras in practice, which also has the advantage of independence of the camera internal calibration, as shown in the text.

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 (which is perhaps the most frequently encountered) occurs when the two cameras are aligned. Alignment means that the optical axes of the two cameras are parallel, but the baseline not necessarily aligned with the horizontal or vertical axes of each of the cameras. In fact, the baseline can be any line in $X-Y$ plane. In this case, $\boldsymbol{R}=\boldsymbol{I}$, where $\boldsymbol{I}$ is an 3×3 identity matrix. Thus, Eq. 32 is simplified as:

$$h_i = \frac{x_i' - x_i''}{x_i' - x_i} H = \frac{y_i' - y_i''}{y_i' - y_i} H \tag{33}$$

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:

- The first pair of images is assumed to have only ground plane points. The overconstrained linear system (Eq. 18) can be solved using least mean squares.

- For current frame i, use the current estimate of the state vector up to the last frame ${\boldsymbol{A}_{i-1}}^{best}$ to estimate the height of each point j in the current frame i, $\hat{h_{ij}}$, then form a ground plane point subset:

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

  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 ${\boldsymbol{A}_i}^{best}$ and its covariance matrix ${\boldsymbol{\Lambda}_i}^{best}$ using a Kalman Filter:

$$A_i^{best} = (\Lambda_i^{best})^{-1}[(\Lambda_{i-1}^{best})^{-1} A_{i-1}^{best} + \Lambda_i^{-1} A_i] \tag{34}$$

$$\Lambda_i^{best} = [(\Lambda_i^{best})^{-1} + \Lambda_i^{-1}]^{-1} \tag{35}$$

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

There is another point that needs to be addressed. The above algorithm works well if the ground plane is flat globally. 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 show that this modified version works much better for real road scenes.

4.2 Error Analysis

In this section, the stability of the estimate of the height h_i for each point $\boldsymbol{P}_i$ based on Eq. 31 is analyzed. In general, there are four sources of error which contribute to the error in h_i:

- measurement error of the physical height of the cameras, H;
- localization error of the 2D image points $\boldsymbol{p}_i, \boldsymbol{p}_i', \boldsymbol{p}_i''$;
- deviations of the ground plane points from the reference plane;
- measurement error of the relative rotation between the two cameras;

The relative error of camera height H is on the order of a percent or two in practice. Since h_i is linearly proportional to H, the contribution to the relative error in h_i will be the same. Thus, we are not taking this into account. The contribution of the error of localization of the image points is complicated, especially the error of $\boldsymbol{p}_i''$, which is a function of the state vector in the Kalman Filter, as is the contribution of the error of the deviations of the ground plane points from the reference plane. To simplify the problem, we are not considering these last two error sources here. Instead, we leave the analysis of these errors under different

Gaussian noise levels to Section 5 based on simulation. Thus, the following error analysis only considers the error in relative rotation. Since it is assumed that the stereo cameras are aligned, i.e. the relative rotation is approximately zero, our analysis is done only in this case. An analysis for the general case of relative rotation is similar.

Expanding Eq. 31 in terms of Taylor's series at $\boldsymbol{R} = \boldsymbol{I}$, we have:

$$h_i = \left[\frac{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i' - (\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i''}{(\boldsymbol{R}_3 \cdot \boldsymbol{p}_i)\boldsymbol{p}_i' - \boldsymbol{R}\boldsymbol{p}_i} H\right]_{\boldsymbol{R}=\boldsymbol{I}} + \left[\frac{\partial h_i}{\partial \boldsymbol{R}}\right]_{\boldsymbol{R}=\boldsymbol{I}} (\boldsymbol{R} - \boldsymbol{I}) + O((\boldsymbol{R} - \boldsymbol{I})^2) \qquad (36)$$

Let us define the rotation matrix as:

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

By ignoring the higher order terms in Eq. 36, we have the error of the estimated height:

$$\begin{aligned} \triangle h_i &= \left[\frac{\partial h_i}{\partial \boldsymbol{R}}\right]_{\boldsymbol{R}=\boldsymbol{I}} (\boldsymbol{R} - \boldsymbol{I}) \\ &= (x_i(x_i \triangle r_{31} + y_i \triangle r_{32} + \triangle r_{33}) - (x_i \triangle r_{11} + y_i \triangle r_{12} + \triangle r_{13}))(x_i - x_i')^{-2}(x_i'' - x_i')H \end{aligned} \qquad (37)$$

There are nine elements in the matrix $r_{ij}, i, j = 1, 2, 3$. However, there are only three degrees of freedom for a rotation: the camera relative roll/pan/tilt angles. This makes it possible to express the relative error in height from Eq. 37 as (see [12]):

$$\frac{\triangle h_i}{H} \sim \frac{x_i(x_i'' - x_i')}{(x_i - x_i')^2}$$

Clearly, the relative error increases as depth increases, because the term $x_i - x_i'$ is inversely proportional to depth. However, since we always focus our attention on the central part of each image, and since $|x_i'' - x_i'| \ll |x_i - x_i'|$, for typical values within our focus-of-attention window as $x_i, y_i \sim 10^2$, $|x_i - x_i'| \sim 10^2$, the relative error is:

$$|\frac{\triangle h_i}{H}| \sim 10^{-2}$$

The experimental results presented in Section 5 indicate that this bound is consistent with the actual relative error. A simulation analysis for the performance of this algorithm under different levels of noise, as well as the false-positive and false-negative probabilities with respect to the different thresholds of heights is also presented there.

5 Experiments

This section compares the three algorithms with respect to robustness in the presence of noise in real and simulated image sequences. We address the question of what is the smallest obstacle that can be reliably detected for a given level of noise. The simulation only adds noise of the type produced by small scale deviations from the ground plane. The real image sequences, of course, include all normal sources of noise such as misalignment of cameras, errors in camera calibration and knowledge of the ground plane as well as bumps, indentations and errors in the optic flow/disparity.

The simulated data consisted of 10 ground plane points and an obstacle point at a distance of 20 ft. from the camera. The height of the obstacle point was varied from 0 ft. to 6.5 ft. The height of the camera was 3.55 ft., which is the actual height of the camera when mounted on the vehicle used to generate the experimental results. In addition, the displacement vectors were multiplied by Gaussian noise, which is equivalent to random fluctuations in the heights of the ground plane points, and thus could be interpreted as bumps and depressions in the gound plane. For the given height of the camera and a $\pm 10.0\%$ noise level, the maximum variation in the height of the ground plane points induced by this noise would be 0.355 ft. The simulation was done to *determine for each algorithm and each level of noise, the smallest obstacle that could be detected.* For an obstacle to be detectable, there must be a decision rule which will find all positive instances and will not produce too many false alarms. In other words, it is safe to allow a few false positives, but the probability of a false negative (failure to detect an obstacle) must be so close to zero as to never arise in practice.

The simulation results for the **KGP** algorithm are shown in Fig. 2. The statistic for detecting obstacle points is the ratio value $\frac{\sigma_{min}(\boldsymbol{D})}{\sigma_{min}(\boldsymbol{Db})}$. The graph shows how this ratio varies with the height of the obstacle point and the level of noise. In order for an obstacle point of

a given height (or greater) to be detectable, one should be able to draw a horizontal line that separates the ratio value for that height from the ratio value when the height is zero. We can see clearly from Fig.2 that for those points with heights greater than 1 ft., the ratio values are very close to 1, and are almost unaffected by noise. However, for levels of noise greater than $\pm 10.0\%$, it will be impossible to choose a threshold that would distinguish ground-plane points from obstacle points. Even for smaller levels of noise, it will be impossible to distinguish ground-plane points from obstacle points with heights less than 1 ft. Based on the simulation, a threshold of this ratio value between 5 and 10 would be able to correctly detect obstacles with height above 1 ft. up to $\pm 5\%$ noise. For obstacles with heights less than 1 ft., even $\pm 2.0\%$ noise would be a problem. It seems from the results that the ratio increases slightly with the height of the obstacle. This needs to be investigated further.

Fig. 3 shows a similar simulation for the **UGP** algorithm. Clearly, the general performance of this algorithm with respect to increased noise levels is worse than that of **KGP**. This is because the **KGP** used knowledge about the environment which was valid in the simulation. However, for points higher than 2 ft., the **UGP** algorithm can still correctly detect them, up to noise level of $\pm 5.0\%$. Satisfactory threshold values are between 5 and 10. For obstacle points with heights less than 2 ft., no satisfactory threshold can be found when the noise level is larger than $\pm 1.0\%$.

Fig.4 shows the simulation results for the **EGP** algorithm. The plot shows the relative error of the estimated height as a percentage of the actual 3D camera height. From this figure, we can see that with the increase in noise level, the relative error increases also, which is consistent with our intuition. The robustness of this algorithm can also be seen from this simulation. When noise is increased to $\pm 10.0\%$ level, the relative error in height is still within $\pm 5.0\%$. Note that in this Figure, the same noise was used in each of the experimental runs; only the amplitude of the noise changed. To determine the probabilities of false positives and false negatives, we ran our simulation 10 times with different seeds for each obstacle height and for each noise level. The false-positive probability is defined as the probability of the maximum height of ground plane points being above the threshold. False-negative probability is defined as the probability of the height of the obstacle point being below the threshold. Fig.5 shows the frequencies of false positives and false negatives as a function of

threshold for different obstacle heights and different noise levels. We are interested in finding a threshold such that the probabilities of false negatives and false positives are as small as possible. In general, one might allow some small number of false positives in order to detect smaller obstacles. For this paper, only ten trials were run for each of the parameter settings, so we looked for a threshold such that no false positives or negatives were obtained. For example, Fig.5(a) shows these frequencies for a noise level of $\pm 1.0\%$ and for a variety of obstacle heights. A satisfactory threshold is 0.03 ft. because the frequency of false positive curve is zero for this value. If the obstacle height is 0.05 ft. or more, then the frequency of false negatives is also zero, and that is the smallest obstacle that can be effectively detected in this case[3]. On the other hand, if the noise level is increased to $\pm 10.0\%$, and the frequency of a false positive is still required to be zero, then one can only detect obstacles that are higher than 0.45 ft. Note this is a big improvement over the 1 ft. height for **KGP** and 2 ft. for **UGP**. This fact indicates that **EGP** is the most robust algorithm of the three with respect to small-scale variation in the ground height.

All the algorithms have been tested for real image data. Fig.6 shows the left image of a hallway scene with some boxes as obstacles. The height of cameras, H, is 3.55 ft. above the ground plane. The right image is similar and is not shown. We arbitrarily chose 38 feature points in this image, as shown in the picture, and obtained their flow values with respect to the right image by Anandan's algorithm[1]. Using all 38 points in the linear system, the singular values of the matrices $\boldsymbol{D}$ and $[\boldsymbol{Db}]$ of **KGP** are listed in Table 1. From the table, $\sigma_{min}(\boldsymbol{D}) = 0.01689$ and $\sigma_{min}(\boldsymbol{Db}) = 0.01667$, respectively. Thus, the ratio value is 1.01, which is very close to 1. On the other hand, using only the ground plane points, the singular values are shown in Table 2. Now $\sigma_{min}(\boldsymbol{D}) = 0.01202$ and $\sigma_{min}(\boldsymbol{Db}) = 0.001744$, and their ratio value is 6.897.

We also tested the same point set for the **UGP** algorithm. With this algorithm, we don't need the camera internal parameters, nor do we need the external calibration information. Table 3 shows the SV's for the two matrices of the linear system with all the 38 points. The ratio of the two minimum SV's of the two matrices is 1.025. Table 4 shows the SV's for the

[3]The observation that no false negatives occurred in 10 trials means that one can state that the probability of such an event is at most 0.26 with a confidence level of 0.95. One would need more trials to get a better upper bound on this probability.

two matrices of the linear system with only the ground plane points. The ratio value this time is 6.594. Both algorithms were able to detect obstacles. Looking at the values, it seems that they performed about the same. This suggests that sources of error other than those in the variation in heights of ground plane points were significant and calibration and *a priori* knowledge of the ground plane may not be useful in improving qualitative obstacle detection in this case.

Fig. 7 is a sample of a sequence stereo picture of a real road scene with obstacles, taken by the stereo cameras onboard the Mobile Perception Lab[10]. The height of cameras, H, is 7.8 ft. Again, here we only show the left image for each stereo frame. **EGP** was run on this sequence. The first three image pairs Fig. 7(a), (b), (c) are used to give an initial estimate of the state vector which includes information about the ground plane. It is assumed that there are no obstacles in these scenes. Fig. 7(d) is a frame with five cones (ground truth height is 2.35 ft.) and a box (ground truth height is 1.50 ft.) on the road as obstacles. Fig. 7(e) is the next frame with the same obstacles. Table 5 lists the results of the **EGP** algorithm for frames (d) and (e). The goal is to show how the iterative estimation of the ground plane and other parameters improves with the number of frames and how this reduces the errors in the estimation of the heights of the six obstacle points that were detected. The first pair of columns shows the estimates of the absolute and relative errors (respectively) in the heights of the obstacles using the initial estimate of the state vector from the first three frames. The second pair of columns shows the estimates based on the updated state vector from frame (d). The third pair of columns shows the estimates of the heights of the points in frame (e) using the state vector estimated from frame (d). The last pair of columns shows the estimates of the points in frame (e) based on the state vector updated from frame (e). From the data shown in this table, we can see that both absolute and relative errors monotonically decrease as the vehicle approaches the obstacles, and more frames are used. The relative error is of the same order as in the simulation.

6 Conclusion

The work presented in this paper compares three different algorithms for obstacle detection using pairs of images. These algorithms use different information about the environment.

The **KGP** assumes knowledge of the ground plane and camera parameters. The **UGP** only assumes that the ground plane can be approximated by some plane. The **EGP** uses Kalman filtering to estimate the ground plane. The simulation results predict that just on the basis of noise in the heights, if the ground plane really were known precisely, the first algorithm would be better than the second. However, the experiments on indoor scenes show that the performance of the two algorithms is comparable. One explanation is that errors in estimating the camera parameters and ground plane are significant compared to the errors from variation in the heights of ground plane points, or that quantization errors dominate. 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.

In terms of robustness with respect to ground plane variation, the simulation shows 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 either of their heights 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.

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] P. Anandan. *Measuring Visual Motion from Image Sequences.* Ph.D. Thesis, COINS TR 87-21, 1987.

[2] M.J. Daily, J.G. Harris, and K. Reiser. Detecting obstacles in range imagery. In *Proc. IUW.* Morgan Haufmann, 1987.

[3] M.J. Daily, J.G. Harris, and K. Reiser. An operational perception system for cross-country navigation. In *Proc. IUW*. Morgan Haufmann, 1988.

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

[5] 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.

[6] O.D. Faugeras, Q.-T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In *Proc. 2nd ECCV*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[7] 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.

[8] G.H. Golub and C.F.V. Loan. *Matrix Computations, 2nd Ed.* The Johns Hopkins University Press, 1989.

[9] 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.

[10] 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.

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

[12] R. Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images.* COINS TR92-04, Univ. Mass., 1992.

[13] L.H. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *Proc. CVPR*. IEEE, June 1992.

[14] L.H. Matthies and P. Grandjean. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. In *Proc. CVPR*. IEEE, June 1993.

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

[16] U. Solder and V. Graefe. Object detection in real time. *SPIE Mobile Robots V*, 1388, 1990.

[17] K. Storjohann, T. Zielke, H.A. Mallot, and Von Seelen W. Visual obstacle detection for automatically guided vehicles. In *Proc. ICRA*. IEEE, 1990.

[18] G-S. Young, T-H Hong, M. Herman, and C.S. Yang. Obstacle detection for a vehicle using optical flow. In *SAE Intelligent Vehicle*, Detroit, MI, July 1992. IEEE.

[19] Zhongfei Zhang. Decomposition of **A** matrix. *CMPSCI TR*, 1994.

Table 1
Singular values of the linear system by all points in Fig. 6

SV's of $\boldsymbol{D}$	SV's of $[\boldsymbol{Db}]$
6.226e+0	6.227e+0
6.219e+0	6.219e+0
6.495e−1	6.497e−1
1.095e−1	1.096e−1
8.573e−2	8.723e−2
1.689e−2	2.729e−2
	1.667e−2

Table 2
Singular values of the linear system by all ground plane points in Fig. 6

SV's of $\boldsymbol{D}$	SV's of $[\boldsymbol{Db}]$
4.650e+0	4.651e+0
4.634e+0	4.634e+0
4.487e−1	4.488e−1
4.057e−2	4.507e−2
3.814e−2	4.041e−2
1.202e−2	1.214e−2
	1.744e−3

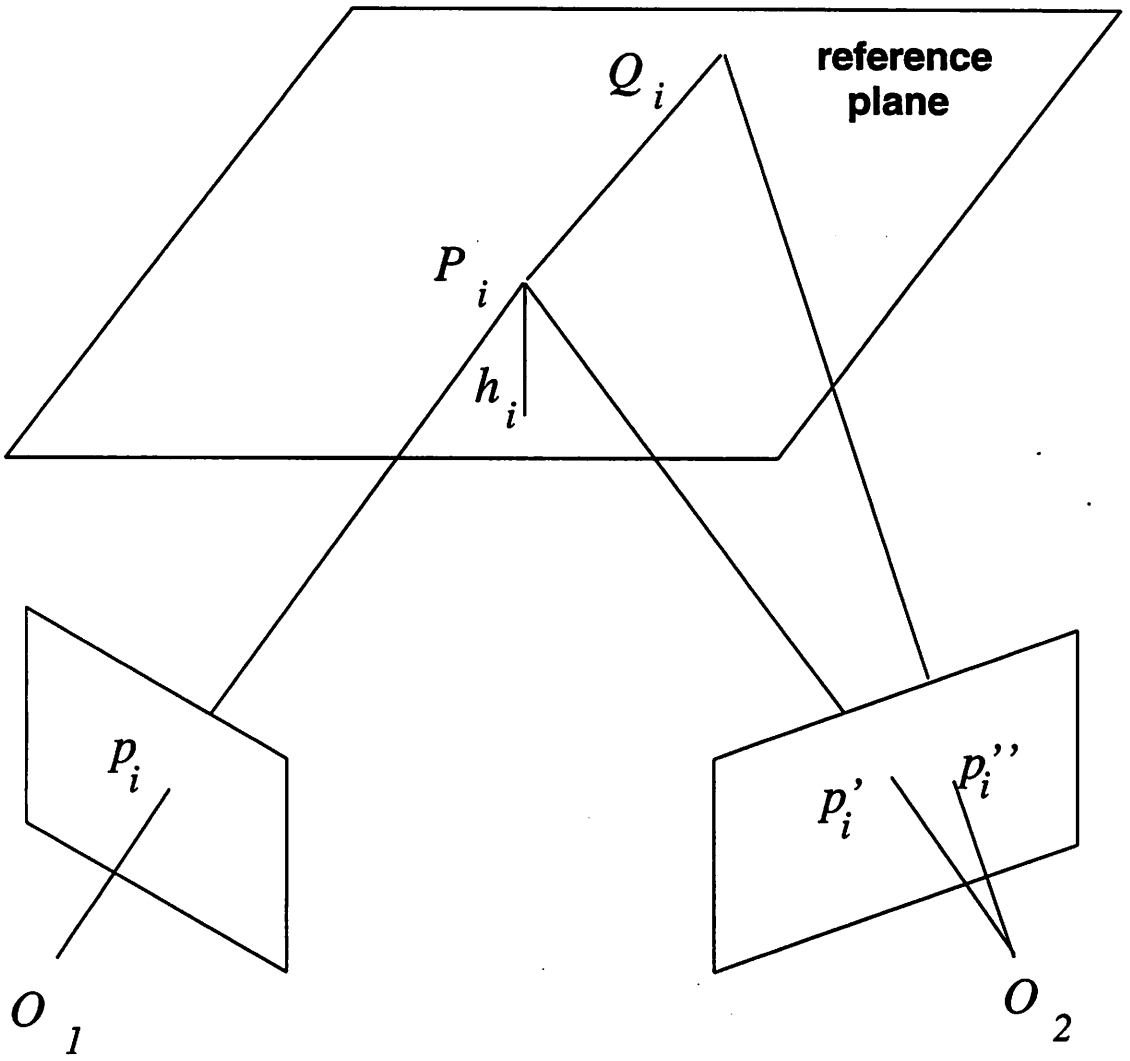

Figure 1: The geometry of an arbitrary 3D point P_i and its corresponding ground plane 3D point Q_i

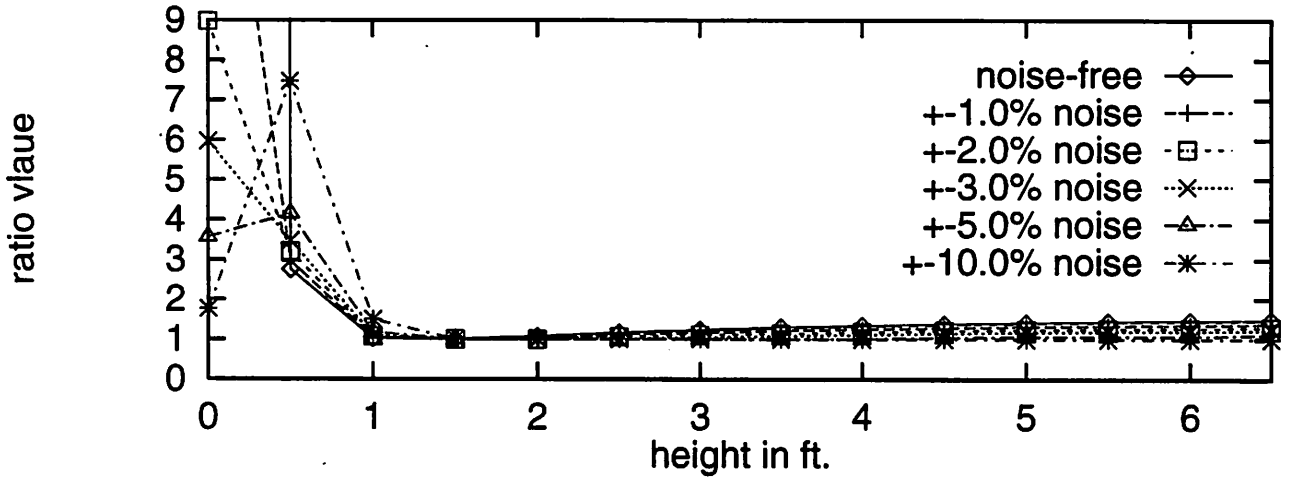

Figure 2: Simulation results of a 3D point located at 20 ft. based on **KGP**.

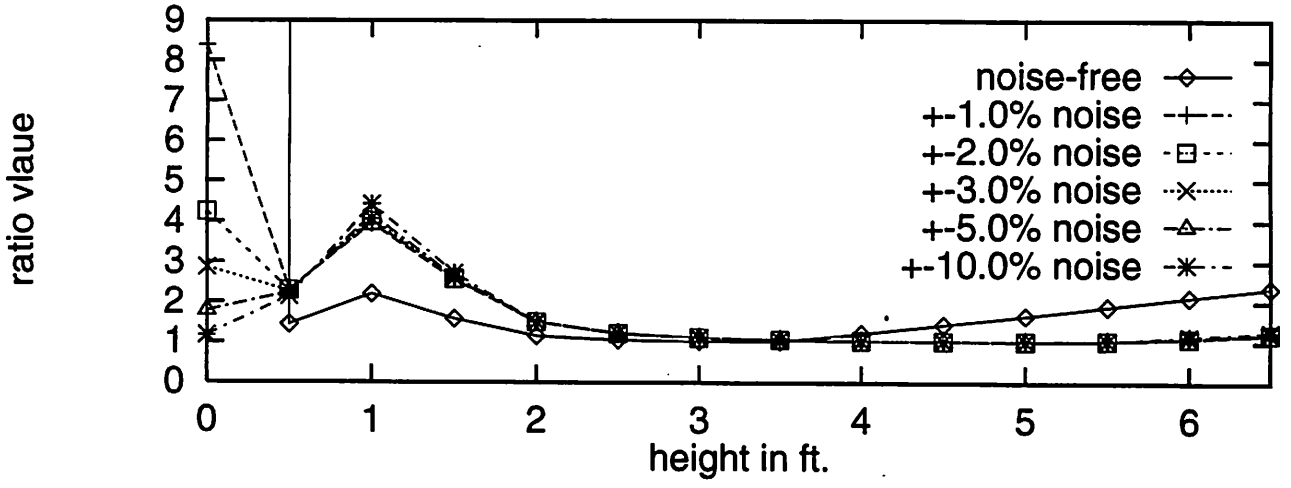

Figure 3: Simulation results of a 3D point located at 20 ft. based on **UGP**.

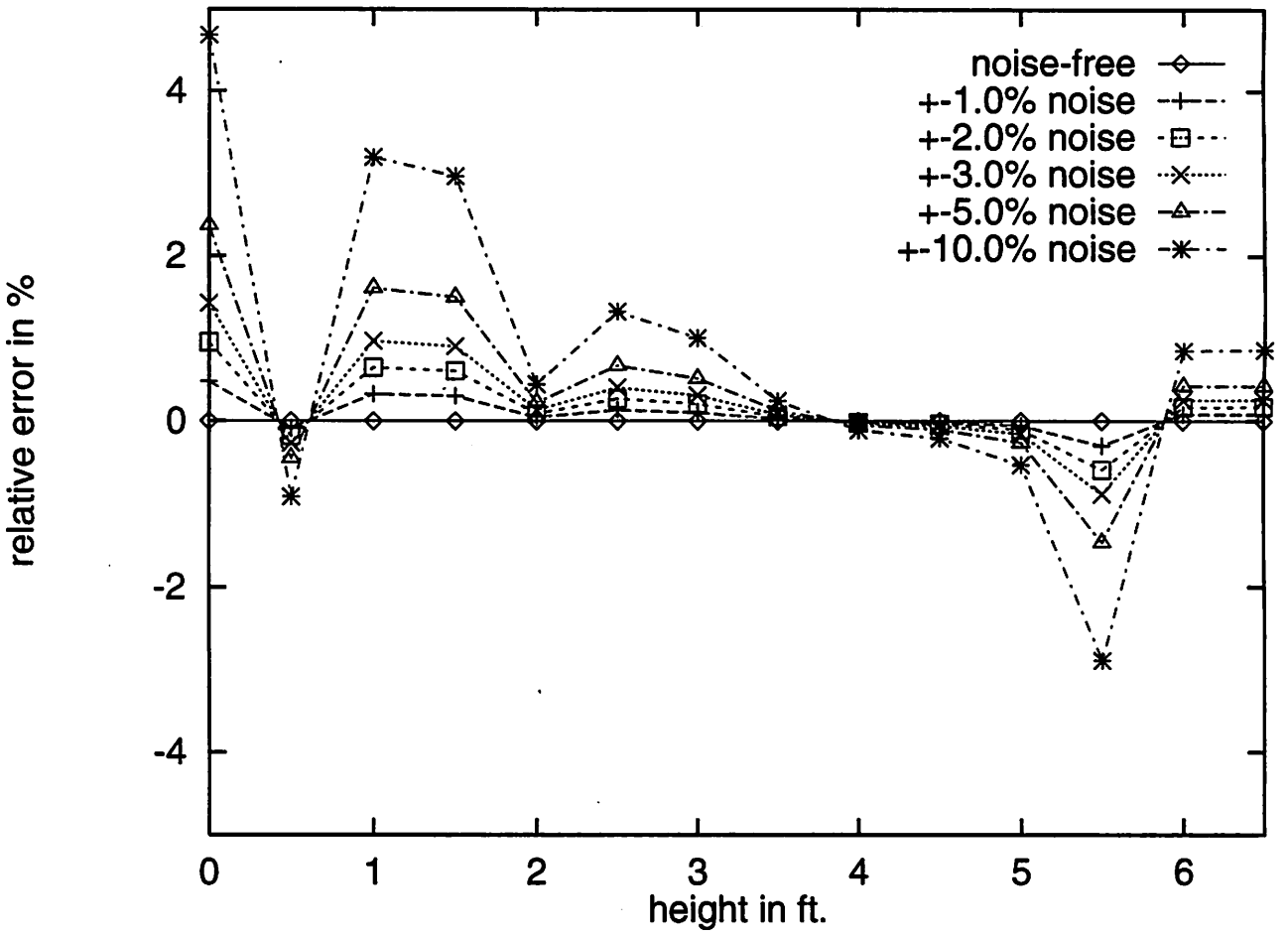

Figure 4: Simulation results of a 3D point located at 20 ft. based on **EGP**.

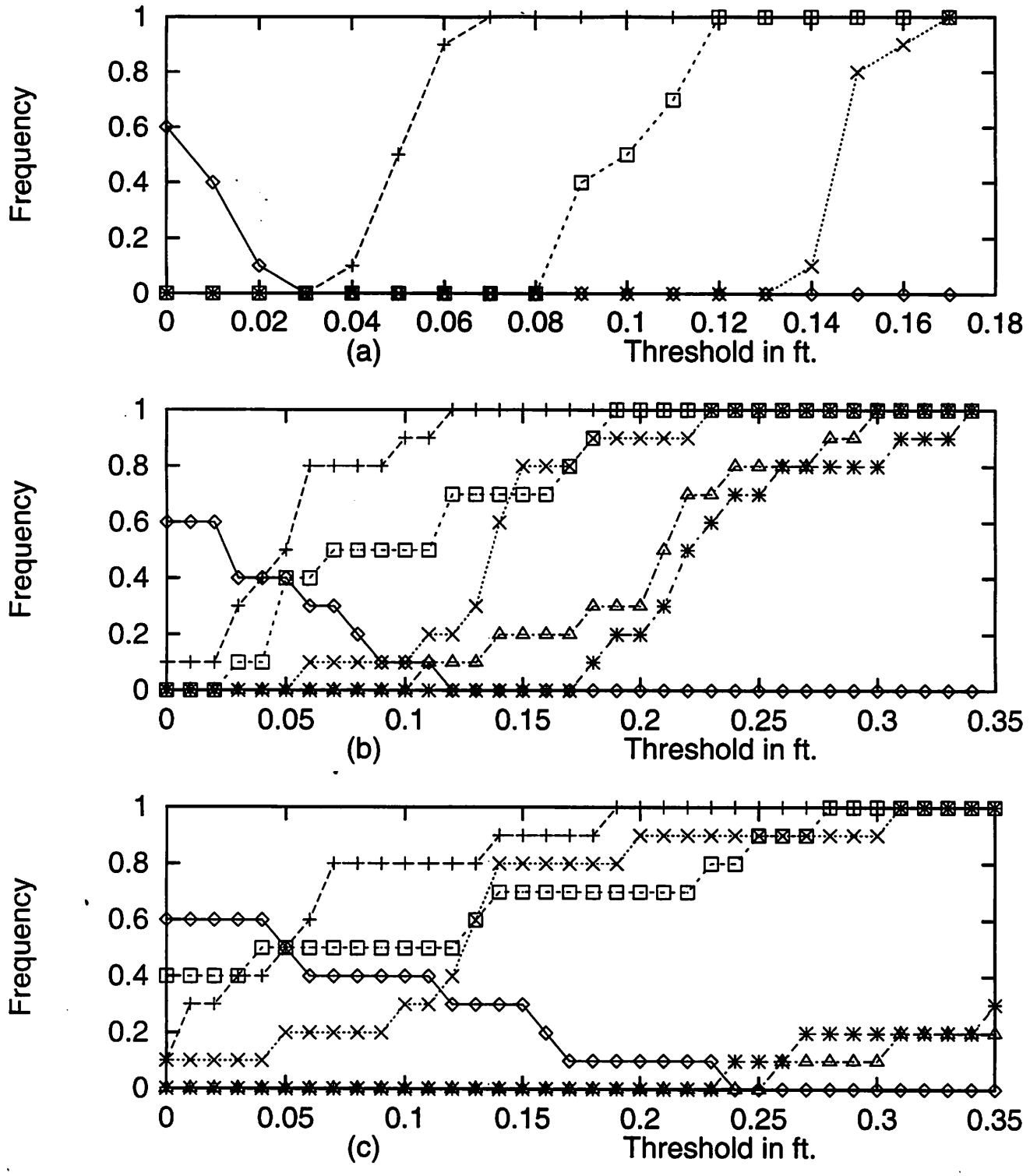

Figure 5: False-positives (f.p.) and false-negatives (f.n.) with respect to threshold for different obstacle heights (o.h.). (a) noise level $\pm 1.0\%$ where $\diamond$: f.p; $+$: f.n. with o.h. 0.05 ft.; $\square$: f.n. with o.h 0.10 ft.; $\times$: f.n. with o.h. 0.15 ft.; (b) noise level $\pm 5.0\%$ where $\diamond$: f.p; $+$: f.n. with o.h. 0.05 ft.; $\square$: f.n. with o.h 0.10 ft.; $\times$: f.n. with o.h. 0.15 ft.; $\triangle$: f.n. with o.h. 0.20 ft.; $*$: f.n. with o.h. 0.25 ft.; (c) noise level $\pm 10.0\%$ where $\diamond$: f.p; $+$: f.n. with o.h. 0.05 ft.; $\square$: f.n. with o.h 0.10 ft.; $\times$: f.n. with o.h. 0.15 ft.; $\triangle$: f.n. with o.h. 0.40 ft.; $*$: f.n. with o.h. 0.45 ft.;

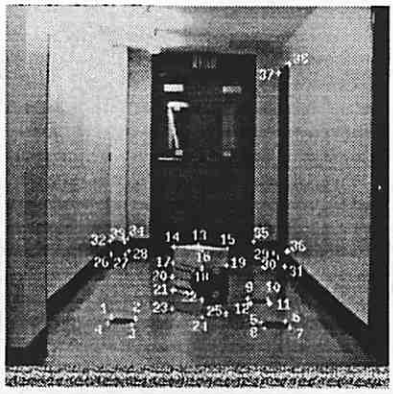

Figure 6: The left image of a hallway box scene

Table 3
Singular values with all points based on Coplanarity

SV's of $\boldsymbol{D}$	SV's of $[\boldsymbol{Db}]$
7.215e+5	7.215e+5
1.754e+5	1.754e+5
1.950e+3	1.950e+3
7.316e+2	7.839e+2
3.008e+2	4.434e+2
1.285e+2	1.286e+2
1.602e+0	2.067e+1
5.468e−1	1.062e+0
	5.335e−1

Table 4
Singular values with all ground plane points based on Coplanarity

SV's of $\boldsymbol{D}$	SV's of $[\boldsymbol{Db}]$
4.672e+5	4.672e+5
6.183e+4	6.183e+4
1.408e+3	1.408e+3
3.838e+2	4.099e+2
1.706e+2	2.469e+2
7.016e+1	7.016e+1
9.909e−1	1.317e+1
2.888e−1	6.133e−1
	4.380e−2

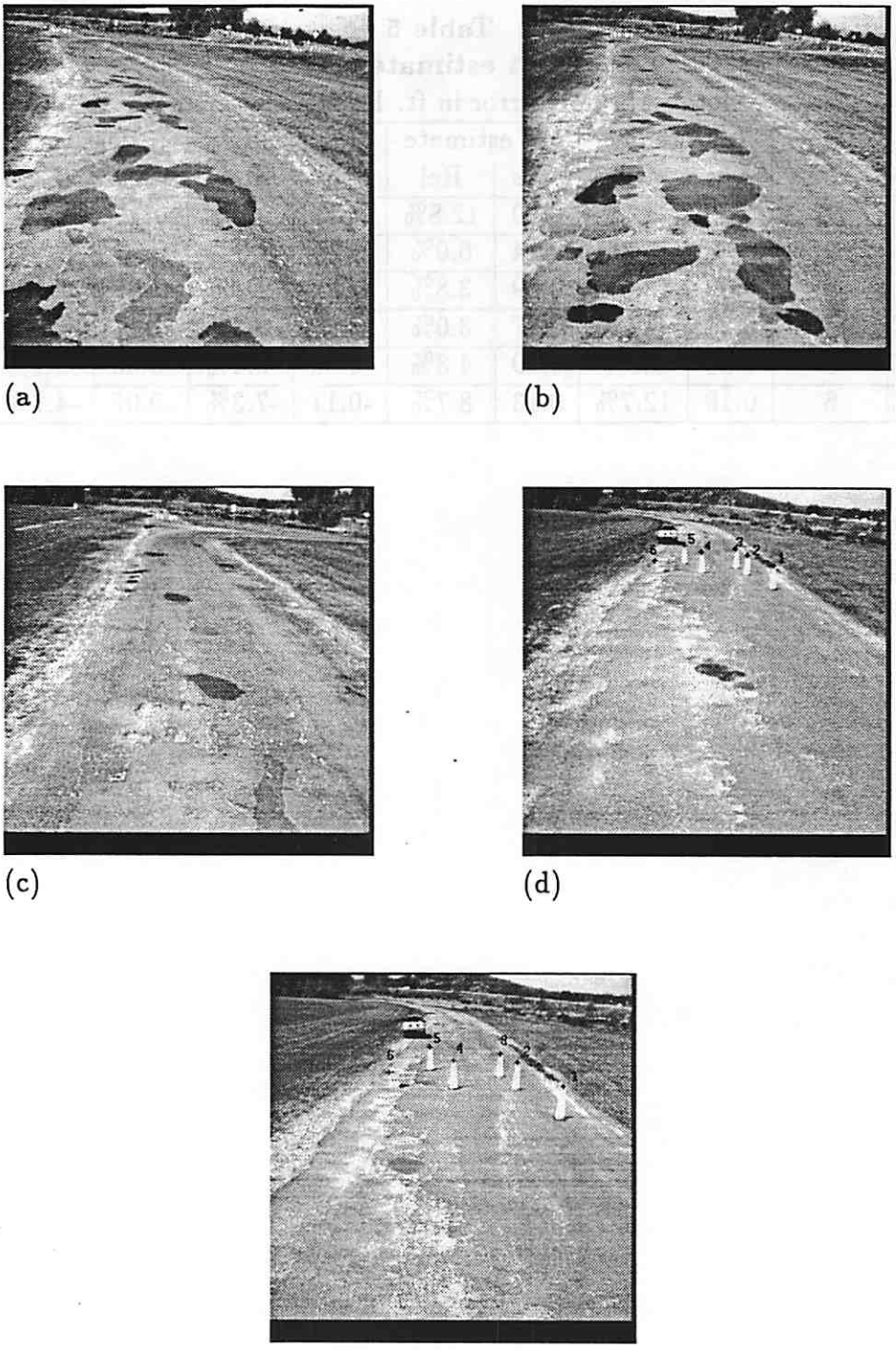

Figure 7: A sample of a stereo sequence. The right images are not shown here. (a) - (c) are the first three images which show the scene without obstacle. (d) is a scene with obstacles. (e) is the next frame with the same obstacles.

Table 5
Height estimate errors
Note: Absolute error in ft. Relative error in %

Point Labels	(d) predict		(d) estimate		(e) predict		(e) estimate	
	Abs	Rel	Abs	Rel	Abs	Rel	Abs	Rel
1	0.50	21.3%	0.30	12.8%	0.23	9.8%	0.11	4.7%
2	0.36	15.3%	0.14	6.0%	0.20	8.5%	0.13	5.5%
3	0.32	13.6%	0.09	3.8%	0.21	8.9%	0.15	6.4%
4	0.21	8.9%	0.07	3.0%	-0.01	-0.4%	0.0	0.0
5	0.23	9.8%	0.10	4.3%	-0.08	-3.4%	-0.05	-2.1%
6	0.19	12.7%	0.13	8.7%	-0.11	-7.3%	-0.07	-4.7%