On the Analysis of the Depth Error on the Road Plane for Monocular Vision-Based Robot Navigation*

Dezhen Song[1], Hyunnam Lee[2], and Jingang Yi[3]

[1] CS Dept., Texas A&M University, dzsong@cs.tamu.edu
[2] H. Lee is with Samsung Techwin Robot Business, hyunnamlee@gmail.com
[3] MAE Dept., Rutgers University, jgyi@jove.rutgers.edu

Abstract. A mobile robot equipped with a single camera can take images at different locations to obtain the 3D information of the environment for navigation. The depth information perceived by the robot is critical for obstacle avoidance. Given a calibrated camera, the accuracy of depth computation largely depends on locations where images have been taken. For any given image pair, the depth error in regions close to the camera baseline can be excessively large or even infinite due to the degeneracy introduced by the triangulation in depth computation. Unfortunately, this region often overlaps with the robot's moving direction, which could lead to collisions. To deal with the issue, we analyze depth computation and propose a predictive depth error model as a function of motion parameters. We name the region where the depth error is above a given threshold as an untrusted area. Note that the robot needs to know how its motion affect depth error distribution beforehand, we propose a closed-form model predicting how the untrusted area is distributed on the road plane for given robot/camera positions. The analytical results have been successfully verified in the experiments using a mobile robot.

1 Introduction

Vision-based navigation is preferable because cameras can be very small, passive, and energy-efficient. Using a single camera to assist a mobile robot is the most simplistic configuration and is often adopted in small robots. However, images from cameras contain rich information of the environment, and understanding the imaging data is nontrivial. Extracting geometry information from images is critical for obstacle avoidance. Stereo vision approaches are often employed.

For the monocular system, the stereo information can be constructed using structure from motion (SFM) approach [1]. This method constructs depth information using images taken at different perspectives. Since the robot motion changes camera perspectives, the baseline distance is not limited by the width of the robot and it is desirable for small robots. However, the SFM approach has its own limitation. The depth of obstacles located at

* This work was supported in part by the National Science Foundation under IIS-0534848 and IIS-0643298, and in part by Microsoft Corporation.

the baseline cannot be obtained because the camera centers and obstacle locations are collinear. Unfortunately, if the robot moves along a straight line, its forward direction is always the baseline direction.

Understanding depth error distribution on the road plane is critical for applications such as robot navigation. We model how depth error is distributed on the road plane and partition the road plane using a given error threshold. The predictive closed-form model is a function of robot motion settings and can be used to predict how the region beyond the given error threshold changes on the road plane. Hence the model has the potential to benefit a variety of applications including 1) guiding the robot for mixed initiative motion planning for better sensing and navigation, 2) guiding the selection of visual landmarks for vision-based simultaneous localization and mapping (SLAM), and 3) improving the visual tracking performance for mobile robots.

The proposed predictive depth error distribution model has been tested in physical experiments. The experiments use a mobile robot and artificial obstacles to validate the predictive depth error model. The experimental results have confirmed our analysis.

2 Related Work

Our research is related to monocular vision systems for robots, structure from motion (SFM) [1], and active vision [2–4].

Due to its simple configuration, a monocular vision system is widely used in mobile robots with space and power constraints. The research work in this category can be classified into two types including SLAM and vision-based navigation. SLAM [5–8] focuses on the mapping and localization aspects and is often used in structured indoor environments where there are no global positioning system (GPS) signals to assist robots in navigation. SLAM focuses on identifying and managing landmark/feature points from the scene for map building and localization. Obstacle avoidance is not the concern of SLAM.

Our work focuses on monocular vision-based navigation for obstacle detection and avoidance. Due to the inherent difficulty in understanding the environment using monocular vision, many researchers focus on applying machine learning techniques to assist navigation [9–12]. However, those methods are appearance-based and only utilize color and texture information. Lack of geometry information limits their ability in obstacle detection.

Our work is a geometry-based approach that uses SFM to obtain the information of the environment. SFM can simultaneously estimate both the 3D scene and camera motion information [1]. Since the camera motion information is usually available from on-board sensors such as an inertial measurement unit (IMU) or wheel encoders, the dimensionality of the SFM problem can be reduced to the only estimation of the 3D scene, namely the triangulation computation. The depth error is determined by the image correspon-

dence error and the camera perspectives. To obtain the 3D information, it is necessary to find the corresponding points between the overlapping images. However, due to the fact that images are discrete representations of the environment and the inherent difficulty in image matching, it is unavoidable that matching errors are introduced into the corresponding points [13, 14]. There are many newly developed techniques that can be used to reduce correspondence errors. Such techniques include low-rank approximations [15–17], power factorization [18], closure constraints [19], and covariance-weighted data [20]. In addition, new features, such as planar parallax [21–24] and the probability of correspondence points [25], can be used instead of correspondence points to reduce the correspondence error.

Our work accepts the fact that image correspondence cannot be eliminated completely. We instead study how the depth error is affected by the image correspondence error. Although the variance of the image correspondence error are the same across the image plane [13,14], the variance of depth error is not uniformly distributed across the image coverage [26]. Therefore, robot navigation and camera motion planning should take the depth error distribution information into account. This observation inspires our development.

3 Problem Description

3.1 Coordinate Systems

Our algorithm runs every τ_0 time. In each period, the robot has a trajectory $T(\tau)$, $\tau \in [0, \tau_0]$. The period length τ_0 is a preset parameter depending on the speed of the robot and the computation time necessary for stereo reconstruction. The most common approach to assist robot navigation is to take a frame $\underline{F}$ at $\tau = 0$ and another frame $\overline{F}$ at $\tau = \tau_0$ for the two-view stereo reconstruction. As a convention, we use underline and overline with variables to indicate their correspondence to $\underline{F}$ and $\overline{F}$, respectively. To clarify the problem, we introduce the following right hand coordinate systems as illustrated in Fig. 1.

- World coordinate system (WCS): A fixed 3D Cartesian coordinate system. Its y-axis is the vertical axis, and its x-z plane is the road plane. Trajectory $T(\tau)$ is located in the x-z plane with $T(\tau_0)$ located at the origin of the WCS. Hence, $T(\tau) = [x_w(\tau), z_w(\tau)]^T, 0 \leq \tau \leq \tau_0$ as illustrated in Fig. 1.
- Camera coordinate system (CCS): A 3D Cartesian coordinate system that is attached to a camera mounted on a robot with its origin at the camera optical center. Its z-axis coincides with the optical axis and points to the forward direction of the robot. Its x-axis and y-axis are parallel to the horizontal and vertical directions of the CCD sensor plane, respectively.

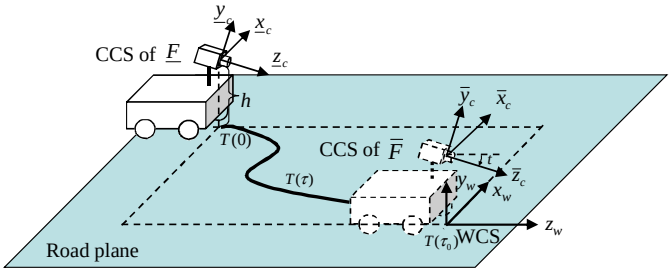


Fig. 1. Definition of coordinate systems and their relationship. The WCS is a fixed coordinate system while a CCS is attached to the moving camera.

- Image coordinate system (ICS): A 2D image coordinate system with the u-axis and v-axis parallel with the horizontal and vertical directions of an image, respectively. Its origin is located at its principal point. Coordinates u and v are discretized pixel readings. When we mention frames such as F, $\underline{F}$ and $\overline{F}$, they are defined in the ICS.

Frames such as $\underline{F}$ and $\overline{F}$ have their corresponding CCSs and ICSs. We use the notation $CCS(F)$ to represent the corresponding CCS for frame F. As illustrated in Fig. 1, the origin of $CCS(\overline{F})$ projects to $T(\tau_0)$ on the road plane, which is the origin of the WCS. The vertical distance between the origins of the $CCS(\overline{F})$ and the WCS is the camera height h. The origin of $CCS(\underline{F})$ projects to $T(0)$ on the road plane.

3.2 Assumptions

- We assume that obstacles in the environment are either static or slow-moving. Therefore, the SFM algorithm can be applied to compute the depth information.
- We assume both intrinsic and extrinsic camera parameters are known either from pre-calibration or camera angular potentiometers and robot motion sensors. The camera has square pixels and zero skew factors, which is valid for most cameras.
- The robot takes frames periodically for the stereo reconstruction. During each period, we assume that the road surface can be approximated by a plane, which is the x-z plane of the WCS as illustrated in Fig. 1.
- We assume that the pixel correspondence error across different frames is uniformly distributed in the ICS. We believe that the pixel correspondence errors do not have an infinite tail distribution in reality and the uniform distribution is a conservative description of the property.
- We assume all CCSs are iso-oriented with the $CCS(\overline{F})$, which is determined by the navigation direction at time τ_0. Although the robot may have different positions and orientations when taking images, we can project the images into the CCSs that are iso-oriented with $CCS(\overline{F})$

using a perspective re-projection because we know accurate camera parameters.

3.3 Problem Context

Frames and Frame Parameters For frames such as $\underline{F}$ and $\overline{F}$, we need to define their corresponding robot locations and camera parameters. As illustrated in Fig. 1, the camera is mounted at a height of h. Hence the camera position is uniquely defined by its coordinates (x_w, h, z_w) in the WCS. In order to have a good coverage of the road, the camera usually tilts towards the ground as illustrated in Fig. 1. The tilt angle is defined as t.

Obstacle-Free Region The previous period provides an obstacle-free road region R_f. The robot needs to stay in R_f and reach $T(\tau_0)$ at the end of the current period.

Region of Interest A camera frame usually covers a wide range, from adjacent regions to an infinite horizon. For navigational purposes, the robot is not interested in regions that are too far away. As illustrated in Fig. 1, the z-axis of the WCS points to the robot's forward direction at time $\tau = \tau_0$ when frame $\overline{F}$ is taken. z_M is defined as the maximal distance that the robot cares about in the next iteration of the algorithm. The region of interest R_i is a subset of camera coverage,

$$R_i = \{(x_w, z_w) | 0 \leq z_w \leq z_M, (x_w, z_w) \in \Pi(\overline{F})\}, \tag{1}$$

where x_w and z_w are defined in the WCS and function $\Pi(\overline{F})$ is the coverage of $\overline{F}$ in the x-z plane of the WCS. Our research problem is to understand how the depth error is associated with objects in R_i. To study how the depth error is distributed on the road plane, we introduce the *untrusted area* below.

3.4 Untrusted Area and Problem Formulation

The computed depth information is not accurate due to the image correspondence error. According to our assumptions, for a given pixel in $\underline{F}$, the corresponding pixel in $\overline{F}$ can be found with an error that is uniformly and independently distributed. Hence, the depth error is also a random variable. Define $e = z_w - \hat{z}_w$ as the depth error, which z_w is the true depth of the corresponding object in the WCS and $\hat{z}_w$ is the depth computed from the stereo reconstruction process. e has a range $|e| \leq |e_\Delta|$. The depth error range e_Δ will be formally defined later. We adopt $|e_\Delta|$ as the metric to characterize the quality of the depth information. $e_t > 0$ is a pre-defined threshold for $|e_\Delta|$. To facilitate robot navigation, we want to ensure that $|e_\Delta| \leq e_t$.

Although the image correspondence error is uniformly and independently distributed in the ICS, the influence of the image correspondence error on the depth is non-uniform due to a nonlinear stereo reconstruction process. For the two camera frames $\underline{F}$ and $\overline{F}$ taken from two different camera perspectives, we can construct the depth map for the overlapping regions of the two frames $\Pi(\underline{F} \cap \overline{F})$. We define the untrusted area $A_u(\underline{F}, \overline{F})$ that is constructed by the image pair $(\underline{F}, \overline{F})$ in the WCS as

$$A_u(\underline{F}, \overline{F}) = \{(x_w, z_w) | (x_w, z_w) \in \Pi(\underline{F} \cap \overline{F}), |e_\Delta(x_w, z_w)| > e_t\}, \tag{2}$$

because we know that the depth information in A_u is untrustworthy due to the excessive $|e_\Delta|$. Our problem is,

Definition 1 *For a given threshold $|e_\Delta|$, a pair of overlapping frames $(\underline{F}, \overline{F})$, and the corresponding camera parameters, compute $A_u(\underline{F}, \overline{F})$.*

The error threshold $|e_\Delta|$ is not necessarily a constant. For example, we define $e_t = \rho z_w$ where ρ is the relative error threshold and $0 < \rho < 1$. The choice of $|e_\Delta|$ and ρ depends on how conservative the motion planning is. A smaller value results in larger A_u and the robot has to travel longer distance to avoid A_u. In our experiments, $\rho = 20\%$ works well for navigation purpose.

4 Analysis of Depth Error

4.1 Computing Depth from Two Views

In stereo vision, 3D information is computed through triangulation under the perspective projection based on the extracted correspondence points from each pair of images [27]. Define $\underline{c}$ and $\overline{c}$ as camera centers for frames $\underline{F}$ and $\overline{F}$, respectively. Define $\underline{P}$ and $\overline{P}$ as the camera projection matrices for $\underline{F}$ and $\overline{F}$, respectively. Since the CCSs of $\underline{F}$ and $\overline{F}$ are iso-oriented and only differ from the WCS by a tilt value t in orientation, the orientation of the WCS with respect to the CCSs can be expressed by a rotation matrix

$$R_X(-t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(t) & s(t) \\ 0 & -s(t) & c(t) \end{bmatrix}.$$

Note that we use $s(\cdot)$ and $c(\cdot)$ to denote $\sin(\cdot)$ and $\cos(\cdot)$, respectively. If CCSs are not iso-oriented, it is not difficult to extend the rotation matrix using Euler angle sets. The origin of the WCS with respect to the CCSs of $\underline{F}$ and $\overline{F}$ are defined as $\underline{W}$ and $\overline{W}$, respectively. Since $T(0) = [x_w(0), z_w(0)]^T$, $T(\tau_0) = [0, 0]^T$, and the camera height is h, the camera center positions with respect to the WCS are $\underline{c} = [x_w(0), h, z_w(0)]^T$ and $\overline{c} = [0, h, 0]^T$, respectively. Then we have,

$$\underline{W} = -R_X(-t)\underline{c}, \text{ and } \overline{W} = -R_X(-t)\overline{c}.$$

Therefore,

$$\underline{P} = K[R_X(-t)|\underline{W}],\ \overline{P} = K[R_X(-t)|\overline{W}], K = diag(f, f, 1),$$

where f is the focal length of the camera divided by the side length of a pixel. Let $\underline{q} = [\underline{u}\ \underline{v}\ 1]^T$ and $\overline{q} = [\overline{u}\ \overline{v}\ 1]^T$ be a pair of corresponding points in $\underline{F}$ and $\overline{F}$, respectively. Define $Q = [x_w, y_w, z_w]^T$ as their corresponding point in WCS. Let $\underline{Q}_c = [\underline{x}_c\ \underline{y}_c\ \underline{z}_c]^T$ and $\overline{Q}_c = [\overline{x}_c\ \overline{y}_c\ \overline{z}_c]^T$ be Q's position in the CCSs of $\underline{F}$ and $\overline{F}$, respectively. Also, we know that $\underline{Q}_c$ and $\overline{Q}_c$ can be expressed as,

$$\underline{Q}_c = R_X(-t)Q + \underline{W},\ \text{and}\ \overline{Q}_c = R_X(-t)Q + \overline{W}. \tag{3}$$

The following holds according to the pin-hole camera model,

$$\underline{q} = \frac{1}{\underline{z}_c}\underline{P}\begin{bmatrix} Q \\ 1 \end{bmatrix} = \frac{1}{\underline{z}_c}K\underline{Q}_c,\ \text{and}\ \overline{q} = \frac{1}{\overline{z}_c}\overline{P}\begin{bmatrix} Q \\ 1 \end{bmatrix} = \frac{1}{\overline{z}_c}K\overline{Q}_c. \tag{4}$$

From (3), we know $\underline{Q}_c = \overline{Q}_c + (\underline{W} - \overline{W})$, namely,

$$\underline{x}_c = \overline{x}_c - x_w(0),\ \ \underline{y}_c = \overline{y}_c - z_w(0)s(t),\ \text{and}\ \underline{z}_c = \overline{z}_c - z_w(0)c(t). \tag{5}$$

From (4) and (5), we obtain,

$$\underline{q} = \frac{1}{\overline{z}_c - z_w(0)c(t)}\left(\overline{z}_c\overline{q} + K(\underline{W} - \overline{W})\right). \tag{6}$$

Since K, $\underline{W}$, $\overline{W}$, $\underline{q}$, and $\overline{q}$ are known, (6) consists of a system of equations with $\overline{z}_c$ as an unknown quantity. There is one unknown variable and a total of two equations (e.g. the first two equations in (6)). This is an overly-determined equation system. A typical approach would be to apply a least-square (LS) method [27]. Using the solution from LS method would result in a high-order polynomial when analyzing the depth error. Solving the high-order polynomial is computationally inefficient. Another method is to simply discard one equation and solve it directly. This method has a speed advantage and its solution quality is slightly worse than that of the LS method. The advantage is that the format of solution can be expressed in simpler format that allows us to derive the depth error distribution. Actually, a worse solution can actually provide a more conservative error bound than that of the LS method. Employing the method and solving the first equation in (6), we have $\overline{z}_c = \frac{x_w(0)f - \underline{u}z_w(0)c(t)}{\overline{u} - \underline{u}}$. From (3), we know $z_w = \overline{z}_c\left(\frac{\overline{v}}{f}s(t) + c(t)\right)$. Hence,

$$z_w = \frac{x_w(0)f - \underline{u}z_w(0)}{\overline{u} - \underline{u}}\left(\frac{\overline{v}}{f}s(t) + c(t)\right). \tag{7}$$

Depth z_w describes the distance from the robot to an obstacle along the z-axis of the WCS. Its error directly affects the robot's collision avoidance performance.

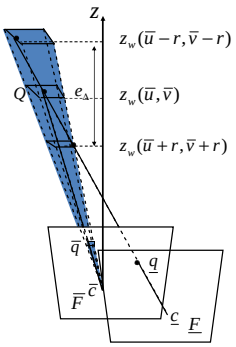


Fig. 2. An illustration of depth error caused by the image correspondence error in $\overline{F}$. The intersection zone between the ray from $\underline{q}$, and the pyramid from $\overline{q}$ is the error range. If the error range projects onto the z axis, it is always bound between $z_w(\overline{u}+r,\overline{v}+r)$ and $z_w(\overline{u}-r,\overline{v}-r)$.

4.2 Estimating the Depth Error Range

For the given pair of corresponding points ($\underline{q}$, $\overline{q}$) from ($\underline{F}$, $\overline{F}$) with camera centers ($\underline{c}$, $\overline{c}$), the geometric interpretation of the above triangulation process is the following. If we back project a ray from $\underline{c}$ through $\underline{q}$, it intersects with the ray generated by back-projecting from $\overline{c}$ through $\overline{q}$, provided that both $\underline{q}$ and $\overline{q}$ are accurate. The intersection point in the 3D space is Q; see Fig. 2.

However, for a given point $\underline{q}$, finding the accurate $\overline{q}$ is unlikely due to noises and pixelization errors. According to our assumptions, the corresponding errors in $\overline{u}$ and $\overline{v}$ are independently distributed according to $U(-r,r)$, where r is usually 0.5-2 pixels in length. This means that $\overline{q}$ is distributed in a small square on $\overline{F}$. When we back project the square, it forms a pyramid in 3D space as illustrated in Fig. 2. When the pyramid meets the ray that is back-projected from $\underline{q}$, it has a range of intersections instead of a single point. The estimated depth z_w is a function of random variables $(\overline{u},\overline{v})$ and can be expressed as $z_w(\overline{u},\overline{v})$. It is apparent that z_w is a random variable that could take any value in this intersection zone.

To compute the intersection zone, we need to compute the intersection points between the ray from $\underline{c}$ through $\underline{q}$ and all four side planes of the pyramid. However, the solution cannot be expressed in a closed-form for further analysis. Instead, we employ the upper and the lower bounds of the length of the intersection zone as illustrated in Fig. 2. Then the bound of the maximum length of the intersection zone is defined as $|e_\Delta|$, where

$$e_\Delta = z_w(\overline{u}+r,\overline{v}+r) - z_w(\overline{u}-r,\overline{v}-r). \tag{8}$$

We skip the process of deriving the bound due to the page limit. The intuition is that any line segment bounded inside the pyramid truncated between plane $z_w(\overline{u}+r,\overline{v}+r)$ and $z_w(\overline{u}-r,\overline{v}-r)$ is shorter than e_Δ. Similarly, another choice is $z_w(\overline{u}-r,\overline{v}+r) - z_w(\overline{u}+r,\overline{v}-r)$. Since both the analysis method and results are similar, we use (8) in the rest of the paper.

$|e_\Delta|$ describes the range of the depth error and is employed as the metric to measure the quality of the stereo reconstruction. To simplify the notation

in computing e_Δ, we define the following intermediate variables for (7).

$$\lambda = \beta\overline{v} + c(t), \quad \zeta_d = \overline{u} - \underline{u}, \quad \beta = \frac{s(t)}{f}, \ \zeta_n = x_w(0)f - \underline{u}z_w(0). \tag{9}$$

Then $z_w = \lambda\frac{\zeta_n}{\zeta_d}$ according to (7) and (9). Substituting them into (8), we have,

$$e_\Delta = (\lambda + r\beta)\frac{\zeta_n}{\zeta_d + r} - (\lambda - r\beta)\frac{\zeta_n}{\zeta_d - r} = \zeta_n\frac{2r(\beta\zeta_d - \lambda)}{\zeta_d^2 - r^2}. \tag{10}$$

Eq. (10) illustrates e_Δ in the ICS. For robot navigation purposes, we are interested in e_Δ in the x-z plane of the WCS. Hence $\underline{u}$, $\overline{u}$ and $\overline{v}$ in (10) should be transformed into functions of x_w and z_w. Recall that $s(\cdot)$ and $c(\cdot)$ denote $\sin(\cdot)$ and $\cos(\cdot)$, respectively. From (4), (7), and (9), we know $\overline{u} = \frac{x_w f}{\overline{z}_c} = f\lambda\frac{x_w}{z_w}$, and $y_w = \left(\frac{\overline{v}}{f}c(t) - s(t)\right)\overline{z}_c + h$. Since we are interested in obstacles on the x-z plane, $y_w = 0$, we have $\overline{v} = \frac{f(z_w s(t) - hc(t))}{z_w c(t) + hs(t)}$. Similarly, from (4), (7), and (9), we know $\underline{u} = \alpha_x x_w + \alpha_0$, where $\alpha_x = \frac{f}{\overline{z}_c - z_w(0)c(t)} = \frac{f\lambda}{z_w - z_w(0)c(t)\lambda}$, and $\alpha_0 = -x_w(0)\alpha_x$. Plugging into (9), we obtain the intermediate variables λ, ζ_n, and ζ_d, in terms of x_w and z_w.

$$\lambda = \frac{z_w}{z_w c(t) + hs(t)}, \ \zeta_n = n_x x_w + n_0, \text{ and } \zeta_d = \frac{n_x\lambda}{z_w}x_w + \frac{n_0\lambda}{z_w},$$

where $n_x = -z_w(0)c(t)\alpha_x$ and $n_0 = x_w(0)z_w\alpha_x/\lambda$. Plugging them into (10), we obtain e_Δ as a function of x_w and z_w,

$$e_\Delta = \frac{2r\beta\lambda z_w(n_x x_w + n_0)^2 - 2r\lambda z_w^2(n_x x_w + n_0)}{\lambda^2(n_x x_w + n_0)^2 - r^2 z_w^2}. \tag{11}$$

For an obstacle located at $(x_w, 0, z_w)$, Eq. (11) allows us to estimate e_Δ. It is clear that the depth error range varies dramatically in different regions, and thus should be considered in robot navigation to avoid obstacles.

4.3 Predicting Untrusted Area

For a given frame pair with the corresponding robot locations, we can partition R_i using a preset depth error threshold $e_t > 0$. We are now ready to predict A_u by computing its boundary using Eq. (11).

Partition R_i According to the Sign of e_Δ. To find the regions corresponding to $|e_\Delta| < e_t$, there are two possible cases to consider: $e_\Delta < 0$ and $e_\Delta > 0$. We can rewrite (11) as,

$$e_\Delta = \frac{2r\lambda z_w(x_w - \mu_{n1})(x_w - \mu_{n2})}{(x_w - \mu_{d1})(x_w - \mu_{d2})}, \tag{12}$$

where

$$\mu_{n1} = \frac{x_w(0)}{z_w(0)\lambda c(t)} z_w, \quad \mu_{n2} = \frac{x_w(0)}{z_w(0)\lambda c(t)} z_w - \frac{z_w(z_w - z_w(0)\lambda c(t))}{f z_w(0)\lambda\beta c(t)},$$
$$\mu_{d1} = \frac{x_w(0)}{z_w(0)\lambda c(t)} z_w + \frac{r z_w(z_w - z_w(0)\lambda c(t))}{f z_w(0)\lambda^2 c(t)},$$
$$\mu_{d2} = \frac{x_w(0)}{z_w(0)\lambda c(t)} z_w - \frac{r z_w(z_w - z_w(0)\lambda c(t))}{f z_w(0)\lambda^2 c(t)}.$$

Recall that t is the camera tilt angle and a typical camera setup has $0 \leq t \leq 30°$. A regular camera would have a focal length of 5-100 mm and pixel side length of 5-10 μm. Therefore, $f \geq 100$. Since $\beta = s(t)/f$,

$$0 < \beta \leq \sin(30°)/100 = 0.005. \tag{13}$$

Also we know that

$$\lambda = \beta\overline{v} + c(t) = s(t)\frac{\overline{v}}{f} + c(t) > \beta \tag{14}$$

because $|\frac{\overline{v}}{f}| < 1$ for any camera with a vertical field of view less than 90°. Combining this information, we have $0 < \beta < r/\lambda$ and $\beta < \lambda$. For obstacles in R_i, $z_w > 0$ according to the definition of the WCS. Also $z_w(0) < 0$ as illustrated in Fig. 1. Hence, we have

$$\frac{z_w(z_w - z_w(0)\lambda c(t))}{f z_w(0)\lambda c(t)} < 0. \tag{15}$$

Combining the inequalities above, we can derive the following relationship:

$$\mu_{d1} < \mu_{n1} < \mu_{d2} < \mu_{n2}. \tag{16}$$

Combining (16) with (12), we have,

$$e_\Delta > 0 \text{ if } \mu_{n1} < x_w < \mu_{d2} \text{ or } x_w < \mu_{d1}, \tag{17}$$
$$e_\Delta < 0 \text{ if } \mu_{d2} < x_w < \mu_{n2} \text{ or } \mu_{d1} < x_w < \mu_{n1}. \tag{18}$$

We ignore the region $x_w > \mu_{n2}$ in $e_\Delta > 0$ as this region is always outside of the camera's coverage.

We are now ready to compute A_u for the two cases defined in (17) and (18).

Computing A_u for $e_\Delta > 0$. This is the case illustrated in Fig. 2(a). Recall that the untrusted area satisfies $e_\Delta > e_t$. It is worth mentioning that the error threshold e_t is usually not a fixed number but a function of z_w. Recall that $e_t = \rho z_w$ where ρ is the relative error threshold. There are two cases: Case (i): $x_w < \mu_{d1}$ and Case (ii): $\mu_{n1} < x_w < \mu_{d2}$.

Case (i): when $x_w < \mu_{d1}$, the denominator of e_Δ in (12) is positive. Plug (12) into $e_\Delta > e_t$, and we have

$$(e_t\lambda^2 - 2r\beta\lambda z_w)n_x^2 x_w^2 + (2(e_t\lambda^2 - 2r\beta\lambda z_w)n_x n_0 + 2r\lambda n_x z_w^2)x_w + (e_t\lambda^2 - 2r\beta\lambda z_w)n_0^2 - e_t r^2 z_w^2 + 2r\lambda n_0 z_w^2 < 0. \quad (19)$$

The solution to the quadratic inequality (19) is

$$\frac{-\kappa_1 - \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} < x_w < \frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2}, \quad (20)$$

where

$$\kappa_2 = (e_t\lambda^2 - 2r\beta\lambda z_w)n_x^2, \quad \kappa_1 = 2(e_t\lambda^2 - 2r\beta\lambda z_w)n_x n_0 + 2r\lambda n_x z_w^2,$$
$$\kappa_0 = (e_t\lambda^2 - 2r\beta\lambda z_w)n_0^2 - e_t r^2 z_w^2 + 2r\lambda n_0 z_w^2.$$

The untrusted area is the region that satisfies (20) and $x_w < \mu_{d1}$. To compute the intersection, we need to understand the relationship between the solution in (20) and the coefficients in (12). Combining them, we know,

$$\mu_{d1} - \frac{-\kappa_1 - \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} = \frac{rz_w(z_w - z_w(0)\lambda c(t))}{fz_w(0)\lambda^2 c(t)}\left(1 - \frac{\lambda + \sqrt{\lambda^2 + \rho^2\lambda^2 - 2r\beta\lambda\rho}}{\rho\lambda - 2r\beta}\right).$$

Notice that $0 < r \leq 2$, $0 < \rho < 1$, β is very small according to (13), and $\lambda > 0$ according to (14). Therefore, $2r\beta$ and $2r\beta\lambda\rho$ are close to zero. Hence, we approximate $\left(1 - \frac{\lambda + \sqrt{\lambda^2 + \rho^2\lambda^2 - 2r\beta\lambda\rho}}{\rho\lambda - 2r\beta}\right) \approx 1 - \frac{2}{\rho} < 0$. Combining this equation with (15), we know,

$$\mu_{d1} > \frac{-\kappa_1 - \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2}. \quad (21)$$

Similarly, we can obtain

$$\mu_{d1} < \frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2}. \quad (22)$$

According to (20), (21), (22), and $x_w < \mu_{d1}$, the untrusted area for this case is given by,

$$\frac{-\kappa_1 - \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} < x_w < \mu_{d1}. \quad (23)$$

Case (ii): when $\mu_{n1} < x_w < \mu_{d2}$, from (16), we know that the denominator of (12) is negative. Hence, $\kappa_2 x_w^2 + \kappa_1 x_w + \kappa_0 > 0$. Similar to the analysis in Case (i), we obtain,

$$\frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} < x_w < \mu_{d2}. \quad (24)$$

Computing A_u for $e_\Delta < 0$. In this case, the untrusted area is the region that satisfies $e_\Delta < -e_t$. There are also two cases including Case (i): $\mu_{d2} < x_w < \mu_{n2}$ and Case (ii): $\mu_{d1} < x_w < \mu_{n1}$. Similar to the analysis in the previous cases, we obtain,

$$\mu_{d2} < x_w < \frac{-\kappa_1' - \sqrt{{\kappa_1'}^2 - 4\kappa_2'\kappa_0'}}{2\kappa_2'} \text{ and } \mu_{d1} < x_w < \frac{-\kappa_1' + \sqrt{{\kappa_1'}^2 - 4\kappa_2'\kappa_0'}}{2\kappa_2'}. \tag{25}$$

where,

$$\kappa_2' = (-e_t\lambda^2 - 2r\beta\lambda z_w)n_x^2, \quad \kappa_1' = 2(-e_t\lambda^2 - 2r\beta\lambda z_w)n_x n_0 + 2r\lambda n_x z_w^2,$$
$$\kappa_0' = (-e_t\lambda^2 - 2r\beta\lambda z_w)n_0^2 + e_t r^2 z_w^2 + 2r\lambda n_0 z_w^2.$$

Computing the Overall A_u. The overall A_u is the union of solution sets of the four cases given by (23), (24), and (25). Let us observe the relationship between the two inner boundaries in A_u,

$$\frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} - \frac{-\kappa_1' + \sqrt{{\kappa_1'}^2 - 4\kappa_2'\kappa_0'}}{2\kappa_2'} \approx 0$$

because $2r\beta\rho\lambda \approx 0$ and $2r\beta \approx 0$. Hence, we have

$$A_u = \Bigg\{ (x_w, z_w) | 0 \le z_w \le z_M, \\ \frac{-\kappa_1 - \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} < x_w < \frac{-\kappa_1' - \sqrt{{\kappa_1'}^2 - 4\kappa_2'\kappa_0'}}{2\kappa_2'} \Bigg\}. \tag{26}$$

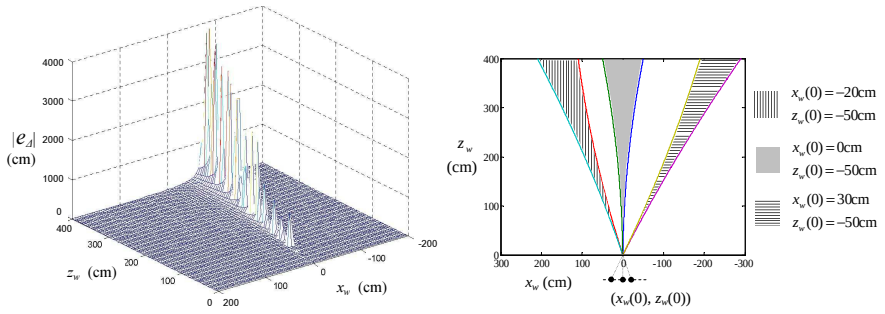
Eq. (26) also tells us how to obtain the boundaries of A_u. Represented as a function of x_w, we define the lower boundary and the upper boundary of A_u as x_w^- and x_w^+, respectively. Hence we have the two boundaries

$$x_w^{\mp}(z_w, x_w(0), z_w(0)) = \frac{x_w(0) z_w}{z_w(0)\lambda c(t)} + \\ \frac{r z_w (z_w - z_w(0)\lambda c(t))}{f z_w(0)\lambda c(t)(\pm e_t\lambda^2 - 2r\beta\lambda z_w)} (z_w\lambda + \sqrt{\lambda^2 z_w^2 + e_t^2\lambda^2 \mp 2re_t\beta\lambda z_w}). \tag{27}$$

5 Experiments

We have verified our analysis for the depth error estimation using a three-wheeled mobile robot. The robot has two front driving wheels and one rear

castor. The robot is 30 cm long, 30 cm wide, 33 cm tall and can travel at a speed of 25 cm/s with a 25 lbs payload. It is also equipped with two wheel encoders and a digital compass. The camera mounted on the robot is a Canon VCC4 pan-tilt-zoom camera with a 47.5° horizontal field of view. The camera mounting height $h = 44$ cm. The intrinsic camera parameters are estimated using the Matlab calibration toolbox [28]. During the experiment, we set $z_M = 4$ m and $t = 15°$ according to our robot and camera configurations. We conducted the experiments in the H. R. Bright Bldg. at Texas A&M University. The obstacles used in the experiments are books and blocks with a size of 20 cm × 14.5 cm × 10 cm.



(a) (b)

Fig. 3. (a) An illustration of $|e_\Delta|$. Robot positions are set to be $x_w(0) = 10$ cm, $z_w(0) = -50$ cm. (b) A_us with different robot positions $(x_w(0), z_w(0))$, which are the black dots in the figure. We set the threshold $e_t = 0.2 z_w$.

Fig. 3(a) illustrates how $|e_\Delta|$ is distributed on the road plane $y_w = 0$ according to our analysis. The 3D mesh is just an approximation of actual $|e_\Delta|$ distribution because it is generated by a finite set of testing locations. The illustration avoids the points on the baseline because the corresponding error range is infinite. It is apparent that the depth error is excessive in the area that is close to the camera baseline.

The second test is to show to how the different camera perspectives affect the location of A_u. Fig. 3(b) gives three examples of A_u for different camera perspectives $(x_w(0), z_w(0))$. It is clear that the selection of perspective can determine the location of A_u.

We also compared the depth error for objects inside and outside A_u in actual robot navigation. To facilitate the comparison, we defined the relative depth error in percentage $e_r = \frac{|e|}{z_w} \times 100$, where z_w is the measured depth that is used as a ground truth. We compare e_r for objects inside and outside the A_u for two scenarios: (a) the different depth of objects and (b) different robot positions as illustrated in Fig. 4. In (a), in each trial, the testing objects are randomly placed with a fixed depth. In (b), we change the relative position

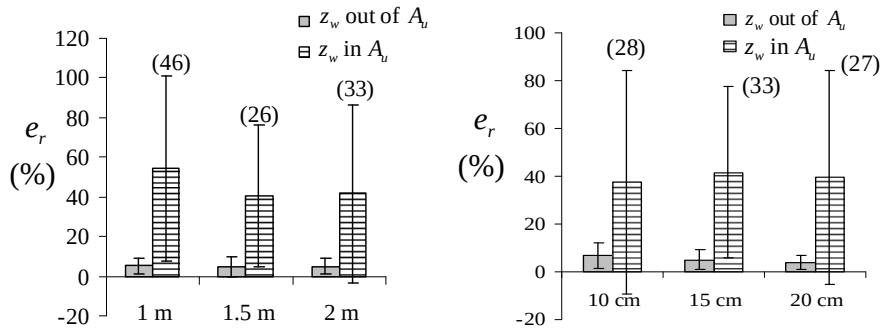


Fig. 4. The effectiveness of depth error reduction. The height of the bar is the mean value of e_r and the vertical interval represents the variance of e_r. The number in the parenthesis is the total number of trials.

between two camera perspectives to verify the depth error with respect to A_u. Obstacles are randomly placed in each trial. The accurate total number of trials for each setup is shown above the bars in the figures. In both (a) and (b), we first compute the obstacle depth using stereo vision and then compare it with the measured ground truth by computing e_r. Note that the mean and the variance of e_r are significantly reduced if the robot stays outside A_u.

6 Conclusion and Future Work

We analyzed the depth error range distribution across the camera coverage for a mobile robot equipped with a single camera. For SFM-based stereo vision for navigation, we showed that the depth error can be excessively large and hence cause collisions in robot navigation. We defined and modeled the untrusted area where the depth error range is beyond a preset threshold. Physical experiment results confirmed our analysis. In the future, we will apply the analysis into a new robot motion planning algorithm that will purposefully generate trajectories to avoid the untrusted area. The introduction of the untrusted area will help us to add more camera perspectives for the SFM. The introduction of the predictive model of the untrusted area opens a door to add depth-error aware planning into a variety of applications involving the monocular vision system. It is possible to use the untrusted area to guide the visual landmark selection for SLAM. Similarly, the untrusted area can be used to improve visual tracking performance when the robot plans to follow a moving target.

Acknowledgement

W thank R. Volz and R. Gutierrez-Osuna for their insightful discussions. Thanks for Y. Xu, C. Kim, H. Wang, N. Qin, B. Fine, and T. Southard for their inputs and contributions to the Netbot Laboratory, Texas A&M University.

References

1. F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 492–504, May 1996.
2. R. Bajcsy, "Active perception," in *Proceedings of the IEEE*, vol. 76, no. 8, August 1988, pp. 996–1005.
3. B. Zavidovique, "First steps of robotic perception: The turning point of the 1990s," in *Proceedings of the IEEE*, vol. 90, no. 7, July 2002, pp. 1094–1112.
4. K. Tarabanis, P. Allen, and R. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, February 1995.
5. E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Monocular vision based slam for mobile robots," in *The 18th International Conference on Pattern Recognition*, August 2006, pp. 1027–1031.
6. Z. Chen, R. Rodrigo, and J. Samarabandu, "Implementation of an update scheme for monocular visual slam," in *International Conference on Information and Automation*, December 2006, pp. 212–217.
7. E. Mortard, B. Raducanu, V. Cadenat, and J. Vitria, "Incremental on-line topological map learning for a visual homing application," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 2049–2054.
8. T. Lemaire and S. Lacroix, "Monocular-vision based slam using line segments," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 2791–2796.
9. E. Royer, J. Bom, B. T. Michel Dhome, M. Lhuillier, and F. Marmoiton, "Outdoor autonomous navigation using monocular vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2005, pp. 1253–1258.
10. Z. Chen and S. T. Birchfield, "Qualitative vision-based mobile robot navigation," in *IEEE International Conference on Robotics and Automation, Olrando, FL*, May 2006, pp. 2686– 2692.
11. J. Michels, A. Saxena, and A. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *22nd International Conference on Machine Learning*, August 2005, pp. 593 – 600.
12. D. Song, H. Lee, J. Yi, and A. Levandowski, "Vision-based motion planning for an autonomous motorcycle on ill-structured roads," *Autonomous Robots*, vol. 23, no. 3, pp. 197–212, October 2007.
13. A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, June 1995.

14. T. Jebara, A. azarbayejani, and A. Pentland, "3D structure from 2D motion," in *IEEE Signal Processing Magazine*, vol. 16, no. 3, May 1999, pp. 66–84.
15. C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, November 1992.
16. D. Martinec and T. Pajdla, "3D recontruction by fitting low-rank matrices with missing data," in *IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA*, June 2005, pp. 198–205.
17. S. Brandt, "Closed-form solutions for affine reconstruction under missing data," in *7th European Conference on Computer Vision, Copenhagen, Denmark*, May 2002, pp. 109–114.
18. R. Hartley and F. Schaffalizky, "Powerfactorization: 3D reconstruction with missing or uncertain data," in *Australia-Japan Advanced Workshop on Computer Vision*, September 2003.
19. N. Guilbert and A. Bartoli, "Batch recovery of multiple views with missing data using direct sparse solvers," in *British Machine Vision Conference, Norwich, UK*, September 2003.
20. P. Anandan and M. Irani, "Factorization with uncertainty," *International Journal of Computer Vision*, vol. 49, no. 3, pp. 101–116, October 2002.
21. B. Triggs, "Plane+parallax, tensors and factorization," in *6th European Conference on Computer Vision, Dublin, Ireland*, June 2000, pp. 522 – 538.
22. M. Irani, P. Anandan, and M. Cohen, "Direct recovery of planar-parallax from multiple frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1528–1534, November 2002.
23. C. Rother and S. Carlsson, "Linear multi view reconstruction and camera recovery using a reference plane," *International Journal of Computer Vision*, vol. 49, no. 3, pp. 117–141, October 2002.
24. A. Bartoli and P. Sturm, "Contrained structure and motion from multiple uncalibrated views of a piecewise planar scene," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 45–64, April 2003.
25. F. Dellaert, S. M. Seitz, C. E. Thorpe, and S.Thrun, "Structure from motion without correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC*, June 2000, pp. 557 – 564.
26. A. K. R. Chowdhury and R. Chellappa, "Statistical bias in 3-D reconstruction from a monocular video," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1057 – 1062, August 2005.
27. R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*. Cambridge University Press, 2003.
28. J.-Y. Bouguet, "Camera calibration toolbox for matlab," Website, 2007, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.