The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

On the Error Analysis of Vertical Line Pair-based Monocular Visual Odometry in Urban Area

Ji Zhang and Dezhen Song

Abstract— **When a robot travels in urban area, Global Positional System (GPS) signals might be obstructed by buildings. Hence visual odometry is a choice. We notice that the vertical edges from high buildings and poles of street lights are a very stable set of features that can be easily extracted. Thus, we develop a monocular vision-based odometry system that utilizes the vertical edges from the scene to estimate the robot ego-motion. Since it only takes a single vertical line pair to estimate the robot ego-motion on the road plane, here we model the ego-motion estimation process and analyze how the choice of different vertical line pair impacts the accuracy of the ego-motion estimation process. The resulting closed form error model can assist to choose an appropriate pair of vertical lines to reduce the error in computation. We have implemented the proposed method and validated the error analysis results in physical experiments.**

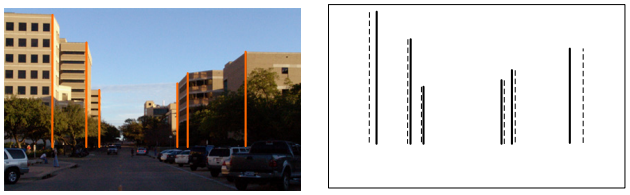
I. Introduction

When a robot travels in urban area, high buildings often block the view of sky and Global Positional System (GPS) signals are sometimes unavailable. Since wheel-encoder based or inertial measurement unit (IMU)-based dead reckoning methods cannot provide sufficient accuracy for navigation needs, visual odometry is a natural choice. Realizing that the vertical edges in the scene, such as the vertical boundaries of buildings and poles of street lights, provide a stable and rich set of features that are easy to extract and recognize, we are interested in developing visual odometry techniques utilizing the vertical line features.

As illustrated in Fig. 1, we start with the simplest configuration where the robot only has one camera. When the robot/camera travels on the streets, images are taken with vertical lines extracted from the scene. We compare the translation of the corresponding vertical lines' movements in adjacent image frames and estimate the robot ego-motion. Since it only takes a single vertical line pair to estimate the robot ego-motion on the road plane, here we model the ego-motion estimation process and analyze how the choice of different vertical line pair impacts the accuracy of the ego-motion estimation process. The resulting closed form error model can assist to choose an appropriate pair of vertical lines in the scene to achieve the best visual odometry accuracy. We have implemented the proposed method and validated the error analysis results in physical experiments.

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

J. Zhang and D. Song are with Computer Science and Engineering Department, Texas A&M University, College Station, TX 77843. Email: jizhang@cse.tamu.edu and dzsong@cse.tamu.edu.



(a) (b)

Fig. 1. An illustration of monocular visual odometry using vertical lines. (a) An image frame taken by the robot with vertical lines highlighted in orange color. (b) Corresponding vertical lines in two consecutive frames after the robot moving forward along the optical axis direction of the camera by 10 meters. The corresponding vertical lines before and after the movement are represented in solid and dash lines, respectively. The robot can utilize the vertical lines' displacements to estimate its ego-motion.

II. Related Work

Using one or more cameras to assist robot navigation has become a popular research direction in the past decade. Depending on the purpose of the system, researchers focus on different aspects such as vision-based navigation [1] in general, simultaneous localization and mapping (SLAM) [2] and visual odometry [3]. In this paper, our work only focuses on visual odometry.

A line is a special type of edge in feature extraction. Using edge features to assist robots in navigation is a popular approach. Edges are robust features that are not sensitive to lighting conditions and shadow. The property makes them suitable to outdoor applications. Edges can be easily extracted from images using edge detectors [4]. In vision-based navigation, edge features are widely used in lane detection [5], road following [6], road edge detection [7], and vehicle motion planning [8], [9]. Inspired by the success of applications in different domains, we zoom in vertical lines, which are a very unique set of edges that exist in urban applications. It becomes possible to precisely and automatically extract these features and hence open the possibility for odometry applications where the accuracy requirement for features are more critical than other applications.

Visual odometry is to estimate vehicle ego-motion using images taken by the onboard camera. It is often used in locations where GPS signals are challenged or not available. Successful applications of visual odometry include unmanned aerial vehicles [10], unmanned underwater vehicles [11], legged robots [12], and ground mobile robots where the wheels often skid [13], [14]. The application of visual odometry on Mars rovers is a famous example [15]–[17]. One challenge of visual odometry is to autonomously identify rigid/static features from the scene. Utilizing the

special structure of the scene, we focus on applications for ground robots navigating in urban area and hence propose to use vertical lines to decrease the difficulty of identifying features.

An early work by Zhou and Li [18] shows that the vertical lines can be automatically extracted even the camera optical axis is not perfectly horizontal. Hence the road plane can be estimated. Building on this, we develop visual odometry techniques using vertical lines and analyze how different vertical line features impact the accuracy of visual odometry.

III. Problem Definition

The robot periodically takes frames to calculate its displacement in each step, which is the ego-motion estimation. To define the problem and focus our effort on the most relevant issues, we have the following assumptions.

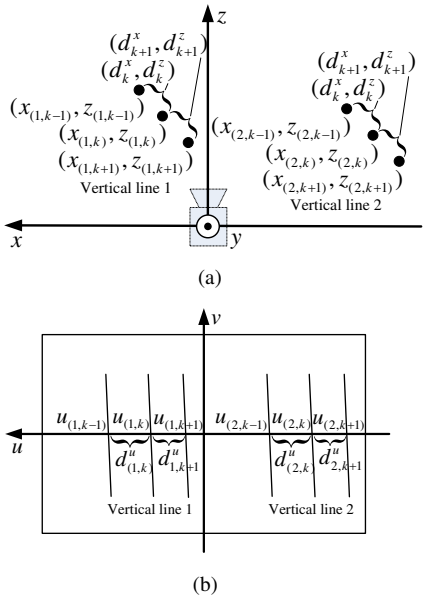


Fig. 2. Superimposed camera coordinate systems (a) and image coordinate systems (b) for vertical lines 1 and 2 over frames $k-1$, k, $k+1$, respectively.

A. Assumptions

1) We assume that the robot displacement of the initial step is known as a reference. This is the requirement for the monocular vision system. Otherwise the ego-motion estimation is only up to similarity.
2) We assume that the robot moves on a flat ground that could be approximated by a plane, which is named as the road plane.
3) We assume that vertical lines, such as poles and buildings' vertical edges, are stationary, hence have the same angle to the road plane. The vertical lines are not necessarily perpendicular to the road plane because the road plane is not necessarily horizontal.
4) We assume that the intrinsic camera parameters and camera orientations are known from pre-calibration or potentiometers. The calibration also removes lens distortion. The camera has square pixels and a zero skew factor. For simplicity of the analysis, we assume that the camera focal length is one unit, and the image planes are always perpendicular to the road plane, and parallel to each other. For other cases, we can always use a homography matrix to rotate them to satisfy the condition. Since the camera orientation is known, the homography matrix can be easily constructed [19].

B. Notations and Coordinate Systems

In this paper, all the coordinate systems are right hand systems. For the camera coordinate system (CCS), we define z-axis as the camera optical axis, and y-axis to point upward toward sky. The corresponding image coordinate systems (ICS) is defined on the image plane parallel to the $x-y$ plane of CCS with its u-axis and v-axis parallel to x-axis and y-axis, respectively. The camera principle axis intersects ICS at its origin on the image plane. To maintain a right hand coordinate system, the x-axis of CCS and its corresponding u-axis in ICS have to point left as illustrated in Fig. 2.

Since the image planes are perpendicular to the road plane, and parallel to each other, the CCSs are iso-oriented during the computation, therefore the displacement of the robot on the road plane in different CCSs is equivalent to the displacement of the vertical lines in a fixed CCS in the opposite direction. Fig. 2(a) illustrates the superimposed CCSs for three consecutive frames $k-1$, k and $k+1$, respectively. At time k, $k \in \mathbb{N}^+$, let $(x_{(i,k-1)}, z_{(i,k-1)})$, $(x_{(i,k)}, z_{(i,k)})$, and $(x_{(i,k+1)}, z_{(i,k+1)})$ be the (x, z) coordinate of the intersection between the corresponding vertical line i, $i = 1, 2$ and $x-z$ plane for frames $k-1$, k, and $k+1$, respectively. Let (d^x_k, d^z_k) be the vertical line pair's displacement from frame $k-1$ to k, we have $d^x_k = x_{(i,k)} - x_{(i,k-1)}$, $d^z_k = z_{(i,k)} - z_{(i,k-1)}$.

Fig. 2(b) shows the corresponding superimposed ICSs for frames $k-1$, k and $k+1$. Let $u_{(i,k-1)}$, $u_{(i,k)}$, and $u_{(i,k+1)}$ be the u-coordinate of the intersections between vertical line i and u-axis for frames $k-1$, k, and $k+1$, respectively. With the above notations and coordinate systems defined, we are ready to describe our task.

C. Problem Description

If the robot knows its displacement of the pervious step: $\mathbf{d}_k = [d^x_k, d^z_k]^T$, it can calculate the displacement of the current step: $\mathbf{d}_{k+1} = [d^x_{k+1}, d^z_{k+1}]^T$ using the vertical line pair's positions in the three images, $\mathbf{u}_i = [u_{(i,k-1)}, u_{(i,k)}, u_{(i,k+1)}]^T$, $i = 1, 2$, as follows,

$$\mathbf{d}_{k+1} = \boldsymbol{F}(\mathbf{d}_k, \mathbf{u}_1, \mathbf{u}_2), \tag{1}$$

where function $\boldsymbol{F}(\cdot)$ will be determined later in the paper.

Eq. (1) provides a recursive format for us to estimate the robot ego-motion represented by $\mathbf{d}_{k+1}$. However, in each step of calculation, errors are brought into the system. We do not know the actual value of $\mathbf{d}_k, \mathbf{u}_1$, and $\mathbf{u}_2$. Instead, we have their measured value $\hat{\mathbf{d}}_k, \hat{\mathbf{u}}_1$, and $\hat{\mathbf{u}}_2$ with corresponding error $\mathbf{e}^{\mathbf{d}}_k = \hat{\mathbf{d}}_k - \mathbf{d}_k, \mathbf{e}^{\mathbf{u}}_1 = \hat{\mathbf{u}}_1 - \mathbf{u}_1$, and $\mathbf{e}^{\mathbf{u}}_2 = \hat{\mathbf{u}}_2 - \mathbf{u}_2$. As a convention in the paper, error value e^a of a variable a is defined as $e^a = \hat{a} - a$. Hence, (1) becomes

$$\mathbf{d}_{k+1} + \mathbf{e}^{\mathbf{d}}_{k+1} = \boldsymbol{F}(\mathbf{d}_k + \mathbf{e}^{\mathbf{d}}_k, \mathbf{u}_1 + \mathbf{e}^{\mathbf{u}}_1, \mathbf{u}_2 + \mathbf{e}^{\mathbf{u}}_2). \tag{2}$$

Since we are interested in how errors propagate in the computation, we want to derive the following relationship from (2),

$$\mathbf{e}^{\mathbf{d}}_{k+1} = \boldsymbol{G}(\mathbf{e}^{\mathbf{d}}_{k}, \mathbf{e}^{\mathbf{u}}_{1}, \mathbf{e}^{\mathbf{u}}_{2}). \tag{3}$$

When errors are small, function $\boldsymbol{G}$ can be approximated by a linear expression

$$\mathbf{e}^{\mathbf{d}}_{k+1} = \left[\mathbf{P}_{2\times 2} \mid \mathbf{Q}_{2\times 3} \mid \mathbf{R}_{2\times 3}\right] \begin{bmatrix} \mathbf{e}^{\mathbf{d}}_{k} \\ \mathbf{e}^{\mathbf{u}}_{1} \\ \mathbf{e}^{\mathbf{u}}_{2} \end{bmatrix}, \tag{4}$$

where $\mathbf{P} = \partial \boldsymbol{F}/\partial \mathbf{e}^{\mathbf{d}}_{k}$, $\mathbf{Q} = \partial \boldsymbol{F}/\partial \mathbf{e}^{\mathbf{u}}_{1}$, and $\mathbf{R} = \partial \boldsymbol{F}/\partial \mathbf{e}^{\mathbf{u}}_{2}$ are Jacobian matrices. Hence our problem can be defined as

Definition 1: Given $\mathbf{d}_k$, $\mathbf{u}_1$, and $\mathbf{u}_2$, derive $\boldsymbol{F}(\cdot)$, and compute $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$. Based on the results, perform sensitivity analysis to study error propagation.

IV. Modeling Error Propagation

To solve the problem, we begin with deriving the expression of $\boldsymbol{F}(\cdot)$ in (1).

A. Deriving F: Incremental Displacement Computing

Since the camera has square pixels, a zero skew factor, and the focal length of one unit, we can use the pin hole camera model to obtain the following relationship between $(x_{(i,k)},\ z_{(i,k)})$ and $u_{(i,k)}$,

$$u_{(i,k)} = \frac{x_{(i,k)}}{z_{(i,k)}}, \quad i = 1, 2. \tag{5}$$

Combining (5) with $x_{(1,k-1)} = x_{(1,k)} - d^x_k$, $x_{(1,k+1)} = x_{(1,k)} + d^x_{k+1}$, $z_{(1,k-1)} = z_{(1,k)} - d^z_k$, $z_{(1,k+1)} = z_{(1,k)} + d^z_{k+1}$, we have

$$u_{(1,k-1)} = \frac{x_{(1,k-1)}}{z_{(1,k-1)}} = \frac{(x_{(1,k)} - d^x_k)}{(z_{(1,k)} - d^z_k)}, \tag{6}$$

$$u_{(1,k)} = \frac{x_{(1,k)}}{z_{(1,k)}}, \tag{7}$$

$$u_{(1,k+1)} = \frac{x_{(1,k+1)}}{z_{(1,k+1)}} = \frac{(x_{(1,k)} + d^x_{k+1})}{(z_{(1,k)} + d^x_{k+1})}. \tag{8}$$

Combining (6-8) to eliminate $x_{(1,k)}$ and $z_{(1,k)}$, we have

$$d^x_{k+1} + a_1 d^z_{k+1} = b_1, \tag{9}$$

where $a_1 = -u_{(1,k+1)}$ and $b_1 = \frac{u_{(1,k+1)} - u_{(1,k)}}{u_{(1,k)} - u_{(1,k-1)}}(d^x_k - u_{(1,k-1)} d^z_k)$.

Similarly, we have the following for vertical line 2,

$$d^x_{k+1} + a_2 d^z_{k+1} = b_2, \tag{10}$$

where $a_2 = -u_{(2,k+1)}$ and $b_2 = \frac{u_{(2,k+1)} - u_{(2,k)}}{u_{(2,k)} - u_{(2,k-1)}}(d^x_k - u_{(2,k-1)} d^z_k)$.

Combine (9) and (10), we have the $\boldsymbol{F}(\cdot)$ function

$$\mathbf{d}_{k+1} = \mathbf{M}^{-1}_{k+1} \mathbf{M}_k \mathbf{d}_k, \tag{11}$$

where

$$\mathbf{M}_k = \begin{bmatrix} u_{(1,k+1)} - u_{(1,k)} & -u_{(1,k-1)}(u_{(1,k+1)} - u_{(1,k)}) \\ u_{(2,k+1)} - u_{(2,k)} & -u_{(2,k-1)}(u_{(2,k+1)} - u_{(2,k)}) \end{bmatrix},$$

$$\mathbf{M}_{k+1} = \begin{bmatrix} u_{(1,k)} - u_{(1,k-1)} & -u_{(1,k+1)}(u_{(1,k)} - u_{(1,k-1)}) \\ u_{(2,k)} - u_{(2,k-1)} & -u_{(2,k+1)}(u_{(2,k)} - u_{(2,k-1)}) \end{bmatrix}.$$

B. Computing Jacobian Matrices

Now we can compute Jacobian matrices to see how errors get propagated. It is possible to take an algebraic approach by directly computing Jacobian. However, it is more intuitive to use a geometric approach, which helps understand the error propagation process.

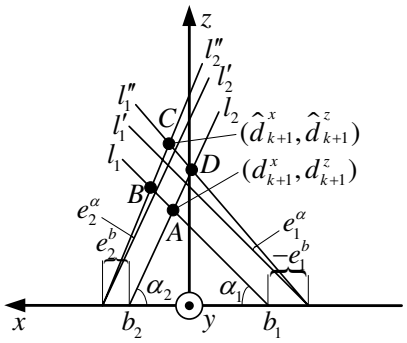


Fig. 3. Computing Jacobian matrices using a geometric approach.

Fig. 3 illustrates the geometric approach. Let l_1 be the line described by (9), which intersects with x-axis at $x = b_1$ with angle α_1. Recalling a_1 defined in (9), we have $\tan\alpha_1 = -1/a_1 = 1/u_{(1,k+1)}$. Similarly, let l_2 be the line described by (10), which intersects with x-axis at $x = b_2$ with angle α_2. Also, we have $\tan\alpha_2 = 1/a_2 = -1/u_{(2,k+1)}$. l_1 and l_2 intersect at point A, which is the robot displacement $\mathbf{d}_{k+1}$.

Let e^α_1, e^b_1 be the parameter errors of α_1, b_1, respectively. Due to the existence of e^b_1, l_1 shifts to l'_1, where l'_1 is a line parallel to l_1. Due to the existence of e^α_1, l'_1 shifts to l''_1, where l''_1 is a line intersects with l'_1 on x. Let e^α_2, e^b_2 be the parameter errors of α_2, b_2, respectively. Similarly, we have lines l'_2 and l''_2. Accordingly, the intersection between l_1, l_2 becomes that of l''_1, l''_2, locating at point C, which is the estimated displacement $\hat{\mathbf{d}}_{k+1}$. The difference between C and A is the robot ego-motion estimation error $\mathbf{e}^{\mathbf{d}}_{k+1}$.

Let B be the intersection between l_1 and l''_2, and D be the intersection between l''_1 and l_2. Since e^α_1 and e^α_2 are both very small, we can approximate $ABCD$ as a parallelogram. Thus, we have

$$e^x_{k+1} = |\overline{AB}|\cos\alpha_1 - |\overline{AD}|\cos\alpha_2, \tag{12}$$

$$e^z_{k+1} = |\overline{AB}|\sin\alpha_1 + |\overline{AD}|\sin\alpha_2. \tag{13}$$

From the geometry relationship, we have

$$|\overline{AD}| = -\frac{e^b_1 \sin(\alpha_1)}{\sin(\alpha_1 + \alpha_2)} + \frac{e^\alpha_1 (b_2 - b_1)\sin(\alpha_2)}{\sin^2(\alpha_1 + \alpha_2)}. \tag{14}$$

Let e^a_1 be the parameter error of a_1 in (9), Since $a_1 = -1/\tan\alpha_1$, we have $e^a_1 = e^\alpha_1/\sin^2\alpha_1$. At the same time, using $u_{(1,k+1)} = 1/\tan\alpha_1$ and $u_{(2,k+1)} = -1/\tan\alpha_2$, (14) becomes

$$|\overline{AD}| = \eta/\sin\alpha_2, \tag{15}$$

where $\eta = \frac{-e_1^b}{u_{(1,k+1)}-u_{(2,k+1)}} + \frac{e_1^a(b_2-b_1)}{(u_{(1,k+1)}-u_{(2,k+1)})^2}$. Similarly, we have

$$|\overline{AB}| = \mu / \sin\alpha_1, \tag{16}$$

where $\mu = \frac{e_2^b}{u_{(1,k+1)}-u_{(2,k+1)}} + \frac{e_2^a(b_2-b_1)}{(u_{(1,k+1)}-u_{(2,k+1)})^2}$. Substituting (15) and (16) into (12) and (13), and using $u_{(1,k+1)} = 1/\tan\alpha_1$ and $u_{(2,k+1)} = -1/\tan\alpha_2$, we have

$$e_{k+1}^x = \mu u_{(1,k+1)} + \eta u_{(2,k+1)}, \tag{17}$$

$$e_{k+1}^z = \mu + \eta. \tag{18}$$

Recalling a_1 in (9), we have

$$e_1^a = -e_{(1,k+1)}^u. \tag{19}$$

From (6), (7), we have $u_{(1,k)} - u_{(1,k-1)} = (d_k^x - u_{(1,k-1)}d_k^z)/z_{(1,k)}$. Let $d_{(i,k)}^u$ be the displacement of vertical line i from frame $k-1$ to k, we have $d_{(i,k)}^u = u_{(i,k)} - u_{(i,k-1)}$. After derivation of b_1 in (9) and substitution of the above equations, we have

$$\begin{aligned} e_1^b =& e_{(1,k+1)}^u z_{(1,k)} - e_{(1,k)}^u z_{(1,k)}(1 + \frac{d_{(1,k+1)}^u}{d_{(1,k)}^u}) \\ &+ e_{(1,k-1)}^u z_{(1,k-1)} \frac{d_{(1,k+1)}^u}{d_{(1,k)}^u} \\ &+ (e_k^x - u_{(1,k-1)}e_k^z)\frac{d_{(1,k+1)}^u}{d_{(1,k)}^u}. \end{aligned} \tag{20}$$

Substituting (19), (20), b_1, and b_2 into the expression of η in (15), and applying the same substitution that we used in (20), we have the expression of η, and similarly, the expression of μ in (16). Then, we substitute η and μ into (17) and (18). Finally, we can obtain the Jacobian matrices as shown in (21-23).

V. Sensitivity Analysis

With Jacobian matrices ready, we can analyze how errors are introduced and propagated over the computation. The first analysis we conduct is to study which dimension of the ego-motion estimation error $\mathbf{d}_{k+1}$ is more suspectable to the error introduced by edge detection error. In this case, matrix $\mathbf{Q}$ is scrutinized. We have the following result.

Theorem 1: Let $Q_{i,j}$ be the (i,j)-th entry of $\mathbf{Q}$. If the camera horizontal field of view (HFOV) is no bigger than $50°$, then $|Q_{1,j}/Q_{2,j}| \leq 0.46$, $j = 1,2,3$.

Proof: From (22), we have

$$Q_{1,j}/Q_{2,j} = u_{(2,k+1)}, \quad j = 1,2,3. \tag{24}$$

Since the HFOV is no bigger than $50°$, we have

$$-\tan 25° \leq \frac{x_{(2,k+1)}}{z_{(2,k+1)}} \leq \tan 25°. \tag{25}$$

Combining (25) with (5), we have

$$-0.46 \leq u_{(2,k+1)} \leq 0.46. \tag{26}$$

Thus

$$|Q_{1,j}/Q_{2,j}| \leq 0.46, \quad j = 1,2,3. \tag{27}$$

■

This theorem indicates that the introduced error in x-direction is smaller than that in z-direction. The result could also be explained by Fig. 3, where point C only moves inside $\angle BAD$. Since the HFOV is no bigger than $50°$, angles α_1 and α_2 are bounded inside set $[65°, 90°]$. Hence the quadrilateral $ABCD$ is long in z-direction and narrow in x-direction. Since a regular camera has HFOV less than $50°$, the conclusion is that the depth error is at least twice more than the lateral error.

Another interesting question is how the ego-motion estimation error $\mathbf{e}_{k+1}^{\mathbf{d}}$ relates to the position of the vertical line pair. In other words, if there are many vertical line pairs available in the scene, we need to find the pair that provides the most accurate ego-motion estimation. Define $\delta_{k+1}^u = u_{(1,k+1)} - u_{(2,k+1)}$ as the distance between the two vertical lines in ICS. Recall that $z_{(1,k+1)}$ is the depth of vertical line 1. We have,

Theorem 2: $\partial|Q_{2,j}|/\partial|\delta_{k+1}^u| \leq 0$, $\partial|Q_{i,j}|/\partial z_{(1,k+1)} \geq 0$, $i = 1,2$, $j = 1,2,3$.

Proof: The first step is to prove $\partial|Q_{2,j}|/\partial|\delta_{k+1}^u| \leq 0$, $j = 1,2,3$. We prove the inequality for the case of $j = 1$ because other cases can be proved similarly. From (22), we have

$$Q_{2,1} = -\frac{d_{(1,k+1)}^u z_{(1,k-1)}}{d_{(1,k)}^u (u_{(1,k+1)} - u_{(2,k+1)})}. \tag{28}$$

Since $\delta_{k+1}^u = u_{(1,k+1)} - u_{(2,k+1)}$, we have

$$\frac{\partial Q_{2,1}}{\partial \delta_{k+1}^u} = \frac{d_{(1,k+1)}^u z_{(1,k-1)}}{d_{(1,k)}^u {\delta_{k+1}^u}^2} = -\frac{Q_{2,1}}{\delta_{k+1}^u}. \tag{29}$$

For the case when $Q_{2,1} \geq 0$ and $\delta_{k+1}^u > 0$, or the case when $Q_{2,1} < 0$ and $\delta_{k+1}^u < 0$, we have

$$\partial|Q_{2,1}|/\partial\left|\delta_{k+1}^u\right| = -Q_{2,1}/\delta_{k+1}^u \leq 0. \tag{30}$$

For the case when $Q_{2,1} \geq 0$ and $\delta_{k+1}^u < 0$, or the case when $Q_{2,1} < 0$ and $\delta_{k+1}^u > 0$, we have

$$\partial|Q_{2,1}|/\partial\left|\delta_{k+1}^u\right| = Q_{2,1}/\delta_{k+1}^u \leq 0. \tag{31}$$

Thus,

$$\partial|Q_{2,1}|/\partial\left|\delta_{k+1}^u\right| \leq 0. \tag{32}$$

The second step is to prove $\partial|Q_{i,j}|/\partial z_{(1,k+1)} \geq 0$, $i = 1,2$, $j = 1,2,3$. We prove the inequality for $i = 1$ and $j = 1$. All other cases with different i and j values can be proved similarly. From (22), we have

$$Q_{1,1} = -\frac{d_{(1,k+1)}^u u_{(2,k+1)} z_{(1,k-1)}}{d_{(1,k)}^u (u_{(1,k+1)} - u_{(2,k+1)})}. \tag{33}$$

Substituting $z_{(1,k-1)} = z_{(1,k+1)} - d_k^z - d_{k+1}^z$ into (33), we have

$$\frac{\partial Q_{1,1}}{\partial z_{(1,k+1)}} = \frac{-d_{(1,k+1)}^u u_{(2,k+1)}}{d_{(1,k)}^u (u_{(1,k+1)} - u_{(2,k+1)})} = \frac{Q_{1,1}}{z_{(1,k-1)}}. \tag{34}$$

Since $z_{(1,k-1)} > 0$, when $Q_{1,1} \geq 0$, we have

$$\partial|Q_{1,1}|/\partial z_{(1,k+1)} = Q_{1,1}/z_{(1,k-1)} \geq 0. \tag{35}$$

$$\mathbf{P} = \frac{-1}{u_{(1,k+1)} - u_{(2,k+1)}} \begin{bmatrix} \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} u_{(2,k+1)} - \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} u_{(1,k+1)} & -\frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} u_{(2,k+1)} u_{(1,k-1)} + \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} u_{(1,k+1)} u_{(2,k-1)} \\ \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} - \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} & -\frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} u_{(1,k-1)} + \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} u_{(2,k-1)} \end{bmatrix}, \tag{21}$$

$$\mathbf{Q} = \frac{-1}{u_{(1,k+1)} - u_{(2,k+1)}} \begin{bmatrix} \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} u_{(2,k+1)} z_{(1,k-1)} & -(1 + \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}}) u_{(2,k+1)} z_{(1,k)} & u_{(2,k+1)} z_{(1,k+1)} \\ \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}} z_{(1,k-1)} & -(1 + \frac{d^u_{(1,k+1)}}{d^u_{(1,k)}}) z_{(1,k)} & z_{(1,k+1)} \end{bmatrix}, \tag{22}$$

$$\mathbf{R} = \frac{1}{u_{(1,k+1)} - u_{(2,k+1)}} \begin{bmatrix} \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} u_{(1,k+1)} z_{(2,k-1)} & -(1 + \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}}) u_{(1,k+1)} z_{(2,k)} & u_{(1,k+1)} z_{(2,k+1)} \\ \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}} z_{(2,k-1)} & -(1 + \frac{d^u_{(2,k+1)}}{d^u_{(2,k)}}) z_{(2,k)} & z_{(2,k+1)} \end{bmatrix}. \tag{23}$$

When $Q_{1,1} < 0$ we have

$$\partial |Q_{1,1}| / \partial z_{(1,k+1)} = -Q_{1,1} / z_{(1,k-1)} > 0. \tag{36}$$

Thus,

$$\partial |Q_{1,1}| / \partial z_{(1,k+1)} \geq 0. \tag{37}$$

■

This theorem indicates that the ego-motion estimation error $\mathbf{d}_{k+1}$ grows as the depth of the vertical line $z_{(1,k+1)}$ increases. Also the depth error, which is the z-direction of $\mathbf{d}_{k+1}$, decreases as $|\delta^u_{k+1}|$ increases. From Theorem 1, we know that the depth error dominates the lateral error. Therefore, choosing the vertical line pair with short depth and a large distance between the two lines can improve the accuracy of the ego-motion estimation.

VI. Experiments

We use a Sony DSC-F828 Camera mounted on a four-wheeled robot in the experiment. The robot has a dimension of $43.7 \times 31.0 \text{cm}^2$. Its wheel diameter is 17.2cm. The camera HFOV is set to 50° with a resolution of 640×480 pixels.

Fig. 4. Experiment setup. We use the vertical edges on the frontal plane of the interdisciplinary life sciences building on Texas A&M University campus. The vertical edges are numbered in pairs. This is the closest view to the building among all the image frames.

As illustrated in Fig. 4, we place the camera/robot in front of a building on Texas A&M University campus. We use eight pairs of vertical lines on the frontal plane of the building. In Fig. 4, two lines with the same number belong to the same pair. The relative distance between the two lines in each pair is defined as δ. During the experiment the camera has to face the frontal plane to obtain edge position readings. Hence, we use z-direction to refer to the direction perpendicular to the frontal plane of the building and x-direction to the direction parallel to the frontal plane of the building.

During each trial, the robot moves along a straight line with 11 incremental steps and the step length of 0.5 meters. The robot takes one image at each of the 12 positions introduced by the 11-step movement. The robot displacement of the first step is given as a reference. For the subsequent 10 steps, we compute the robot ego-motion using (11) and compare it with the actual measurements from tape measurer for each step. Each trial is the average of the outcome of the 10 steps.

To facilitate the comparison we use relative error metrics. Let d_x and d_z be the robot real displacements in x- and z-directions, respectively. d_x and d_z are obtained by using a tape measurer. Let $\hat{d}_x$ and $\hat{d}_z$ be the estimated displacements from the algorithm. The two relative error metrics used to measure the ego-motion estimation error are defined as

$$\varepsilon_x = |\hat{d}_x - d_x| / \sqrt{d_x^2 + d_z^2}, \tag{38}$$

$$\varepsilon_z = |\hat{d}_z - d_z| / \sqrt{d_x^2 + d_z^2}. \tag{39}$$

The combination of four different experimental setups is tested in different trials:

1) Two different robot headings including x- and z-directions,
2) Eight different relative distance settings δ between the vertical lines,
3) Eight different depth of vertical lines z. The initial positions of the robot with respect to the frontal plane is from 8 different depth settings ranging from 35m to 70m with 5m intervals, and
4) Camera rotation vs. no camera rotation. For cases without camera rotation, we adjust camera pan and tilt in the experiment to force CCSs to be iso-oriented. For cases with camera rotation, we introduce CCSs with $\pm 10^\circ$ orientational difference.

Therefore, we have conducted a total of 256 trials in the experiments.

Fig. 5 illustrates experiment results. As shown in Fig. 5(a), regardless of the robot moving directions, the ego-motion estimation error in depth direction ε_z is always over two times bigger than that in lateral direction ε_x, which confirms Theorem 1.

Since the depth error ε_z is the dominating error, we only compare the depth error. Fig. 5(c) illustrates how ε_z changes with respect to different δ settings. It is clear that as δ

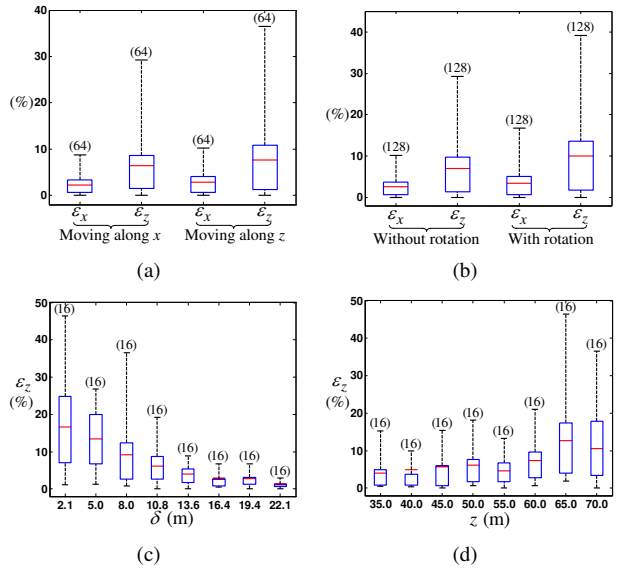


Fig. 5. Statistical experiment results. Note that the red line is the mean value, the blue box represents the population ranging from 25 percentile to 75 percentile, and the black dashed intervals indicate the data range. (a) ε_z vs. ε_x. (b) Camera rotation vs. no camera rotation. (c) ε_z vs. δ. (d) ε_z vs. z. The number in the parenthesis is the number of trials used to compute the statistics.

increases, ε_z decreases. Fig. 5(d) illustrates how ε_z changes as z changes. There is a trend that ε_z decreases as z decreases although the trend is not clear when z is relatively small since factors other than z dominate the error. These results confirm Theorem 2.

Additionally, Fig. 5(b) illustrates how camera rotation might impact ε_x and ε_z. It is clear that there is no significant difference between the two cases for either ε_x or ε_z. This result shows that assuming the CCSs are iso-oriented during the computation is reasonable.

VII. Conclusion and Future Work

We developed a monocular vision-based odometry system that utilizes the vertical edges from the scene to estimate the robot ego-motion. We modeled the ego-motion estimation process and analyzed how the choice of different vertical line pair impacts the accuracy of the ego-motion estimation process. The resulting closed form error model can assist to choose an appropriate pair of vertical lines to reduce the error in computation. Two conclusions have been identified through the analysis: 1) the depth error is always a dominating error for regular cameras with HFOV less than 50°, and 2) a vertical line pair that is closer to the camera and has longer relative distance between the two vertical lines gives more accurate ego-motion estimation. We have implemented the proposed method and validated the error analysis results in physical experiments.

In the future, we will build on the approach by actively selecting a subset of vertical lines from a potential large number of vertical lines to compose multiple vertical line pairs to achieve the optimal estimation accuracy. We will also work on approaches that utilize both vertical lines and horizontal lines that provide possibility for vertical motion estimation.

Acknowledgement

We thank Dr. J. Yi for his insightful inputs and discussions. Thanks Y. Xu, C. Kim, A. Aghamohammadi, and Z. Bing for their contributions to NetBot Laboratory in Texas A&M University.

References

[1] D. Song, H. Lee, and J. Yi, "On the analysis of the depth error on the road plane for monocular vision-based robot navigation," in *The Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR), Dec. 7-9, 2008, Guanajuato, Mxico*, 2008.

[2] A. Davison, L. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[3] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015–1026, Oct. 2008.

[4] D. Ziou and S. Tabbone, "Edge detection techniques - an overview," *International Journal of Pattern Recognition and Image Analysis*, vol. 8, no. 4, pp. 537–559, 1998.

[5] A. Broggi and S. Berte, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *Journal of Artificial Intelligence Research*, vol. 3, pp. 325–348, 1995.

[6] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transaction on Image Processing*, vol. 7, no. 1, pp. 62–81, January 1998.

[7] M. Ekinci, F. W. J. Gibbs, and B. T. Thomas, "Knowledge-based navigation for autonomous road vehicles," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 8, no. 1, pp. 1–29, May 2000.

[8] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 309– 318, December 2004.

[9] 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, Oct. 2007.

[10] O. Amidi, T. Kanade, and J. Miller, "Vision-based autonomous helicopter research at carnegie mellon robotics institute 1991-1997," in *American Helicopter Society International Conference*, Heli, japan, April 1998.

[11] R. Marks, H. Wang, M. Lee, and S. Rock, "Automatic visual station keeping of an underwater robot," in *Oceans Engineering for Today's Technology and Tomorrow's Preservation*, vol. 2, Brest, France, Sep. 1994, pp. 137–142.

[12] R. Ozawa, Y. Takaoka, Y. Kida, and et al, "Using visual odometry to create 3D maps for online footstep planning," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Oct. 2005, pp. 2643–2648.

[13] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep. 2004, pp. 4007–4012.

[14] M. Agrawal and K. Konolige, "Rough terrain visual odometry," in *International Conference on Advanced Robotics*, Aug. 2007.

[15] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 2, pp. 169–186, March 2007.

[16] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, Oct. 2005, pp. 903– 910.

[17] D. Helmick, Y. Cheng, and D. Clouse, "Path following using visual odometry for a mars rover in high-slip environments," in *2004 IEEE Aerospace Conference*, vol. 2, March 2004, pp. 772–789.

[18] J. Zhou and B. Li, "Exploiting vertical lines in vision-based navigation for mobile robot platforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, April 2007, pp. I–465–I–468.

[19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, 2nd Edition*. Cambridge University Press, 2004.