

# NAVIGATIONAL DRIFT ANALYSIS FOR VISUAL ODOMETRY

Hongtao LIU, Ruyi JIANG, Weng HU, Shigang WANG\*

*School of Mechanical Engineering  
Shanghai Jiao Tong University  
Dongchuan Road 800  
Shanghai, China*

*e-mail: liuhongtaosdu@163.com, jiangrui83@gmail.com,  
hwhust@gmail.com, wangshigang@sjtu.edu.cn*

**Abstract.** Visual odometry estimates a robot’s ego-motion with cameras installed on itself. With the advantages brought by camera being a sensor, visual odometry has been widely adopted in robotics and navigation fields. Drift (or error accumulation) from relative motion concatenation is an intrinsic problem of visual odometry in long-range navigation, as visual odometry is a sensor based on relative measurements. General error analysis using “mean” and “covariance” of positional error in each axis is not fully capable to describe the behavior of drift. Moreover, no theoretic drift analysis is available for performance evaluation and algorithms comparison. Drift distribution is established in the paper, as a function of the covariance matrix from positional error propagation model. To validate the drift model, experiment with a specific setting is conducted.

**Keywords:** Visual odometry, error analysis, motion concatenation, ego-motion, navigational drift

**Mathematics Subject Classification 2010:** 62L12

## 1 INTRODUCTION

Visual odometry uses camera(s) to incrementally calculate a robot’s motion between time steps, and finally localize the robot. In general, it can determine ego-motion in all six degrees of freedom in the 3D world. Compared with other sensors used

---

\* Corresponding author

in robot navigation (e.g., odometry, GPS and IMU), visual odometry has its advantages, and has already been widely tested or applied in many fields. For driver assistance or autonomous driving, ego-motion of a vehicle can be obtained by analyzing video inputs from the camera(s) installed in the vehicle (e.g., [1]). Visual odometry is also a popular choice in Simultaneous Localization and Mapping (SLAM) to obtain motion trajectory of a robot [2]. By looking downward, visual odometry assists an unmanned helicopter to calculate its own moving trajectory when GPS is unavailable [3]. It is even applied in underwater situations to help a robot to localize itself [4].

As a sensor based on relative measurement, visual odometry estimates motion by the following two steps: relative motion estimation between time steps, and motion concatenation. For relative motion estimation, various algorithms have been tested to implement visual odometry, using monocular [5, 6] or stereo [3, 7, 8], perspective or omnidirectional [5, 6, 9] cameras. Among these, the most popular framework for visual odometry is based on feature matching and tracking (e.g., [10, 11, 12]), as shown in Figure 1. While considering that feature-based methods are sensitive to systematic errors in intrinsic and extrinsic camera parameters, appearance-based visual odometry uses appearance of world to extract motion information (e.g., [5]). Recently, a specific quadrifocal method using dense pixel matching from temporal stereo was also tested for visual odometry with very accurate results reported [1]. Furthermore, many other sensors can also be integrated with cameras to provide more accurate results for long-range navigation.



Figure 1. A general visual odometry framework based on feature matching and tracking

## 1.1 Motivation

The idea of this paper is motivated by the following considerations:

1. Currently, attention has been widely paid to suppress navigational drift in visual odometry, but methods to “see” drift output from algorithms are still limited to experiments by running the algorithms. There are many visual odometry algorithms with no discussion of analytical error propagation process, which prevents the development of visual odometry from performance evaluation.
2. As a new and promising sensor, visual odometry needs a methodology for systematic and comparative analysis of its drift, in order to quantify the performance of various algorithms.
3. When some allowable drift or “error” is given for a specific trajectory, how to design or select a proper visual odometry algorithm? What is the maximum measurement error allowed for every step? It is an inverse problem of general error propagation.

## 1.2 Related Works

The paper is mainly on error accumulation during motion concatenation step in visual odometry. Drift or error accumulation has long been recognized in visual odometry, and various methods have been suggested in order to suppress its effect, such as frame selection techniques (e.g., “Fire-walls” [11] or similarly keyframe [6, 13]), sparse bundle adjustment [8], more accurate spatial uncertainty models [7], loop-closing [9], and most promisingly integrating with other navigation sensors (e.g., inertial sensor [14], low-cost GPS [15], absolute orientation sensor [16]). All of these techniques observe performance improvements in long-range navigation.

Drift analysis and evaluation are mainly based on experimental results in the literature, by running algorithms on some testing roads and comparing the estimated trajectory with the ground truth, which is usually obtained from GPS, while, for presenting the results, charts with the estimated and real trajectory are usually provided to give a direct comparison. However, methods for detailed analysis of drift are limited in the literature.

End-point absolute drift distance or drift ratio (ratio of the final drift value to the traveled distance), is a common choice (e.g., [1, 4, 5, 6, 7, 11, 15, 16, 17]) to measure the drift when running a visual odometry algorithm over some distance, from tens or hundreds of meters to several kilometers. Their drawbacks are closely related to the properties of drift. First, drift will not increase linearly with the distance traveled, which was stated by Clark [16] and will be further proved in this paper. Thus, when running algorithms on some distance, drift value and ratio will change with different distances traveled. Moreover, running the same algorithms on the same trajectory repeatedly will produce quite different drift value or ratio. The reason is that drift is a random process, and it will not always increase, but sometimes also decrease at some places, as errors in different motion vectors will compensate to some extent. Thus, using end-point values (the final drift value, and the final traveled distance) is inappropriate to model the whole random process.

Andrea [12] adopted a method different from the above, which is to measure the drift distance induced within a predefined traverse (e.g., 100 m). For a whole trajectory, drift statistics can be obtained from all the traverse segments. Later, Navid [17] improved the method to measure drift for several distance intervals (e.g., 10 m, 20 m, 50 m, and 100 m) to show the tendencies of the drift increment as distance increased. Both of these methods analyze drift in a more detailed scale than offset ratio, but still the dependence of drift on the traveled trajectory is not considered.

As a main contribution of the paper, we will study theoretical analysis of drift for navigation with visual odometry. Based on this, a new drift evaluation method using drift distribution will be introduced.

## 1.3 Organization

The paper is organized as follows. Section 2 discusses a well-adopted linear error propagation model. The output of this error model is the second-moment (“co-

variance”) of error distribution in each axis. Based on this, Section 3 presents a general drift model for visual odometry, which is the main contribution of the paper. Section 4 validates the drift model using a specific experiment setting. Finally, conclusions are provided in Section 5.

## 2 ERROR DYNAMICS

Spatial error propagation as widely used in robotics and navigation fields can be similarly applied in visual odometry, though it uses cameras as sensor for calculating relative motion between time steps. This section will briefly discuss such a general linearized error propagation method as used in [18, 19].

### 2.1 Notations

In this paper, capital letter (e.g.,  $M$ ) signifies matrices, lowercase (e.g.,  $x$ ) means a scalar, and bolded lowercase (e.g.,  $\mathbf{p}$ ) indicates a vector. Values for different time steps are down-right indexed. Transposition of a matrix or vector is upper-right indicated by  $T$ . For random variables,  $E(\cdot)$  is the expectation. For a matrix,  $\det(\cdot)$  is the determinant. Note that camera pose and relative motion are both represented by  $4 \times 4$  homogeneous matrices.

The origin of the vehicle coordinate system locates on the center of the camera(s) installed (left camera for stereo situation). The  $z$ -axis points to the front of a vehicle, the  $x$ -axis to the left, and the  $y$ -axis upwards (right hand system). Accordingly, the rotational angles along  $x$ -,  $y$ -, and  $z$ - axes are respectively named as *pitch*, *yaw*, and *roll*. The initial camera position before moving is regarded as the global coordinate system.

### 2.2 Linearized Error Propagation

Differential homogeneous transforms are used to represent the spatial error of camera poses estimated. The error is assumed to be small enough that a first-order model will provide a good approximation. The assumption is reasonable, as end-point drift ratio is generally around 1% ~ 5% in the literature. The estimated pose of camera  $G$  is considered to be a nominal pose  $\hat{G}$  with a small perturbation  $\delta G$ :

$$G = \hat{G}(I + \delta G) \quad (1)$$

where  $\delta G$  is the differential matrix representing the uncertainty associated with the camera pose and is written as:

$$\delta G = \begin{bmatrix} 0 & -\delta\phi & \delta\theta & \delta x \\ \delta\phi & 0 & -\delta\psi & \delta y \\ -\delta\theta & \delta\psi & 0 & \delta z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

where  $\delta x, \delta y, \delta z$  are the positional errors and  $\delta\psi, \delta\theta$  and  $\delta\phi$  are the orientational errors along the  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively. Note that the spatial uncertainties are associated with camera at each time step, the error variables must be expressed with respect to the nominal pose coordinate frames established at each time step. With the representation in Equation (1), the probabilistic property of the uncertainty representation is modeled by the error vector,  $(\delta x, \delta y, \delta z, \delta\psi, \delta\theta, \delta\phi)^T$  in  $\delta G$ . Also, motion estimated  $M$  for every time step is modeled in the same way.

Error propagation of visual odometry is then to find the total error of the estimated camera pose in a compound spatial relation, corresponding to motion concatenation in every time step. At time  $t$ , we have the estimated camera pose  $G_{t-1}$  and relative motion  $M_{t-1,t}$  with their corresponding uncertainties using Equation (1) as, respectively

$$G_{t-1} = \hat{G}_{t-1}(I + \delta G_{t-1}) \quad \text{and} \quad M_{t-1,t} = \hat{M}_{t-1,t}(I + \delta M_{t-1,t}). \tag{3}$$

The spatial propagation is then to calculate the camera pose uncertainty at time  $t$  by motion concatenation as<sup>1</sup>

$$G_t = G_{t-1} \cdot M_{t-1,t} = \hat{G}_{t-1} \hat{M}_{t-1,t} (I + \dots + \hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t} + \delta M_{t-1,t} + \hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t} \delta M_{t-1,t}). \tag{4}$$

With the first-order approximation, it will be

$$G_t = \hat{G}_{t-1} \hat{M}_{t-1,t} (I + \hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t} + \delta M_{t-1,t}). \tag{5}$$

The differential matrix of  $G_t$  becomes:

$$\delta G_t = \hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t} + \delta M_{t-1,t} \tag{6}$$

where  $\hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t}$  is the transformation of camera pose's differential matrix  $\delta G_{t-1}$  to the reference frame of camera pose  $G_t$ . As the error vectors of  $\delta G_{t-1}$  and  $\delta M_{t-1,t}$  are independent, the covariance of the camera pose error at time  $t$  can be easily computed to be

$$\Sigma_{G_t} = J \Sigma_{G_{t-1}} J^T + \Sigma_{M_{t-1,t}} \tag{7}$$

where  $\Sigma_{G_{t-1}}$  and  $\Sigma_{M_{t-1,t}}$  are the covariance matrices of the error vectors of  $\delta G_{t-1}$  and  $\delta M_{t-1,t}$  respectively.  $J$  is the  $6 \times 6$  Jacobian matrix of  $\hat{M}_{t-1,t}$ , and can be expressed as:

$$J = \begin{bmatrix} R^T & R^T D^T \\ 0 & R^T \end{bmatrix} \tag{8}$$

---

<sup>1</sup> Right multiplication of the motion is because the estimated motion is based on the camera's coordinate system, while not the global one.

where  $R$  is the rotational sub matrix of  $M_{t-1,t}$ , and

$$D = \begin{bmatrix} 0 & -z_n & y_n \\ z_n & 0 & -x_n \\ -y_n & x_n & 0 \end{bmatrix} \quad (9)$$

and  $\mathbf{t} = (x_n, y_n, z_n)^T$  is the translational vector in  $M_{t-1,t}$ .

### 3 DRIFT ANALYSIS IN VISUAL ODOMETRY

Though slightly different in the form, error propagation as described in Section 2 has been widely adopted in robotics and navigation fields. Covariance of the state vector is the usual output of the error analysis. With such covariance matrix, error distribution of every state element becomes evident. However, specifically for visual odometry, error analysis can be improved. For this purpose, drift is analyzed in this section.

#### 3.1 Drift Distribution Model

Drift  $s$  is the 3D Euclidean distance between the nominal destination traveled by the camera and that estimated by the visual odometry algorithm. It is a function of camera pose's error vector, defined as

$$s = \sqrt{(\delta x)^2 + (\delta y)^2 + (\delta z)^2}. \quad (10)$$

Note that only three translational error elements are included in the drift definition, while the rotational ones are taken as intrinsic. The explanations are as follows. Firstly, in visual odometry for ground vehicle, though error in a vehicle's yaw angle may increase boundlessly, its pitch and roll angle will generally be small and physically bounded by the ground plane, and will not accumulate infinitely<sup>2</sup>. Secondly, the final positional drift is caused partly by accumulated angular error, and partly by accumulated translational error. Clark [16] proved that rotational error will contribute more than translational one to the final positional drift. Overall, the positional drift we analyzed in the paper includes a large part of contribution from the rotational drift. Finally, it is also a common practice to analyze the final drift using positional distance.

For theoretic analysis, one assumption is made that the variables  $\delta x, \delta y$  and  $\delta z$  follow Gaussian distribution with a mean of zero and covariance as described by Equation (7). The validation of the assumption is experimentally proved by Williams [2], and still based on the following reasons. First, many of the noise input disturbances (e.g., feature localization uncertainty, calibration uncertainty)

---

<sup>2</sup> Visual odometry used in helicopters, it will be a different story as no ground plane is available in air.

encountered in visual odometry are approximately Gaussian; and Gaussian inputs to linear systems will still be Gaussian. Equation (5) can be regarded as linearization, as error vectors from camera pose and motion are linearized for the new camera pose; and the deviation from the Gaussian condition in Equation (5) is caused only by the inexactness of the approximation in linearizing. Secondly, Central-Limit Theorem proves that the total input effect on the system tends towards a Gaussian condition as the number of independent inputs becomes larger (see [20], page 369). For large scale visual odometry, the number of inputs is usually very large (over hundreds or thousands). Finally, convolution of two Gaussian distributions will still be Gaussian distributions [21]. As we have approximated the original motion concatenation as a linear subsystem convolution, so for the Gaussian distribution inputs of the linear system, the convolution output will still be Gaussian.

Applying the Gaussian assumption, the joint probability density function for the error vector elements  $(\delta x, \delta y, \delta z)$  may be written as

$$p(\delta x, \delta y, \delta z) = \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{\det(\bar{P})}} \exp \left\{ -\frac{1}{2} [\delta x, \delta y, \delta z]^T \bar{P}^{-1} [\delta x, \delta y, \delta z] \right\} \quad (11)$$

where

$$\bar{P} = \begin{pmatrix} \sigma_{\delta x}^2 & r_{\delta x \delta y} & r_{\delta x \delta z} \\ r_{\delta x \delta y} & \sigma_{\delta y}^2 & r_{\delta y \delta z} \\ r_{\delta x \delta z} & r_{\delta y \delta z} & \sigma_{\delta z}^2 \end{pmatrix} \quad (12)$$

with

$$\begin{aligned} \sigma_{\delta x}^2 &\triangleq E(\delta x^2) & \sigma_{\delta y}^2 &\triangleq E(\delta y^2) & \sigma_{\delta z}^2 &\triangleq E(\delta z^2) \\ r_{\delta x \delta y} &\triangleq E(\delta x \delta y) & r_{\delta y \delta z} &\triangleq E(\delta y \delta z) & r_{\delta x \delta z} &\triangleq E(\delta x \delta z). \end{aligned}$$

$\bar{P}$  is a sub-matrix from the error covariance matrix  $\Sigma$  in Equation (7), as not all the elements of the error vector are needed for drift.

Drift  $s$  as a random variable expressed by Equation (10) is practical for Monte Carlo simulations, but too complicated for statistical analysis, as  $\delta x, \delta y$  and  $\delta z$  are mutually correlated. For statistical analysis, a linear transformation as presented in Appendix A is used to transform three correlated norm variables  $(\delta x, \delta y, \delta z)$  into three uncorrelated  $(\bar{\delta} x, \bar{\delta} y, \bar{\delta} z)$ . Then

$$s = \sqrt{(\bar{\delta} x)^2 + (\bar{\delta} y)^2 + (\bar{\delta} z)^2}. \quad (13)$$

Now with the independent norm variables  $\bar{\delta} x, \bar{\delta} y$  and  $\bar{\delta} z$ , and their variances  $\bar{\sigma}_x^2, \bar{\sigma}_y^2$  and  $\bar{\sigma}_z^2$ , their probability density function is as follows (take  $\bar{\delta} x$  as an example):

$$p_x(\bar{\delta} x) = \frac{1}{\sqrt{2\pi} \bar{\sigma}_x} \exp \left\{ -\frac{\bar{\delta} x^2}{2\bar{\sigma}_x^2} \right\}. \quad (14)$$

As  $\bar{\delta} x, \bar{\delta} y$  and  $\bar{\delta} z$  are uncorrelated, then their joint probability density function is

$$p_{xyz}(\bar{\delta} x, \bar{\delta} y, \bar{\delta} z) = p_x(\bar{\delta} x) \cdot p_y(\bar{\delta} y) \cdot p_z(\bar{\delta} z). \quad (15)$$

Finally the probability function for  $s$  is

$$\begin{aligned}
 P(s) &= P\left(\sqrt{\bar{\delta}x^2 + \bar{\delta}y^2 + \bar{\delta}z^2} \leq s\right) \\
 &= \iiint_I p_{xyz}(\bar{\delta}x, \bar{\delta}y, \bar{\delta}z) d_{\bar{\delta}x} d_{\bar{\delta}y} d_{\bar{\delta}z}
 \end{aligned}
 \tag{16}$$

where  $I$  is the integration region  $\sqrt{\bar{\delta}x^2 + \bar{\delta}y^2 + \bar{\delta}z^2} \leq s$ . Using spherical coordinate system, we will have

$$\begin{aligned}
 P(s) &= \int_0^s \int_0^\pi \int_0^{2\pi} \frac{r^2 \sin \varphi}{(2\pi)^{\frac{3}{2}} \bar{\sigma}_x \bar{\sigma}_y \bar{\sigma}_z} \exp\left\{-\frac{r^2}{2} \left(\frac{\sin^2 \varphi \cos^2 \theta}{\bar{\sigma}_x^2} \right. \right. \\
 &\quad \left. \left. + \frac{\sin^2 \varphi \sin^2 \theta}{\bar{\sigma}_y^2} + \frac{\cos^2 \varphi}{\bar{\sigma}_z^2}\right)\right\} dr d\varphi d\theta
 \end{aligned}
 \tag{17}$$

and, the probability density function for drift  $s$  will be

$$\begin{aligned}
 p(s) &= \int_0^\pi \int_0^{2\pi} \frac{s^2 \sin \varphi}{(2\pi)^{\frac{3}{2}} \bar{\sigma}_x \bar{\sigma}_y \bar{\sigma}_z} \exp\left\{-\frac{s^2}{2} \left(\frac{\sin^2 \varphi \cos^2 \theta}{\bar{\sigma}_x^2} \right. \right. \\
 &\quad \left. \left. + \frac{\sin^2 \varphi \sin^2 \theta}{\bar{\sigma}_y^2} + \frac{\cos^2 \varphi}{\bar{\sigma}_z^2}\right)\right\} d\varphi d\theta.
 \end{aligned}
 \tag{18}$$

The drift distribution in Equation (18) is calculated following drift definition, and covariance matrix from the general error propagation as discussed in Section 2.2. For theoretical analysis, the main step is to transform the obtained correlated error vector elements to be uncorrelated following Appendix A. Thus, a key point of the drift distribution model is that drift distribution is closely related to the specific trajectory, and the position in the trajectory.

It should be pointed out that the definite integral in Equation (18) can be solved analytically only when

$$\bar{\sigma}_x^2 = \bar{\sigma}_y^2 = \bar{\sigma}_z^2.
 \tag{19}$$

Otherwise, the integration may be performed numerically.

If

$$\bar{\sigma}_x^2 = \bar{\sigma}_y^2 = \bar{\sigma}_z^2 = \bar{\sigma}^2
 \tag{20}$$

drift  $s$  will follow Maxwell-Boltzmann Distribution, with density function as

$$p(s) = \sqrt{\frac{2}{\pi}} \frac{1}{\bar{\sigma}^3} s^2 \exp\left(-\frac{s^2}{2\bar{\sigma}^2}\right).
 \tag{21}$$

With the drift distribution density function such as Equation (21), some more interesting quantities can be derived. The *most probable drift*,  $s_p$ , is the drift most

likely to happen and corresponds to the maximum value of  $p(s)$ . To find it, we calculate  $dp/ds$ , set it to zero and solve for  $s$  as

$$\frac{dp(s)}{ds} = 0 \tag{22}$$

which yields

$$s_p = \sqrt{2}\sigma. \tag{23}$$

The *mean drift*,  $E(s)$ , is the mathematical average of the drift distribution

$$E(s) = \int_0^\infty sp(s) ds = \frac{2}{\sqrt{\pi}}s_p. \tag{24}$$

The *root mean square drift*,  $s_{rms}$ , is the square root of the average square drift:

$$s_{rms} = \left( \int_0^\infty s^2p(s) ds \right)^{1/2} = \sqrt{\frac{3}{2}}s_p. \tag{25}$$

The typical drift is related as follows

$$0.886E(s) = s_p < E(s) < s_{rms} = 1.085E(s). \tag{26}$$

### 3.2 Drift Propagation

Recall the differential camera pose matrix in Equation (6) as

$$\delta G_t = \hat{M}_{t-1,t}^{-1} \delta G_{t-1} \hat{M}_{t-1,t} + \delta M_{t-1,t}.$$

From this equation, we can derive the drift propagation equation as follows. For notational simplicity, time indexes in variables are removed; then camera poses at time  $t - 1$  and  $t$  are represented by  $G$  and  $G'$  respectively, with relative motion matrix between them as  $M$ . So the motion concatenation is  $G' = GM$ .

Write the motion matrix into general rotation and translation form; we can have

$$M = \begin{bmatrix} R_m & \mathbf{t}_m \\ \mathbf{0} & 1 \end{bmatrix} \tag{27}$$

and its differential matrix will be

$$\delta M = \begin{bmatrix} \delta R_m & \delta \mathbf{t}_m \\ \mathbf{0} & 0 \end{bmatrix}. \tag{28}$$

Camera pose  $G$  and its differential matrix will have the similar form.

Then from Equation (6), we will have

$$\begin{bmatrix} \delta R' & \delta \mathbf{t}' \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} R_m^T & -R_m^T \mathbf{t}_m \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta R & \delta \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix} \cdot \begin{bmatrix} R_m & \mathbf{t}_m \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} \delta R_m & \delta \mathbf{t}_m \\ \mathbf{0} & 0 \end{bmatrix}. \tag{29}$$

After simple matrix and vector multiplication, we will have the following equation from Equation (29):

$$\delta \mathbf{t}' = R_m^T \delta R \mathbf{t}_m + R_m^T \delta \mathbf{t} + \delta \mathbf{t}_m \quad (30)$$

where  $\delta \mathbf{t}' = [\delta x', \delta y', \delta z']^T$  is the translational error at time  $t$ .

From drift definition in Equation (10), it is obvious that

$$s'^2 = \delta \mathbf{t}'^T \delta \mathbf{t}'. \quad (31)$$

Substituting Equation (30) into Equation (31), note the relation that

$$(\mathbf{t}_m)^T \delta R^T \delta \mathbf{t} + \delta \mathbf{t}^T \delta R \mathbf{t}_m = 0 \quad (32)$$

as  $\delta R^T = -\delta R$ . And similarly,

$$(\mathbf{t}_m)^T \delta R^T R_m \delta \mathbf{t}_m + (\delta \mathbf{t}_m)^T R_m^T \delta R \mathbf{t}_m = 0. \quad (33)$$

Finally, we have

$$s'^2 = s^2 + \delta \mathbf{t}_m^T \delta \mathbf{t}_m + (\delta R \mathbf{t}_m)^T \delta R \mathbf{t}_m + 2\delta \mathbf{t}_m^T R_m^T \delta \mathbf{t}. \quad (34)$$

It is evident from Equation (34) that the drift after motion concatenation  $s'$  is composed of four parts: the previous drift  $s$ , the translational error in relative motion measurement  $\delta \mathbf{t}_m$ , error from the previous rotational drift by affecting the translational motion  $\delta R \mathbf{t}_m$ , and rotated translational error  $\delta \mathbf{t}_m^T R_m^T \delta \mathbf{t}$ .

Another point to emphasize is that drift will not always increase. From Equation (34), it is clear that the first three items in the right part are all contribute to the increase of drift, while the sign of the last item is undetermined. An example of drift decrease can be observed in Figures 5, 6, 7 in the experiment section.

### 3.3 Using Drift Distribution as Error Quantification for Visual Odometry

The most popular way for experimental performance evaluation of visual odometry is the drift value or drift ratio after running the algorithm on a testing trajectory. The drawbacks of such method are as follows. Firstly, as error propagation is a random process, drift value from running the algorithm once cannot tell the real performance. Secondly, as the drift is dependent on the trajectory traveled, different places on the trajectory will have different drift values, possibly larger or smaller. Finally, it is impossible to compare performance of various visual odometry algorithms based on the drift value.

Drift distribution such as Equation (21) is suggested to evaluate performance of various visual odometry. Compared with covariance, the advantages brought by drift distribution are as follows. Firstly, directly observed from Equation (10), drift provides an overall description of error for visual odometry. With this description

as an index, different visual odometry algorithms can be compared. As error distribution may be different for each state elements, comparing error propagation for visual odometry directly by covariance matrix is not feasible. Secondly, using drift distribution as the output of visual odometry experiment will present the overall error propagation of a specific visual odometry algorithm on the testing road by running only once. Finally, drift distribution facilitates the inverse problem as discussed in Section 1.1. In conclusion, experiment will prove that drift distribution from running the algorithm on a testing trajectory will not only tell the performance of the algorithm, but can also be used for comparison with other algorithms.

### 4 EXPERIMENT

Experiments are conducted to illustrate the validation of the established drift model. Moreover, some important facts of the drift in visual odometry are also revealed from the experimental results. We report about experiments which use a specific setting. Note that though there are several visual odometry frameworks as reviewed in Section 1, we adopt the widely used one as in Figure 1. The analysis method is of not different for other frameworks.

#### 4.1 Experiment Setting

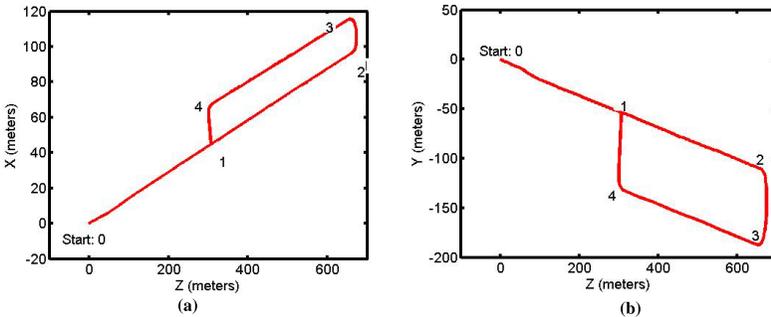


Figure 2. The camera’s real trajectory (0 → 1 → 2 → 3 → 4 → 1) for simulation in a) XZ plane and b) XY plane. Note that the trajectory (1 → 2 → 3 → 4 → 1) forms a close-looping

We combine real dataset with simulation technology to conduct the experiment. A real trajectory<sup>3</sup> from Malaga dataset [22] is adopted, with centimeter-accuracy ground truth provided by DGPS. Moreover, we manually add another segment of the trajectory to form a local close-looping in order to test systematic error propagation. In Figure 2, the trajectory (0 → 1 → 2 → 3 → 4) is from DGPS, and the trajectory

<sup>3</sup> The trajectory has been transformed to left camera’s trajectory.

(4  $\rightarrow$  1) is manually connected local close-looping. Totally, the trajectory is sampled 2574 times, with a driving distance about 1.2 km.

Following the framework as in Figure 1, we simulate feature registration, instead of using real stereo frames of the dataset. The details of simulation will be explained in Section 4.1.1. Recall that the first essential step of visual odometry is relative motion estimation, which is modeled as an *absolute orientation problem* in [8, 23], and a well-known SVD (Singular Value Decomposition)-based solution [23] is adopted here.

All using SVD-solver for relative motion estimation, three different drift suppression schemes during motion concatenation are implemented for comparison. The first one, named DIRECT, directly concatenates the estimated relative motion in every time step without any suppression. As error will accumulate well fast, DIRECT will generally not be adopted in real visual odometry algorithm. However, it is used here as a baseline for drift performance comparison. The second one, named KEYFRAME, uses key-frame selection for drift minimization; relative motion vectors will only concatenate between key frames, thus drift will be suppressed as the number of concatenation for a trajectory is decreased. KEYFRAME is simple, easy to implement, and widely adopted in literatures (e.g., [11]). The third one, named SBA, adopts sparse bundle adjustment for minimizing errors both in features registered, as well as the motion between frames. For accurate visual odometry, SBA is a common choice (e.g., [8]).

#### 4.1.1 Simulated Feature Registration

Paras	image width ( $w$ )	image height ( $h$ )	focal length ( $f$ )
Values	1024	768	796
Paras	base-line ( $b$ )	principal center ( $\bar{u}_0, \bar{v}_0$ )	feature number ( $n$ )
Values	0.8	(512, 384)	200

Table 1. Parameters used in simulation. Note “base-line” is in meters, “feature number” has no unit, and others are in pixels

Feature registration includes spatial feature detection, and sequential feature tracking. We use a similar method as in [10] to simulate features in stereo according to the camera’s real relative motion from the ground truth. The parameters and their values used are listed in Table 1. For camera’s real relative rotational and translational motion  $\bar{R}$  and  $\bar{\mathbf{t}}$  from time  $t$  to  $t + 1$ , we simulate features as in Algorithm 1. Note that systematic error and random error can be simulated separately or simultaneously.

The benefits of combining real trajectory with simulated feature registration are mainly threefold. As the main reason, error sources (systematic and random) can be controlled to take effect with simulation. Also, though outliers in feature registration commonly exist in real situation and contribute to the estimation error, they are unpredictable and currently cannot be modeled. While using simulation,

there will be no outliers in feature registration at all. Thus, robust regression, as a necessary step with real feature registration, can be avoided with the simulated data. Finally, realistic motion conditions are considered, as real trajectory contains various motion types (e.g., constant speed, turning, acceleration, vibration, and so on).

**Step 1:** Randomly generate  $n$  points in the left frame at  $t$

$$\{{}^t\bar{p}_i(u, v) | 0 \leq u \leq w, 0 \leq v \leq h, i = 1, \dots, n\}$$

**Step 2:** Randomly generate float disparity value  $d$  within some range

$$\{d_i | d \in [10 \dots 80], i = 1, \dots, n\}$$

**Step 3:** With  ${}^t\bar{p}$  and  $d$ , and using triangulation, we obtain 3D points;

$$\{{}^t\bar{\mathbf{p}}_i(x, y, z), i = 1, \dots, n\}$$

as well as points in the right frame at  $t$ ;

$$\{{}^t\bar{q}_i(u, v), i = 1, \dots, n\}$$

**Step 4:** With camera motion  $\bar{R}$  and  $\bar{\mathbf{t}}$  from  $t$  to  $t + 1$ , 3D points move;

$${}^{t+1}\bar{\mathbf{p}}_i = \bar{R}^T \cdot {}^t\bar{\mathbf{p}}_i - \bar{R}^T \cdot \bar{\mathbf{t}}$$

**Step 5:** Project the 3D points into stereo frames at  $t + 1$ , we will have;

$$\{{}^{t+1}\bar{p}_i(u, v) | i = 1, \dots, n\} \text{ and } \{{}^{t+1}\bar{q}_i(u, v) | i = 1, \dots, n\}$$

**Step 6a for systematic noise;**  
 Add zero-mean gaussian noise to the parameters  $f, b, u_0$  and  $v_0$ ;

**Step 6b for random noise;**  
 Add zero-mean gaussian noise with variance  $\sigma^2$  to  $u$  of image points in stereo at  $t$  and  $t + 1$ ;

$$\begin{matrix} {}^t p_i \sim \mathcal{N}({}^t\bar{p}_i, \sigma^2) & {}^t q_i \sim \mathcal{N}({}^t\bar{q}_i, \sigma^2) \\ {}^{t+1} p_i \sim \mathcal{N}({}^{t+1}\bar{p}_i, \sigma^2) & {}^{t+1} q_i \sim \mathcal{N}({}^{t+1}\bar{q}_i, \sigma^2) \end{matrix}$$

**Algorithm 1:** Simulated stereo feature matching and tracking

### 4.1.2 Experiment Results

Many error sources will affect the accuracy of the adopted SVD-based algorithm, such as calibration errors in camera's intrinsic parameters, feature localization uncertainty and the number of features.

**General error propagation.** Errors in camera’s intrinsic parameters ( $f, b, u_0, v_0$ ) will cause systematic error in visual odometry. We simulate errors in each intrinsic parameters as Step 6a of Algorithm 1 to see the effect in visual odometry on the close-looping. The estimated trajectory as well as the positional error from systematic errors are shown in Figure 3. It is obvious from the results that errors in different parameters will cause distinct systematic error in visual odometry, not only scale error (e.g., errors caused by  $b$ ) or constant bias but also some unrecognizable errors (e.g., errors by  $(u_0, v_0)$ ). Moreover, the relationship between errors in parameters and systematic error in visual odometry is also dependent on the algorithm adopted.

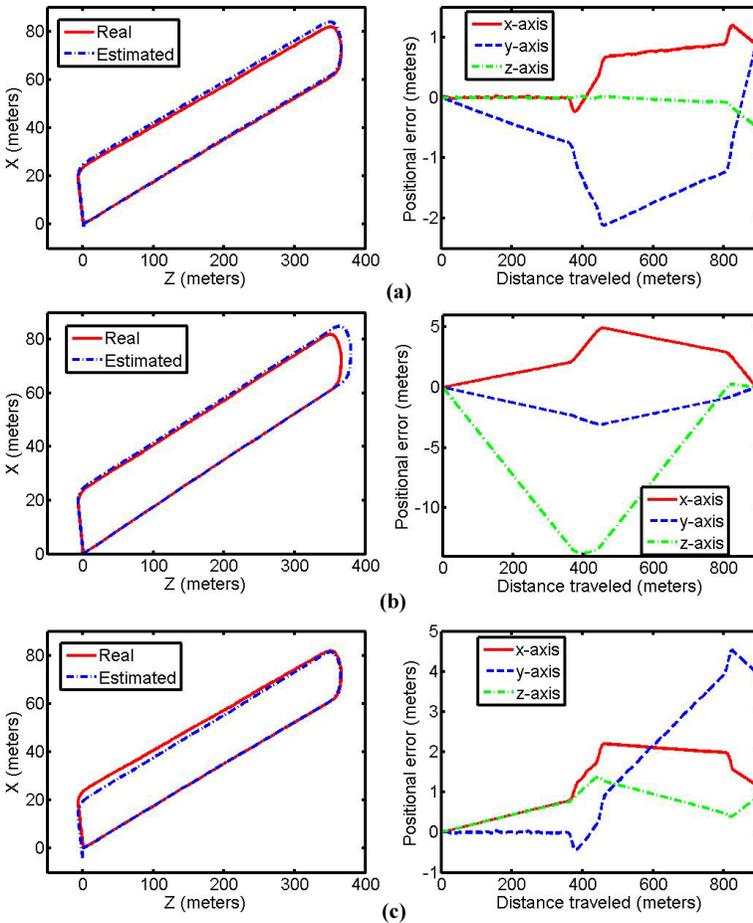


Figure 3. The estimated trajectory (left), as well as the positional error (right) for DIRECT visual odometry caused by the simulated systematic error in camera’s intrinsic parameters a)  $f$ , b)  $b$ , and c)  $u_0, v_0$ . Note that the closed trajectory ( $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ ) as in Figure 2 is used

We simulate errors in feature localization only as Step 6b of Algorithm 1 to see the effect of random noise in visual odometry. With feature localization uncertainty modeled as zero-mean Gaussian noise with standard deviation value 0.1, the trajectory estimated, as well as the accumulated error in position are shown in Figure 4. Note that the result shown in Figure 4 is only one running of the algorithm on the testing trajectory, which may be quite different from another running, as error accumulation is a random process.

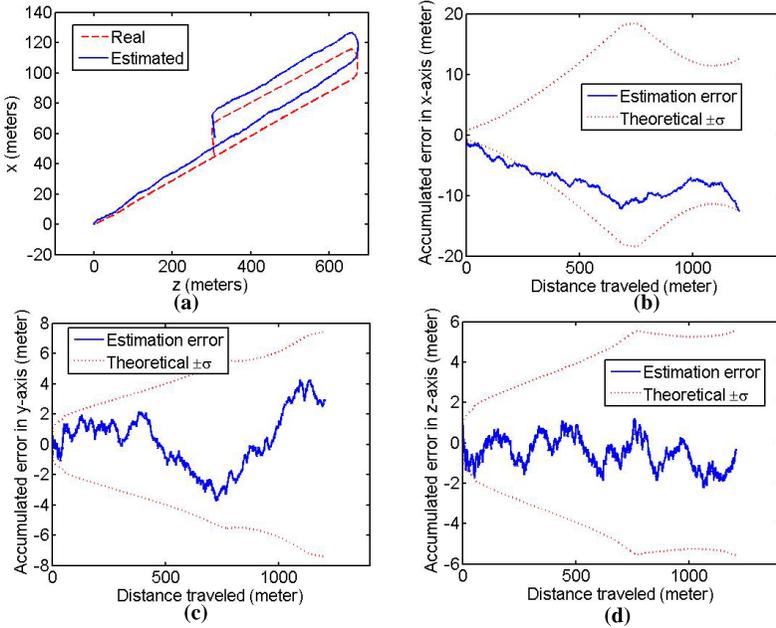


Figure 4. Visual odometry results and positional errors using DIRECT algorithm with simulated random noise in feature localization. a) The real and estimated trajectories. b), c), d) Positional error with standard deviation ( $\pm\sigma$ ) in  $x$ -,  $y$ -,  $z$ -axis, respectively

**Linearized error dynamics.** Now we adopt the model as developed in the paper to analyze the error propagation in visual odometry. Before that, we need to get the covariance matrix  $Q$  in every estimation step from Monte Carlo simulation. With assumption that the noise in each relative motion element is uncorrelated, we can obtain  $Q$  from the statistics of all the relative motion measurement.

The experiment on drift distribution estimation using DIRECT, KEYFRAME and SBA are not meant to compare the drift values between them, as they are evident. Drift distribution from both Monte Carlo simulation and the theoretical model developed in the paper are compared to prove the point that the latter one

is of validity. Also, another point to validate is that the drift distribution model is applicable to various visual odometry algorithms, as long as motion concatenation step is used.

In Figure 4, the standard deviation for each positional errors from matrix  $P$  as in Equation (7) are presented. These results are the popular outputs from error analysis. Note that as variances of the positional error in each axis are not the same, drift distribution will be calculated as Equation (18) while not Equation (21).

The similar linearized error propagation models have been used in robotics and navigation, and their correctness has been widely tested. For validating the drift analysis results, Monte Carlo simulation is used again. The drift distribution will be specific for an algorithm, though each running of the algorithm will present different drift values. For the three algorithms (DIRECT, KEYFRAME and SBA), we run the SVD-algorithm with the same feature localization uncertainty as random noise for 1000 times, as well as the respective drift suppression method. The results are presented in Figures 5, 6, 7. From these results, we can have a direct comparison of performance for visual odometry algorithms adopting the above drift suppression methods in navigating the trajectory. Firstly, the magnitude of mean drift value in

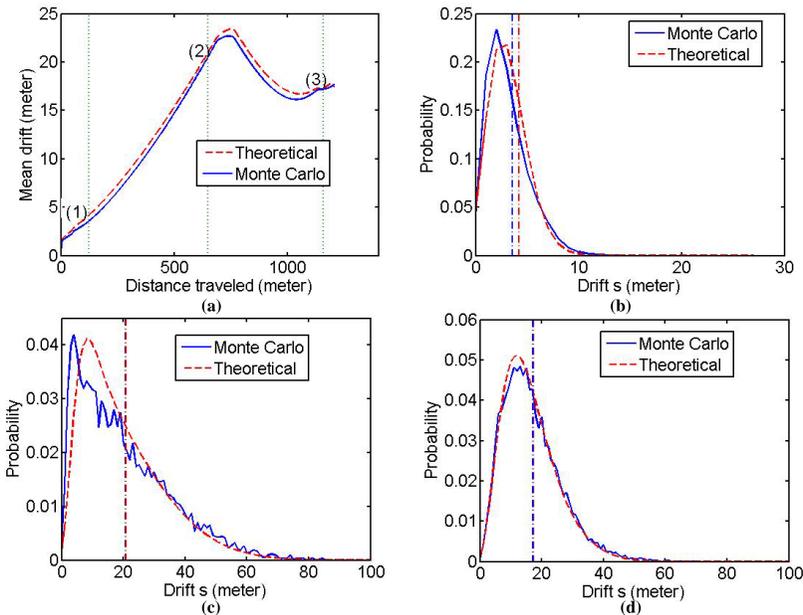


Figure 5. Drift analysis for DIRECT algorithm. Theoretical drift analysis results are compared with that of Monte Carlo simulation. a) The mean drift with the distance traveled. Three randomly selected places (1), (2), (3) are marked where specific drift distributions are presented in b), c), d), respectively. Note that the vertical dot-dashed lines in b), c), d) are the mean drift values of the distribution

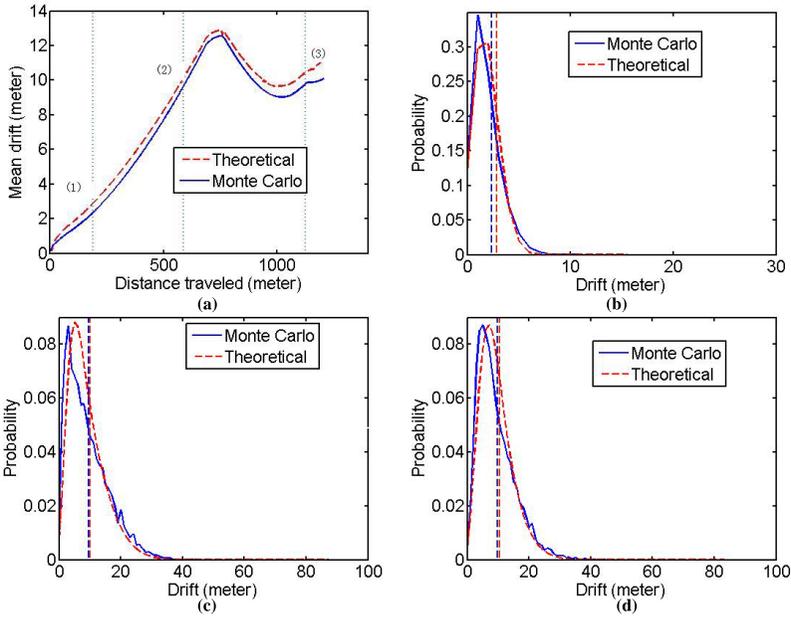


Figure 6. Drift analysis for KEYFRAME algorithm. Theoretical drift analysis results are compared with those of Monte Carlo simulation. a) The mean drift with the distance traveled. Three randomly selected places (1), (2), (3) are marked where specific drift distributions are presented in b), c), d), respectively. Note that the vertical dot-dashed lines in b), c), d) are the mean drift values of the distribution

these figures shows as expected that SBA has the smallest drift value, while DIRECT the largest. Moreover, different distance in the trajectory may have different drift distribution, as presented in b), c), d) of the these figures. Finally, for a specific drift suppression method, drift property is unique, with the measurement covariance in each step being the same. Thus, performance evaluation of visual odometry using various drift suppression methods is capable using drift propagation model as developed.

### 5 CONCLUSION

This paper proposes navigational drift analysis in visual odometry. Instead of modeling error in each axis, positional drift is modeled to provide an overall description of the navigational error. Moreover, with error distribution in each axis, we obtain theoretical drift propagation as a trajectory-dependent distance function. With such drift propagation model, visual odometry algorithms can be compared and evaluated in navigational accuracy.

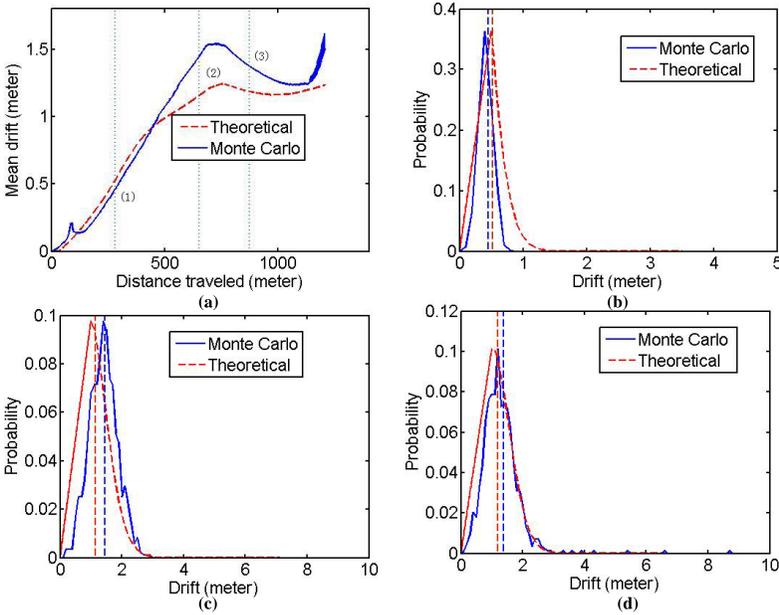


Figure 7. Drift analysis for SBA algorithm. Theoretical drift analysis results are compared with those of Monte Carlo simulation. a) The mean drift with the distance traveled. Three randomly selected places (1), (2), (3) are marked where specific drift distributions are presented in b), c), d), respectively. Note that the vertical dot-dashed lines in b), c), d) are the mean drift values of the distribution

**Acknowledgments**

This work is supported by the National Natural Science Foundation of China under Grant 50875169.

**A APPENDIX: TRANSFORM OF THREE CORRELATED NORM RANDOM VARIABLES INTO UNCORRELATED**

Assume zero-mean normal variables  $\delta_x, \delta_y$  and  $\delta_z$ , with covariance matrix  $M$  known as

$$M = \begin{pmatrix} E(\delta_x\delta_x) & E(\delta_x\delta_y) & E(\delta_x\delta_z) \\ E(\delta_y\delta_x) & E(\delta_y\delta_y) & E(\delta_y\delta_z) \\ E(\delta_z\delta_x) & E(\delta_z\delta_y) & E(\delta_z\delta_z) \end{pmatrix}. \tag{35}$$

Note that  $E(\delta_x\delta_y), E(\delta_x\delta_z)$  and  $E(\delta_y\delta_z)$  are not zero, so  $\delta_x, \delta_y$  and  $\delta_z$  are correlated.

Consider a linear transformation  $L_{3 \times 3}$  defined by

$$\begin{pmatrix} \bar{\delta}_x \\ \bar{\delta}_y \\ \bar{\delta}_z \end{pmatrix} = L_{3 \times 3} \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_z \end{pmatrix}. \tag{36}$$

Then we have

$$\begin{pmatrix} \bar{\delta}_x \\ \bar{\delta}_y \\ \bar{\delta}_z \end{pmatrix} \cdot (\bar{\delta}_x \bar{\delta}_y \bar{\delta}_z) = L \cdot \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_z \end{pmatrix} \cdot (\delta_x \delta_y \delta_z) \cdot L^T. \tag{37}$$

In order to be uncorrelated among  $\bar{\delta}_x$ ,  $\bar{\delta}_y$  and  $\bar{\delta}_z$ , so

$$\begin{aligned} E(\bar{\delta}_x \bar{\delta}_y) &= 0 \\ E(\bar{\delta}_y \bar{\delta}_z) &= 0 \\ E(\bar{\delta}_z \bar{\delta}_x) &= 0. \end{aligned} \tag{38}$$

Taking expectation of Equation (37), it will be

$$\bar{M} = L \cdot M \cdot L^T \tag{39}$$

where

$$\bar{M} = \begin{pmatrix} \bar{\sigma}_x^2 & 0 & 0 \\ 0 & \bar{\sigma}_y^2 & 0 \\ 0 & 0 & \bar{\sigma}_z^2 \end{pmatrix} \tag{40}$$

with

$$\bar{\sigma}_x^2 = E(\bar{\delta}_x^2) \quad \bar{\sigma}_y^2 = E(\bar{\delta}_y^2) \quad \bar{\sigma}_z^2 = E(\bar{\delta}_z^2). \tag{41}$$

As  $M$  is a real symmetric matrix, and  $\bar{M}$  is a diagonal matrix, the transformation  $L$  can be easily obtained from Equation (39). The variances of the normal variables  $\bar{\delta}_x$ ,  $\bar{\delta}_y$  and  $\bar{\delta}_z$  are  $\bar{\sigma}_x^2$ ,  $\bar{\sigma}_y^2$  and  $\bar{\sigma}_z^2$ , which are the eigenvalues of  $M$ . Assuming  $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$  are the corresponding eigenvectors of  $M$ , then  $L = (\mathbf{l}_1 \ \mathbf{l}_2 \ \mathbf{l}_3)$  is the required linear transformation from three correlated normal variables to three uncorrelated.

**REFERENCES**

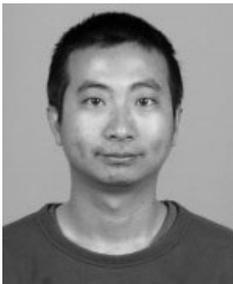
[1] COMPORT, A. I.—MALIS, E.—RIVES P.: Real-Time Quadrifocal Visual Odometry. International Journal of Robotics Research, Vol. 29, 2010, pp. 245–266.  
 [2] WILLIAMS, B.—REID, I.: On Combining Visual Slam and Visual Odometry. Proc. IEEE Conf. on Robotics and Automation 2010, pp. 3494–3500.  
 [3] KELLY, J.—SUKHATME, G.S.: An Experimental Study of Aerial Stereo Visual Odometry. 6<sup>th</sup> IFAC Symposium on Intelligent Autonomous Vehicles 2007.

- [4] CORKE, P.—DETWILER, C.—DUNBABIN, M. et al.: Experiments with Underwater Robot Localization and Tracking. *IEEE Int. Conf. Robotics Automation 2007*, pp. 4556–4561.
- [5] SCARAMUZZA, D.—SIEGWART, R.: Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE Trans. Robotics*, Vol. 24, 2008, pp. 1015–1026.
- [6] TARDIF, J. P.—PAVLIDIS, Y.—DANILIDIS, K.: Monocular Visual Odometry in Urban Environments Using an Omnidirectional Camera. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Philadelphia, PA, USA 2008, pp. 2531–2538.
- [7] DUBBELMAN, G.—GROEN, F. C. A.: Bias Reduction for Stereo Motion Estimation With Applications to Large Scale Visual Odometry. *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pp. 2222–2229.
- [8] SÜNDEHAUF, N.—PROTZEL, P.: Towards Using Sparse Bundle Adjustment for Robust Stereo Odometry in Outdoor Terrain. *Towards Autonomous Robotic Systems 2006*, pp. 206–213.
- [9] SCARAMUZZA, D.—FRAUNDORFER, F.—POLLEFEYS, M.: Closing the Loop in Appearance-Guided Omnidirectional Visual Odometry by Using Vocabulary Trees. *Journal of Robotics and Autonomous Systems*, Vol. 58, 2010, No. 6, pp. 820–827.
- [10] BANDINO, H.: Binocular Ego-Motion Estimation for Automotive Applications. Ph.D. thesis, Frankfurt/Main University 2008.
- [11] NISTÉR, D.—NARODITSKY, O.—BERGEN, J.: Visual Odometry. *IEEE Conf. Computer Vision Pattern Recognition*, Vol. 1, 2004, pp. 652–659.
- [12] JOHNSON, A. E.—GOLDBERG, S. B.—MATTHIES, L. H.: Robust and Efficient Stereo Feature Tracking for Visual Odometry. *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA 2008, pp. 39–46.
- [13] ROYER, E.—LHULLIER, M.—DHOME, M.—LAVEST, J. M.: Monocular Vision for Mobile Robot Localization and Autonomous Navigation. *International Journal of Computer Vision*, Vol. 74, 2007, pp. 237–260.
- [14] READY, B. B.—TAYLOR, C. N.: Inertially Aided Visual Odometry for Miniature air Vehicles in GPS-Denies Environments. *Journal of Intelligent Robotic Systems*, Vol. 55, 2009, pp. 203–221.
- [15] AGRAWAL, M.—KONOLIGE, K.: Rough Terrain Visual Odometry. *Int. Conf. Automatic Robotics 2007*.
- [16] OLSON, C. F.—MATTHIES, L. H.—SCHOPPERS, M.—MAIMONE, M. W.: Stereo Ego-Motion Improvements for Robust Rover Navigation. *IEEE Int. Conf. Robotics Automation 2001*, pp. 1099–1104.
- [17] NOURANI-VATANI, N.—ROBERTS, J.—SRINIVASAN, M. V.: Practical Visual Odometry for Car-Like Vehicles. In *Proceedings of IEEE International Conference on Robotics and Automation*, Kobe, Japan 2009, pp. 3551–3557.
- [18] SMITH, R. C.—CHEESEMAN, P.: On the Representation and Estimation of Spatial Uncertainty. *International Journal of Robotics Research*, Vol. 5, 1986, No. 4, pp. 56–68.

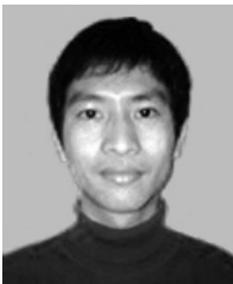
- [19] SU, S. F.—LEE, C. S. G.: Manipulation and Propagation of Uncertainty and Verification of Applicability of Actions in Assembly Tasks. *IEEE Transactions on System, Man, and Cybernetics*, Vol. 22, 1992, No. 6, pp. 1376–1389.
- [20] HENDRICKS, E.—JANNERUP, O. E.: *Linear Systems Control: Deterministic and Stochastic Methods*. Springer 2008.
- [21] ROWEIS, S.—GHAHRAMANI, Z.: A Unifying Review of Linear Gaussian Models. *Neural Computation*, Vol. 11, 1999, No. 2, pp. 305–345.
- [22] BLANCO, J. L.—MORENO, F. A.—GONZALEZ, J.: A Collection of Outdoor Robotic Datasets with Centimeter-Accuracy Ground Truth. *Automatic Robotics*, Vol. 27, 2009, pp. 327–351.
- [23] UMEYAMA, S.: Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, No. 4, pp. 376–380.



**Hongtao LIU** received the Bachelor and Master of Mechanical Engineering from Shandong University, China in 2003 and 2006, respectively. Currently, he is a Ph. D. candidate in Shanghai Jiao Tong University, China. His research interests include machine vision based measurement and pattern recognition.



**Ruyi JIANG** received the Bachelor of Mechanical Engineering from Zhengzhou University, China in 2005, and Ph. D. degree in 2012 from Shanghai Jiao Tong University, China. His research interests include vision-based road modeling, visual SLAM and robotics.



**Wen HU** received the Bachelor and Master in Mechanical Engineering from Huazhong University of Science and Technology, China in 2004 and 2007, respectively. Currently, he is a Ph. D. candidate in Shanghai Jiao Tong University, China. His research interests include machine vision based measurement and machine learning.



**Shigang WANG** received his Ph. D. degree from HuaZhong University of Science and Technology (HUST), Wuhan, China in 1993, as a postdoctoral fellow in the Department of Mechanical Engineering at HUST from 1993 to 1995. Working in the School of Mechanical Engineering, SJTU since 1995, he is now a Chief Professor and the Director of the Institute of Mechatronics and Automation Technology. He is also the Director of Research Center for Microelectronic Equipment at SJTU. His research interests include machine vision and pattern recognition, robotics, design and control of complex mech-electrical systems etc.