Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Gaussian particle filter based pose and motion estimation*

WU Xue-dong†[1,2], SONG Zhi-huan[1]

(*[1]State Key Lab. of Industrial Control Technology, Institute of Industrial Process Control, Zhejiang University, Hangzhou 310027, China*)

(*[2]Department of Electronic Information and Electrical Engineering, Fujian University of Technology, Fuzhou 350014, China*)

†E-mail: woolcn@163.com

Received Jan. 14, 2007; revision accepted June 22, 2007

Abstract: Determination of relative three-dimensional (3D) position, orientation, and relative motion between two reference frames is an important problem in robotic guidance, manipulation, and assembly as well as in other fields such as photogrammetry. A solution to pose and motion estimation problem that uses two-dimensional (2D) intensity images from a single camera is desirable for real-time applications. The difficulty in performing this measurement is that the process of projecting 3D object features to 2D images is a nonlinear transformation. In this paper, the 3D transformation is modeled as a nonlinear stochastic system with the state estimation providing six degrees-of-freedom motion and position values, using line features in image plane as measuring inputs and dual quaternion to represent both rotation and translation in a unified notation. A filtering method called the Gaussian particle filter (GPF) based on the particle filtering concept is presented for 3D pose and motion estimation of a moving target from monocular image sequences. The method has been implemented with simulated data, and simulation results are provided along with comparisons to the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) to show the relative advantages of the GPF. Simulation results showed that GPF is a superior alternative to EKF and UKF.

Key words: Gaussian particle filter (GPF), Pose and motion estimation, Line features, Monocular vision, Extended Kalman filter (EKF), Unscented Kalman filter (UKF), Dual quaternion

doi:10.1631/jzus.2007.A1604 **Document code:** A **CLC number:** TP391

INTRODUCTION

The pose and motion estimation or determination problem has been a subject of considerable research for many years in computer vision, photogrammetry, and robotics with many solutions having been proposed (Shakernia *et al.*, 1999; Blais and Beraldin, 2001; Takahashi and Ghosh, 2001; Porta and Kröse, 2003; Sarcinetti-Filho *et al.*, 2003; Kim *et al.*, 2005; Miro *et al.*, 2005; Tomono, 2005; Shademan and Janabi-Sharifi, 2005). The methods available are generally deterministic and use single-vision cameras (Kim *et al.*, 2005), stereo-vision cameras (Tomono, 2005), or more direct 3D measuring techniques, such as range images from laser (Diosi *et al.*, 2005), ultrasonic (Hazas and Ward, 2002; Dijk *et al.*, 2003; Lin *et al.*, 2003), infrared sensors (Aytac and Barshan, 2002) and structured lighting devices (Pages *et al.*, 2005).

Recently, much work has been done for determining the 3D pose of a moving target from a monocular sequence of images. It is advantageous because a standard video camera is low in cost, and setup and calibration are simple, physical space requirements are small, reliability is high, and low-cost hardware is available for digitizing and processing the images (Kim *et al.*, 2005). Most measurement techniques for image-based pose and motion estimation can be classified into two major categories. These categories are point-based (Ansar and Daniilidis, 2003; Deng *et al.*, 2005; Zhang *et al.*, 2006) and model-based methods (Youngrock *et al.*, 2003) using higher-order geometric primitives. Each type involves acquiring an image, and then processing that image to arrive at a value for the pose and motion.

* Project (No. 2006J0017) supported by the Natural Science Foundation of Fujian Province, China

However, the main body of research of image-based pose and motion estimation has been devoted to point features based algorithms. An alternative would be a line features based algorithm such as those provided by (Chang and Tsai, 1999; Andreff *et al.*, 2002; Rehbinder and Ghosh, 2003; Aggarwal and Karl, 2006). Line features are present in many scenes and objects to a greater extent than point features, moreover, they may be more visible, as well, under a wider range of lighting and environmental conditions than points, and they are more robust than point features (Rehbinder and Ghosh, 2003).

Quaternion is limited in the sense that only rotation is represented in a full 3D transformation, and translation must be dealt with separately. To describe the relative translation and rotation between two coordinates, the usual way is by means of homogeneous transformation matrix, however, the dual number concept may be applied to quaternion as well as vectors and matrices. Chen (1991) introduced the screw theory in the hand-eye calibration, which is the first simultaneous consideration of rotation and translation in a geometric way; later, Daniilidis (1999) used dual quaternion to represent both 3D rotation and 3D translation in a unified notation.

For most nonlinear models, closed-form analytic expression for the posterior distributions do not exist in general. The extended Kalman filter (EKF) is a typical nonlinear estimation method and has been successfully implemented in some problems (Burl, 1993), but in others, it diverges or provides poor approximations. This is especially emphasized when the model is highly nonlinear. Efforts to improve the EKF have led to the new filters recently, like the unscented Kalman filter (UKF) by Julier and Uhlmann (2004), which uses deterministic sets of points in the space of state variable to obtain more accurate approximations to the mean and covariance than the EKF. However, divergence can still occur in some nonlinear problems for UKF. In order to overcome these flaws, Kotecha and Djuric (2003) proposed a new nonlinear estimation method called the Gaussian particle filter (GPF), which is used for tracking filtering and predictive distributions encountered in dynamic state-space models. The models are characterized with additive Gaussian noises, but the functions that appear in the process and observation equations are nonlinear functions. The underlying assumption is that the predictive and filtering distributions can be approximated as Gaussians. Unlike the EKF, which also assumes that these distributions are Gaussians and employs linearization of the functions in the process and observation equations, the GPF updates the Gaussian approximations by using particles that are propagated through the process and observation equations without approximations.

Therefore, in this paper, we explore the potential benefits of the GPF over the more traditional EKF and UKF for pose and motion estimation (Wu *et al.*, 2006). The remainder of this paper is organized as follows. Section 2 gives line transformations with unit dual quaternion. The model of pose and motion estimation is described in Section 3. Section 4 introduces the EKF, UKF and GPF formulations used in our study, and Section 5 describes our experimental methodology and presents the experimental results. Section 6 concludes the paper.

LINE TRANSFORMATIONS WITH UNIT DUAL QUATERNION

We will show how a line transformation can be written with the dual-quaternion product in this section.

A line in space with direction $\boldsymbol{l}$ through a point $\boldsymbol{P}$ can be represented with the 6-tuple ($\boldsymbol{l}$, $\boldsymbol{m}$), where $\boldsymbol{m}$ is the unit normal vector and $\boldsymbol{m}=\boldsymbol{P}\times\boldsymbol{l}$. The constraints $\boldsymbol{l}\cdot\boldsymbol{m}=0$ and $|\boldsymbol{l}|=1$ guarantee that the degree of freedom of an arbitrary line in space is 4.

Applying a rotation $\boldsymbol{R}$ and a translation $\boldsymbol{t}$ to a given line ($\boldsymbol{l}_b$, $\boldsymbol{m}_b$), we can obtain the transformed line ($\boldsymbol{l}_a$, $\boldsymbol{m}_a$):

$$\begin{cases} \boldsymbol{l}_a = \boldsymbol{R}\boldsymbol{l}_b, \\ \boldsymbol{m}_a = \boldsymbol{P}_a \times \boldsymbol{l}_a = (\boldsymbol{R}\boldsymbol{P}_b + \boldsymbol{t}) \times \boldsymbol{R}\boldsymbol{l}_b \\ \quad\;\; = \boldsymbol{R}(\boldsymbol{P}_b \times \boldsymbol{l}_b) + \boldsymbol{t} \times \boldsymbol{R}\boldsymbol{l}_b = \boldsymbol{R}\boldsymbol{m}_b + \boldsymbol{t} \times \boldsymbol{R}\boldsymbol{l}_b. \end{cases} \tag{1}$$

We change from vector to quaternion notation, which means that $\boldsymbol{l}$ is represented by a quaternion with zero scalar part $l=(0, \boldsymbol{l})$. The terms containing rotation can be easily written with quaternion. The cross-product is tackled with the identity (in this paper, q^* is the conjugate of quaternion q) (Daniilidis, 1999):

$$(0, \boldsymbol{t} \times \boldsymbol{q}) = (qt^* + tq)/2, \tag{2}$$

where t is the translation quaternion (0, $\boldsymbol{t}$) and q is the rotation quaternion (0, $\boldsymbol{q}$). Using the identity Eq.(2), we can obtain

$$\begin{cases} l_a = q l_b q^*, \\ m_a = q m_b q^* + (q l_b q^* t^* + t q l_b q^*)/2. \end{cases} \tag{3}$$

We define a new quaternion $\bar{q} = tq/2$ and a dual quaternion $\hat{q} = q + \in \bar{q}$. It can be easily shown that Eq.(3) is equivalent to

$$l_a + \in m_a = (q + \in \bar{q})(l_b + \in m_b)(q^* + \in \bar{q}^*). \tag{4}$$

Denoting also the lines by dual quaternion $\hat{l}_a$ and $\hat{l}_b$, we obtain

$$\hat{l}_a = \hat{q}\hat{l}_b\hat{q}^*. \tag{5}$$

This formula resembles the rotation of points with quaternion. Lines can thus be transformed using a single operation in a non-abelian ring of dual quaternion. The norm is

$$\begin{aligned} |\hat{q}|^2 &= \hat{q}\hat{q}^* = qq^* + \in (q\bar{q}^* + \bar{q}q^*) \\ &= qq^* + \in (qq^* t^* + tqq^*)/2 = 1, \end{aligned} \tag{6}$$

hence $\hat{q}$ is a unit quaternion. From the definition of quaternion $\bar{q}$ and dual quaternion $\hat{q}$, we can obtain

$$\hat{q} = q + \in \frac{t}{2} q = \left(1 + \in \frac{t}{2}\right) q. \tag{7}$$

We know from Eq.(7) that the unit dual quaternion $\hat{q}$ can be written as the concatenation of a pure translation unit dual quaternion and a pure rotational quaternion with dual part being equal to zero.

SYSTEM MODEL OF POSE AND MOTION ESTIMATION

Consider the following general nonlinear model of a dynamic system whose states are to be estimated:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{v}_k), \quad \boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{w}_k), \tag{8}$$

where $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are Gaussian distribution independent of current and past states.

State transition equation

The state assignment estimates the transformation between the camera and the object reference frames and the first derivatives of this transformation. The assignment is based on the dual quaternion representation of the 3D transformation. Similar to the approach given by Broida *et al.*(1990), the state variable assignment with a known object geometry is

$$\begin{aligned} \boldsymbol{x}_k = [&t_x \quad t_y \quad t_z \quad q_0 \quad q_1 \quad q_2 \quad q_3 \\ &v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^{\mathrm{T}}, \end{aligned} \tag{9}$$

where thirteen state variables are presented, t_i and v_i (i=x,y,z) are the linear translation and linear velocity, respectively; q_j (j=0,1,2,3) is the rotational quaternion, and ω_i is the rotational velocity in each axis. Translation, rather than the dual part of the dual quaternion, is estimated in the state vector since the dual part can be readily calculated from the translation and rotational real quaternion as given by Eq.(7). The first derivative is

$$\dot{\hat{q}} = \dot{q} + \in \left(\frac{\dot{t}}{2} q + \frac{t}{2} \dot{q}\right). \tag{10}$$

Chou (1992) gave the relation between quaternion angular velocity and the spatial angular velocity

$$\boldsymbol{\Omega} = [0 \quad \omega_x \quad \omega_y \quad \omega_z]^{\mathrm{T}} = 2\dot{q}q^*. \tag{11}$$

$\boldsymbol{\Omega}$ is a vector quaternion where the vector portion is the angular velocity about the axis. Solving for $\dot{q}$

$$\dot{q} = \boldsymbol{\Omega} q/2. \tag{12}$$

Since the quaternion has four parameters to represent rotation, additional degree of freedom is present. As a result, normalization of the quaternion to unit magnitude is performed after each iteration.

The state transition function $f(\boldsymbol{x}_k, \boldsymbol{v}_k)$ extrapolates from the state at time interval k to the next state at time interval k+1. The linear and angular velocities are assumed constant so that $\omega_i(k+1)=\omega_i(k)$ and $v_i(k+1)=v_i(k)$. The quaternion propagation is described by Eq.(12), and thus the solution is when all

ω_i are constant, after simplification

$$q(t_{k+1})=\left[I\cos(|\omega|\tau/2)+\boldsymbol{\Omega}\frac{2}{|\omega|}\sin(|\omega|\tau/2)\right]q(t_k)$$
$$=Q_t q(t_k),\qquad \tau \text{ is the sampling time,}\qquad (13)$$

the complete state transition comes to be

$$\boldsymbol{x}_{k+1}=[t_x+\tau v_x \quad t_y+\tau v_y \quad t_z+\tau v_z \quad Q_t q(t_k) \quad v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^{\mathrm{T}}. \qquad (14)$$

Measurement equation

A pinhole camera model as shown in Fig.1 is used, where the lens center is the camera reference origin. The first step is to transform the object coordinates to the camera reference. Next, the x and y coordinates of the object projected onto the image plane are found. Given image coordinates $\boldsymbol{p}(x, y)$ and camera coordinates $\boldsymbol{P}(x_{\mathrm{c}}, y_{\mathrm{c}}, z_{\mathrm{c}})$, these relations are

$$x=Fx_{\mathrm{c}}/z_{\mathrm{c}},\ y=Fy_{\mathrm{c}}/z_{\mathrm{c}}, \qquad (15)$$

where F is the focal length.

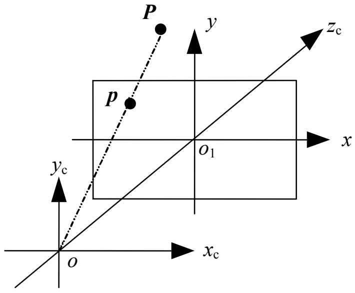

Fig.1 The pinhole camera model

The result of the perspective projection from the 3D lines is a set of dual vector quaternion coplanar lines located in the image plane. A format is needed to compare these transformed lines with line features measured from the acquired images. The format used for representing these lines is an (x_{lp}, y_{lp}) point called line point, which is defined as the intersection of the line feature with a line passing through the image origin that is perpendicular to the line feature. The definition of the line point on the image plane is illustrated in Fig.2.

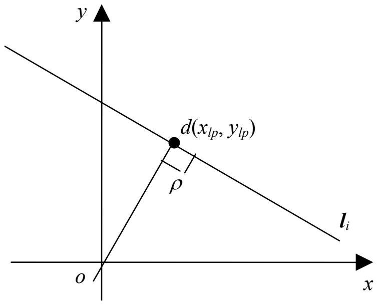

Fig.2 Definition of the line point in 2D image plane

The projected line lies in a plane defined by the 3D line and the projection center. This plane is described by

$$m_x x_{\mathrm{c}}+m_y y_{\mathrm{c}}+m_z z_{\mathrm{c}}=0, \qquad (16)$$

which intersects the image plane at $z_{\mathrm{c}}=F$. The result is an equation of the projected line in the $z_{\mathrm{c}}=F$ plane:

$$m_x x_i+m_y y_i+m_z F=0, \qquad (17)$$

where x_i and y_i are the image plane coordinates. The direction vector of the image line is

$$\boldsymbol{l}_i=\left[-\frac{m_y}{\sqrt{m_x^2+m_y^2}} \quad \frac{m_x}{\sqrt{m_x^2+m_y^2}} \quad 0\right]^{\mathrm{T}}. \qquad (18)$$

The line point is calculated from the dual vector image line as

$$x_{lp}=l_{iy}m_{iz},\ y_{lp}=-l_{ix}m_{iz}. \qquad (19)$$

In terms of 3D dual vector components,

$$x_{lp}=F\frac{m_x m_z}{m_x^2+m_y^2},\ y_{lp}=F\frac{m_y m_z}{m_x^2+m_y^2}. \qquad (20)$$

Measurement equation $h(\boldsymbol{x}_k, \boldsymbol{w}_k)$ is comprised of the line point equations obtained from each line given in Eq.(20). For the parameter $\boldsymbol{m}$ in measurement Eq.(20), when the initial normal vector $\boldsymbol{m}_{\mathrm{m}}$ is given, the transformed normal vector $\boldsymbol{m}$ can be computed from equation quaternion multiplication.

NONLINEAR FILTER METHODS

We present a brief description of the EKF, UKF and GPF based on (Julier and Uhlmann, 2004) and (Kotecha and Djuric, 2003) for better understanding of the paper in this section.

Extended Kalman filter

The EKF is a minimum mean-square-error (MMSE) estimator based on the Taylor series expansion of nonlinear function. Using only the linear expansion terms, it is easy to derive the following update equations for the mean and covariance of the Gaussian approximation to the posterior distribution of the states:

1. Initialize with

$$\bar{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0],\ \boldsymbol{P}_0 = E[(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^{\mathrm{T}}].$$

2. For k=1, 2, …, M

(1) Time update

$$\begin{cases} \bar{\boldsymbol{x}}_{k|(k-1)} = f(\hat{\boldsymbol{x}}_{k-1}, 0), \\ \boldsymbol{P}_{k|(k-1)} = \boldsymbol{F}_k \boldsymbol{P}_{k-1} F_k^{\mathrm{T}} + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^{\mathrm{T}}. \end{cases} \tag{21}$$

(2) Measurement update equations

$$\begin{cases} \boldsymbol{K}_k = \boldsymbol{P}_{k|(k-1)} \boldsymbol{H}_k^{\mathrm{T}} \left[\boldsymbol{U}_k \boldsymbol{R}_k \boldsymbol{U}_k^{\mathrm{T}} + \boldsymbol{H}_k \boldsymbol{P}_{k|(k-1)} \boldsymbol{H}_k^{\mathrm{T}} \right]^{-1}, \\ \hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_{k|(k-1)} + \boldsymbol{K}_k (\boldsymbol{y}_k - g(\bar{\boldsymbol{x}}_{k|(k-1)}, 0)), \\ \boldsymbol{P}_k = \boldsymbol{P}_{k|(k-1)} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k|(k-1)}, \end{cases} \tag{22}$$

where $\boldsymbol{Q}_k$ is the process noise covariance, $\boldsymbol{R}_k$ is the measurement noise covariance, and $\boldsymbol{K}_k$ is the Kalman gain. $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ are the Jacobians of the process model and $\boldsymbol{H}_k$ and $\boldsymbol{U}_k$ are the Jacobians of the observed model.

Unscented Kalman filter

The complete UKF algorithm that updates the mean and covariance of the Gaussian approximation to the posterior distribution of the states is given as follows [More details can be found in (Julier and Uhlmann, 2004)]:

1. Initialize with

$$\bar{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0],\ \boldsymbol{P}_0 = E[(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^{\mathrm{T}}].$$

2. For k=1, 2, …, M

(1) Calculate the sigma points and the weights

$$\begin{cases} \chi_{0,k-1} = \bar{\boldsymbol{x}}, \\ \chi_{i,k-1} = \bar{\boldsymbol{x}} + \left(\sqrt{(n_{\boldsymbol{x}} + \lambda)\boldsymbol{P}_k} \right)_i,\ i = 1, \cdots, n_{\boldsymbol{x}}, \\ \chi_{i,k-1} = \bar{\boldsymbol{x}} - \left(\sqrt{(n_{\boldsymbol{x}} + \lambda)\boldsymbol{P}_k} \right)_i,\ i = n_{\boldsymbol{x}} + 1, \cdots, 2n_{\boldsymbol{x}}, \\ W_0^{(m)} = \lambda / (n_{\boldsymbol{x}} + \lambda), \\ W_0^{(c)} = \lambda / (n_{\boldsymbol{x}} + \lambda) + (1 - \alpha^2 + \beta), \\ W_i^{(m)} = W_i^{(c)} = 1/[2(n_{\boldsymbol{x}} + \lambda)],\ i = 1, \cdots, 2n_{\boldsymbol{x}}, \end{cases} \tag{23}$$

where $n_{\boldsymbol{x}}$ is the dimension of the state vector, α determines the spread of the sigma points around $\bar{\boldsymbol{x}}$ and is usually set to a small positive value. κ is a secondary scaling parameter which is usually set to 0, and β is used to incorporate prior knowledge of the distribution of $\boldsymbol{x}$ (for Gaussian distributions, β=2 is optimal). $\left(\sqrt{(n_{\boldsymbol{x}} + \lambda)\boldsymbol{P}_k} \right)_i$ is the ith row of the matrix square root. λ is defined by $\lambda=\alpha^2(n_{\boldsymbol{x}}+\kappa)-n_{\boldsymbol{x}}$.

(2) Time update

$$\begin{cases} \chi_{i,k|(k-1)} = f(\chi_{i,(k-1)}),\ \bar{\boldsymbol{x}}_{k|(k-1)} = \sum_{i=0}^{2n_{\boldsymbol{x}}} W_i^{(m)} \chi_{i,k|(k-1)}, \\ \boldsymbol{P}_{k|(k-1)} = \sum_{i=0}^{2n_a} W_i^{(c)} [\chi_{i,k|(k-1)}^x - \bar{\boldsymbol{x}}_{k|(k-1)}][\chi_{i,k|(k-1)}^x - \bar{\boldsymbol{x}}_{k|(k-1)}]^{\mathrm{T}} + \boldsymbol{Q}_k, \\ \gamma_{i,k|(k-1)} = h(\chi_{i,k|(k-1)}^x, \chi_{i,k-1}^n),\ \bar{\boldsymbol{y}}_{k|(k-1)} = \sum_{i=0}^{2n_{\boldsymbol{x}}} W_i^{(m)} \gamma_{i,k|(k-1)}. \end{cases} \tag{24}$$

(3) Measurement update equations

$$\begin{cases} \boldsymbol{P}_{y_k y_k} = \sum_{i=0}^{2n_{\boldsymbol{x}}} W_i^{(c)} [\gamma_{i,k|(k-1)} - \bar{\boldsymbol{y}}_{k|(k-1)}][\gamma_{i,k|(k-1)} - \bar{\boldsymbol{y}}_{k|(k-1)}]^{\mathrm{T}} + \boldsymbol{R}_k, \\ \boldsymbol{P}_{x_k y_k} = \sum_{i=0}^{2n_{\boldsymbol{x}}} W_i^{(c)} [\chi_{i,k|(k-1)} - \bar{\boldsymbol{x}}_{k|(k-1)}][\gamma_{i,k|(k-1)} - \bar{\boldsymbol{y}}_{k|(k-1)}]^{\mathrm{T}}, \\ \boldsymbol{K}_k = \boldsymbol{P}_{x_k y_k} \boldsymbol{P}_{y_k y_k}^{-1},\ \bar{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_{k|(k-1)} + \boldsymbol{K}_k (\boldsymbol{y}_k - \bar{\boldsymbol{y}}_{k|(k-1)}), \\ \boldsymbol{P}_k = \boldsymbol{P}_{k|(k-1)} - \boldsymbol{K}_k \boldsymbol{P}_{y_k y_k} \boldsymbol{K}_k^{\mathrm{T}}, \end{cases} \tag{25}$$

where $\boldsymbol{Q}_k$ is the process noise covariance, $\boldsymbol{R}_k$ is the measurement noise covariance, and $\boldsymbol{K}_k$ is the Kalman gain.

Gaussian particle filter

From Corollary 1 of (Kotecha and Djuric, 2003), we may deduce that GPF provides better approximations to the integrations involved in the update processes than EKF and UKF in the presence of severe nonlinearities. The complete GPF algorithm that updates the mean and covariance of the Gaussian approximation to the posterior distribution of the states is given as follows [More details can be found in (Kotecha and Djuric, 2003)]:

1. Initialize with

$$\bar{\boldsymbol{x}}_0=E[\boldsymbol{x}_0],\ \boldsymbol{P}_0=E[(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_0)(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_0)^{\mathrm{T}}].$$

2. GPF measurement update algorithm

(1) Draw samples from the important function $q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ and denote them as $\{\boldsymbol{x}_k^{(j)}\}_{j=1}^N$ (N is the number of sample particles).

(2) Obtain the weights by

$$w_k^{(i)}=\frac{p(\boldsymbol{y}_k\mid \boldsymbol{x}_{k|k}^{(i)})N(\boldsymbol{x}_k^{(i)};\hat{\boldsymbol{x}}_{k|(k-1)},\boldsymbol{P}_{k|(k-1)}^{\boldsymbol{xx}})}{q(\boldsymbol{x}_{k|k}^{(i)};\boldsymbol{x}_k\mid \boldsymbol{y}_{1:k})}. \tag{26}$$

(3) Normalize the weights as

$$\tilde{w}_k^{(j)}=w_k^{(j)}\Big/\sum\nolimits_{j=1}^{N}w_k^{(j)}. \tag{27}$$

(4) Estimate the mean and covariance by

$$\begin{cases}\hat{\boldsymbol{x}}_{k|k}=\sum\limits_{i=1}^{N}\tilde{w}_k^{(i)}\boldsymbol{x}_{k|k}^{(i)},\\ \boldsymbol{P}_{k|k}^{\boldsymbol{xx}}=\sum\limits_{i=1}^{N}\tilde{w}_k^{(i)}(\boldsymbol{x}_{k|k}^{(i)}-\hat{\boldsymbol{x}}_{k|k})(\boldsymbol{x}_{k|k}^{(i)}-\hat{\boldsymbol{x}}_{k|k})^{\mathrm{T}}.\end{cases} \tag{28}$$

3. GPF time update algorithm

(1) Retain weighted samples of $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ obtained in the measurement update and denote them as $\{\boldsymbol{x}_k^{(j)},w_k^{(j)}\}_{j=1}^N$.

(2) For j=1,…,N, sample from $p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k=\boldsymbol{x}_k^{(j)})$ to obtain $\{\boldsymbol{x}_{k+1}^{(j)}\}_{j=1}^N$.

(3) Compute the mean and covariance as

$$\begin{cases}\hat{\boldsymbol{x}}_{(k+1)|k}=\sum\limits_{i=1}^{N}w_k^{(i)}\boldsymbol{x}_{(k+1)|k}^{(i)},\\ \boldsymbol{P}_{(k+1)|k}^{\boldsymbol{xx}}=\sum\limits_{i=1}^{N}w_k^{(i)}\boldsymbol{x}_{(k+1)|k}^{(i)}(\boldsymbol{x}_{(k+1)|k}^{(i)})^{\mathrm{T}}-\hat{\boldsymbol{x}}_{(k+1)|k}\hat{\boldsymbol{x}}_{(k+1)|k}^{\mathrm{T}}.\end{cases} \tag{29}$$

EXPERIMENTS

To demonstrate the pose and motion estimation performance of EKF, UKF and GPF, a target object with four coplanar points in a rectangular pattern is simulated, and the dimensions and the shape of the four points target is shown in Fig.3. The noisy image plane feature locations are used as inputs along with a priori knowledge, and a pinhole camera model with a known effective focal length is used in the simulation.

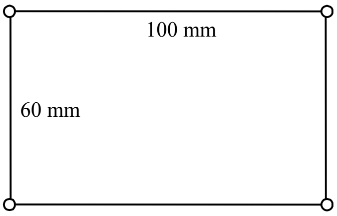

Fig.3 Four-point coplanar target used in the simulation experiment

The assumed initial state and the true initial state are shown in Table 1 (the assumed initial state is far from the true initial state). The initial error covariance matrix $\boldsymbol{P}_0=E[(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_0)(\boldsymbol{x}_0-\bar{\boldsymbol{x}}_0)^{\mathrm{T}}]$ and the diagonal elements of process noise matrix $\boldsymbol{Q}_k$ are 10^{-5}, the off-diagonal elements of $\boldsymbol{Q}_k$ are zero. The input noise is Gaussian with a standard deviation of 0.02 mm. Based on this noise level, the measurement error covariance matrix has diagonal elements of 0.0004 mm^2 for each measured variable.

We compare the estimation performance of the EKF, UKF and GPF based on the following metrics:

$$MSE_x=\frac{1}{n}\sum_{k=1}^{n}(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k)^{\mathrm{T}}. \tag{30}$$

Table 1 True and assumed initial states

Initial state	Translation (mm)			Quaternion				Linear velocity (mm/s)			Rotational velocity (rad/s)		
	t_x	t_y	t_z	q_0	q_1	q_2	q_3	v_x	v_y	v_z	ω_x	ω_y	ω_z
True	10	10	1000	1	0	0	0	−5	2	−5	−0.03	0.05	−0.20
Assumed	0	0	500	0.9	0.1	0.1	0.1	0.01	0.01	0.01	0.01	0.01	0.01

To enable a fair result of the estimation proposed in this paper, the estimation is averaged across a Monte Carlo simulation consisting of 50 runs. Each run is carried out with a different noise sample, and the simulation is run for 30 s with a measurement interval of 0.1 s. The pose and motion estimation results with *N*=200 particles are shown in Fig.4. Fig.5 shows the *MSE* for 50 random realizations with *N*=200 particles. The average *MSE* is plotted in Fig.6. We can see from Fig.6 that GPF performs better than EKF and UKF except that the performance of GPF is slightly worse than UKF on q_0 and q_3. A comparison of computation time is also shown in Fig.7 for simulation implemented on a 1.5 GHz Intel Pentium processor using MATLAB. Note that as expected, the computation time for GPF is much higher than that for EKF and UKF. However, much reduction in computation time can be expected for the GPF when it is implemented in parallel computer (Kotecha and Djuric, 2003).

CONCLUSION

An imaging technique with a single camera along with a reference object to calculate estimations for relative six degrees-of-freedom position and orientation as well as the associated velocity estimation is proposed in this paper. The system model is based on line features and a dual quaternion parameterization for the 3D transformation. The indirect measurement solutions of pose and motion from monocular vision are presented based on EKF, UKF and GPF respectively with simulation data. From the simulation results, we can see that GPF has improved performance over that of EKF and UKF for monocular vision-based pose and motion estimation.

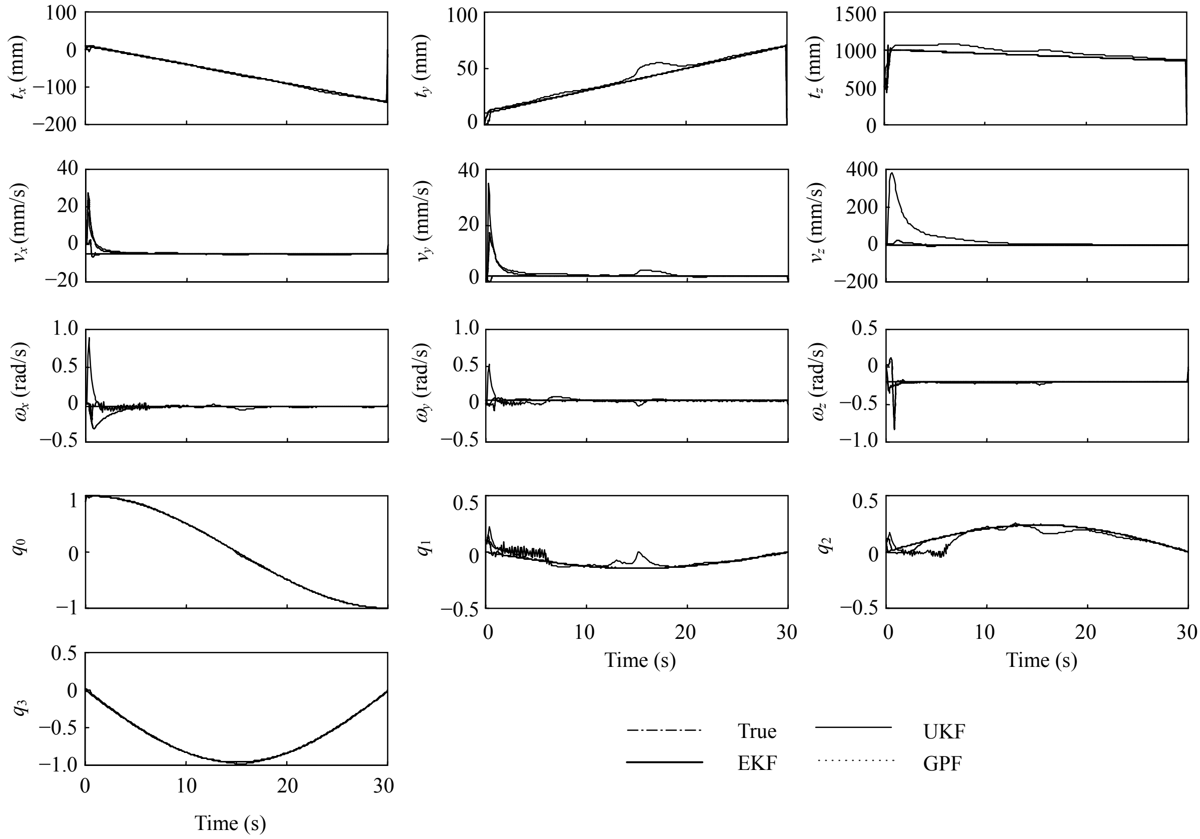

Fig.4 Pose and motion estimation results (average of 50 runs) with initialized value [0 0 500 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.9 0.1 0.1]. "True" refers to the true state

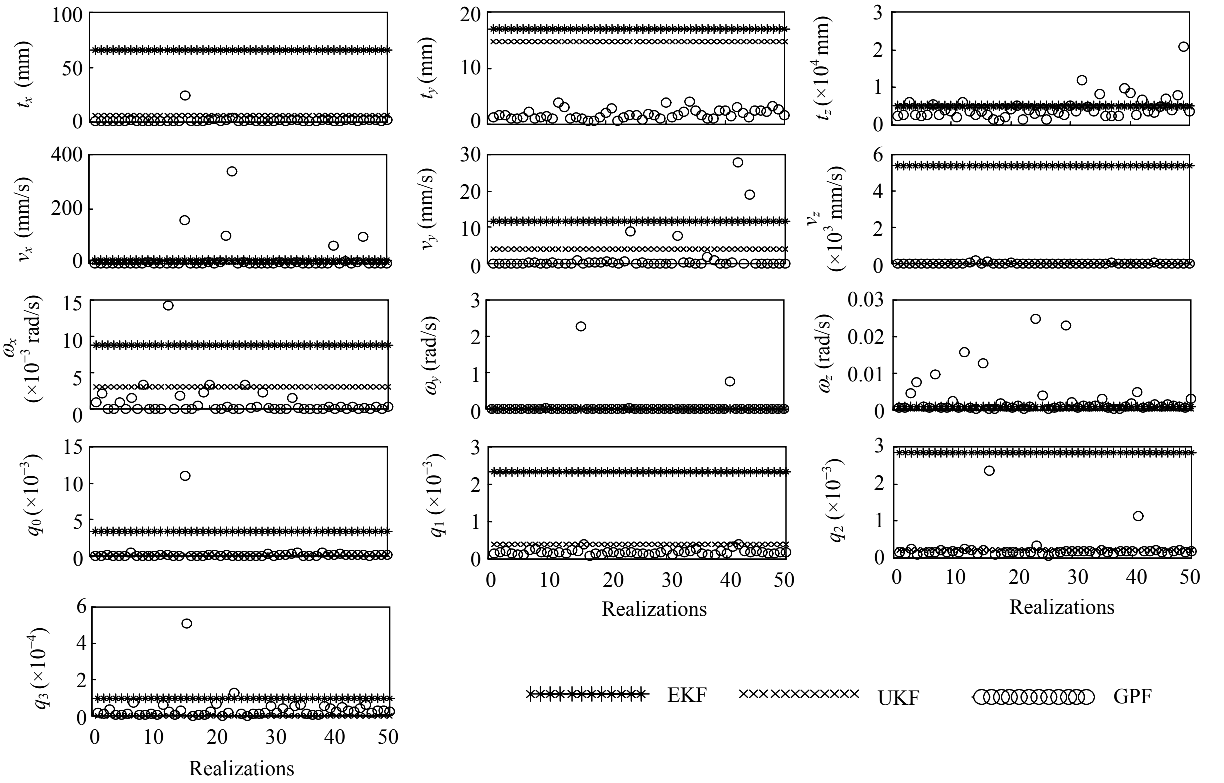

Fig.5 Mean square error comparison of EKF, UKF and GPF with 50 random realizations

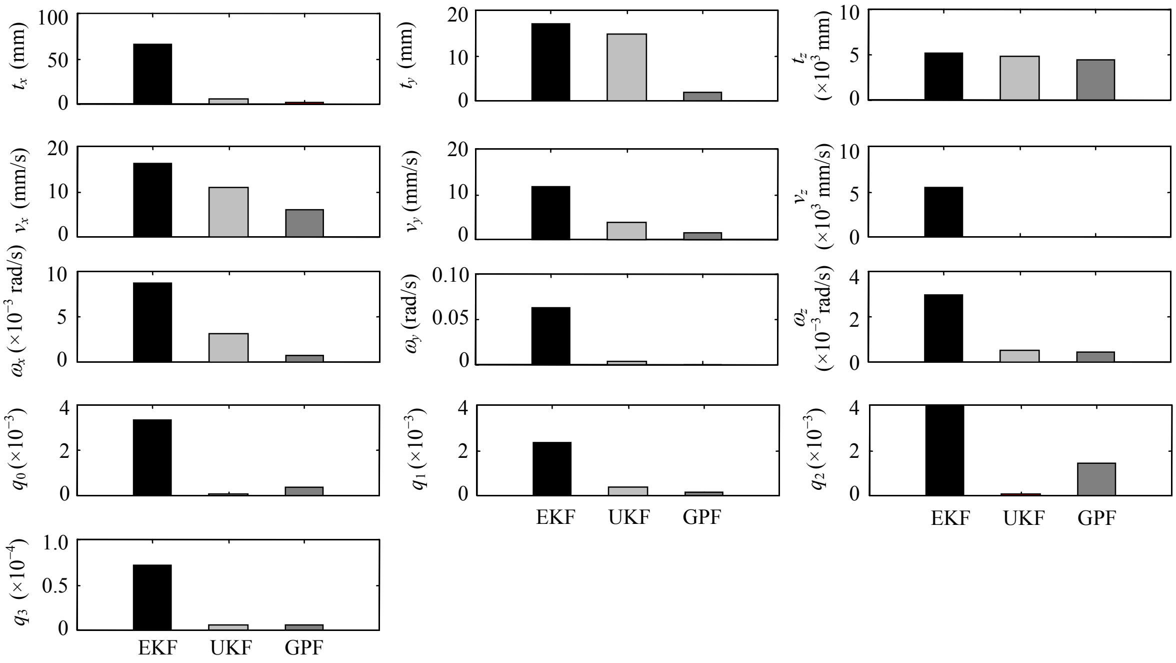

Fig.6 Performance comparison of EKF, UKF and GPF with average mean square error of 50 realizations

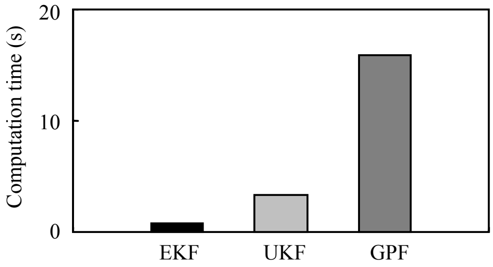

Fig.7 Computation time (per realization) comparison of EKF, UKF and GPF

References

Aggarwal, N., Karl, W.C., 2006. Line detection in images through regularized hough transform. *IEEE Trans. on Image Processing*, **15**:582-591. [doi:10.1109/TIP.2005.863021]

Andreff, N., Espiau, B., Horaud, R., 2002. Visual servoing from lines. *Int. J. Robot. Res.*, **21**:679-700. [doi:10.1177/027836402761412430]

Ansar, A., Daniilidis, K., 2003. Linear pose estimation from points or lines. *IEEE Trans. on Pattern Anal. Machine Intell.*, **25**:578-589. [doi:10.1109/TPAMI.2003.1195992]

Aytac, T., Barshan, B., 2002. Differentiation and localization of targets using infrared sensors. *Opt. Commun.*, **210**:25-35. [doi:10.1016/S0030-4018(02)01732-7]

Blais, F., Beraldin, J.A., 2001. Comparison of Pose Estimation Methods of a 3D Laser Tracking System Using Triangulation and Photogrammetry Techniques. SPIE Proc., Electronic Imaging, Videometrics and Optical Methods for 3D Shape Measurement VII. San Jose, California, USA, **4309**:185-194.

Broida, J., Chandrashekhar, S., Chellappa, R., 1990. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. on Aeros. Electron. Syst.*, **26**:639-656. [doi:10.1109/7.55557]

Burl, J.B., 1993. A reduced order extended Kalman filter for sequential images containing a moving object. *IEEE Trans. on Image Processing*, **2**:285-295. [doi:10.1109/83.236537]

Chang, C.C., Tsai, W.H., 1999. Reliable determination of object pose from line features by hypothesis testing. *IEEE Trans. on Pattern Anal. Machine Intell.*, **21**:1235-1241. [doi:10.1109/34.809118]

Chen, H.H., 1991. A Screw Motion Approach to Uniqueness Analysis of Head-eye Geometry. IEEE Conf. on Computer Vision and Pattern Recognition. Mau, KI, USA, p.145-151. [doi:10.1109/CVPR.1991.139677]

Chou, J.C.K., 1992. Quaternion kinematic and dynamic differential equations. *IEEE Trans. on Rob. Autom.*, **8**:53-64. [doi:10.1109/70.127239]

Daniilidis, K., 1999. Hand-eye calibration using dual quaternions. *Int. J. Robot. Res.*, **18**:286-298. [doi:10.1177/02783649922066213]

Deng, L.F., Wilson, W.J., Janabi-Sharifi, F., 2005. Decoupled EKF for Simultaneous Target Model and Relative Pose Estimation Using Feature Points. IEEE Conf. on Control Application. Toronto, Canada, p.749-754.

Dijk, E., Berkel, K., Aarts, R., Loenen, E., 2003. Ultrasonic 3D Position Estimation Using a Single Base Station. First European Symposium on Ambient Intelligence. Veldhoven, the Netherlands, p.133-148.

Diosi, A., Taylor, G., Kleeman, L., 2005. Interactive SLAM using Laser and Advanced Sonar. IEEE Conf. on Robotics and Automation. Barcelona, Spain, p.1103-1108.

Hazas, M., Ward, A., 2002. A Novel Broadband Ultrasonic Location System. Conf. on Ubiquitous Computing. Göteborg, Sweden, p.264-280.

Julier, S.J., Uhlmann, J.K., 2004. Unscented filtering and nonlinear estimation. *Proc. IEEE*, **92**:401-422. [doi:10.1109/JPROC.2003.823141]

Kim, G.H., Kim, J.S., Hong, K.S., 2005. Vision-based Simultaneous Localization and Mapping with Two Cameras. IEEE Conf. on Intelligent and Systems. Alberta, Canada, p.1671-1676.

Kotecha, J.H., Djuric, P.M., 2003. Gaussian particle filtering. *IEEE Trans. on Signal Processing*, **51**:2592-2601. [doi:10.1109/TSP.2003.816758]

Lin, H.H., Tsai, C.C., Hsu, J.C., Chang, C.F., 2003. Ultrasonic Self-localization and Pose Tracking of an Autonomous Mobile Robot Via Fuzzy Adaptive Extended Information Filtering. IEEE Conf. on Robotics and Automation, Taipei, Taiwan, p.1283-1290.

Miro, J.V., Dissanayake, G., Zhou, W.Z., 2005. Vision-based SLAM Using Natural Features in Indoor Environments. Conf. on Intelligent Sensors, Sensor Networks and Information Processing. Melbourne, Australia, p.151-156.

Pages, J., Collewet, C., Chaumette, F., Salvi, J., 2005. Visual Servoing by Means of Structured Light for Plane-to-Plane Positioning. Technical Report 5579, INRIA.

Porta, J.M., Kröse, B.J.A, 2003. Vision-based Localization for Mobile Platforms. First European Symp. on Ambient Intelligence. Veldhoven, the Netherlands, p.209-219.

Rehbinder, H., Ghosh, B.K., 2003. Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Trans. on Autom. Control*, **48**:186-199. [doi:10.1109/TAC.2002.808464]

Sarcinetti-Filho, M., Bastos-Filho, T., Freitas, R., 2003. Mobile Robot Navigation Via Reference Recognition Based on Ultrasonic Sensing and Monocular Vision. Conf. on Advanced Robotics. Coimbra, Portugal, p.204-209.

Shademan, A., Janabi-Sharifi, F., 2005. Sensitivity Analysis of EKF and Iterated EKF Pose Estimation for Position-based Visual Servoing. IEEE Conf. on Control Applications. Toronto, Canada, p.755-760.

Shakernia, O., Ma, Y., Koo, T., Sastry, S., 1999. Landing an unmanned air vehicle: vision based motion estimation and nonlinear control. *Asian J. Control*, **1**:128-145.

Takahashi, S., Ghosh, B.K., 2001. Motion and Shape Parameters Identification with Vision and Range. American Control Conf., **6**:4626-4631.

Tomono, M., 2005. 3D Localization and Mapping Using a Single Camera Based on Structure-from-Motion with Automatic Baseline Selection. IEEE Conf. on Robotics and Automation. Barcelona, Spain, p.3342-3347.

Wu, X.D., Jiang, X.H., Zheng, R.J., Huang, J.S., 2006. An Application of Unscented Kalman Filter for Pose and Motion Estimation Based on Monocular Vision. IEEE Int. Symp. on Industrial Electronics, **4**:2614-2619.

Youngrock, Y., DeSouza, G.N., Kak, A.C., 2003. Real-time Tracking and Pose Estimation for Industrial Objects Using Geometric Features. IEEE Conf. on Robotics and Automation. Taipei, Taiwan, p.3473-3478.

Zhang, X.Y., Liu, Y.C., Huang, T.S., 2006. Motion analysis of articulated objects from monocular images. *IEEE Trans. on Pattern Anal. Machine Intell.*, **28**:625-663. [doi:10.1109/TPAMI.2006.78]