Geometric particle swarm optimization for robust visual ego-motion estimation via particle filtering

Young Ki Baik, Junghyun Kwon, Hee Seok Lee, Kyoung Mu Lee

Dept. of Electrical and Computer Engineering, ASRI, Seoul National University, Korea

1. Introduction

Visual ego-motion estimation or visual odometry is the process to continuously estimate the 3-D camera pose based on 2-D image sequences captured by a camera. Visual ego-motion estimation plays an important role in various computer vision and robotics applications such as visual simultaneous localization and mapping (SLAM) and augmented reality [1–6].

Visual ego-motion estimation can be formulated as a state estimation problem and effectively solved by Bayesian filtering methods [7–9]. Particle filtering also known as sequential Monte Carlo [10,11] is one of the popular Bayesian filtering methods for visual ego-motion estimation because of its inherent ability to deal with nonlinearity present in camera projection [12–14].

How to choose the importance function is one of crucial factors for the performance of particle filtering [10]. The most simple choice is to use the state transition density determined by the state equation as the importance function. The well-known problem of the sampling from the state transition density is the waste of particles due to the fact that the recent measurement is not considered in particle sampling.

To improve the sampling efficiency and overall performance of particle filtering-based visual ego-motion estimation, we should utilize the optimal importance function that minimizes the variance of particle weights by explicitly considering the recent measurement [10]. Since the closed form of optimal importance function is not available for general nonlinear systems, its adequate approximation is required. In practice, the most feasible way is to approximate the optimal importance function as Gaussian distribution based on the local linearization of a measurement function [10].

However, there is an inherent problem in the local linearization-based approximation of the optimal importance function: there always exists a local linearization error for nonlinear measurement functions. When the state predicted by the state equation is fairly close to the true state, the linearization error is negligible and the optimal importance function can be approximated accurately. However, if the predicted state is far from the true state, the optimal importance function cannot be approximated accurately because of a large local linearization error. Fig. 1 graphically illustrates this local linearization error arising in approximating the optimal importance function.

To overcome the limitation of local linearization-based approximation, the unscented particle filter (UPF) [15] was proposed. The UPF employs a deterministic sampling approach called the unscented transformation (UT) [16,17] to estimate the mean and covariance of optimal importance function using a small number of samples called sigma points. Whereas the local linearization relies on the first order approximation, the unscented transformation can estimate the covariance accurately up to the third order [18] and thus approximate...
However, (b) when error is negligible and the optimal importance function can be approximated accurately.

The principal idea of PSO is to use interactions between particles sharing information between themselves to find the global optimum using particles exploring the solution space. Each particle explores the solution space

the optimal importance function more accurately. However, it cannot be a perfect solution because it still depends on the Gaussian approximation.

In this paper, we focus on abrupt camera motion that can result in large error in approximating the optimal importance function for visual ego-motion estimation. We define the abrupt camera motion as the motion that cannot be well predicted by the state equation. The most probable case is to perform visual ego-motion estimation using a handheld camera without additional motion sensors (e.g., web-cams). Since there is no available motion estimate, it is usual to use smooth motion models such as constant velocity for the motion dynamics. In this case, the sudden and sharp camera motion especially in orientation between adjacent frames shall not be predicted well by the smooth motion model and result in large error in the optimal importance function approximation. Even with additional motion sensors such as gyros and actuator information of mobile robots, the state can be predicted poorly. One of examples of such cases is to perform visual ego-motion estimation using cameras mounted on mobile robots moving on quite uneven terrain.

The first contribution of this paper is to present a novel particle filtering-based visual ego-motion estimation algorithm especially robust to abrupt camera motion. To obtain particles distributed consistently with the recent measurement even for the case of abrupt camera motion, we propose to use multi-layered importance sampling via particle swarm optimization (PSO), which is a meta-heuristic optimization algorithm to effectively find the global optimum using particles sharing information between themselves [19,20].

In our proposed multi-layered importance sampling, the particles sampled from a motion model are moved to higher likelihood region iteratively via PSO. The particles are made to be more consistent with the recent measurement by PSO and can be considered to be similar to those sampled from the optimal importance function. Since our proposed approach does not require the measurement equation approximation and the Gaussian assumption, we can obtain superior visual ego-motion estimation results in the case of abrupt camera motion compared with the local linearization and UT-based approximation methods.

Since we employ PSO which is an iterative method, efficient convergence is an important issue toward a real-time visual ego-motion estimation system. The second contribution of this paper is to reformulate the conventional vector space PSO algorithm in terms of the geometry of the special Euclidean group SE(3), which is a Lie group corresponding to the space of camera poses. By employing the proposed geometric PSO on SE(3), less iterations are needed for particle convergence than the vector space PSO, and it results in a real-time implementation of our proposed visual ego-motion estimation algorithm.

PSO has been already applied to various vision problems such as visual tracking [21,22] and visual SLAM [23,24]. Here, we concentrate on PSO for visual ego-motion estimation-related problems. Ref. [23] proposed PSO-FastSLAM, where PSO is used to enhance the resampling performance of the ordinary FastSLAM method used for visual SLAM. Hence, PSO-FastSLAM cannot properly deal with the abrupt camera motion. In Ref. [24], PSO is used to cope with the linearization error problem and overcome the ambiguity of repeated ceiling patterns by adopting the concept of multi-swarms. Ref. [24] can be considered to be similar to ours as Ref. [24] employs PSO to reduce the linearization error in approximating the optimal importance function. However, Ref. [24] is much more restricted than ours because it relies on the odometry information. Furthermore, both Ref. [23] and Ref. [24] only deal with 3-DOF ego-motion estimation problems, i.e., the state space is confined to a 2-D plane. Unlike Ref. [23] and Ref. [24], we deal with general 6-DOF ego-motion estimation problems without odometry information. To the best of our knowledge, our proposed algorithm is the first one that employs PSO in a 6-DOF visual ego-motion estimation system, where the unavailable odometry information brings much more difficulties.

We demonstrate the robustness of the proposed visual ego-motion estimation algorithm to abrupt camera motion via various experiments with simulated and real sequences. We first show that our proposed algorithm yields superior performance in the presence of abrupt camera motion to the local linearization and UT-based importance functions via simulated experiments. We also verify that the proposed geometric PSO is more efficient than the conventional vector space PSO. Via experiments with the real sequences where abrupt camera motion is prominent, we show that our proposed visual ego-motion estimation algorithm can yield quite accurate results in real-time despite abrupt camera motion.

Our visual ego-motion estimation algorithm is quite similar to particle filtering-based SLAM systems [25,26,12–14] because we build a map of landmarks and update it via extended Kalman filter (EKF) based on the estimated camera pose. The important difference between our algorithm and complete SLAM algorithms is that there is no loop closing in our algorithm. Our visual ego-motion algorithm can be equipped with a loop closing algorithm to be used for visual SLAM problems. Alternatively, our proposed multi-layered importance sampling via geometric PSO on SE(3) can be easily plugged into any existing particle filtering-based visual SLAM system to enhance the robustness to the abrupt camera motion.

The paper is organized as follows. In Section II, we first briefly review the conventional PSO on a vector space and then present our geometric PSO on SE(3). In Section III, we present our visual ego-motion estimation framework including multi-layered importance sampling using our geometric PSO. In Section IV, we verify the feasibility of the proposed visual ego-motion estimation algorithm via various experiments with the simulated and real sequences, while Section V concludes with a summary.

2. Geometric PSO on SE(3)

In this section, we present our geometric PSO algorithm formulated in consideration of the geometry of SE(3). We begin with a brief review of conventional PSO on a vector space. The more thorough review of PSO can be found in Ref. [20].

2.1. PSO on a vector space

The principal idea of PSO is to use interactions between particles sharing information between themselves to find the global optimum efficiently. Consider a set of particles $x^e = \{x^e_1, \ldots, x^e_N\}$ randomly distributed on the solution space. Each particle $x^e \in \mathbb{R}^N$ explores the solution space

**Fig. 1.** Graphical illustration of a local linearization error in approximating the optimal importance function (g(X); a nonlinear measurement function, X;: a predicted camera state by a motion model, z: a predicted measurement at X, z: an observed measurement, X: a real camera state). (a) When X is fairly close to the true state X, the linearization error is negligible and the optimal importance function can be approximated accurately. However, (b) when X is much different from X, the optimal importance function cannot be approximated accurately because of a large linearization error at X.
to find the global optimum based on two relative position vectors
\( \mathbf{p}_{gb} - \mathbf{x}_i \) and \( \mathbf{p}_{ib} - \mathbf{x}_i \), where \( \mathbf{p}_{gb} \) is a globally best particle and \( \mathbf{p}_{ib} \) is
the best record of each particle \( \mathbf{x}_i \).

Each particle computes its velocity \( \mathbf{v} \in \mathbb{R}^N \), which indicates
where to go, by summing the two relative position vectors as

\[
\mathbf{v} \leftarrow \mathbf{w} \cdot \mathbf{v} + c_1 \cdot \mathbf{r}_1 \odot (\mathbf{p}_{ib} - \mathbf{x}_i) + c_2 \cdot \mathbf{r}_2 \odot (\mathbf{p}_{gb} - \mathbf{x}_i),
\]

where \( \mathbf{w} \) is the inertia, \( c_1 \) and \( c_2 \) are the weighting coefficients
for the two relative position vectors, and \( \odot \) represents the component-wise
multiplication. \( \mathbf{r}_1 \) and \( \mathbf{r}_2 \in \mathbb{R}^N \) are random vectors drawn from a uniform
distribution on the interval between 0 and 1, which gives

\[
\mathbf{v} \leftarrow \mathbf{v} + \mathbf{v}', \quad \text{where } \mathbf{v}' \text{ is a stochastic component.}
\]

Then each particle \( \mathbf{x}_i \) moves to a new
position using \( \mathbf{v} \) as

\[
\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}.
\]

Finally, \( \mathbf{p}_{gb} \) and \( \mathbf{p}_{ib} \) are newly updated as

\[
\mathbf{p}_{ib} \leftarrow \mathbf{x}_i \quad \text{if } f(\mathbf{x}_i) > f(\mathbf{p}_{ib}),
\]

\[
\mathbf{p}_{gb} \leftarrow \mathbf{x}_i \quad \text{if } f(\mathbf{x}_i) > f(\mathbf{p}_{gb}).
\]

where \( f \) is a fitness function, i.e., an objective function.

By repeating Eqs. (1), (2), and (3) for a number of times, particles can
explore the solution space efficiently and convergence can be guaranteed.
In order to avoid convergence to local optima, anti-converging particles
called quantum particles are employed. At the end of a single PSO iter-
ation, quantum particles \( \mathbf{x}_q \) are randomly generated as

\[
\mathbf{x}_q \sim \mathbb{B}(\mathbf{p}_{gb}, \mathbf{r}_{cloud}),
\]

where \( \mathbb{B}(\mathbf{p}_{gb}, \mathbf{r}_{cloud}) \) represents the particle cloud centered at \( \mathbf{p}_{gb} \) within
radius \( \mathbf{r}_{cloud} \). Newly generated quantum particles are also evaluated by a

stochasticity in optimization. Then each particle \( \mathbf{x}_i \) moves to a new
position using \( \mathbf{v} \) as

\[
\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}'.
\]

Finally, \( \mathbf{p}_{gb} \) and \( \mathbf{p}_{ib} \) are newly updated as

\[
\mathbf{p}_{ib} \leftarrow \mathbf{x}_i \quad \text{if } f(\mathbf{x}_i) > f(\mathbf{p}_{ib}),
\]

\[
\mathbf{p}_{gb} \leftarrow \mathbf{x}_i \quad \text{if } f(\mathbf{x}_i) > f(\mathbf{p}_{gb}).
\]

where \( f \) is a fitness function, i.e., an objective function.

By repeating Eqs. (1), (2), and (3) for a number of times, particles can
explore the solution space efficiently and convergence can be guaranteed.
In order to avoid convergence to local optima, anti-converging particles
called quantum particles are employed. At the end of a single PSO iter-
ation, quantum particles \( \mathbf{x}_q \) are randomly generated as

\[
\mathbf{x}_q \sim \mathbb{B}(\mathbf{p}_{gb}, \mathbf{r}_{cloud}),
\]

where \( \mathbb{B}(\mathbf{p}_{gb}, \mathbf{r}_{cloud}) \) represents the particle cloud centered at \( \mathbf{p}_{gb} \) within
radius \( \mathbf{r}_{cloud} \). Newly generated quantum particles are also evaluated by a
fitness function and can be selected as the global best particle \( p_{gb} \) according to its fitness value. If any of quantum particles finds the better position and selected as \( p_{gb} \), then other particles can escape the local optima and converge to the global optimum consequently. It is usual to call normal particles neutral particles when quantum particles are employed. An example of optimization using PSO with quantum particles is shown in Fig. 2.

2.2. The special Euclidean group \( \text{SE}(3) \)

A camera pose can be represented by a rigid body transformation matrix of the form \(
\begin{bmatrix}
R & t \\
0 & 1
\end{bmatrix}
\) in the homogeneous coordinates, where \( R \) is a \( 3 \times 3 \) rotation matrix and \( t \) is a \( \mathbb{R}^3 \) vector. The rigid body transformation matrices and the rotation matrices can be identified with matrix Lie groups, i.e., the special Euclidean group \( \text{SE}(3) \) and the special orthogonal group \( \text{SO}(3) \), respectively. \( \text{SO}(3) \) and \( \text{SE}(3) \) can be formally defined as

\[
\text{SO}(3) = \left\{ R \in \mathbb{R}^{3\times3} | R^TR = I, \det(R) = +1 \right\}
\]

and

\[
\text{SE}(3) = \left[ \begin{array}{c}
R \\
t
\end{array} \right].
\]

where \( R \in \text{SO}(3) \) and \( t \in \mathbb{R}^3 \).

A Lie group is a differentiable manifold possessing group structure with smooth product and inverse operations. The Lie algebra associated with a Lie group is defined as a tangent vector space at the identity of a Lie group. The Lie algebras associated with \( \text{SE}(3) \) and \( \text{SO}(3) \) are denoted by \( \text{SE}(3) \) and \( \text{SO}(3) \), respectively. A Lie group and its Lie algebra can be related via the exponential map, i.e., \( \exp: \text{so}(3) \to \text{SO}(3) \) and \( \exp: \text{se}(3) \to \text{SE}(3) \). The log map is defined as the inverse of the exponential map. For matrix Lie groups, the matrix exponential and log give the exponential and log maps. \( \text{SO}(3) \) is a set of \( 3 \times 3 \) skew symmetric matrices of the form \( \omega = \begin{bmatrix}
0 & -\omega_3 & \omega_2 \\
\omega_3 & 0 & -\omega_1 \\
-\omega_2 & \omega_1 & 0
\end{bmatrix} \), and \( \text{se}(3) \) is given by \( \begin{bmatrix}
\omega \\
v
\end{bmatrix} \) with \( \omega \in \text{so}(3) \) and \( v \in \mathbb{R}^3 \).

In this paper, we represent the camera pose as \( \text{SE}(3) \) itself not employing vector parametrization of rotation matrices such as Euler angles. By \( \text{SE}(3) \) representation, we do not suffer from a singularity problem inherent in any kind of \( \mathbb{R}^3 \) vector parametrization of rotation matrices. However, since \( \text{SE}(3) \) is a curved space not a flat vector space, we should reformulate PSO in consideration of the geometry of \( \text{SE}(3) \) for better optimization behavior.

2.3. PSO on \( \text{SE}(3) \)

In order to formulate a geometric PSO algorithm on \( \text{SE}(3) \), we first consider a general Riemannian manifold case. What we have to consider when formulating PSO on a Riemannian manifold is that the difference between particles should be calculated as the minimal geodesic distance on a manifold. Since the Riemannian log and exponential maps are derived from the minimal geodesics on the Riemannian manifold, we can represent the difference between the elements on the Riemannian manifold as the one on its tangent vector space obtained via the Riemannian log map. In this manner, a particle can be considered to be...
a point \( X' \) on a manifold, and its velocity \( V' \) can be defined as the one on the tangent vector space of \( X' \).

The difference between \( X' \) and the individual best \( P_\theta \) can be identified as a vector on the tangent space at \( X' \) obtained via \( \log_{X'}(P_\theta) \). The difference between \( X' \) and the global best \( P_{gb} \) also can be represented as \( \log_{X'}(P_{gb}) \). Then the velocity \( V' \) is obtained by \( \log_{X'}(P_\theta) \) and \( \log_{X'}(P_{gb}) \) similarly to Eq. (1) and the particle update with \( V' \) is realized via \( \exp_{X'} \), the Riemannian exponential map at \( X' \), as \( \exp_{X'}(V') \). Fig. 3 depicts this geometric PSO procedure on a general Riemannian manifold. The procedure of PSO on manifolds has some similarity to the nonlinear mean shift on manifolds of Ref. [27] since the required operations are done on the tangent vector space of a manifold.

However, it is not straightforward to directly apply this geometric PSO on a general Riemannian manifold to \( \text{SE}(3) \). The first requirement of geometric PSO is to obtain the Riemannian exponential and log maps on a specific manifold. Since the minimal geodesics on \( \text{SE}(3) \) is given by the union of the respective geodesics on \( \text{SO}(3) \) and \( \mathbb{R}^3 \) [28], it is hard to obtain a single expression of the Riemannian exponential and log maps for \( \text{SE}(3) \). Fortunately, the Riemannian exponential and log maps for \( \text{SO}(3) \) are simply given by the left and right translations of \( \exp \) and \( \log \), which are the matrix exponential and log, thus. We can perform geometric PSO on \( \text{SE}(3) \) appropriately by splitting \( X' \in \text{SE}(3) \) into \( R' \in \text{SO}(3) \) and \( t' \in \mathbb{R}^3 \).

The calculations of particle velocity \( V_k' \in \mathbb{R}^3 \) and particle update for \( R' \) are given by

\[
V_k' \leftarrow \begin{bmatrix} \dot{V}_k' + c_1 r_k^r \odot \log (R_k' R_k) + c_2 r_k^s \odot \log (R_k' R_{gb}) \end{bmatrix},
\]

\[
R' \leftarrow R' \exp(V_k'),
\]

where \( R_{gb} \) and \( R_{gb} \) are the rotation parts of the individual and global best particles, respectively. Note that the differences are calculated by \( \log \) after multiplying \( R' \) to \( R_{gb} \) and \( R_{gb} \). Then the exponential of the resulting vector on the tangent vector space is multiplied by \( R' \). This is to apply the exact Riemannian exponential and log maps of \( \text{SO}(3) \). In Eq. (7), \( r_k^r \) and \( r_k^s \) represent \( \mathbb{R}^3 \) uniform random vectors, and \( \odot \) represents the component-wise multiplication with \( \mathbb{R}^3 \) elements represented in \( \mathbb{R}^3 \) column vectors with respect to basis elements of \( \mathbb{R}^3 \). The velocity calculation and particle update equations for \( t' \) can be represented by the ordinary PSO algorithm as

\[
t_k' \leftarrow \begin{bmatrix} \dot{t}_k' + c_1 r_k^r \odot \left( t_{gb} - t_k \right) + c_2 r_k^s \odot \left( t_{gb} - t_k \right) \end{bmatrix},
\]

\[
t' \leftarrow t' + V_k',
\]

where \( t_{gb} \) and \( t_{gb} \) are the translation parts of the individual and global best particles, respectively. \( r_k^r \) and \( r_k^s \) represent \( \mathbb{R}^3 \) uniform random vectors on \( \mathbb{R}^3 \).

The quantum particles around \( R_{gb} \) are generated by \( R_{gb} \cdot \exp(e) \), where \( e \in \mathbb{R}^3 \) is composed of a uniform random vector on \( \mathbb{R}^3 \) with the zero mean. Similarly, those for \( t_{gb} \) are generated from the uniform distribution with the mean \( t_{gb} \). After particle update, the fitness function evaluation is performed at the new position determined by merging the newly updated \( R' \) and \( t' \) into \( X' \).

3. Visual ego-motion estimation framework

We now present our proposed visual ego-motion estimation framework based on particle filtering and PSO. The distinct feature of our framework is the multi-layered importance sampling via our geometric PSO on \( \text{SE}(3) \), which gives robustness to abrupt camera motion. Our system uses only images from hand-held cameras as sensor input. We consider both monocular and binocular cases with known intrinsic camera parameters. The overall procedure of the proposed framework is summarized in Algorithm 1.

### Algorithm 1. Overall algorithm

1. Initialization
   a. Set the number of particles as \( N \) and \( k = 0 \).
   b. Set the number of PSO iteration as \( M \).
   c. For \( i = 1, \ldots, N \), set \( X_0 = i \), and \( A_i = 0 \).
   d. Initialize landmarks with the detected FAST corners.

2. Multi-layered importance sampling via PSO on \( \text{SE}(3) \)
   a. Set \( k = k + 1 \).
   b. Perform FAST corner detection and data association.
   c. For \( i = 1, \ldots, N \), draw \( X_k - \rho(X_k | i - 1) \).
   d. Perform geometric PSO on \( \text{SE}(3) \) with \( M \) iterations.
      - Update the particles via Eqs. (7) and (8).
      - Evaluate \( f(X_k) \) and update \( P_{gb} \) and \( P_b \).
      - Draw the quantum particles \( X_k^q \).
      - Evaluate \( f(X_k^q) \) and update \( P_{gb} \).
      - If Eq. (15) is satisfied, finish the iteration.
   e. For \( i = 1, \ldots, N \), compute \( A_i \) with Eq. (10).

3. Landmark update
   a. Resample \( X_k, A_i \) and the associated landmarks according to the normalized \( w_k \) and set \( w_k = 1 \).

4. Resampling
   a. Initialize landmarks from new FAST corners.

5. Go to Step 2.

3.1. State equation

To represent the camera pose as \( \text{SE}(3) \), we need a state equation properly defined on \( \text{SE}(3) \). The discretized stochastic differential equation on \( \text{SE}(3) \) previously employed in Ref. [29] is suitable for our purpose. Since we assume hand-held cameras whose odometry information is not available, we employ the auto-regressive (AR) process on \( \text{SE}(3) \) as the state dynamics, which is similar to the smooth motion models used in other previous work [8,14]. Though the smooth motion model is employed, our framework can work well even with abrupt camera motion by multi-layered importance sampling via geometric

![Initial position](image)

Fig. 7. Setup for visual ego-motion estimation experiments with known landmarks. A stereo camera is initially placed on the initial position and suddenly moves to another place on the sphere. Since the camera viewing direction is always fixed to the sphere center, the camera pose can be represented by two angle parameters, \( \theta \) and \( \phi \). The radius \( d \) is 5 m.
PSO on SE(3) that will be detailed in Section III-D. Our state equation on SE(3) is expressed as

\[ X_k = X_{k-1} \exp (A_{k-1} + dW_k \sqrt{\Delta t}) , \]

where \( X_k \in SE(3) \), \( dW_k \) is the Wiener process noise on SE(3) with a covariance \( \Sigma_w \in \mathbb{R}^{6 \times 6} \), and \( A_{k-1} \) is the AR state dynamics term calculated as

\[ A_{k-1} = a \log (X_{k-2}^{-1} X_{k-1}) \]

with the scalar AR process parameter \( a \) usually set to be smaller than 1.

### 3.2. Measurement equation

In our framework, we regard a landmark as a 3-D point and represent it as a 6-D vector using the inverse-depth parametrization since it allows us to deal with 3-D points at infinity and initialize landmarks using a single image from a monocular camera [30].

With the inverse-depth parametrization, a landmark \( L \) is represented as a 6-D vector, which is composed of an initial 3-D point \( \hat{r} \), orientation \( (\hat{\theta}, \hat{\phi}) \), and inverse depth \( \hat{\rho} \), with a covariance \( \Sigma_L \in \mathbb{R}^{6 \times 6} \) representing its uncertainty. Then measurement \( y_k \) from \( X_k \) is the 2-D image coordinates of the camera projection of a landmark, which is represented as

\[ y_k = g(X_k^{-1} h(L)) + n_k , \]

where \( h \) is a function transforming a 6-D \( L \) into a 3-D position in homogeneous coordinates, \( g \) is a camera projection function with the known internal parameters, and \( n_k \) is Gaussian measurement noise with a covariance \( \Sigma_y \in \mathbb{R}^{2 \times 2} \). Then the likelihood distribution is given as

\[ p(y_k | X_k) = N(y_k; \hat{y}_k, Q) , \]

where \( y_k \) is an observed measurement and \( \hat{y}_k \) is the predicted measurement from \( X_k \). Note that if there are \( N \) landmarks in a scene, \( y_k \) becomes a \( \mathbb{R}^{2N} \) vector in the monocular case.

In the binocular camera case, it is possible to use depth information which is computed by stereo triangulation as Ref. [13]. Such depth information, however, is sensitive to noise, especially when a landmark is far from the camera. Thus, we use horizontal differences of stereo pair points as measurement directly without triangulation. Accordingly, we still represent a landmark using the inverse-depth parametrization even in the binocular camera case. In this case, we set a value of a diagonal term of \( \Sigma_L \) related to the inverse depth \( \hat{\rho} \) in inverse proportion to the disparity of \( L \) at initialization because a possible error for a landmark with smaller disparity will be greater than for one with larger disparity. Note that this over-parametrization in the binocular case is intended not to rely on triangulation and the uncertainty of \( \hat{\rho} \) is usually much smaller than the monococular case.

In the binocular camera case, we use the same projection function \( g \) for both left and right cameras with the assumption of the rectified binocular camera. Therefore, the measurement \( \bar{y}_k \) from \( X_k \) in the binocular case becomes a \( \mathbb{R}^4 \) vector composed of projections of a landmark \( L \) and can be expressed as

\[ \bar{y}_k = \left[ g(X_k^{-1} h(L)) \ g(X_k^{-1} h(L)) \right] + \pi_k , \]

where \( T_{LR} \in SE(3) \) represents coordinate transformation from a left camera to a right one, and \( \pi_k \) is Gaussian measurement noise with a covariance \( \Sigma_y \in \mathbb{R}^{4 \times 4} \). Since we rectify the stereo images to have epipolar lines parallel to the horizontal axis, we set the values of \( \Sigma_y \) related to the vertical coordinates of a right camera image as nearly 0. It means that the degrees of freedom of \( \bar{y}_k \) is actually 3. Note that the state \( \dot{X}_k \) represents the left camera pose. If there are \( N \) landmarks in a scene, \( \bar{y}_k \) becomes a \( \mathbb{R}^{4N} \) vector.

### Table 1

<table>
<thead>
<tr>
<th>Numbers of particles used for the smooth camera motion test.</th>
<th>The number of particles ( N_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF-VE</td>
<td>500(A1), 1000(A2), 1500(A3)</td>
</tr>
<tr>
<td>PSO-VE</td>
<td>70(B1), 100(B2), 200(B3), 300(B4), 400(B5)</td>
</tr>
<tr>
<td>LIN-VE</td>
<td>100(C1), 200(C2), 400(C3), 600(C4), 800(C5)</td>
</tr>
<tr>
<td>UPF-VE</td>
<td>50(D1), 100(D2), 200(D3), 300(D4)</td>
</tr>
</tbody>
</table>
vector. Fig. 4 depicts the measurement and the state transition model in the binocular camera case.

3.3. Landmark initialization and data association

At the initial frame, we first detect the FAST corners [31] above a certain threshold from an image. A \(31 \times 31\) patch centered at the selected corner on the image is extracted with landmark initialization and used in data association in future frames. At the future frames, the extracted landmark images are transformed up to the scale and rotation according to the predicted camera pose, and then landmarks are associated with the detected FAST corners on new input images by a normalized cross correlation (NCC) value. To minimize the boundary effect caused by image transformation, only a smaller \(21 \times 21\) patch of the transformed image is considered in calculating NCC. We newly initialize landmarks in data association in future frames. At the future frames, the extracted \(31 \times 31\) patch centered at the selected landmark is considered in calculating NCC. We newly initialize landmarks based on the remaining corners. We maintain only the most recent 500 landmarks in a built map for computational efficiency.

In the binocular camera case, we detect the FAST corners for both images and perform stereo matching based on the corners on the reference camera (a left camera in our implementation). We apply a simple template matching method based on the sum of absolute difference (SAD), and use rectified images to reduce the search range to horizontal lines. Then the so called LR check follows to assure the matching consistency, which enhances the performance of data association. The overall procedure of the data association is briefly shown in Fig. 5.

3.4. Multi-layered importance sampling via PSO

In our multi-layered importance sampling, the state transition density \(p(X_k|X_{k-1})\) is used as the initial importance function. At time \(k\), we first sample particles \(X_k^i\) via Eq. (9). Then we perform our geometric PSO on SE(3) to move the initially sampled particles \(X_k^i\) to a higher likelihood region. A single iteration of PSO can be regarded as a single layer of multi-layered importance sampling.

In the PSO iteration, we first set \(P_{gb}\) and \(P_{ib}\) according to the fitness function \(f(X_k^i)\), which is defined as

\[
f(X_k^i) = -\frac{1}{N_i} \|y_k - y_k^i\|^2,
\]

where \(N_i\) is the number of matched landmarks. By maximizing the fitness value \(f(X_k^i)\) via geometric PSO on SE(3), we can obtain the particles distributed consistently with the recent measurement \(y_k\).

It is important for PSO performance how to appropriately set the values for \(w\), \(c_1\), and \(c_2\) in Eqs. (7) and (8). \(c_1\) and \(c_2\) control the relative attraction to the \(P_{gb}\) and \(P_{ib}\), respectively, and it is general to set \(c_1 = c_2\). The convergence speed is mainly affected by \(w\) which has a similar role to the temperature in the simulated annealing. Although smaller values of \(w\) assure faster convergence, the exploring ability is diminished and it can lead to convergence to local optima. On the other hand, too large values of \(w\) slow down the convergence speed. Fig. 6 shows an example that how the fitness values change in accordance with different values of the inertia parameter \(w\). We empirically set \(w = 0.5\) and \(c_1 = c_2 = \frac{1}{2}\) to yield the best performance in our experiments.

The geometric PSO iteration is terminated when the following condition is satisfied:

\[
f(\tilde{P}_{gb}) - f(P_{wor}) < \tau,
\]

where \(P_{wor}\) denotes the global worst particle at each iteration. We empirically set \(\tau = 1\). We also limit the maximum number of iteration to 15 in order to ensure real-time performance of the proposed algorithm.

After the particle update, we generate quantum particles as many as 20\% of ordinary neutral particles with a radius \(r_{cloud}\) around \(P_{gb}\). Here, we set \(r_{cloud}\) to be equal to the diagonal terms of the state covariance \(\Sigma_y\). The quantum particles generated at each iteration play the important role to find the global optimum even in the case of abrupt camera motion. When the multi-layered importance sampling via geometric PSO is finished, \(\tilde{A}_k\) are updated with the finally obtained particles \(X_k^i\) via Eq. (10).

After the multi-layered importance sampling via geometric PSO on SE(3), the landmarks are updated by EKF based on the finally obtained camera pose \(X_k^i\) and the measurement \(y_k\). Although we adopt the notion of multi-layered importance sampling, the weights should be calculated differently from generic multi-layered importance sampling of Ref. [32]. In Ref. [32], it is assumed that different measurements are available for each layer. However, in our framework, each geometric PSO iteration is performed with the same measurement \(y_k\).

The simplest way to deal with this problem is to assume that we actually have another state equation which propagates particles from \(X_k^i_{k-1}\) to the finally obtained \(X_k^i\) via geometric PSO and we sample particles from such state transition density. In this case, the particle weight calculation is simply given by

\[
w_k = w_{k-1} \cdot P(y_k|X_k^i)
\]

as in the usual case of state transition density-based importance functions. Finally, the optimal state estimation \(X_k^i\) is computed by the sample mean of SE(3) particles following the formula given in, e.g., Ref. [33].

Fig. 9. Ego-motion estimation errors of (a) position and (b) orientation for the smooth camera motion by PF-VE, LIN-VE, UPF-VE, and PSO-VE.
4. Experiments

In this section, we verify the feasibility and robustness of our proposed visual ego-motion estimation algorithm via various experiments with simulated and real sequences. The experiments were performed on an Intel Core-2 Quad 2.4 GHz processor with C++ implementation.

4.1. Experiments with simulated sequences

We demonstrate the superior performance of our visual ego-motion estimation algorithm (PSO-VE) to other algorithms for the simulated data set containing abrupt camera motions. The algorithms compared with ours are particle filtering-based visual ego-motion estimation ones employing different importance functions based on the state transition density (PF-VE), local linearization (LIN-VE), and unscented transformation (UPF-VE). For all the compared algorithms, instead of conventional vector space state equations, we use the same state equations on SE(3) as ours for fair performance comparison. We also show that our geometric PSO on SE(3) yields better optimization performance than the conventional vector space PSO.

4.1.1. Ego-motion estimation with known landmarks

We first performed visual ego-motion estimation with exactly known landmarks in order to better quantify ego-motion estimation accuracy of the proposed algorithm. A binocular stereo camera is placed on the sphere and takes the images of nine landmarks as depicted in Fig. 7. The camera viewing direction is always fixed to the sphere center. In this case, we can represent various camera pose changes including both translation and orientation by adjusting only 2 angle parameters, $\theta$ and $\phi$, as shown in Fig. 7. By varying these two parameters, we can quantify the influence of a local linearization error to the ego-motion estimation accuracy.

Supposing the camera suddenly moves between the two adjacent frames from its initial position to another place on the sphere, we compared the ego-motion estimation performance of LIN-VE, UPF-VE, and PSO-VE. We assumed that the landmark positions are exactly known and the data association is perfect. In order to make the computation time of each algorithm comparable, we used 800, 400, and 400 particles for LIN-VE, UPF-VE, and PSO-VE, respectively.

Fig. 8 shows the errors of ego-motion estimation by each algorithm. The errors were averaged over 10 independent runs for each algorithm. From Fig. 8, we can see that all algorithms yielded quite small errors for small $\theta$ and $\phi$, which can be regarded as smooth camera pose change. However, as $\theta$ and $\phi$ increased, the errors of LIN-VE and UPF-VE also gradually increased. It is clear that this increasing error is due to the increasing approximation error for the relatively large camera pose change. Note that the errors of PSO-VE are consistently small despite the large camera pose change. From these results, we can conclude that PSO-VE can provide much more robust ego-motion estimation performance than LIN-VE and UPF-VE in the case of abrupt camera motion.

4.1.2. Ego-motion estimation with unknown landmarks

We now demonstrate the overall performance of our proposed visual ego-motion estimation algorithm, PSO-VE, for simulated sequences without the prior knowledge of landmarks. We again compare our ego-motion estimation performance with PF-VE, LIN-VE and UPF-VE.

Fig. 10. Ego-motion estimation results for the simulated sequence of abrupt camera motion by (a) LIN-VE, (b) UPF-VE, and (c) PSO-VE, respectively. The ground truth camera trajectory and landmarks are shown in white and gray, respectively, while the estimated camera trajectory and landmarks are shown in cyan and green, respectively.

Fig. 11. Estimated position errors for the simulated sequence of abrupt camera motion by LIN-VE (cases I, II, and III), UPF-VE (case IV), and PSO-VE (case V).
Assuming a 9 m by 8 m rectangular room where landmarks are placed near the walls, we generated binocular stereo camera trajectories and corresponding measurement sequences with known camera parameters. For measurement noise $n_k$, we used the identity matrix as the covariance, i.e., the standard deviation for each image coordinates is 1 pixel. As aforementioned, the measurement noise for the vertical coordinates of a right camera was set to nearly 0. We assumed that the camera was set to nearly 0. We assumed that data association is always successful. We used two different camera trajectories: one is a smooth elliptic trajectory within predefined rectangular room and the other contains temporary abrupt camera motion with the same elliptic trajectory.

### 4.1.2.1. Smooth camera motion

For the smooth camera motion trajectory, we set the same state covariances for all algorithms. To take into account different computational complexities of algorithms, we used various numbers of particles for each algorithm as shown in Table 1. Note that we additionally included the algorithm employing the state transition density-based importance function, PF-VE.

Fig. 9 shows the results for the smooth camera motion. The errors were averaged over 30 independent runs for each algorithm and corresponding error bars are shown together. We can clearly see that LIN-VE, UPF-VE, and PSO-VE yielded quite accurate ego-motion estimation results for the smooth camera motion except for PF-VE.

### 4.1.2.2. Abrupt camera motion

Ego-motion estimation results for the abrupt camera motion trajectory by each algorithm with the same state covariance are shown in Fig. 10. In order to make the computation time for each algorithm similar, we used 800 particles for LIN-VE and 400 particles for UPF-VE and PSO-VE. When a binocular camera made abrupt camera motion suddenly, LIN-VE failed to yield accurate ego-motion estimation results because of a large local linearization error as shown in Fig. 10(a). However, as shown in Fig. 10(c), PSO-VE was able to yield accurate ego-motion estimation results owing to the multi-layered importance sampling via geometric PSO despite the sudden abrupt camera motion. The ego-motion estimation result by UPF-VE shown in Fig. 10(b) is better than LIN-VE, but its accuracy is still worse than that of PSO-VE.

One can claim that such sudden abrupt camera motion can be properly dealt with by using more particles with larger state covariances. However, it cannot be a perfect solution because there always exists a large local linearization error caused by an inaccurate state dynamic model without odometry information in the case of sudden abrupt camera motion. This can be verified from the results shown in Fig. 11. As shown in Fig. 11 (see also Extension 1), the estimated position errors by LIN-VE with more particles and larger state covariances (case II and III) are still greater than that of PSO-VE. It is worth to note that the error by LIN-VE with larger state covariance (case III) is worse than that by LIN-VE with smaller state covariance (case II). This phenomenon is also inevitable for UPF-VE except PSO-VE that does not rely on approximation.

### 4.1.3. Geometric PSO vs vector space PSO

We also compare the performance of our geometric PSO with that of the conventional vector space PSO. For a vector space PSO, we use a 6-D vector representation of $SE(3)$, which is composed of a 3-D translational vector and a 3-D rotation vector corresponding to the three independent values of a skew symmetric matrix obtained by taking matrix log of a rotation matrix. We analyzed the results of both geometric PSO and vector space PSO for the simulated smooth motion sequence by setting the maximum number of iteration to 100.

The analysis result is summarized in Table 2. As shown in Table 2, the computation time required to perform a single geometric PSO iteration is greater than the vector space PSO because of required matrix exponential and log operations. However, our geometric PSO requires less iterations until convergence than the vector space PSO, and it results in faster processing time per frame than the vector space PSO case. Although the means of $f(P_{wst})$, the fitness value of the worst particle, for the entire frames for both cases are quite similar, the mean of $f(P_{wst})$, the fitness value of the best particle, for the vector space is much worse than that for our geometric PSO. From these results, it can be confirmed that our geometric PSO shows the better optimization behavior for our visual ego-motion estimation purpose than the vector space PSO in terms of both overall computational efficiency and optimization ability.

---

**Table 2**

Performance comparison between the vector space PSO and geometric PSO for experiments with the simulated smooth camera motion sequence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Vector space PSO</th>
<th>Geometric PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time per iteration</td>
<td>1.96 ms</td>
<td>2.58 ms</td>
</tr>
<tr>
<td>Iterations per frame</td>
<td>35.05</td>
<td>16.72</td>
</tr>
<tr>
<td>Time per frame</td>
<td>68.70 ms</td>
<td>43.14 ms</td>
</tr>
<tr>
<td>Mean of $f(P_{wst})$</td>
<td>$-0.1783$</td>
<td>$-0.1779$</td>
</tr>
<tr>
<td>Mean of $f(P_{wst})$</td>
<td>$-1.9516$</td>
<td>$-0.9219$</td>
</tr>
</tbody>
</table>

---

**Table 3**

Processing time (ms) per frame of PSO-VE with 200 particles and 25 landmarks in an image frame. CE: corner extraction, DA: data association, IS: importance sampling, LU: landmark update.

<table>
<thead>
<tr>
<th>Camera</th>
<th>CE &amp; DA</th>
<th>IS</th>
<th>LU</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monocular</td>
<td>5.7</td>
<td>20.8</td>
<td>7.1</td>
<td>33.6</td>
</tr>
<tr>
<td>Binocular</td>
<td>6.3</td>
<td>36.9</td>
<td>12.2</td>
<td>55.4</td>
</tr>
</tbody>
</table>

Fig. 12. Ego-motion estimation results and reconstructed 3-D maps by PSO-VE with monocular (left) and binocular (right) cameras for “Lab”.
We now demonstrate the overall performance of the proposed visual ego-motion estimation framework especially robust to abrupt camera motion. We use our own “Lab”, “Office”, “Campus I” and “Campus II” obtained by a Point Grey’s Bumblebee binocular stereo camera (BB-HICOL-60) and the publicly available “New College” data set [34]. Our C++ implementation can run in 20–30 fps with 200 particles yielding satisfactory results for these real sequences.

4.2.1. Experiment 1: Lab
We first test the overall performance of PSO-VE using “Lab” where the camera motion is relatively smooth. The number of particles used is 200 for both monocular and binocular stereo cases. From the results shown in Fig. 12, we can regard that PSO-VE with only 200 particles yields successful ego-motion estimation results for both monocular and binocular cases because the reconstructed landmarks clearly show the rectangular shape of the lab environment. PSO-VE can run in real time with 200 particles and 25 landmarks in an image frame as shown in Table 3. The results for “Lab” can be seen in Extension 1.

4.2.2. Experiment 2: Office
We also test the robustness of PSO-VE to the abrupt camera motion using “Office” where the abrupt camera motion occurs several times. We use 200, 400 and 200 particles for PSO-VE, LIN-VE and UPF-VE, respectively. The results are shown in Fig. 13. In spite of the abrupt camera motion, we can see that PSO-VE can accurately estimate the ego-motion as shown in the bottom row of Fig. 13, owing to the multi-layered importance sampling via the proposed geometric PSO on SE(3). Contrarily, the ego-motion estimation of LIN-VE is quite erroneous as shown in the second row because of the large linearization error for the abrupt camera motion. The results for “Office” also can be seen in Extension 1.

---

Fig. 13. Ego-motion estimation results for “Office” (top row) by LIN-VE (second row), UPF-VE (third row) and PSO-VE (bottom row), respectively.

Fig. 14. Left: ego-motion estimation results by PSO-VE for “Campus I”. Right: the estimated camera trajectory overlaid on the aerial photo. The camera trajectory length is approximately 100 m.
4.2.3. Experiment 3: Campus I

Fig. 14 shows our ego-motion estimation results for “Campus I” (Extension 1), where the abrupt camera motion caused by fast walking with holding a binocular camera by hand is present. Observing the estimated trajectory overlaid on the aerial photo, it seems that PSO-VE yielded reasonably accurate ego-motion estimation results owing to the proposed multi-layered importance sampling via geometric PSO despite the abrupt camera motion.

4.2.4. Experiment 4: Campus II

There also exists abrupt camera motion mainly caused by many speed bumps in “Campus II”, which was captured during a 2 km long car ride. Since we drove a car along the beltway, we can precisely obtain the ground truth camera trajectory from the aerial photo. The estimated camera trajectory by PSO-VE for “Campus II” is shown in Fig. 15. Comparing the estimated camera trajectory with the ground truth, we can see that PSO-VE yielded accurate ego-motion estimation results for a
Fig. 17. Estimated camera trajectories for “New College” by (a) PSO-VE, (b) UPF-VE, and (c) R-SLAM, respectively. The visual odometry given in Ref. [34] is also shown in (d) for comparison purpose. The result by R-SLAM is taken from Ref. [35].
Fig. 18. Estimated camera trajectories for “New College” by PSO-VE with (a) 2× downsampling and (b) 3× downsampling and by UPF-VE with (c) 2× downsampling and (d) 3× downsampling.
quite long camera trajectory with a small error at the latter part despite the severe abrupt camera motion. A result for “Campus II” can be seen in Extension 1.

4.2.5. Experiment 5: New College

We finally test our proposed framework on “New College” whose camera trajectory length is approximately 2.2 km. In “New College”, a binocular camera moves smoothly along a circulating trajectory. The estimated camera trajectory for “New College” by PSO-VE is shown in Fig. 16. We can compare the accuracy of the estimated camera trajectory with the result of R-SLAM [35], which is a state-of-the-art bundle adjustment-based large scale SLAM method using a binocular stereo camera. Fig. 17 shows the results for “New College” by PSO-VE, UPF-VE, and R-SLAM. From Fig. 17, we can clearly see that the accuracy of PSO-VE is comparable to that of R-SLAM for “New College” and much better than the visual odometry given in Ref. [35]. Note that our result for “New College” is also clearly better than that of UPF-VE. A result for “New College” can be seen in Extension 1.

We additionally test the robustness of PSO-VE to abrupt camera motion by downsampling the original “New College” data set. Fig. 18 shows the results for “New College” by PSO-VE and UPF-VE with 2× and 3× downsampling, which means that we use only every other and third images of the original data set. We can see that the result by PSO-VE with 2× downsampling is still acceptable while UPF-VE failed to yield satisfactory results with 2× downsampling. Although the result by PSO-VE with 3× downsampling is somewhat erroneous, it is far better even than the result by UPF-VE with 2× downsampling. This robustness of PSO-VE superior to UPF-VE stems from our multi-layered importance sampling via geometric PSO.

To verify the effectiveness of quantum particles for the performance of PSO-VE, we checked the results for “New College” by PSO-VE with and without quantum particles. Table 4 summarizes the comparison results. From Table 4, we can see that good best particle update is rarely done by quantum particles for the normal speed sequence and there is no noticeable performance difference between with and without quantum particles. However, for the 2× and 3× downsampled sequences, the role of quantum particles is much clearer. As shown in Table 4, the number of global best particle update by quantum particles increased and all the other indicators to represent the optimization performance such as the fitness values of best and worst particles and required number of iterations also improved as a result.

4.2.6. Statistics of camera motion abruptness

In this subsection, we summarize how much each real test sequence contains the abrupt camera motion. As there is no ground truth for the real test sequences, we instead use our estimated ego-motion results as the ground truth. Since our AR process-based state dynamics assumes smooth camera velocities, i.e., small accelerations, a camera motion with a high acceleration can be considered as the abrupt motion. Therefore, we calculate accelerations for each frame and determine frames whose $L^2$ norm of accelerations are above the thresholds as the abrupt camera motion frames. We set the thresholds as 0.1 m/frame$^2$ and 0.1 rad/frame$^2$ for position and orientation, respectively.

Table 5 shows the ratio of the abrupt camera motion frames to the total number of frames for each test sequence. In the table, “Mean” and “Var.” represent the mean and variance of $L^2$ norm of accelerations of the abrupt camera motion frames. For “New College”, we can verify from the increasing ratio of the abrupt camera motion frames that the camera motion abruptness increases with the down sampling rate.

5. Conclusions

In this paper, we have proposed a new particle filtering-based visual ego-motion estimation framework especially robust to abrupt camera motion. We achieved the robustness to abrupt camera motion via multi-layered importance sampling based on our geometric PSO on SE(3). We experimentally demonstrated that our proposed framework yielded superior ego-motion estimation performance to the local linearization and UT-based importance functions in the case of abrupt motion. We also verified that the optimization performance of our geometric PSO was better than that of conventional vector space PSO. Moreover, we finally showed that our framework could run in real time to yield successful results for real sequences where severe abrupt camera motion is present.

References


Table 4

Table 4: Comparison of results for “New College” by PSO-VE with and without quantum particles. PSO and QSO represent the cases of without and with quantum particles, respectively. $N_{iter}$ and $N_{quant}$ are the average number of iterations per frame and average number of global best particle updates by quantum particles per frame.

<table>
<thead>
<tr>
<th></th>
<th>$N_{iter}$</th>
<th>Best fitness</th>
<th>Worst fitness</th>
<th>$N_{quant}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO(1×)</td>
<td>12.29597</td>
<td>−2.588798</td>
<td>−4.17150</td>
<td>0</td>
</tr>
<tr>
<td>QSO(1×)</td>
<td>12.51971</td>
<td>−2.604593</td>
<td>−3.33276</td>
<td>0.635540</td>
</tr>
<tr>
<td>PSO(2×)</td>
<td>16.17239</td>
<td>−3.127934</td>
<td>−5.46793</td>
<td>0</td>
</tr>
<tr>
<td>QSO(2×)</td>
<td>14.84872</td>
<td>−2.581024</td>
<td>−4.18913</td>
<td>2.086005</td>
</tr>
<tr>
<td>PSO(3×)</td>
<td>22.66101</td>
<td>−4.483217</td>
<td>−15.78914</td>
<td>0</td>
</tr>
<tr>
<td>QSO(3×)</td>
<td>19.23316</td>
<td>−3.267125</td>
<td>−12.51971</td>
<td>3.840407</td>
</tr>
</tbody>
</table>

Table 5

Table 5: Statistics of the camera motion abruptness for each real test sequence.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Motion</th>
<th>% of abrupt camera motion frames</th>
<th>Mean</th>
<th>Var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab</td>
<td>Position</td>
<td>0.28</td>
<td>0.1191</td>
<td>0.0004</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Office</td>
<td>Position</td>
<td>2.11</td>
<td>0.2187</td>
<td>0.0103</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>4.21</td>
<td>0.1356</td>
<td>0.0012</td>
</tr>
<tr>
<td>Campus I</td>
<td>Position</td>
<td>55.15</td>
<td>0.3055</td>
<td>0.0869</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>26.21</td>
<td>0.6447</td>
<td>2.0623</td>
</tr>
<tr>
<td>Campus II</td>
<td>Position</td>
<td>47.31</td>
<td>0.2376</td>
<td>0.0381</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>5.02</td>
<td>0.1542</td>
<td>0.0037</td>
</tr>
<tr>
<td>New College</td>
<td>Position</td>
<td>0.18</td>
<td>0.2180</td>
<td>0.0199</td>
</tr>
<tr>
<td>(×1)</td>
<td>Orientation</td>
<td>0.01</td>
<td>0.1904</td>
<td>0.0040</td>
</tr>
<tr>
<td>New College</td>
<td>Position</td>
<td>1.03</td>
<td>0.1550</td>
<td>0.0054</td>
</tr>
<tr>
<td>(×2)</td>
<td>Orientation</td>
<td>0.65</td>
<td>0.1234</td>
<td>0.0005</td>
</tr>
<tr>
<td>New College</td>
<td>Position</td>
<td>5.18</td>
<td>0.1937</td>
<td>0.2863</td>
</tr>
<tr>
<td>(×3)</td>
<td>Orientation</td>
<td>3.65</td>
<td>0.1457</td>
<td>0.0022</td>
</tr>
</tbody>
</table>


