



An improved iterative closest point algorithm based on the particle filter and K-means clustering for fine model matching

Ahmad Reza Saleh¹ · Hamid Reza Momeni¹

Accepted: 17 November 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

The rigid matching of two geometric clouds is vital in the computer vision and its intelligent applications, such as computational geometry, robotics, shape modelling, surface reconstruction and mapping, and many other fields. The variants of the iterative closest point algorithm were employed as the most noticeable matching algorithm. In traditional ICP algorithms applications for symmetrical geometry matching, the initial uncertainty and the multiple local minima of the distance function adversely affect the alignment process, which leads to weak performance, such as incorrect correspondence, narrow convergence region, and instability. In this study, the novel algorithm fused the ICP algorithm, particle filter and K-means clustering to correctly estimate the transformation ICP parameters. Further guide to initial values of parameters and their covariance obtained by k-means clustering. Then, a particle filter was implemented to estimate accurate values and perform global optimization. In the introduced PF-ICP algorithm, the alignment parameters: rotation angles, scale factor, and translation, were defined as particles elements optimized using a sequential importance resampling (SIR) particle filter. The proposed algorithm was implemented on a medical robot FPGA board and applied to “three symmetrical models” and “noisy and poor datasets.” The calculated variances and estimated parameters were compared with four modified ICP methods. The results show a significantly increasing accuracy and convergence region with an acceptable speed for the practical conditions.

Keywords Matching algorithm · Iterative closest point · Particle filter · Global optimization · K-means clustering · Measurement uncertainty

1 Introduction

Finding a rigid transformation to match a dataset (as measured/scanned data) with a model point set or map is a fundamental problem in computer vision, computational geometry, robotics, shape modelling/analysis, surface reconstruction and mapping, augmented reality and many other fields. 3D registration is based on variants of the iterative closest point (ICP) algorithm. ICP was presented by Besl and McKay in 1992 [1]. In addition, the iterative closest contour point algorithm (ICCP) was eventuated from the ICP method,

proposed by Kamgar-Parsi 1999 [2], to solve the gravity navigation problem.

The ICP starts with two meshes as model set and dataset with an initial setting for their relatively rigid-body transformation that repeatedly refines the transformation by obtaining pairs of corresponding points on the model and data. An incremental transformation is found, such that the point-to-point distance is minimized to a local minimum. Owing to the ICP algorithm that can be made efficient and reliable, it has widely considered and adopted in different applications. So, the researchers have focused on addressing the shortcomings and generated variant methods. Variant improvement of the ICP algorithm classifies aspects, such as convergence speed, stability, tolerance of noise or outlier, and maximum initial misalignment. Performance developments can be achieved by modifying the algorithm in three stages:

✉ Hamid Reza Momeni
Research552@gmail.com; momeni_h@modares.ac.ir
Ahmad Reza Saleh
ahmadreza_saleh@modares.ac.ir

¹ Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Jalal Ale Ahmad, P. O. Box: 14115-111, Tehran, Iran

1. Selecting samples of point sets in one or both meshes that may have imperfection data;

2. Corresponding the points to samples in the model, weighting pairs and rejecting outliers;
3. Defining an error metric due to the point pairs, estimating the rigid transformer, and minimizing the ICP covariance.

For each of these stages, there were different improvements in the performance and reliability of the algorithm. The first issue with ICP is the selection of an input sample points. The sample selection and matching sequence length affect the algorithm's performance. In some studies, all available points are used [1], while in others, some points are used [3] or uniform random sampling is used at each iteration [4]. Xiao et al. generated a matching sequence method based on the coding principle for an ICCP geomagnetic matching algorithm [5]. They presented an adaptive approach for choosing appropriate matching lengths and matching points in a highly dynamic environment. Also, about the analysis of the measurement data, Zhao et al. studied the geomagnetic map data analysis based on random error modelling and filtering [6].

The next stage is related to assigning weights to pairs or eliminating outlier pairs. This stage may significantly affect minimization and solve several heuristics to prune or reweight the correspondences. Besl and Kamgar used the closest point in another mesh as the corresponding point for each sample. The other ICP variants employ the closest compatible point, normal shooting, and normal shooting to a compatible point [7, 8], and their computation is accelerated using a k-d tree. Also, the data point projection to the model mesh from the model mesh point of view can be used. Figure 1a shows that the closest point matching algorithm can perform wrong pairing in the presence of noise and outliers, then it converges slowly. But Fig. 1b represents that the "projection" matching methods are less sensitive to the presence of noise.

Bouaziz et al. [9] proposed a formulation of the ICP algorithm that avoids difficulties related to dealing with outliers and incomplete data using sparsity-inducing norm optimization. Guo et al. [10] introduced an adaptive weight vector that eliminated the negative effect of pairs with the most significant registration errors and the retained pairs assigned suitable weights to suppress the noise and outliers.

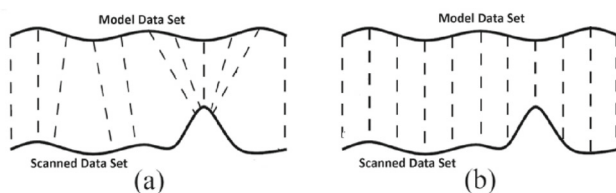


Fig. 1 The pairing of the two cloud data with **a** closest point matching algorithm and **b** projection matching method, in presence of noise and outlier data

The final step of the ICP algorithm, is to minimize the error metric cost function and the optimization approach to minimize the distance error. For example, Besl and McKay's original work minimized the distance between two correspondence points. In contrast, the second method minimizes the distance between a data point and the corresponding model point plane, which is perpendicular to its normal plane. Generally, distance point-to-plane minimization converges faster and with greater accuracy, but not necessarily a better region of convergence [11].

It has also been shown that with a trimmed square distance function, more computational capability, and auxiliary data, more convergence speed and stability could be achieved [12] or by cancelling the greater distance pairs [13]. In [14], a new symmetric distance metric presented to achieve a better convergence region and a higher speed. The research in [15] extended the point-to-hyperplane distance function and combined various position estimation algorithms to minimize different metric errors.

In addition, recent studies have attempted to accelerate the traditional ICP method. The Levenberg–Marquardt algorithm to minimize registration was employed to accelerate and yield a basin of convergence that directly minimized the energy function as a nonlinear optimization. In [16, 17], Anderson's acceleration was employed to speed up convergence. In addition, in considerable research, an established numerical technique based on Welsch's function was presented, which efficiently minimizes a simple quadratic function [18]. Yue et al. [19] completed point cloud registration through key point extraction, local point-pair feature matching, and coarse and fine registration. They verified the performance of the method by evaluating its RMSE and MAE. However, the uncertainty of the feature fitting affects the estimated parameters accuracy [20, 21].

Owing to the strong nonlinearity of the objective function of matching in some terrain types, linearization methods, such as the extended Kalman filter, are often not well suited for this problem, and the effectiveness of the probabilistic approach is slightly better [22].

Probabilistic matching algorithms obtain higher performance and more robust and global results with a higher computational cost. They are often based on techniques such as histograms, sequential Monte Carlo, and particle filters.

The key contribution of this work is the novel fusion of the particle filter and ICP algorithm to estimate registration parameters and to overcome the weak performance of traditional ICP method in the symmetrical geometry matching related to local minima of the cost function and sensor uncertainty.

The proposed approach defined a sequential Monte Carlo process that enabled importance sampling of each of the registration parameters according to the collected error statistics. The PF-ICP method implemented the particle filter algorithm

to estimate the global optimization and accurate values of the ICP transformation parameters. The presented PF-ICP algorithm employed the k-means clustering method to further guide the parameter initialization step and to obtain the ICP parameters covariance.

This article sections included: related works, required fundamentals (A. the ICP algorithm, B. the covariance of ICP, and C. the SIR particle filter), the introduced PF-ICP method, with four parts: A. definitions, B. initialization, C. covariance estimation, and D. particle filter estimation. Finally, the results of the two case studies for different conditions were explained and concluded.

2 Related works

Combesa and Primab [16] presented a registration method for nonlinear 3D point sets. They considered the points of the dataset as draws of the Gaussian mixture model in which the centres are the transformed points of the model set. In the next step, a maximum likelihood estimation of the model parameters is performed, using the expectation–maximization (EM) algorithm. Xiao et al. [23] adopted a probability data association (PDA) algorithm, based on considering constraints of a vehicle’s kinematics, to solve the problem of geomagnetic matching in an interference environment. Rowekamper et al. [24] analysed the accuracy of an integrated laser-based positioning system for mobile platforms. Also, Censi [25] and Maken et al. [26] introduced an algorithm to estimate the uncertainty of the ICP transformation parameters and to align two point clouds. Maken used a gradient-based optimization of the distance cost function to develop a Stein variational inference framework for multi-modal distribution models.

Because modern computer computational power warrants performance vs. complexity, we improved the ICP matching algorithm with a particle filter for the complexity model condition. In this paper, the introduced PF-ICP algorithm determined the transformation parameters (rotation, translation, and scaling) by k-means clustering initialization and sequential importance resampling (SIR) particle filter. The developed PF-ICP algorithm could answer very reliable and comprehensive for local minimums, caused by symmetric spatial and measurement noise.

3 Fundamentals

The fundamental algorithms required for the implementation of PF-ICP method are as follows:

3.1 The iterative closest point algorithm

The iterative closest point (ICP), or the iterative correspondence point algorithm, is an alignment method in which the two point sets are registered based on the function of either

point-to-point or point-to-plane distance. These two sets may be represented as point sets, triangle sets, line segment sets, parametric surfaces, implicit surfaces, implicit curves, and parametric curves [27–31]. The most common method is based on the iterative closest point algorithm introduced by [1]. In this method, the points in one cloud and their nearest points in another cloud are repeatedly paired as the corresponding points. Then, an incremental transformation that minimizes the distance of the point pair is calculated.

Let the model shape be represented as a set of points, $D = \{d_1, d_2, \dots, d_{N_D}\}$, and N_D be the number of points in the model database, and the scanned or scene shape be considered as a set of points, $M = \{m_1, m_2, \dots, m_{N_M}\}$. If the defined correspondents are the closest points, the relative transformation can be obtained as follows: For every point m_i , in the scanned shape $M = \{m_i\}$, $i = 1, \dots, N_M$, we seek the closest point model D based on using the Euclidean distance. For a given scanned point m_i and the model points set D , the Euclidean distance between m_i and D can be found as:

$$d(m_i, D) = \min_{k=1 \dots N_D} d(m_i, d_k) = \min_{k=1 \dots N_D} \|m_i - d_k\| \quad (1)$$

The closest point of the model set, $d_j \in D$, satisfies the equality:

$$d(m_i, d_j) = d(m_i, D); j = \arg \min_{k=1 \dots N_D} d(m_i, d_k);$$

Also, the closest point in the model set D that yields the minimum distance will be denoted by y_i .

$$d(m_i, y_i) = d(m_i, D)$$

where the m_i corresponds to y_i . Let Y define the resulting set of closest points and $C(\cdot)$ be the closest point operator.

$$Y = C(M, D) \quad (2)$$

$$Y = \{y_i\}_{i=1}^{N_M}, M = \{m_i\}_{i=1}^{N_M}, D = \{d_i\}_{i=1}^{N_D}, Y \subseteq D$$

The optimal registration parameters (scaling, rotation, and translation) can be found that match the scanned points M to the closest model points Y by minimizing the total error.

$$E = \sum_{i=1}^{N_M} \|e_i\|^2, e_i = \{y_i - [sR(m_i) + T]\} \quad (3)$$

The ICP algorithm outline is as follows:

1. Initialize registration parameters: rotation ($R, R \in \{\mathbb{R}^{3 \times 3} \mid R^T R = I, \det(R) = +1\}$), translation ($T = 0, T \in \mathbb{R}^3$), and scaling ($s = 1, s \in \mathbb{R}$) and registration error; Error = ∞ .

2. For each scanned point set, find the corresponding closest point in the model point set.
3. Calculate registration parameters for given point correspondents from step 2.
4. Apply the alignment to the scanned point set.
5. Calculate the registration error between the currently aligned scanned set and the model set.
6. If error > threshold, return to step 2; otherwise, return with new scanned points set (Fig. 2).

Algorithm 1: ICP_Matching

[Rotation, Translate, Scale, dist] =
 ICP_matching(D_database, M_Scanned)
 $k = 0, m_i^{(0)} = R^{(0)}m_i + t^{(0)}$
repeat
 $k = k + 1$
 $y_i^{(k-1)} = C(m_i^{(k-1)}, D)$
 $[R^*, T^*, s^*] = \arg \min_{R, T, s} \sum_{i=1}^N \|sRm_i^{(k-1)} + T - y_i^{(k-1)}\|^2$
 $m_i^{(k)} = sR^*m_i^{(k-1)} + T^*$
 $R^{(k)} = R^*R^{(k-1)}$
 $T^{(k)} = R^*T^{(k-1)} + T^*$
 $s^{(k)} = s^*s^{(k-1)}$
until convergence and Error < Threshold
return $R^{(k)}, T^{(k)}, s^{(k)}$

Four major estimating algorithms for 3D rigid-body transformations were reviewed and compared by [32]. These algorithms are used when there is an established correspondence between data cloud M and D. The optimal registration is obtained from linear algebra singular value decomposition (SVD) method. First, the scan point sets $\{d_i\}$ and the model point sets $\{m_i\}$ should have the same centroid in the rotation matrix calculation. The least-squares equation for the centroid datasets is:

$$\Sigma^2 = \sum_{i=1}^N \|d_{ci} - \hat{R}m_{ci}\|^2 = \sum_{i=1}^N (d_{ci}^T d_{ci} + m_{ci}^T m_{ci} - 2d_{ci}^T \hat{R}m_{ci}) \tag{4}$$

The minimum of this equation occurs when maximizing the *Trace*($\hat{R} \cdot C$), where C is a correlation matrix defined by:

$$C = \sum_{i=1}^N [m_i d_i^T] - \mu_m \mu_d^T = \sum_{i=1}^N m_{ci} d_{ci}^T \tag{5}$$

If the SVD of C is given by $C = U \Lambda V^T$, R must be $\hat{R} = VU^T$, for trace maximization. In addition, the optimal

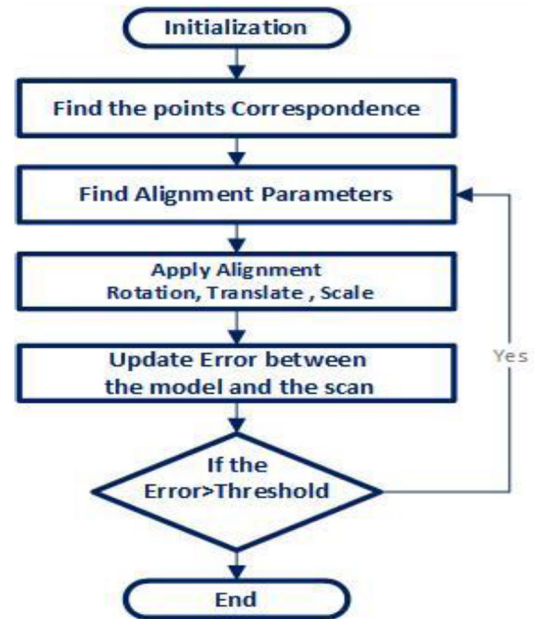


Fig. 2 The iterative closest point flowchart

translation was computed from the alignment of the D centroids with the rotated centroid of M. The algorithm-SVD matching is presented in the following:

Algorithm 2: SVD_Matching

[Rotation, Translate, Scale] =
 SVD_Matching({D_database}, {M_Scanned})

- Calculate the centroid of datasets M and D

$$d_{ci} = d_i - \mu_d, \quad \mu_d = \frac{1}{N_D} \sum_{i=1}^{N_D} d_i$$

$$m_{ci} = m_i - \mu_m, \quad \mu_m = \frac{1}{N_M} \sum_{i=1}^{N_M} m_i$$

- Compute the

$$C = \sum_{i=1}^N m_{ci} d_{ci}^T$$

- Obtain the rotation matrix R by SVD method:

$$C = U \Lambda V^T, \quad R = VU^T.$$

- The scale factor can be calculated by:

$$s = \sqrt{\frac{\sum_{i=1}^{N_M} d_{ci} R(m_{ci})}{\sum_{i=1}^{N_D} \|d_{ci}\|^2}} \approx \sqrt{\frac{\sum_{i=1}^{N_M} \|m_{ci}\|^2}{\sum_{i=1}^{N_D} \|d_{ci}\|^2}}$$

- the optimal translation aligns the centroids of the datasets

$$T = \mu_d - sR(\mu_m)$$

- **Return** s, T, R ;
-

Therefore, for the case of established correspondence between data cloud M and D, the above SVD matching method would be used for calculating the ICP parameters. At the initialization step of the presented approach, a correspondence relationship is considered between the centroids of two datasets clusters and will be executed the SVD matching algorithm [33].

When the probability distribution function of the ICP parameters is necessary, the covariance is often obtained based on the calculation of the Hessian Matrix [24, 34–36]. In this method, the Hessian and covariance matrices are obtained from:

$$H = \frac{dj^2(\hat{X})}{d\hat{X}^2} \tag{6}$$

$$\text{cov}(\hat{x}) \cong \left(\frac{\partial^2 J}{\partial x^2} \right)^{-1} \frac{\partial^2 J^T}{\partial z \partial x}, \text{cov}(z) \cong \frac{\partial^2 J^T}{\partial z \partial x} \left(\frac{\partial^2 J}{\partial x^2} \right)^{-1}$$

where x is the model parameter, z is the observation matrix, and $J(z, x)$ is the error function. The Hessian matrix can be obtained once the ICP algorithm determines the minimum [35]. For more details, we can see [37, 38].

3.2 The particle filter—sequential Monte Carlo estimation

The particle filter as a sequential state estimation method was presented in 1993 for Bayesian nonlinear filtering problems. The particle filter is also known as bootstrap filtering, sequential Monte Carlo algorithm [39]. To date, the particle filter is developing and evolving; thus, it has become a relatively mature theory. A summary of the particle filter algorithm framework is presented herein, and the sequential importance resampling (SIR) particle filter algorithm is defined [40, 41].

The particle filter steps consist of:

1. Select a proposal distribution $q(x_{k+1}|x_{1:k}, y_{k+1})$, resampling method, and the number of particles N .
2. Generate and initiate.
 $x_1^i \sim p_{x0}, i = 1, \dots, N$ and $w_{1|0}^i = 1/N$.
3. Measurement update: For:
 $i = 1, \dots, N, w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k|x_k^i)$
That the normalization weight coefficient c_k is calculated by:

$$c_k = \sum_{i=1}^N w_{k|k-1}^i p(y_k|x_k^i) \tag{7}$$

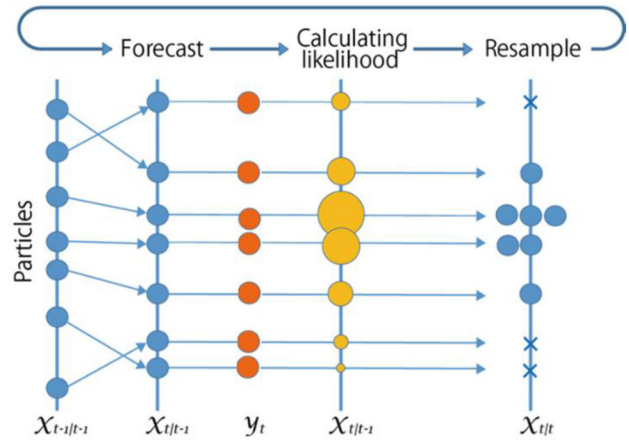


Fig. 3 Sequential importance resampling particle propagation

- (4) Estimation: The filtering probability is obtained by:

$$\hat{p}(x_{1:k}|y_{1:k}) = \sum_{i=1}^N w_{k|k}^i \delta(x_{1:k} - x_{1:k}^i) \tag{8}$$

That the mean could be calculated by:

$$\hat{x}_{1:k} \approx \sum_{i=1}^N w_{k|k}^i x_{1:k}^i. \tag{9}$$

- (5) Resampling:

At each arbitrary time, obtain N samples with super-sedence from the set $\{x_{1:k}^i\}_{i=1}^N$ with probability, $w_{k|k}^i = 1/N$.

- (6) Time update:

Produce predictions with the proposal distribution and correct for the importance weight.

$$x_{k+1}^i \sim q(x_{k+1}|x_k^i, y_{k+1});$$

$$w_{k+1|k}^i = w_{k|k}^i \frac{p(x_{k+1}^i|x_k^i)}{q(x_{k+1}^i|x_k^i, y_{k+1})}; \tag{10}$$

Also, the SIR particle filter algorithm is implemented as follows:

As shown in Fig. 3, the nonlinear filter infers states from observations in a likelihood or Bayesian framework. The state posterior distribution is computed or approximated using all available observations at that iteration. Then, the sample quantity and weight of particles are assigned with the same weight value and more quantity for more likelihood estimation.

Algorithm 3: SIR_Particle_Filter

```

[ $\{x_k^i, w_k^i\}_{i=1}^{N_s}$ ] = SIR_Particle_Filter( $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k$ )
  FOR i = 1: Ns
    Draw  $x_k^i \sim p(x_k^i | x_{k-1}^i)$ ;
    Obtain  $w_k^i = p(z_k | x_k^i)$ ;
  END FOR
  S = SUM [ $\{w_k^i\}_{i=1}^{N_s}$ ]; Sum of total weight
  FOR i = 1: Ns
    Normalization:  $w_k^i = S^{-1} w_k^i$ ;
  END
  Sequential Importance Resampling
  Initialize the CDF:  $c_1 = 0$ ;
  FOR i = 2: Ns
    Build CDF:  $c_i = c_{i-1} + w_k^i$ ;
  END
  Start i = 1 of the CDF:
  Draw initial point:  $u_1 \sim U[0, N_s^{-1}]$ ;
  FOR j = 1: Ns
    Movement along the CDF:
     $u_j = u_1 + N_s^{-1}(j - 1)$ ;
    WHILE  $u_j > c_i$     i = i + 1;  END
    Assignment of the sample:  $x_k^j = x_k^i$ ;
    Assignment of the weight:  $w_k^j = N_s^{-1}$ ;
    Assignment of the parent:  $i^j = i$ ;
  END FOR.
  RETURN {  $x_k, w_k$  }

```

3.3 The proposed particle filter–ICP algorithm

The proposed algorithm establishes a fine matching of two geometric data points with symmetric spatial distribution. Its advantage is the reliability of reaching the correct matching and the accuracy of the final result for a “difficult” geometry with a symmetric spatial distribution. The main idea of the algorithm is to represent the estimation belief by a probability distribution over the possible region for the ICP transformation parameters and update the estimation belief whenever each iteration is performed [40].

First, the algorithm estimates the initial value and uncertainty as a Gaussian distribution with a mean and variance. The initial values are required for the ICP algorithm. The probability distribution associated with data uncertainty is required for the particle filter. By applying sensor fusion techniques in nonlinear conditions, such as the particle filtering methods, almost a maximum likelihood estimation is

obtained by probabilistic filtering. These probabilities are typically assumed to be normally distributed, so only the mean and covariance matrix are calculated. The main methods introduced for ICP covariance calculation are based on the Hessian matrix method. The researches such as [35, 36, 38, 40] show that the ICP covariance depends not only on sensor noise characteristics but also on the geometry of the environment. Here, a statistical method that approximates the mean and variance of the ICP matching parameters is introduced.

In the second step of the proposed algorithm, a particle filter is defined to estimate the transformation ICP parameters (rotation angles, translation, and scaling) as particle elements. According to the presented PF-ICP algorithm, four strategic concepts and significant achievements were observed.

- (1) In the scanning process, the sensors always impose errors, such as noise, scale factor, bias, and drift, on the measurement data. These errors correspond to and are consistent with the ICP transfer parameters, and the effects of these errors on the parameters can be modelled in the same way.
- (2) Owing to the linear transformation in the regression process, the relative position constraints of the measurement points were held constant. The main advantage of the ICP algorithm is that this concept is also used here.
- (3) In linear equation solving related to determining ICP transformation parameters and their optimization, the existence of outliers measurement data and overlap model data does not pose a serious constraint.
- (4) A valuable property of the PF-ICP algorithm is that it can universally approximate the state probability distributions and escape from the local minimum. The particle filter algorithm arranges its computational particles in global regions with high probability, where things really matter.

Based on the aforementioned advantages, this method has high reliability and a very convincing performance in the presence of uncertainty and nonlinearity, which will be shown in the next sections.

3.4 Definitions

Consider two corresponding point sets $M = \{m_i, i = 1..N_M\}$ for the measurement point cloud and $D = \{d_i, i = 1..N_D\}$ database point cloud (could consider $N_M = N_D = N$) and the rigid transformation s, R, T between them, such that they are related by:

$$d_i = sRm_i + T + V_i \quad (11)$$

where R is known as a standard 3×3 rotation matrix, T contains a 3D translation vector, V_i is noise or uncertainty and the optimal parameters of transformation $[\hat{s}, \hat{R}, \hat{T}]$ that match the set $\{m_i\}$ onto $\{d_i\}$ typically are related to minimizing the least squares error function typically given by:

$$\Sigma^2 = \sum_{i=1}^N \|d_i - \hat{s} * \hat{R} * m_i - \hat{T}\| \tag{12}$$

3.5 Clustering and parameters initialization

The initialization is very important for the variations of ICP algorithms. The more correctly the ICP parameter initialization is performed, the fewer problems related to local minima arise. In order to reduce the mathematical calculation and increase the speed and integrity, the data clouds are first classified by the k-means clustering method. In the PF-ICP initialization, the k-means algorithm performs two-data cloud clustering. The initial values of the transformation parameters are then obtained by matching the centroids of the two data clusters. K numbers of sequential partitions or clusters are expressed as follows:

$$\begin{aligned} Clustering(M) &= \{M_1, M_2, \dots, M_K\}; Size(M_K) = n_{M_K}; \\ Clustering(D) &= \{D_1, D_2, \dots, D_K\}; Size(D_K) = n_{D_K}; \end{aligned} \tag{13}$$

Because of the scanning method, it is assumed that there is a generally (and not exactly) corresponding relation between the two sets, which leads to an established correspondence relationship between the centroids of the clustered subsets and:

$$D_i = Pair(M_i)$$

Based on the total number of cloud points, K was selected such that the number of cluster members could be between 10 and 100. The centroids of these clusters are employed for initializing the ICP transformation parameters. To initialize the ICP transformation parameters, the centroids of all measurement cloud M clusters and database cloud D clusters were calculated.

$$\begin{aligned} U_M &= \{u_{M1}, u_{M2}, u_{M3}, \dots, u_{MK}\}, \text{ that } u_{Mi} = Centroid(M_i) \\ U_D &= \{u_{D1}, u_{D2}, u_{D3}, \dots, u_{DK}\}, \text{ that } u_{Di} = Centroid(D_i) \end{aligned} \tag{14}$$

Algorithm 4: Clustering

```

[ {D}, {M}, {U_D}, {U_M} ] =
Clustering(D_Model, M_Scanned, K)
Get K, M, D.
{M_1, M_2, ... M_K} = Kmean_Clustering(M);
{D_1, D_2, ... D_K} = Kmean_Clustering(D);
{M_1, M_2, ... M_K} = Sequential_Sort(M_1, M_2, ... M_K);
{D_1, D_2, ... D_K} = Sequential_Sort(D_1, D_2, ... D_K);
FOR i = 1: K
    D_i = Pair( M_i );
END FOR
FOR i = 1: K
    u_{M_i} = Centroid(M_i);
    u_{D_i} = Centroid(D_i);
END FOR
U_M = {u_{M_i}}; U_D = {u_{D_i}};
Return [ {D}, {M}, {U_D}, {U_M} ];
    
```

The SVD matching algorithm calculates the initial estimate of the transformation parameters with centroids U_M and U_D inputs.

$$[S_{int}, R_{int}, T_{int}] = SVD_matching(U_D, U_M); \tag{15}$$

The results of R_{int} (initial rotation), T_{int} (initial translate), and S_{int} (initial scale factor) are used as the initial values of PF-ICP transformation parameters. In addition, the clustering could be executed several times, and the average of the centroids used for the initialization. The members of the generated clusters are employed for the estimation of the ICP parameters covariance and the calculation of the particle filter weights, which are presented in the following sections.

3.6 The PF-ICP parameters covariance estimation

As previously mentioned, to apply sensor fusion techniques, based on maximum likelihood estimation, the probability distribution of the error related to the applied data is required. Various techniques have been proposed to obtain this covariance, which are often based on the calculation of the Hessian matrix [25, 40]. These techniques require numerous mathematical calculations. To reduce these calculations, the k-means clustering and ICP matching were used to approximate the ICP covariance. The results were compared with those obtained using Censi's method.

For this purpose, n_C subsets of cloud points M are created such that their first member is randomly selected from M_1 , the first cluster of cloud points M , the second member of them is selected from M_2 , the second cluster of cloud points $M...$ and their K -th member is selected from M_k , the K -th

cluster of cloud points M . Thus, the n_C subsets of M with K members are built. All members of the subsets are selected from a uniform stochastic distribution. This can be rewritten as follows:

$$\begin{aligned} \mathbf{m}_1 &= \{m_{1,i} | m_{1,i} \in M_i, i : 1..K\} \\ \mathbf{m}_{n_C} &= \{m_{n_C,i} | m_{n_C,i} \in M_i, i : 1..K\} \end{aligned} \quad (16)$$

Then, for each subset \mathbf{m}_i and model dataset D , the ICP matching algorithm is executed to determine the ICP transformation parameters, the values of (R rotation matrix, T translate vector, S Scalar scale factor).

$$(R_i, T_i, S_i, dist_i) = ICP(\mathbf{m}_i, D), \text{ for } i : 1..n_C; \quad (17)$$

Parameter $dist$ is the value of distance function of \mathbf{m}_i and D . Thus by repeating these calculations for n_C subsets, the n_C estimations are obtained for transformation parameters, and an estimate of the mean and covariance of the transformation parameters can be calculated.

Algorithm 5: Approximate_ICP_Variance

```

[ $\sigma_\varphi, \sigma_\theta, \sigma_\psi, \sigma_T, \sigma_s$ ] =
Appx_ICP_Var( $D\_Model, \{M_i\_Clusters\}, n_C$ )
  { $M_i$ } = { $M_i\_Clusters$ };
  FOR j = 1:  $n_C$ 
    FOR i = 1:  $K$ 
       $m_{i,j} = \text{Rand\_Select}\{m_{i,j} | m_{i,j} \in M_i\}$ ;
    END FOR
     $\mathbf{m}_j = \{m_{i,j}\}$ ;
  END FOR
  FOR i = 1:  $n_C$ 
    [ $R_i, T_i, S_i, dist_i$ ] =  $ICP\_matching(\mathbf{m}_i, D)$ ;
    [ $\varphi_i, \theta_i, \psi_i$ ] =  $Transform(R_i)$  ;
  END FOR
   $\bar{\varphi} = \text{Mean}(\varphi_i), \sigma_\varphi = \text{Variance}(\varphi_i)$ ;
   $\bar{\theta} = \text{Mean}(\theta_i), \sigma_\theta = \text{Variance}(\theta_i)$ ;
   $\bar{\psi} = \text{Mean}(\psi_i), \sigma_\psi = \text{Variance}(\psi_i)$ ;
   $\bar{T} = \text{Mean}(T_i), \sigma_T = \text{Variance}(T_i)$ ;
   $\bar{s} = \text{Mean}(s_i), \sigma_s = \text{Variance}(s_i)$ ;

```

Return $\sigma_\varphi, \sigma_\theta, \sigma_\psi, \sigma_T, \sigma_s$;

3.7 The particle filter implementation

A particle filter can be implemented according to the values obtained for the probability distribution model related to the parameters of the ICP matching algorithm.

First, according to the 2D or 3D space defined for matching, a four- or seven-state particle filter can be defined. In the two-dimensional, the particles have a rotation angle and two translate variables, and a scale variable. Which is defined as:

$$\begin{aligned} x_i &= \{a_i, b_i, s_i, \theta_i\}; \\ T_i &= [a_i, b_i]; \quad Scale_i = s_i; \quad R_i = R(\theta_i); \end{aligned} \quad (18)$$

In the 3D space, the states have three translate variables in three directions and a scale variable and three rotation angles. Which is defined as follows:

$$\begin{aligned} x_i &= \{a_i, b_i, c_i, s_i, \varphi_i, \theta_i, \psi_i\}; \\ T_i &= [a_i, b_i, c_i]; \quad Scale_i = s_i; \quad R_i = R(\varphi_i, \theta_i, \psi_i); \end{aligned} \quad (19)$$

As mentioned before, in the particle filter, the more processing data, the greater computational volume required. In each particle filter iteration, the estimation algorithm propagates the particles forward to reach their optimum values and the error distance function is minimized. According to the initial values calculated in the previous section as well as the obtained variance values, a region for particle propagation can be determined (2sigma-standard deviation interval around the mean). The particles are then propagated as uniform random distribution in this region, the value of the distance function for each particle is calculated, and the weights of the particles are defined. This weighting process assumes a Gaussian function of the computed errors and is given by:

$$w_i = \frac{1}{\sqrt{2\pi Q}} \exp\left(-\frac{(D - s_i R_i M - T_i)^2}{2Q}\right) \quad (20)$$

```

i = 1 :  $n_P$ ; total_number_of_particles
 $w_i = \text{The\_Particles\_Weight}$ ;
 $Q = ICP\_variance$ ;
 $T_i = [a_i, b_i, c_i]; \quad Scale_i = s_i; \quad R_i = R(\varphi_i, \theta_i, \psi_i)$ ;

```

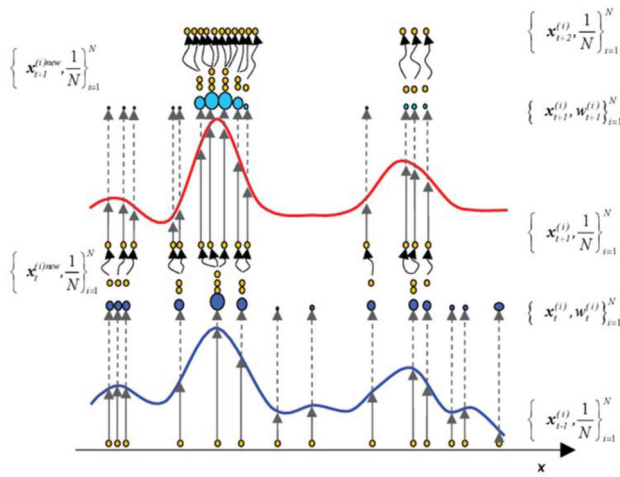



Fig. 4 The proposed PF-ICP algorithm: particle propagation, likelihood calculation, and resampling

After weighing the particles, their sum is normalized to one.

$$W_i = \frac{w_i}{\sum_{i=1}^{nSMC} w_i}; \tag{21}$$

The weight functions related to the particles and propagation regions are updated according to the uncertainty for each iteration, and new propagation regions are defined around the new means according to new standard deviation intervals. Then, the resampling step is performed based on the new propagation if necessary. For the resampling scheme, the sequential importance resampling (SIR) technique [42] was used here. The operation of SIR particle filter algorithm is demonstrated in Fig. 4.

The number of samples was adapted online, invoking large sample sets only when necessary to avoid the local minimum. As a result, the samples with low-importance are omitted, and the particles with high-importance weights are multiplied by the resampling step. When the PF-ICP sampling step was terminated, the resulting particles were used in the next computational step. The sample set size was selected based on the required accuracy and computational load capability, and it was shown that the variance of the importance sampler converges to zero at a rate of $1/\sqrt{N}$ [46]. The proposed PF-ICP matching algorithm flowchart is as follows (Fig. 5).

Algorithm 6: The proposed PF-ICP Algorithm

$[Rot, Trans, Scale, \{M_matched\}, w_best, dist] =$
 $SMC_ICP(D_Model, M_Scanned, K, n, Thrs)$

n_p : the number of Particles;
 $Thrs = [Thrs_max, Thrs_Avg]$;
 The threshold contains the acceptable maximum of error and the acceptable total average of error.

K : the number of clusters;
 $\{M_matched\}$: the transformed of scanned data;
 w_best : The weight of best particle;
 $dist$: the value of distance function.

1. Get Measurement data cloud and Database: M, D
2. Clustering

$$[\{D\}, \{U_D\}] = Clustering(D_Model, K);$$

$$[\{M\}, \{U_M\}] = Clustering(M_Scanned, K);$$

3. Initialization:

$$[s_{int}, R_{int}, T_{int}] = SVD_matching(U_D, U_M);$$

4. Define ICP_Parameters as Particles

$$x_k = \{a_k, b_k, c_k, s_k, \varphi_k, \theta_k, \psi_k\};$$

$$T_k = [a_k, b_k, c_k]; Scale_k = s_k; R_k = R(\varphi_k, \theta_k, \psi_k);$$

$$[a_k, b_k, c_k, s_k, \varphi_k, \theta_k, \psi_k] = Transform(R_k, T_k, Scale_k);$$

5. Covariance Estimation

$$[Q] = Appx_ICP_Var(D_Model, \{M_i\}, n_C)$$

6. Particle Filter and Resampling

$$w_i = \frac{1}{\sqrt{2\pi Q}} \exp\left(-\frac{(D - s_i R_i M - T_i)^2}{2Q}\right)$$

$$W_i = w_i / \sum_{i=1}^{nSMC} w_i;$$

$$[\{x_k^i, w_k^i\}_{i=1}^{N_i}] =$$

$SIR_Particle_Filter(\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_i}, z_k);$

7. Estimate Parameters and Error Distance Calculation

$$x_{1:k} \approx \sum_{i=1}^N w_k^i x_{1:k}^i;$$

$$dist = \sum_{i=1}^N \|d_i - s(x) \cdot \mathbf{R}(x) \cdot m_i - \mathbf{T}(x)\|$$

8. If $(Avg_dist < Thrs_Avg \text{ and } Max_dist < Thrs_max)$ or the maximum iteration;

else return to (3).

Return $\hat{\mathbf{R}}^{(i)}, \hat{\mathbf{T}}^{(i)}, \hat{\mathbf{s}}^{(i)}, \mathbf{w}^{(i)}, dist^{(i)}$

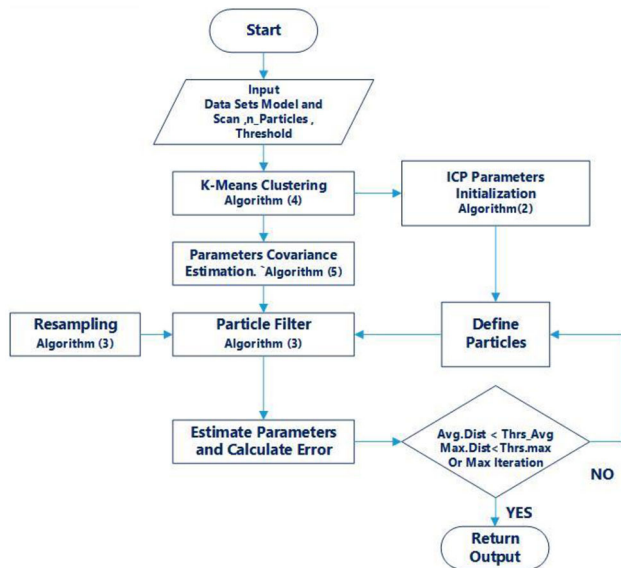


Fig. 5 The proposed PF-ICP matching algorithm chart

4 Experiments and results

In this section, the capabilities of the proposed method to reach the global optimum value of the ICP parameters are shown by using three models with complexity and highly symmetrical spatial. Under different conditions, the PF-ICP algorithm determines the optimal matching parameters. The first model dataset consisted of three-dimensional continuous data strings, and the other model contained a three-dimensional data surface and model-3 as real 3D scan. The use of the synthetic model examples provides the possibility to artificially the maximum symmetry creation. These examples ensure the performance of the algorithm in complex conditions and obtain the necessary assurance for implementation and utilization steps.

The PF-ICP algorithm was implemented on an XC6SLX45-2FGG484C FPGA, which is employed in laboratory medical-surgery robots, in Fig. 6. The model map was recorded in the FPGA, and the scanned data was generated or simulated by MATLAB and then inputted with a PC USB port. The FPGA board executed the matching algorithm to compensation positioning error and the result sent back to the PC.

As shown in Fig. 7, model-1 datasets are contained continuous data strings in three-dimensional space, and the performance of the presented algorithm is evaluated for sequence strings of data sample matching. In this model, the string of samples is considered similar to a spiral because the traditional ICP algorithms suffer from the local minimum in the objective function minimizing these shape types and do not obtain the correct answer that is discussed.

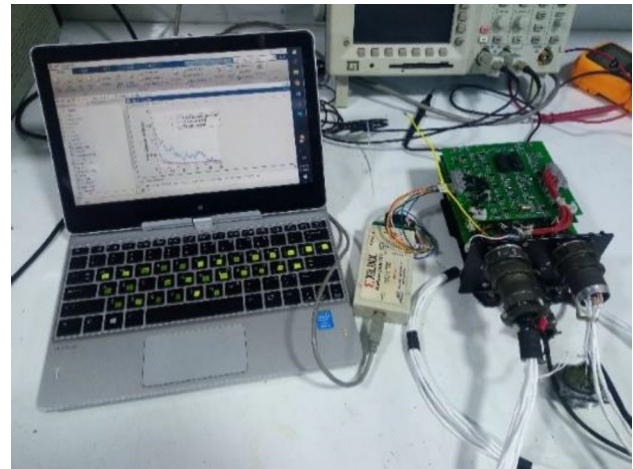


Fig. 6 The implemented algorithm on the lab surgery robot FPGA board



Fig. 7 The 3D Continuous Strings Symmetrical model-1 dataset:

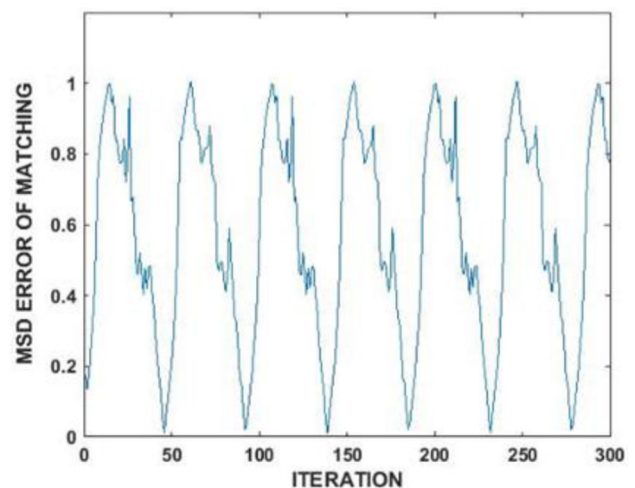


Fig. 8 The MSD value of the distance function related to the ICP algorithm output of model-1

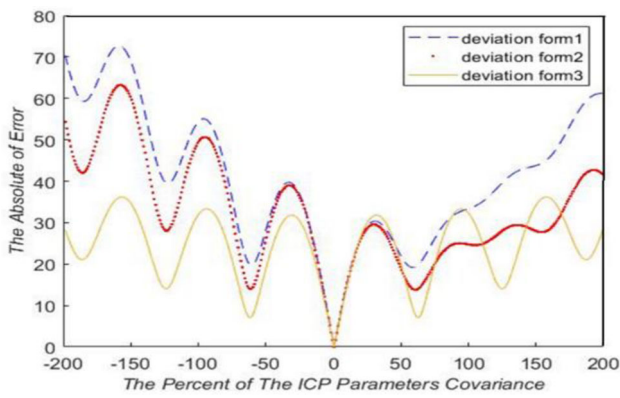


Fig. 9 The absolute of distance between the origin model-1 and its deviated point sets scan-1 respect to the percent of parameters standard deviation (σ), Form1: The parameters φ , T_z are drifted, Form2: The parameters ψ , φ , T_x , Form3: The ψ They are drifted

The following are the results of a study of the model-1 matching, instead of converging toward the optimal parameters and an error function of less than 0.001, the traditional ICP algorithm fluctuated and failed to find the overall minimum (Fig. 8).

To further explain this matter and investigate how to form a local minimum, a geometric distance function is drawn based on different drift values of the spatial transformation parameters. In model-1, the distance function is drawn for the deviation of the Z-axis rotation φ , T_z spatial transformation parameters, form-1 in Fig. 9. Also, in Fig. 9 form-3, the objective function of the matching error exhibits similar behaviour for the various drifts of the ψ parameters.

As shown, the local minimum is created at some points, which most optimization methods, such as gradient descent-based methods, mistakenly present these points as the solution to the problem, whereas the correct answer is far away from it. With the same aim and similarly, if the data of scan-1 is regenerated with several known values of the transformation parameters, then the traditional ICP method is applied to match the scan and the model. The value of the real distance between the matched datasets and the main model datasets are plotted in terms of input transformation parameter drift values, in Fig. 10.

As shown in Fig. 10, owing to the complexity and geometric symmetry of the model data, the ICP algorithm cannot calculate the values of the transformation parameters correctly and the MSE of the spatial distance is not less than the threshold of acceptable error value. Note that this case study uses the same points for the model set and the scan set; then, correct matching occurs when the distance function or error is zero. Therefore, when the error is not zero, the ICP algorithm is placed at local minimum and the estimated parameters values are incorrect.

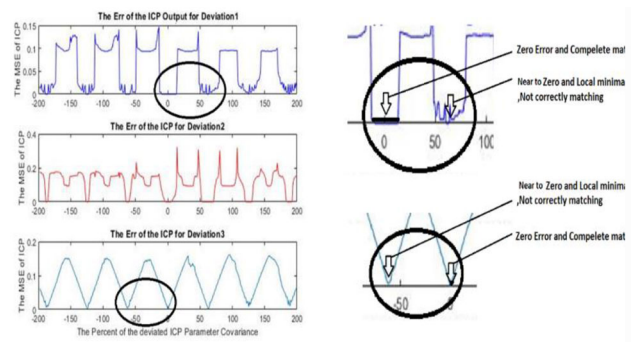


Fig. 10 The MSE of ICP matching algorithm that it is applied for the three deviation of Fig. 7 for the model-1, respect to percent of parameters standard deviation

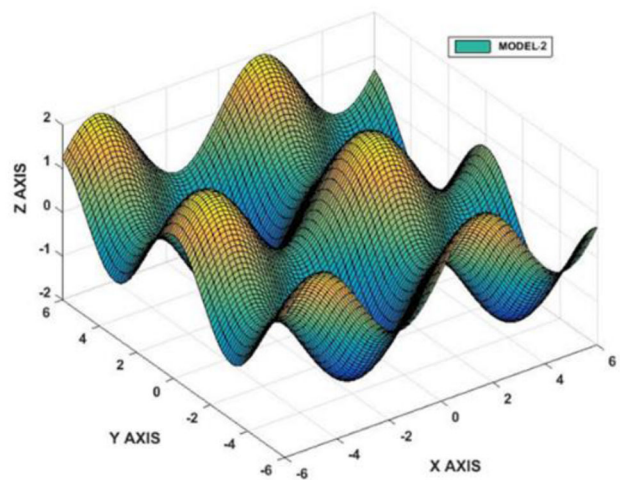


Fig. 11 The model-2 datasets, the 3D mesh of surfaces;

In the other study model, that applicable in geophysics, geomagnetic studies and earth geometry, the model dataset is the 3D mesh of surfaces. Owing to existing symmetries and local similarities, if the proper initial values are not used, the traditional ICP algorithms will make a mistake in the distance function minimization and will have a local minimum and wrong results will be obtained (Fig. 11).

For model-2, the behaviour of the absolute distance function and the performance of the ICP algorithm can be investigated against different drifts of displacements and rotations (Fig. 12).

As shown, owing to the complexity and geometric symmetry in the model-2, the ICP algorithm can only calculate the values of the transformation parameters in narrow range of parameter deviations, and it is often placed in the local minimum where the distance value is very close to the global minimum value.

According to these explanations, the presented PF-ICP method was implemented for these two models and its results

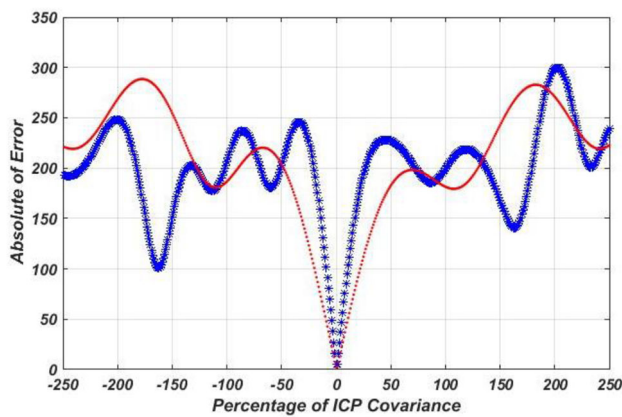


Fig. 12 The absolute of distance between the origin model-2 and its deviated point sets scan-2 respect to the percent of parameters standard deviation(σ), Thin Line: The parameters θ are drifted, Thick Line: The θ, ψ, s parameters are drifted

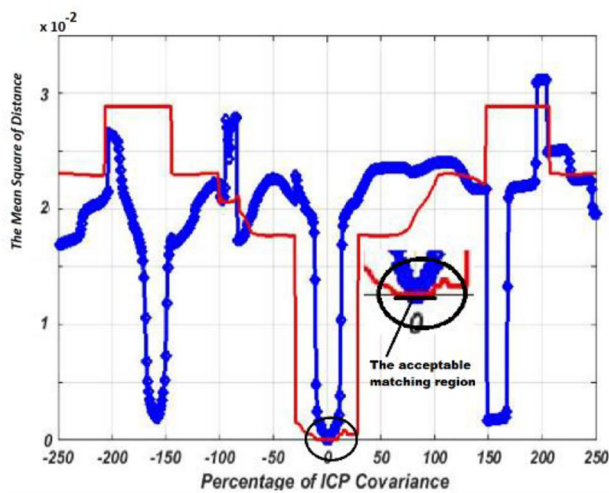


Fig. 13 The MSE of ICP matching algorithm that is applied for the two deviation (the drifted model-2), respect to percent of parameters standard deviation (σ)

were compared with those of the two improved ICP methods. We use these algorithm as state-of-the-art and reliable method in our research studies of shape matching. These ICP methods were improved with Welsch's robust criterion functions [43, 44] and Cauchy's function [43] and [45] by Bergström. In addition, to further evaluate the performance of the proposed method under different conditions, the proposed algorithm was implemented separately under thin-scanted and noisy measurement data conditions.

The scanned datasets of the models are shown in Fig. 13, and the values of the rotation, translate, and scale parameters related to the spatial transformation were chosen according to the covariance values of the ICP parameters in such a way that the condition of being in the local minimum exists. These parameters values are (Table 1; Fig. 14):

Table 1 The value of transformation parameters

	Translate [T_x, T_y, T_z]	Rotation(deg.) [φ, θ, ψ]	Scale
Scan-1	$[-0.9928, -0.5570, -0.4844]$	$[-29.4659, -18.6011, -29.4659]$	0.7143
Scan-2	$[-2.2487, -1.6495, -3.3125]$	$[5.2258, -16.4071, -17.9510]$	0.8333

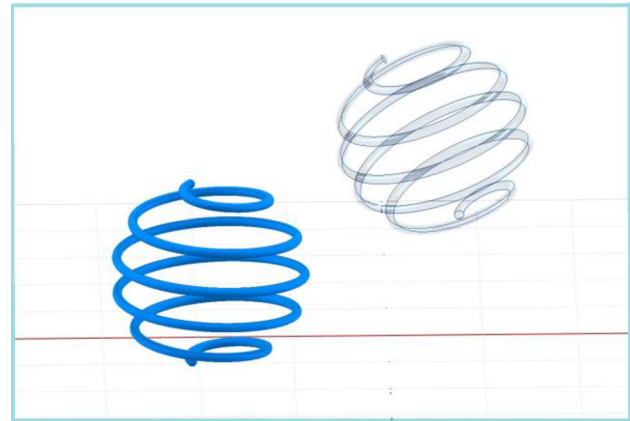


Fig. 14 The 3D scan and the 3D model-1

The following results of the present PF-ICP method and two other improved ICP methods for the matching parameters are provided.

As shown in Table 2, the initial values and variance values of the parameters have been calculated for two clustering $k = 10, k = 100$. The results are very satisfactory. The estimated initial values are very close to the target value, and the maximum difference and the largest variance are around the z -axis, because the maximum symmetry is around this axis.

The calculated values of the variances are also smaller than the Hessian method values, which provide better conditions for the filtering stage. Also, as we expected, the actual values of the parameters are in the range of one sigma around the estimated initial value.

In Table 3, as in the previous table, the values of clustering $k = 100$ have acceptable answers and close to target values. In addition, as we expected, the actual values of the parameters are in the range of one sigma around the estimated initial value. The calculated variance values are smaller compared to the Hessian method values, too.

Table 4 presents and compares the three different algorithms results for model-1 and scan-1 matching. Also, in the last column, the results of the proposed PF-ICP algorithm have been compared with the results of two variants of ICP and shown that the matching performance is better

Table 2 The result of the ICP initialization algorithm and covariance approximation, model-1

Model-1 ICP parameter	Hessian covariance	Clustering with k = 10		Clustering with k = 100		Calculated variance with k = 100 vs Hessian covariance ratio
		Average value	Variance	Average value	Variance	
φ (deg)	1.745	- 28.596	5.652	- 29.343	1.410	0.81
θ (deg)	2.342	- 16.789	5.577	- 18.735	1.456	0.62
ψ (deg)	17.45	- 31.168	36.67	- 26.103	17.05	0.97
tx	0.335	- 0.8370	0.409	- 0.7717	0.389	1.16
ty	0.733	- 0.4091	0.560	- 0.5091	0.537	0.73
tz	0.055	- 0.4803	0.116	- 0.4835	0.030	0.54
$scale$	0.032	0.7191	0.022	0.7124	0.011	0.34

Table 3 The result of the ICP initialization algorithm and covariance approximation, model-2

Model-2 ICP parameter	Hessian covariance	Clustering with k = 10		Clustering with k = 100		Calculated variance with k = 100 vs Hessian covariance RATIO
		Average value	Variance	Average value	Variance	
φ (deg)	8.431	- 37.060	83.27	- 11.463	7.175	0.85
θ (deg)	2.873	- 6.703	23.02	- 9.770	2.59	0.90
ψ (deg)	24.62	- 63.950	80.95	- 56.325	22.31	0.90
tx	2.441	- 0.3835	2.17	- 1.055	2.09	0.85
ty	0.332	- 1.6788	0.33	- 1.694	0.29	0.87
tz	2.754	- 1.3646	3.02	- 2.005	2.54	0.92
$scale$	0.025	0.8327	0.02	0.8320	0.02	0.99

Table 4 The result of the proposed PF-ICP algorithm and modified ICP based on Welsch function and Cauchy function method for scanned data of model-1

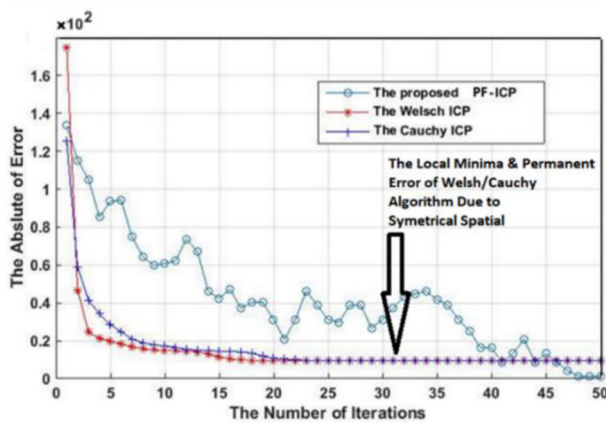
Model-1 ICP parameter	Proposed PF-ICP initiated with K = 100 Clusters and n = 50 particles	Modified ICP based on Welsch function	Modified ICP based on Cauchy function	True value	PF-ICP performance vs. two variants ICP
φ (deg)	- 29.473	- 29.49	- 29.49	- 29.47	Same
θ (deg)	- 18.624	- 18.71	- 18.71	- 18.61	Better
ψ (deg)	- 30.176	7.85	8.05	- 29.47	Very better
tx	- 0.9818	- 0.438	- 0.435	- -0.993	Very better
ty	- 0.5133	- 1.035	- 1.04	- 0.557	Very better
tz	- 0.4865	- 0.472	- 0.472	- 0.484	Better
$scale$	0.7144	0.714	0.714	0.714	Same
MSD error	0.00313	0.0180	0.0190		
No of iteration	50	50	50		

and it much better estimated three parameters. In Table 5, the results of two variants ICP algorithms and the PF-ICP for model-2 and scan-2 are presented and compared. Also, in the last column, the results of the proposed algorithm have been compared with the results of two types of ICP algorithm and shown that it better estimated four parameters.

In the previous methods, the distance function approaches the nearest local minimum point and reaches a local minimum in less number of iterations, and based on that, it presents the ICP parameters that are far from their correct values. The presented PF-ICP method performs more iterations and spends more computations and achieves a much lower

Table 5 The result of the proposed PF-ICP algorithm and modified ICP based on Welsch function and Cauchy function method for scanned data of model-2

Model-2 ICP parameter	Proposed PF-ICP initiated with K = 10 Clusters and n = 50 particles	Modified ICP based on Welsch function	Modified ICP based on Cauchy function	True value	PF-ICP performance vs. modified ICP
$\varphi(deg)$	5.27	4.59	5.58	5.23	Better
$\theta(deg)$	- 16.51	- 13.91	- 16.37	- 16.41	Better
$\psi(deg)$	- 17.36	- 17.55	- 17.44	- 17.95	Same
tx	- 2.195	- 2.848	- 2.195	- 2.249	Same
ty	- 1.643	- 1.967	- 1.628	- 1.649	Better
tz	- 3.353	- 3.751	- 3.337	- 3.312	Better
$scale$	0.829	0.837	0.831	0.833	Same
MSD error	1.144×10^{-3}	0.0199	0.0202		
No of iteration	50	50	50		

**Fig. 15** The value of error function of model-2 for each iteration. **a** The proposed PF-ICP **b** Modified ICP- Welsch function **c** Modified ICP Cauchy function

value of the distance function, which includes the global minimum.

Finally, the parameter values are more correctly and close to the target values are obtained, than the previous methods. The value of error function of model-2 for PF-ICP algorithm and two other variants of ICP algorithm is shown in Fig. 15.

According to the mentioned reference [46], the variance of the importance sampler should converge at a rate of about $1/\sqrt{50}$. It reaches an error of less than 0.01 after about 40 iterations. However, since the ICP algorithm is executed in each iteration and the variance of the parameters and the area of the particles propagation can change based on the spatial environment symmetry, and the number of iterations increased to about 50.

The CPU execution times of the seven execution for presented examples are between 96 and 124 min with Intel Corei3 5Gen CPU and reach to acceptable accuracy (less

than 0.01). In Fig. 16, the graphical results of PF-ICP algorithm and two other variants of ICP algorithm are compared. The effectiveness of the proposed PF-ICP algorithm, correct matching, and its accurate performance are shown.

4.1 Experiment with noisy measurement data

Owing to the presence of noise in the measurement sensors, there is often an uncertainty in the scan data. To better evaluate the performance of the PF-ICP algorithm, noise with stochastic uniform distribution and between the (0, 0.1) range was added to the scan data, as shown in Fig. 17, and the results obtained for the transformation parameters are given below and compared with sparse ICP method [9]. The Sparse ICP is proposed to dealing with outliers and incomplete data and to solve the sensitivity of registration to outliers and missing data often observed in 3D scans (Tables 6 and 7).

When the measurement noise is considered for scanned data, the average error increases, which is an acceptable value. In the last column, the absolute errors between ICP parameters and true value are indicated. As shown, the estimated parameter absolute errors are in acceptable range and this indicates the high capability of the presented algorithm in noisy conditions. Therefore, in this sparse condition, the total MSD distance function remains around 0.035, which could not be achieved with the previous ICP matching algorithm in non-sparse condition.

4.2 Thin measurement datasets 3D scan real experiment

The scanned data as the available data for the model under consideration often do not have the same density and compression in all regions, and even in some places, the amount

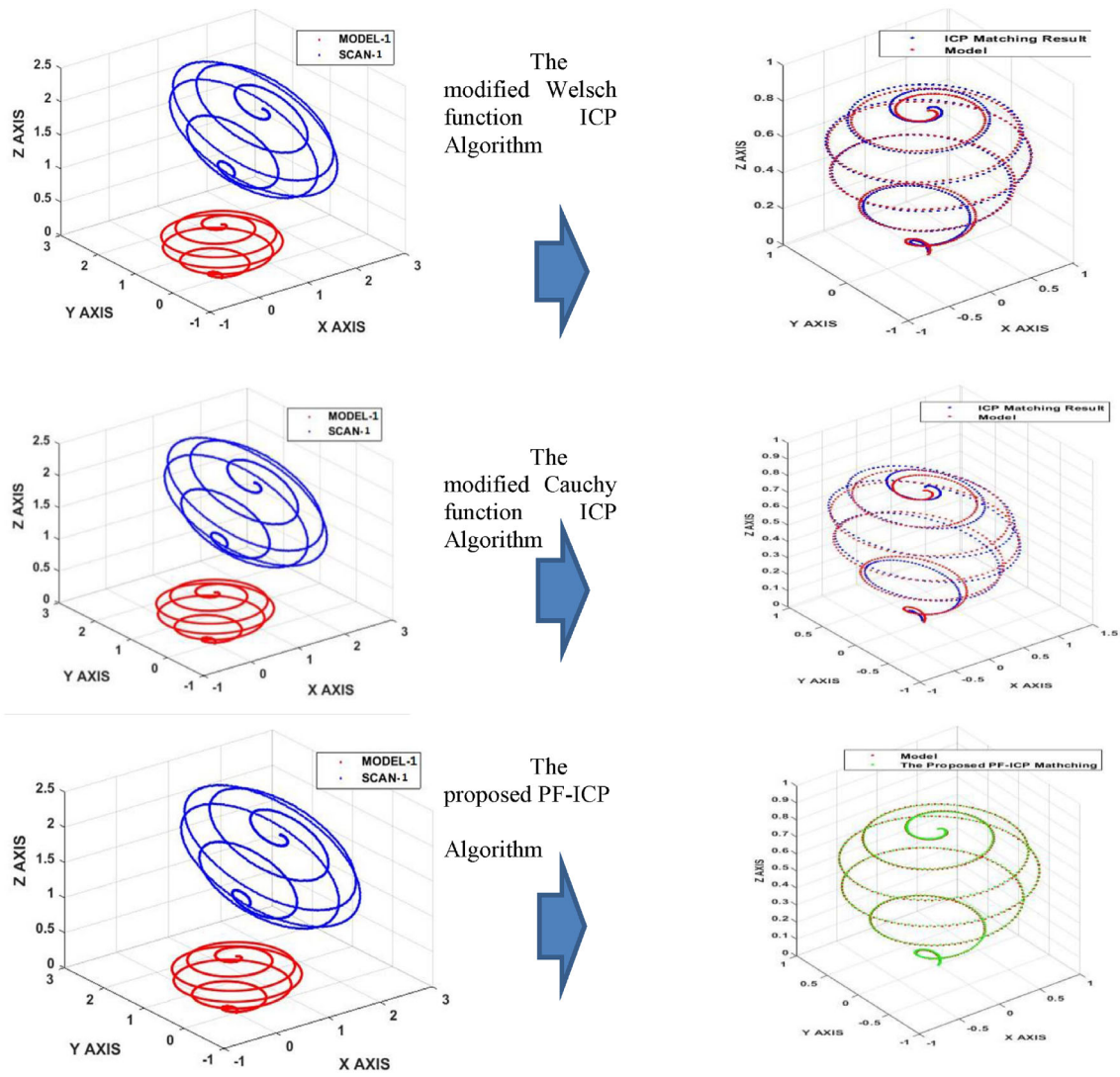


Fig. 16 The effectiveness of the proposed PF-ICP algorithm related to other variant ICP algorithm

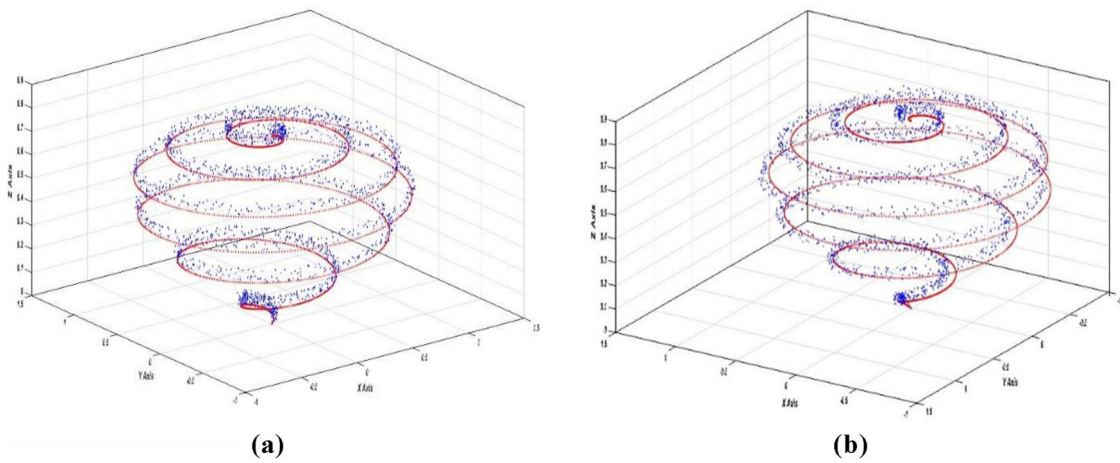


Fig. 17 The noisy scanned data points for model-1: (a) PF-ICP matching algorithm (b) Spars ICP matching algorithm

Table 6 The result of the PF-ICP algorithm for noisy scanned data of model-1

Model-1 ICP parameters	Proposed PF-ICP initiated with K = 100 clusters and estimated with n = 50 particles	Sparse ICP algorithm	True value	PF-ICP performance vs. sparse ICP
$\varphi(deg)$	- 28.938	- 11.2	- 29.47	Better
$\theta(deg)$	- 19.024	- 16.8	- 18.61	Same
$\psi(deg)$	- 30.061	- 6.4	- 29.47	Very better
tx	- 0.9381	- 0.955	- 0.993	Same
ty	- 0.5313	- 0.535	- 0.557	Same
tz	- 0.4685	- 0.471	- 0.484	Same
$scale$	0.7151	0.715	0.714	Same
Total MSD distance function	0.03586	0.1187		
No of iteration	50	50		

Table 7 The result of the PF-ICP algorithm for thin scanned data of model-3

Model-3 ICP parameters	Proposed PF-ICP initiated with K = 200 clusters and estimated with n = 80 particles	EM-ICP algorithm with variance 0.1 mm	True value	PF-ICP performance vs. EM-ICP
$\varphi(deg)$	5.827	7.13	5.23	Better
$\theta(deg)$	- 15.91	- 14.8	- 16.41	Better
$\psi(deg)$	- 19.16	11.75	- 17.95	Very better
tx	- 2.192	- 2.297	- 2.249	Same
ty	- 1.594	- 1.558	- 1.649	Same
tz	- 3.283	- 3.056	- 3.312	Better
$scale$	0.837	0.841	0.833	Same
Total MSD Distance Function	5.314×10^{-3}	0.0845		
No of Iteration	100	100		

of given and available data is minimal and the database distribution is thin. This phenomenon causes a local minimum for the objective distance function such that low-density points lose their effect on the distance function. In the traditional ICP algorithm, the transformation parameters values are miscalculated. The presented PF-ICP algorithm can estimate the transformation parameters related to the global minima in this condition with acceptable accuracy. In this experiment, the symmetric bottle is defined as real model-3, in Fig. 18. The result is compared with the EM-ICP method [16]. The EM-ICP uses the likelihood, the EM (expectation–maximization) algorithm and rigid-body transformations used for large data registration with better computation time and a little carelessness toward finding the best matching and reaching to the global minimum with respect to considered variance.

Therefore, in cases, that there is large data and the accuracy is negligible; the EM-ICP method is used due to less processing time. In addition, in cases with high symmetry

**Fig. 18** Real 3D scan of symmetrical model

and the presence of a complex local minimum in the distance function, the accuracy and correctness of the PF-ICP method is better and more reliable (Fig. 19).

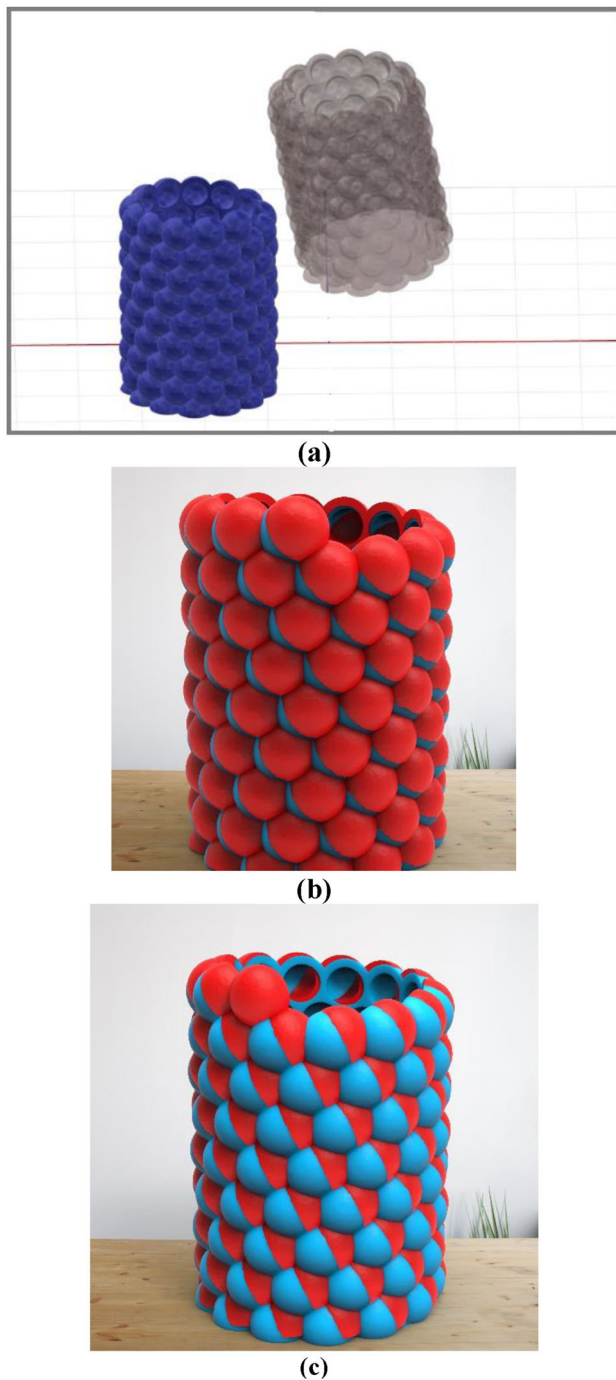


Fig. 19 Thin measurement datasets of real experiment 3D scan. **a** Model data and scanned data **b** PF-ICP Matching algorithm **c** EM-ICP matching algorithm (model is blue, and scan is red)

5 Conclusion

The proposed PF-ICP algorithm is used for computer vision computational geometry, medical robotics positioning, and other intelligent applications. It fuses the ICP algorithm and

the particle filter for fine 3D model matching with local minimum and uncertainty conditions, which leads to overcome the weak performance, such as a narrow convergence region, incorrect correspondence, and instability.

In this method, the initial values and covariance approximation of the ICP parameters were determined using k-means clustering. Then, the particle filter algorithm implemented to obtain the global minimum of the ICP distance function and accurate values of the registration parameters in the presence of complicated local minima and symmetrical geometry and nonlinear conditions. The experimental results of improved PF-ICP matching algorithm show that the PF-ICP yields convergence reliability and correct correspondence. In addition, the presented algorithm performed with insufficient available and sparse measurement data and measurement noise with acceptable accuracy. The performance of the PF-ICP method has been evaluated and proven by comparison with four modified ICP methods.

Authors' contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by ARS and HRM. The first draft of the manuscript was written by ARS, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The data that support the findings of this study are available from the corresponding author upon request. Mail to: research552@gmail.com, momeni_h@modares.ac.ir, ahmadreza_saleh@modres.ac.ir.

Declarations

Conflict of interest The authors whose names are listed immediately below of the title of manuscript certify that they have no relevant financial or non-financial interests to disclose. They have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent licensing arrangements) or non-financial interest (such as personal or professional relationships, affiliations, knowledge, or beliefs) in the subject matter or materials discussed in this manuscript.

Ethics approval The authors affirm that this manuscript does not involve human or animal research subject.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent to publish The authors affirm that this manuscript has not human research participants to provide informed consent for publication.

References:

1. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Sensor fusion IV: control paradigms and data structures, vol. 1611. Spie (1992)
2. Kamgar-Parsi, B., Kamgar-Parsi, B.: Vehicle localization on gravity maps. In: Unmanned Ground Vehicle Technology. International Society for Optics and Photonics, vol. 3693 (1999)
3. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the SIGGRAPH (1994)
4. Masuda, T., Sakaue, K., Yokoya, N.: Registration and Integration of Multiple Range Images for 3-D Model Construction. In: Proceedings of 13th international conference on pattern recognition (CVPR) (1996)
5. Xiao, J., Duan, X., Qi, X.: An adaptive Δ M-ICCP geomagnetic matching algorithm. *J. Navig.* **71**, 649–663 (2018). <https://doi.org/10.1017/S0373463317000844>
6. Zhao, J., Sun, X., Li, M., Zhang, Y.: Random error modeling and compensation of geomagnetic map data. In: 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), pp. 70–73 (2021). <https://doi.org/10.1109/ICPECA51329.2021.9362557>.
7. Meng, Y., Zhang, H.: Registration of point clouds using sample-sphere and adaptive distance restriction. *Vis. Comput.* **27**, 543–553 (2011). <https://doi.org/10.1007/s00371-011-0580-0>
8. Nuchter, A., Lingemann, K., Hertzberg, J.: Cached kd tree search for ICP algorithms. In: Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007). IEEE (2007)
9. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. In: Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing (SGP'13). Eurographics Association, Goslar, DEU, pp. 113–123 (2013). <https://doi.org/10.1111/cgf.12178>
10. Guo, Y., Zhao, L., Shi, Y., Zhang, X., Du, S., Wang, F.: Adaptive weighted robust iterative closest point. *Neurocomputing* **508**, 225–241 (2022). <https://doi.org/10.1016/j.neucom.2022.08.047>
11. Ketty, F., Muriel, P., Eric, M., Luce, M.: Plane-based Accurate Registration of Real-world Point Clouds. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2018–2023 (2021).
12. Dong, J., Peng, Y., Ying, S., Hu, Z.: LieTriCP: an improvement of trimmed iterative closest point algorithm. *Neurocomputing* **140**, 67–76 (2014)
13. Wang, X., Li, Y., Peng, Y., Ying, S.: A coarse-to-fine generalized-ICP algorithm with trimmed strategy. *IEEE Access* **8**, 40692–40703 (2020). <https://doi.org/10.1109/ACCESS.2020.2976132>
14. Li, J., Hu, Q., Zhang, Y., Ai, M.: Robust symmetric iterative closest point. *ISPRS J. Photogramm. Remote Sens.* **185**, 219–231 (2022). <https://doi.org/10.1016/j.isprsjprs.2022.01.019>
15. Ireta Muñoz, F.I., Comport, A.I.: Point-to-hyperplane ICP: fusing different metric measurements for pose estimation. *Adv. Robot.* **32**(4), 161–175 (2018)
16. Combès, B., Prima, S.: An efficient EM-ICP algorithm for non-linear registration of large 3D point sets. *Comput. Vis. Image Underst.* **191**, 102854 (2020). <https://doi.org/10.1016/j.cviu.2019.102854>
17. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* **49**, 1715–1735 (2011). <https://doi.org/10.1137/10078356X>
18. Zhang, J., Yao, Y., Deng, B.: Fast and robust iterative closest point. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 3450–3466 (2022). <https://doi.org/10.1109/TPAMI.2021.3054619>
19. Yue, X., et al.: Coarse-fine point cloud registration based on local point-pair features and the iterative closest point algorithm. *Appl. Intell.* **52**(11), 12569–12583 (2022)
20. Li, M., Zhang, M., Niu, D., et al.: Point set registration based on feature point constraints. *Vis. Comput.* **36**, 1725–1738 (2020). <https://doi.org/10.1007/s00371-019-01771>
21. Basdogan, C., Oztireli, A.: A new feature-based method for robust and efficient rigid-body registration of overlapping point clouds. *Vis. Comput.* **24**, 679–688 (2008). <https://doi.org/10.1007/s00371-008-0248-6>
22. André, M., Lionel, L.: On-manifold probabilistic Iterative Closest Point: application to underwater karst exploration. *Int. J. Robot. Res.* **41**, 875–902 (2022). <https://doi.org/10.1177/02783649221101418>
23. Xiao, J., Duan, X., Qi, X., Liu, Y.: An improved ICP matching algorithm for use in an interference environment during geomagnetic navigation. *J. Navig.* **73**, 56–74 (2020). <https://doi.org/10.1017/S0373463319000535>
24. R'owek'amper, J., Sprunk, C., Tipaldi, G.D., Stachniss, C., Pfaff, P., Burgard, W.: On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In: International Conference on Intelligent Robots and Systems, pp. 3158–3164 (2012). <https://doi.org/10.1109/IROS.2012.6385988>
25. Censi, A.: An accurate closed-form estimate of ICP's covariance. In: Proceedings 2007 IEEE International Conference on Robotics and Automation (ICRA), pp. 3167–3172 (2007). <https://doi.org/10.1109/ROBOT.2007.363961>
26. Maken, F., Ramos, F., Ott, L.: Stein ICP for uncertainty estimation in point cloud matching. *IEEE Robot. Autom.* **7**, 1063–1070 (2021). <https://doi.org/10.1109/LRA.2021.3137503>
27. Hu, L., Xiao, J., Wang, Y.: An automatic 3D registration method for rock mass point clouds based on plane detection and polygon matching. *Vis. Comput.* **36**, 669–681 (2020). <https://doi.org/10.1007/s00371-019-01648-z>
28. Ameer, M., Abbas, M., Miura, K.T., Majeed, A., Nazir, T.: Curve and surface geometric modeling via generalized Bézier-like model. *Mathematics* **10**(7), 1045 (2022). <https://doi.org/10.3390/math10071045>
29. Song, Y., Shen, W., Peng, K.: A novel partial point cloud registration method based on graph attention network. *Vis. Comput.* **39**, 1109–1120 (2023). <https://doi.org/10.1007/s00371-021-02391-0>
30. Majeed, A., Abbas, M., Miura, K.T., Kamran, M., Nazir, T.: Surface modeling from 2D contours with an application to craniofacial fracture construction. *Mathematics* **8**(8), 1246 (2020). <https://doi.org/10.3390/math8081246>
31. Nguyen, M., Yuan, X., Chen, B.: Geometry completion and detail generation by texture synthesis. *Vis. Comput.* **21**, 669–678 (2005). <https://doi.org/10.1007/s00371-005-0315-1>
32. Eggert, D., Lorusso, A., Fisher, R.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vis. Appl.* **9**, 272–290 (1997). <https://doi.org/10.1007/s001380050048>
33. Oomori, S., Nishida, T., Kurogi, S.: Point cloud matching using singular value decomposition. *Artif. Life Robot.* **21**, 149–154 (2016). <https://doi.org/10.1007/s10015-016-0265-x>
34. Saleem, W., Schall, O., Patanè, G., et al.: On stochastic methods for surface reconstruction. *Vis. Comput.* **23**, 381–395 (2007). <https://doi.org/10.1007/s00371-006-0094-3>
35. Prakhya, S.M., Bingbing, L., Rui, Y., Lin, W.: A closed-form estimate of 3D ICP covariance. In: 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 526–529 (2015). <https://doi.org/10.1109/MVA.2015.7153246>
36. Maken, F.A., Ramos, F. and Ott, L.: Estimating motion uncertainty with bayesian icp. In 2020 IEEE International Conference on Robotics and Automation (ICRA), 8602–8608 (2020).

37. Barrie Wetherill, G., Duncombe, P., Kenward, M., Köllerström, J., Paul, S.R., Vowden, B.J.: Regression Analysis with Applications. Chapman & Hall, London (1986)
38. Segal, A., Haehnel, D., Thrun, S.: Generalized-icp. In: Robotics: science and systems, vol. 2, no. 4, p. 435 (2009)
39. Gustafsson, F.: Particle filter theory and practice with positioning applications. IEEE Aerosp. Electron. Syst. Mag. **25**, 53–82 (2010). <https://doi.org/10.1109/MAES.2010.5546308>
40. Ying, W., Sun, Sh.: An improved Monte Carlo localization using optimized iterative closest point for mobile robots. Cognitive Comput. Syst. **4**(1), 20–30 (2022)
41. Speekenbrink, M.: A tutorial on particle filters. J. Math. Psychol. **73**, 140–152 (2016). <https://doi.org/10.1016/j.jmp.2016.05.006>
42. Liu, J.S., Chen, R., Logvinenko, T.: A theoretical framework for sequential importance sampling with resampling. In: Sequential Monte Carlo methods in practice, pp. 225–246. Springer, New York (2001).
43. Bengtsson, O., Baerfeldt, A.J.: Robot localization based on scan matching - estimating the covariance matrix for the IDC algorithm. Robot. Auton. Syst. **44**, 29–40 (2003). [https://doi.org/10.1016/S0921-8890\(03\)00008-3](https://doi.org/10.1016/S0921-8890(03)00008-3)
44. Bergström, P., Edlund, O.: Robust registration of surfaces using a refined iterative closest point algorithm with a trust region approach. Numer. Algor. **74**, 755–779 (2017). <https://doi.org/10.1007/s11075-016-0170-3>
45. Bergström, P.: Reliable updates of the transformation in the iterative closest point algorithm. Comput. Optim. Appl. **63**, 543–557 (2016). <https://doi.org/10.1007/s10589-015-9771-3>
46. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte Carlo localization: efficient position estimation for mobile robots. In: Proceedings of the National Conference on Artificial Intelligence, pp. 343–349 (1999)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Ahmad Reza Saleh was born in Tehran, Iran, in 1979. He received his BSc and MSc in Electrical and Electronic Engineering from Sharif University of Technology, Tehran, Iran, in 2005. He is Ph.D. candidate in Intelligent and Data Science from Tarbiat Modares University, Tehran, Iran. His research interests are matching algorithm, geometric/shape modelling, computer vision, pattern recognition, optimization, applied mathematics, intelligent systems, and data science.



Hamid Reza Momeni was born in Komein, Iran, in 1954. He received his BSc in Electrical Engineering from Sharif University of Technology, in 1977, and MSc from University of Wisconsin at Madison, USA, in 1979 and PhD degrees from Imperial College of London, England, in 1987, respectively. His research interests are intelligent systems, pattern recognition, modelling and identification, adaptive control, optimization, robust control, tele-operation systems, and industrial control, instrumentation, automation and navigation and guidance. He is professor and Dean of Faculty of Electrical and Computer Engineering of Tarbiat Modares University, Tehran, Iran.