ISBN: 978-1-948012-08-9

Contents List available at VOLKSON PRESS

Computer Science and Artificial Intelligence(CSAI)

DOI : http://doi.org/10.26480/iscsai.01.2017.45.48

Coarse Pose Registration Using Local Geometric Features

Yan Wang[1,2*], Hongcheng Xu[3], Ge Yu[1,2], Ji Liang[1,2]

[1]Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, South Dengzhuang Road, Beijing, China.
[2]Key Laboratory of Space Utilization, Chinese Academy of Sciences South Dengzhuang Road, Beijing, China.
[3]Institute of Robotics, Beihang University, Xueyuan Road, Beijing, China.
**Corresponding Author Email: wangy@csu.ac.cn*

ARTICLE DETAILS

Article History:

Received 02 october 2017
Accepted 06 october 2017
Available online 11 october 2017

Keywords:

Coarse pose registration, local geometric features, corner edge feature, circle normal feature

ABSTRACT

Coarse pose registration is a crucial step in 6DOF pose estimation. Assuming the objects to be estimated are regular and their 3D models are known, this paper presents local geometric features (LGFs) to estimate the poses of objects in point cloud. Coordinate frames describing the object's pose are defined according to geometric information. Pose estimation are performed by calculating the transformation between the frames in the scene point cloud and the reference model. Two kinds of LGFs are proposed: one is the corner edge feature (CEF) which exploits corner and radiating edges, and the other is the circle normal feature (CNF) which exploits circle center and normal vector of the plane where the circle is located. Included angle sequence of adjacent edges and circle radius, which are invariant to *SE* (3) transformation, are selected to identify CEF and CNF, respectively. Simulation is performed to verify the feasibility of the proposed features.

1. Introduction

Coarse pose registration is a crucial step in 6DOF pose estimation in the field of 3D machine vision. It is usually followed by fine matching which is implemented through iterative closest point (ICP) algorithm. Due to the coupling between position and orientation, a proper coarse registration can accelerate the computation of the refinement, while a bad estimation might lead to wrong or even failed final estimation of the object's pose.

Pose registration problem has been overwhelmingly researched during the past decades. A group of scientists present the whole pipeline of 6D pose estimation and introduce the common used local and global descriptors [1]. Other study concentrates on coarse registration and perform a comprehensive review [2]. According to, coarse registration can be further divided into key point detection, description, and search strategies for correspondence [2]. In the step of detection, a certain set of key points which are more distinctive are selected according to some criterion. For example, Harris 3D detector applies Harris operator to each point and extracts the point with highest Harris value as key points [3]. Descriptor is then defined on these key points to represent the shape characteristics of the object. Depending how the characteristics are represented, descriptor can be categorized into global and local descriptors. Global descriptors try to describe monolithic feature of the object. For example, Ensemble of Shape Functions encodes the distance, angle between points and their combined segments, and area of meshes [4]. Curve Skeleton use skeletal graph to preserve the topological property of the shape [5]. Local descriptors use the neighborhood of key point to construct signature, e.g. Fast Point Feature Histogram and Signature of Histograms of Orientations [6,7]. Searching correspondence is responsible for pairing the scattered descriptors in the scene and reference point cloud so that a transformation can be estimated. Usually multiple sets of descriptors at different viewpoint of the reference point cloud need to be generated, which is trivial and time consuming [8]. As the correspondence grouping is performed on two sets of points, the structure feature of the object and the intrinsic constraints between the points are abandoned, which could lead to wrong or failed pose estimation.

This paper proposes an approach for coarse pose registration using local geometric features (LGFs). Assuming the objects to be estimated are regular and their 3D model are known, the proposed approach defines coordinate frames to describe the object's pose at corners with radiating edges and centers of circular brims. Geometric parameters, which are invariant to *SE* (3) transformation, are selected to identify LGFs. Pairing LGFs detected in the point cloud and that specified according to the 3D models, transformation candidates from the initial pose to the current pose can be directly and easily calculated. using only corner, edge and board points, ICP can quickly filter the candidates and obtain the final coarse estimation.

The rest of the paper is organized as follows. In Section 2, two kinds of local geometric features are presented according to different local structures. Section 3 introduces the pipeline of the proposed coarse registration method. In Section 4, simulation is performed to verify the feasibility of the proposed LGFs. Finally, we conclude the paper in Section 5.

2. LOCAL GEOMETRIC FEATURES

Large part of daily objects are regular and have distinct geometric features, such as corner with radiating edges of a box and circular brim of a cup. Some of the geometric information can be used to define a coordinate frame to directly describe the pose of the object. For example, with the position of the corner and two edge vectors, the pose of the object can be decided. And some information, e.g. circle radius and included angle sequence between the edges, is invariant to *SE* (3) transformation and is suitable to be a descriptor for LGF correspondence grouping between point cloud and 3D model.

In this section, we present two kinds of LGFs according to different local structures. One is the corner edge feature (CEF). CEF utilize the corner and radiating edges to define a coordinate frame. The sequence of included angles between the edges is invariant to *SE* (3) and is chosen to identify the feature. The other is the circle normal feature (CNF). CNF utilizes the center of the circle and the normal of the plane where the circle is located to define the frame. Circle radius is chosen to identify the feature.

2.1 Corner Edge Feature

As shown in Figure 1, for a corner with radiating edges, we can define a frame with the position of the corner and two radiating edge vectors, which describes the pose T of the object. Therefore, we have

$$T=\begin{bmatrix} e_1 & z\times e_1 & z & p_C \\ & 0 & & 1 \end{bmatrix} \quad (1)$$

where p_C is the position of the corner; e_1 denotes the X axis of the defined frame and is an edge vector selected from the radiating edges; z denotes the Z axis and is defined with e_1 and its adjacent edge vectors, e.g. e_2, that is, $z=e_1\times e_2/\|e_1\times e_2\|$.

The included angles between the edges are invariant to *SE* (3) transformation. In order to augment the discrimination, the sequence of the included angles between adjacent radiating edges is selected to identify the feature. The edges are numbered, and the sequence of the included angles is defined corresponding to the edge index sequence, i.e.,

$$\begin{aligned} &\text{edge index sequence}:(1,2,3) \\ &\text{included angle sequence}:(\theta_1,\theta_2,\theta_3) \end{aligned} \quad (2)$$

where $\theta_1,\theta_2,\theta_3$ denotes the included angle between edge 1 and 2, edge 2and 3, and edge 3 and 1.

Both the edge index sequence and included angle sequence are circular linked list, that is, $(\theta_1,\theta_2,\theta_3)$ can also mean $(\theta_2,\theta_3,\theta_1)$ which corresponds to the edge index sequence $(2,3,1)$.

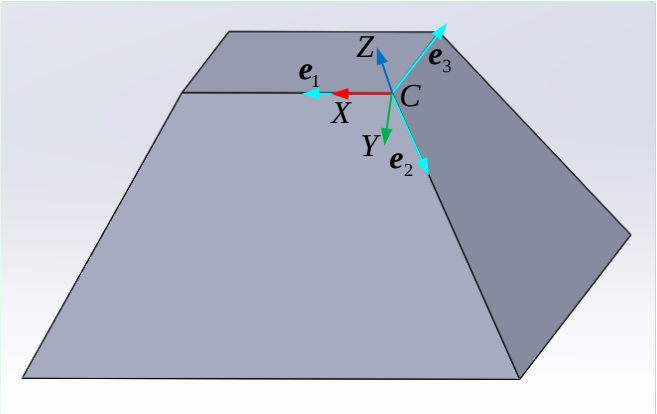


Figure 1: Corner edge feature

It is important to obtain the correct edge index sequence. Given the object's 3D model, we can easily determine the edge index sequence and specify the included angles and edge vectors via CAD software. By contrast, determination of edge index sequence from the scene point cloud is complicated. As the included angles between 3D vectors would mislead the detection, we project the edge vectors into a 2D plane, as shown in Fig. 2. The normal n of the plane is defined as

$$n=\sum e_i \quad (3)$$

Choosing one projected vector as the primary, e.g. e_1', rotate it around the normal n, the meeting sequence with other projected vectors are the correct sequence in 3D space.

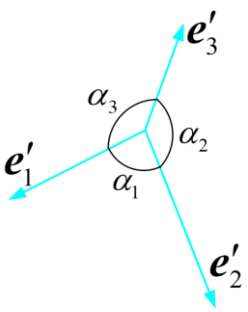


Figure 2: Detection of edge index sequence

The included angle sequence is used to find the correspondence between CEFs in scene point cloud and reference 3D model. However, due to the limited viewport, all the edges radiating from the corner might not be seen by the camera, that is, only partial included angles can be calculated. Nevertheless, with at least two edges, the pose of the object can be decided and CEF is still feasible. For example, if only two edges from corner *C* can be seen, we obtain

$$\begin{aligned} &\text{edge index sequence}:(1,2) \\ &\text{included angle sequence}:(\theta_1,\theta_2) \end{aligned} \quad (4)$$

To avoid effect from the wrong included angle θ_2 between edge 2 and 1, one included angle is discarded from the sequence and then compared with CEFs in the reference 3D model.

With the CEF correspondence between the scene point cloud and the reference 3D model, transformation candidates T can then be calculated.

$$T=T_sT_m^T \quad (5)$$

where T_s is the pose matrix of CEF in the scene point cloud; T_m denotes the pose matrix of the corresponding CEF in the reference 3D model.

2.2 Circle Normal Feature

Apart from corner, circle is another distinct geometric feature, such as the brim of a cup. Similar to CEF, circle normal feature (CNF) uses geometric information to construct a coordinate frame to describe the object's pose as well. The circle center and the normal of the plane where the circle is located are used to construct the coordinate frame. When the object is a solid of revolution, e.g. a vase in Figure 3(a), an arbitrary vector parallel to the plane can be selected as the X axis. On the other side, if the object has principal axes when projected on the plane, we can choose one principal axis as the X axis. For example, a cup in Figure 4(b) has a handle, we can choose the direction pointing from the center to the handle as the X axis. This determine a unique frame and can accelerate coarse pose registration. The radius of the circle is invariant to *SE*(3) transformation and is selected to identify CNF.

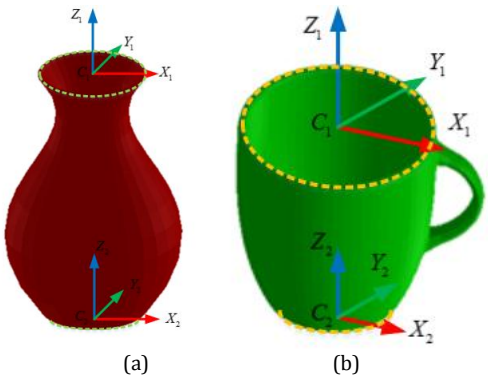


Figure 3: Circle normal feature. (a) CNF in revolved object. (b) CNF with primal axis

CNF is estimated using the edge and board points in the point cloud. After the detection of corner, edge and board points, the number of the extracted points are significantly reduced, compared to the original point cloud. Hence, circles used to specify the parameters of CNF can be quickly estimated using random sample consensus (RANSAC). And the principal axis can be detected using principal components analysis (PCA) [9].

3. COARSE POSE REGISTRATION

As stated in the previous section, both CEF and CNF utilize local geometric information to define coordinate frames so that the pose of the object can be directly described. Coarse registration is performed by calculating the transformation between the matched LGFs in the scene point cloud and the reference model. In this section, we firstly introduce the pipeline of coarse pose registration using LGFs. Then, the detection of corner, edge, and board feature points are introduced. Algorithm for coarse pose registration is presented.

3.1 Coarse Pose Registration Pipeline

Figure 4 shows the block diagram of coarse pose registration pipeline. The scene point cloud is segmented into individual clusters after filtering and segmentation. Corner, edge, and board feature points are detected on each cluster. These points consist of a subset S_f of the original point cloud S. Upon the feature points, edges and circles are extracted using RANSAC. CEF and CNF are then estimated, and coordinate frames describing the pose of the object are constructed. Pairing each LGF estimated in the point cloud to that specified with 3D model, transformation candidates representing the current relative pose of the point cloud with respect to the initial pose are obtained. These candidates will be filtered using ICP algorithm [10]. The feature points S_f which has much smaller number than S are used in ICP, and wrong transformations will cause significant divergence. Therefore, the final coarse pose will be quickly estimated through ICP filtering.

Figure 4: Block diagram of coarse pose registration pipeline

3.2 Detection of Corner, Edge, and Board Feature Points

Feature points detection is critical in the proposed approach. The LGFs are estimated based on the detected feature points, i.e. corner, edge, and board points. In addition, the number of the feature points is much smaller than that of the original point cloud, which will accelerate the filtering of the transformation candidates. Gumhold et al. proposed a method of feature point detection through PCA on the neighborhood of a point, which has been proved to be an efficient method to detect feature points [11]. Here we briefly introduce the method of feature point detection.

For each point p_i in the point cloud S, its *k*-nearest neighbor points N_i need to be gathered. The center location $\boldsymbol{c}_i$ and the correlation matrix M_i can then be given by

$$c_i = \frac{1}{|N_i|}\sum_{\boldsymbol{q}\in N_i} q \tag{6}$$

$$M_i = \frac{1}{|N_i|}\sum_{\boldsymbol{q}\in N_i} (q-c_i)(q-c_i)^T \tag{7}$$

The eigenvectors $\{v_1, v_2, v_3\}$ and the corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$, where $\lambda_1 \le \lambda_2 \le \lambda_3$, of M_i define the correlation ellipsoid which describes the local surface property around the point.

For a point on a flat surface the correlation ellipsoid degenerates to a pancake with $\lambda_1 \approx 0$ and $\lambda_2 \approx \lambda_3$. At a corner point, the correlation ellipsoid has no preference direction and all eigenvalues should be approximately the same. In the case of an edge point the correlation ellipsoid is stretched in the direction of the edge and the eigenvalues obey $\lambda_1 \approx \lambda_2$ and $\lambda_1 + \lambda_2 \approx \lambda_3$. In the case of a border point, the ellipsoid degenerates to an ellipse. The smallest eigenvalue λ_1 is approximately zero and the other two eigenvalues obey $2\lambda_2 \approx \lambda_3$. According to the properties of the correlation ellipsoid at different features, corner, edge, and board points can be extracted from the segmented clusters.

3.3 Algorithm for Coarse Pose Registration

Table 1 shows the pseudo code for the proposed coarse pose registration method. The LGFs in the models are pre-specified. After the estimation of LGFs in the segmented point cloud, each LGF will be paired with each LGF of the models. If they are matched, transformation candidates will be stored as well as the corresponding model. The candidates will be filtered using ICP. For each candidate, the feature points will be transformed. The shortest distance from the point in transformed feature points to the model will be calculated. If it exceeds the threshold, the candidate is not the correct transformation and will be discarded. Due to the property of LGFs that incorrect candidates will cause larges distance, the ICP filtering will run quickly.

Table 1: Pseudo code for coarse pose registration

```
    Data:   Segmented Point Cloud, LGFs in Models, 3D Models
    Result: Coarse Pose Matrix
1   detecet feature points;
2   estimate CEFs and CNFs from the feature points using RANSAC;
    // Pair LGFs in the point cloud to LGFs of each reference model.
3   for scene_lgf in scene_lgf_set
4     for refer_model in refer_model_set
5       for refer_lgf in refer_model.refer_lgf_set
6         if scene_lgf matches refer_lgf then
7           transform_candidate_set.append();
    // ICP Filtering.
8   for transform_candidate in transform_candidate_set
9     transform feature points using transform_candidate;
10    for point in the transformed_feture_points
11      calculate the shortest distance to the corresponding model
12      if distance > distance_threshold then
13        break;
```

4. SIMULATION

In this section, we use BlenSor, which is a software package to simulate various range cameras and can facilitate the development of algorithms on 3D machine vision, to generate point cloud from the model of a scene and estimate the coarse pose of the objects in the scene using the proposed LGFs [12].

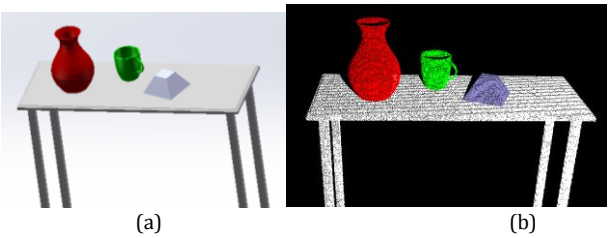
As shown in Figure 5, 3 objects, a vase, a cup and a square frustum, are on the table. They cover the proposed CEF and CNF. Point cloud is generated under noise which is subject to a normal distribution with mean $\mu = 0$ and variance $\sigma = 0.01$. Figure 5(c) shows the detected feature points in the segmented point clouds. The green points with larger size represent board points, and the blue points represent edge points.

Both the vase and the cup are estimated via CNF. Therefore, here we take the vase as an example to demonstrate the efficiency of CNF. According to the output of the proposed coarse estimation algorithm, the pose matrices, with respect to the camera frame, of the matched CNFs in the scene point cloud and the 3D model are as below.

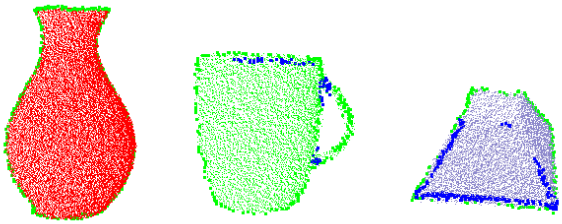
$$T_{vase,scene} = \begin{bmatrix} 0.064 & 0.435 & -0.898 & 0.023 \\ 0.997 & 0 & 0.071 & -0.548 \\ 0.031 & -0.900 & -0.434 & -2.948 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$T_{vase,refer} = \begin{bmatrix} 0 & -0.180 & -0.984 & -0.026 \\ 1 & 0 & 0 & -0.558 \\ 0 & -0.984 & 0.180 & -2.991 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Position error is $p_{vase,error} = [0.049 \quad 0.010 \quad 0.043]^T\ m$. Since the vase is a solid of revolution, we choose the included angle between normal vectors, i.e. the Z axis vectors in Eq. (8) and (9), to represent orientation error. The orientation error is $\theta_{vase,error} = 36.4^\circ$. The large position and orientation error is mainly caused by the inaccurate estimation on the parameters of circular brims.



(a) (b)



(c)

Figure 5: Coarse pose estimation using LGFs. (a) 3D model of the scene. (b) Point cloud with noise. (c) Detected feature points in the segmented point clouds

The square frustum is used to verify the efficiency of CEF. The pose matrices of the matched CEF in the scene point cloud and the 3D model are

$$T_{frust,scene} = \begin{bmatrix} 0.080 & -0.930 & -0.359 & 0.517 \\ 0.927 & 0.202 & -0.317 & 0.102 \\ 0.367 & -0.308 & 0.878 & -3.000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$T_{frust,refer} = \begin{bmatrix} 0.065 & -0.924 & -0.378 & 0.526 \\ 0.933 & 0.190 & -0.305 & 0.080 \\ 0.354 & -0.332 & 0.874 & -3.007 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Position error is $p_{frust,error} = \begin{bmatrix} -0.011 & 0.022 & 0.007 \end{bmatrix}^T m$. Orientation error is represented using axis angle of rotation matrix [13]. The rotation angle from the coordinate frame in the reference model to that in the scene point cloud is $\theta_{frust,error} = 1.5^\circ$. Compared to CNF, the position and orientation error in CEF is much smaller [14].

5. CONCLUSIONS

In this paper, we present two kinds of local geometric features (LGFs) for coarse pose registration. The proposed LGFs utilize the geometric information to define coordinate frames on the estimated objects. Pose registration is performed by calculating the transformation between the frames of the matched LGFs in the scene point cloud and the reference 3D model. How to construct these two LGFs, i.e. corner edge feature (CEF) and circle normal feature (CNF), are introduced. The pipeline of coarse registration using LGFs is presented. Simulation is done and verifies LGFs' feasibility. The simulation results also indicate that the registration error is affected by the estimation accuracy of the parameters of 3D lines and circles. To conquer this, in future work we will optimize RANSAC algorithm so that it can reduce estimation error.

ACKNOWLEDGMENTS

This work is supported by the CSU (Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences) Foundation under Grant No. CSU-QZKT-201702.

REFERENCES

[1] Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Vincze, M. 2012. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. IEEE Robotics and Automation Magazine, 19 (3), 80-91.

[2] Diez, Y., Roure, F., Lladó, X., Salvi, J. 2015. A qualitative review on 3d coarse registration methods. ACM Computing Surveys (CSUR), 47 (3), 45.

[3] Sipiran, I., Bustos, B. 2011. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. The Visual Computer, 11 (2), 963-976.

[4] Wohlkinger, W., Vincze, M. 2011. Ensemble of shape functions for 3-D object classification, in IEEE Int. Conf. Robotics and Biomimetics (IEEE-ROBIO), 2987-2992.

[5] Cornea, N.D., Silver, D., Min, P. 2007. Curve-skeleton properties, applications, and algorithms. IEEE Transactions on Visualization and Computer Graphics, 13 (3), 530-548.

[6] Rusu, R. B., Blodow, N., Beetz, M. 2009. Fast point feature histograms (FPFH) for 3-D registration, in IEEE Int. Conf. Robotics and Automation (ICRA), Kobe, Japan, 3212-3217.

[7] Tombari, F., Salti, S., Stefano, L. 2010. Unique signature of histograms for local surface description, in Proc. 11th European Conf. Computer Vision (ECCV '10), 356–369.

[8] Armenise, V. 2013. Estimation of the 3D Pose of objects in a scene captured with Kinect camera using CAD models. School of Computer Science and Communication.

[9] Bischof, Horst, Leonardis, A., Selb, A. 1999. MDL Principle for Robust Vector Quantisation. Pattern Analysis and Applications, 1 (2), 59-72.

[10] Besl, P. J., McKay, N. D.1992. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (2), 239-256.

[11] Gumhold, S., Wang, X., MacLeod, R. S. 2001. Feature Extraction From Point Clouds. In Proceedings of the 10th International Meshing Roundtable.

[12] Gschwandtner, M., Kwitt, R., Uhl, A., Pree, W. 2011. BlenSor: Blender Sensor Simulation Toolbox. Advances in Visual Computing, 199-208.

[13] Shuster, M. D. 1993. A survey of attitude representations. Navigation, 8 (9), 439–517.

[14] Fischler, M. A., Bolles, R. C. 1981. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24 (6), 381-395.