WEP Warwick Evans Publishing

International Journal of Mechanical and Electrical Engineering

ISSN: 3005-9615 (Print), ISSN: 3005-7132 (Online) | Volume 6, Number 3, Year 2025
DOI: https://doi.org/10.62051/ijmee.v6n3.10
Journal homepage: https://wepub.org/index.php/IJMEE/index

A Machine Vision-Based Method for Pose Estimation of Pin Shafts in Oil Derrick Structures

Qitao Tu, Chong Xie

Southwest Petroleum University, Chengdu Sichuan, 610500, China

ABSTRACT

As a core component in oil drilling operations, the derrick structure of a drilling rig traditionally relies on manual handling during assembly and disassembly, which is complex, inefficient, and poses significant safety risks. To enhance the level of automation, this paper proposes a machine vision-based method for pose recognition of pin shafts in derrick structures. A structured light camera is employed to capture on-site 3D point cloud data, which is then aligned with pre-defined workpiece templates for recognition. The registration process begins with a coarse alignment using the Sample Consensus Initial Alignment (SAC-IA) algorithm based on Fast Point Feature Histograms (FPFH), followed by fine pose estimation through the Iterative Closest Point (ICP) algorithm accelerated by a KD-tree structure. Experimental results demonstrate the effectiveness and practicality of the proposed system in identifying and locating typical components. This method offers reliable technical support for the automated assembly and disassembly of derrick structures and holds promising value for engineering applications.

KEYWORDS

Derrick Pin; FPFH; Model Point Cloud; Registration.

1. INTRODUCTION

The derrick of an oil drilling rig is a critical piece of equipment in drilling operations. During assembly and disassembly, the positioning and handling of pin shafts are typically performed manually, involving complex procedures and high labor intensity. To achieve automation, the automatic localization of pin shafts during derrick disassembly and assembly has become a key technical challenge for improving operational efficiency and ensuring safety.

In recent years, machine vision has emerged as a key technology for three-dimensional object pose recognition and has attracted widespread attention from researchers worldwide[1][2]. Point cloud registration, as one of the core techniques for pose estimation, aims to align point cloud data from different coordinate systems through rigid transformations, thereby enabling accurate pose determination. Registration methods are typically divided into two stages: coarse registration, which provides an initial alignment, and fine registration, which further refines the result to improve accuracy.

In the field of coarse registration, Chua et al.[3] were the first to propose the Point Signature feature, laying the foundation for research on point cloud feature descriptors. Johnson et al.[4][5] introduced the Spin Image descriptor, which offers rotational invariance but suffers from high computational complexity. Rusu et al.[6][7] subsequently proposed the Point Feature Histogram (PFH) and its improved version, the Fast Point Feature Histogram (FPFH). The latter significantly enhances computational efficiency while maintaining strong geometric descriptiveness, making it one of the most widely used local features to date. Coarse registration methods based on FPFH and RANSAC

have been applied in various industrial scenarios[8]; however, their matching accuracy tends to degrade in the presence of noise or occlusion.

In the field of fine registration, the Iterative Closest Point (ICP) algorithm proposed by Besl and McKay[9] has been widely adopted due to its simplicity and high accuracy. However, ICP is highly sensitive to the initial pose and prone to converging to local minima; it is also affected by variations in point cloud density. To address these limitations, researchers have proposed numerous improvements. For example, Zhang et al.[10] introduced a least-squares estimation approach, Trucco et al.[11] proposed a robust iterative method known as RICP, and GP-ICP[12] incorporates geometric primitives and accelerated search strategies. These enhancements effectively mitigate issues such as high computational cost and mismatches, yet challenges such as long computation time and strong dependence on initial alignment still persist.

With the advancement of deep learning, such methods have been increasingly introduced into point cloud registration tasks. End-to-end networks such as PointNet[13] and PointNet++[14] have demonstrated superior performance in feature extraction and matching accuracy. However, these approaches typically require large amounts of annotated data and substantial computational resources, which poses challenges for practical deployment in real-world scenarios.

In summary, although significant progress has been made in improving the accuracy and robustness of point cloud registration, existing methods still face challenges such as unstable matching, insufficient real-time performance, and complex system integration under complex operating conditions. These limitations hinder their direct application to practical tasks such as derrick assembly and disassembly. To address these issues, this paper proposes a pose recognition system for key components based on the registration of template and scene point clouds. The system employs the Sample Consensus Initial Alignment (SAC-IA) algorithm for coarse registration and incorporates a KD-tree-accelerated Iterative Closest Point (ICP) algorithm for precise pose estimation, enabling efficient and reliable identification and localization of critical components in derrick operations.

2. PRINCIPLES AND METHODS

2.1. Model Point Cloud Creation

The model point cloud is generated based on a mesh model, where the three-dimensional geometry of the object is represented by a collection of triangular facets. To simplify the point cloud registration process, a reference coordinate system is established for the model, enabling direct comparison between the target workpiece point cloud and the template point cloud via rigid transformation. The reference frame is defined with the center point of the upper surface of the pin shaft as the origin. After extracting the point cloud from the mesh model, key surfaces of the template are selectively retained to reduce the number of points and improve computational efficiency. Only regions containing distinctive geometric features are preserved, and the resulting point clouds are aggregated to construct a template point cloud library, as illustrated in Fig. 1.

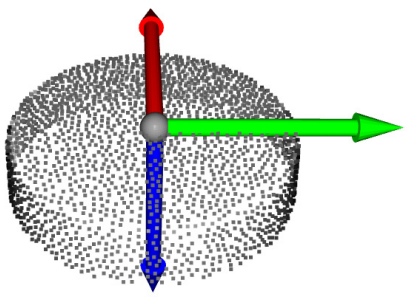


Fig 1. Pin point cloud template diagram

2.2. Workplace Attraction Cloud Preprocessing

2.2.1. Point Cloud Segmentation based on RANSAC Algorithm

Random Sample Consensus (RANSAC)[15] is an iterative algorithm designed to estimate mathematical models from datasets that contain a significant proportion of outliers.

The RANSAC algorithm flow is as follows:

(1) A predefined distance threshold $T$is used as the inlier criterion. Three non-collinear sample points are randomly selected from the point cloud data. Using the spatial coordinates of these three points, the four parameters A,B,C, and D of the plane model are calculated according to Equation (1).

$$Ax + By + Cz + D = 0 \tag{1}$$

(2) The remaining data points are then used to validate the plane model, and the corresponding error values are computed.

(3) The computed errors are compared against the predefined threshold T: if the error for a point is less than T, the point is classified as an inlier, and the number of inliers for the current model is recorded.

(4) This process iterates repeatedly, and whenever the number of inliers in the current iteration exceeds the previously recorded maximum, the optimal model parameters are updated accordingly.

(5) The above steps continue iteratively until a predefined maximum number of iterations is reached. Finally, the algorithm outputs the optimal model parameters corresponding to the largest inlier set and re-estimates the model parameters using these inliers to obtain the final plane model.

The RANSAC algorithm is employed for point cloud segmentation of the scene, with the segmentation results shown in Fig. 2.

(a) Pin Shaft Scene Point Cloud (b) RANSAC-Based Point Cloud

Fig 2. Point cloud segmentation based on RANSAC algorithm

2.2.2. Denoising and Downsampling

Statistical filtering is a point cloud denoising method based on spatial distribution characteristics. By analyzing the spatial distribution of each sample point and its neighboring points within the point cloud data, abnormal points are identified and removed according to a predefined statistical threshold, thereby achieving noise reduction in the point cloud dataset.

For each data point in the point cloud, a spatial search is performed to identify its k nearest neighbors, establishing a local neighborhood relationship. The Euclidean distances between the point and its k neighbors are calculated, and the mean of these distances is computed. Assuming that these distances follow a Gaussian distribution, the mean μ and standard deviation σ of the distances for all n points can be obtained. A threshold is set as s times the standard deviation σ, where s is typically chosen as the filtering distance multiplier. If the average distance of a point exceeds this threshold, it is classified

as an outlier and removed. The algorithm processes the entire point cloud by iterating over all data points, thus completing the statistical filtering of the point cloud.

Due to the large volume of point data, subsequent computations require traversing every point, resulting in increased computational load. Therefore, downsampling of the point cloud data is necessary. In this work, voxel-based downsampling is employed. The main principle involves constructing a three-dimensional voxel grid over the input point cloud, where each voxel has the same size. The centroid of all points within each voxel is computed and used to represent all points inside that voxel. The set of these centroids forms the downsampled point cloud. The process is illustrated in Fig. 3.

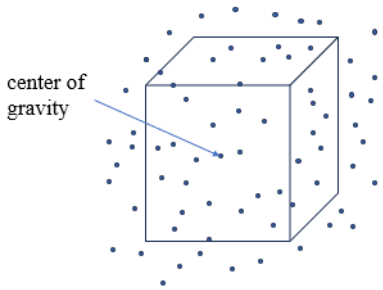


Fig 3. Voxel-Based Downsampling

2.3. Workpiece Pose Estimation

This paper estimates the pose information of the workpiece based on point cloud registration. By aligning the model point cloud with the scene point cloud, the transformation matrix between them can be obtained after registration, which in turn provides the center pose of the pin shaft.

2.3.1. FPFH Feature Extraction

The Fast Point Feature Histograms (FPFH)[16], proposed by Rusu et al., is an improved version of the traditional Point Feature Histogram (PFH) descriptor. The specific steps of the FPFH algorithm are as follows:

(1) For each query point, simplified features are computed for all points within its k-nearest neighbor radius, and these are aggregated to form the Simplified Point Feature Histogram (SPFH).

(2) For each k-nearest neighbor of the query point, the above process is repeated to generate their respective SPFH descriptors.

(3) The SPFH descriptors of the query point and its neighboring points are fused through a weighted averaging method to form the final FPFH descriptor.

$$\mathrm{FPFH}\left(p_q\right)=\mathrm{SPFH}\left(p_q\right)+\frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot\mathrm{SPFH}\left(p_k\right) \tag{2}$$

In the equation, the weight depends on the distance between the point and its neighboring points. The FPFH lookup process is illustrated in Fig. 4.

2.3.2. SAC-IA Algorithm Based on FPFH Features

The SAC-IA algorithm performs initial registration by computing the FPFH feature descriptors of both the template and target point clouds, followed by iterative alignment based on these FPFH features. The basic steps of the algorithm are as follows:

(1) Select i sample points with distinct FPFH features from the target point cloud P, ensuring that the Euclidean distance between any two sampled points exceeds a predefined threshold.

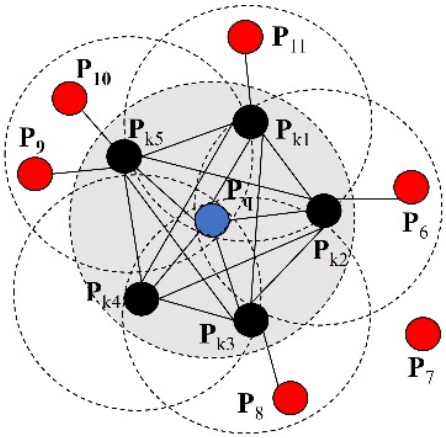


Fig 4. Simplified Illustration of FPFH Neighborhood Search

(2) By comparing the FPFH descriptors of the two point clouds, correspondences are established in the feature space, enabling precise matching between points in the target point cloud P and the template point cloud Q based on FPFH features.

(3) Calculate the transformation matrix between corresponding points of the two point clouds. Using the distance error and the function $\sum_{i=1}^{n} D(e_i)$ as the error distance, set a distance threshold d_{min} Here,e_i represents the error of the i-th pair of corresponding points after transformation, and t denotes a predefined value. The error function is:

$$D(e_i) = \begin{cases} \frac{1}{2} e_i^2 & \| e_i \| \leq t_e \\ \frac{1}{2} t_e (2 \| e_i \| - t_e) & \| e_i \| > t_e \end{cases} \tag{3}$$

2.3.3. ICP Algorithm based on Kd-tree Acceleration

To accelerate the correspondence matching process in the ICP algorithm, this study employs the Kd-tree nearest neighbor search method. The construction process of the Kd-tree is as follows:

(1) For the point cloud set Q, compute the variance of the data along each of the three dimensions: X, Y, and Z. The dimension with the largest variance is selected as the splitting axis. Based on this axis, the points are sorted accordingly.

(2) The median value along the selected splitting axis is computed and used as the threshold t, dividing the entire 3D space into two parts. A node is then created to store the splitting dimension and the corresponding threshold value.

(3) The 3D data is compared against the threshold t along the splitting axis. Data points with values less than t are assigned to the left subtree, while those with values greater than t are assigned to the right subtree.

(4) Steps 1 to 3 are recursively applied to both the left and right subtrees until all subsets can no longer be divided. At that point, the data is stored as leaf nodes.

The ICP algorithm accelerated by Kd-tree can improve the computational efficiency of the registration process and, to some extent, help avoid convergence to local optima.

3. EXPERIMENT AND VALIDATION

3.1. Experimental Platform Setup

The hardware used in the experiment primarily includes the robotic arm, the host control computer, the gripper, and the camera. Specifically, the robotic arm consists of a Han's S30 dual-arm robot. The camera employed is the Cross-View DF-XEMA-D. As illustrated in Fig. 5.

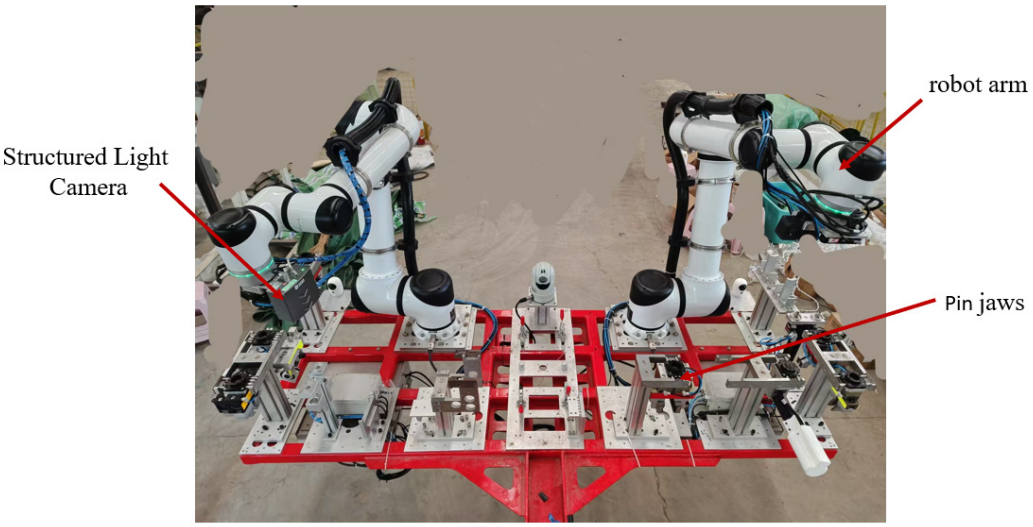


Fig 5. Experimental Setup

To validate the effectiveness of the algorithm for automated target recognition in the dismantling and assembly of derrick components, a derrick pin shaft part was fabricated, with the ear plate designed to mate with the pin shaft, as shown in Fig. 6.

(a) Experimental Test Rig (b) Ear Plate (c) Pin Shaft

Fig 6. Photograph of the Components

3.2. Experimental Procedure

To validate the effectiveness of the algorithm, pose estimation and dismantling experiments were conducted. First, the Cross-View DF-XEMA-D camera was used to capture the scene point cloud of the workpiece, which was then transmitted to the PC. To accelerate computation, the point cloud was preprocessed based on the characteristics of the target point cloud. This preprocessing included point cloud segmentation, statistical filtering, and voxel-based downsampling to reduce the number of points while preserving relevant geometric features. Subsequently, features were extracted from both the template and target point clouds. Coarse registration was performed using the SAC-IA algorithm to obtain an initial transformation matrix, followed by fine registration via the Kd-tree accelerated

ICP algorithm to derive the final transformation matrix. The six-degree-of-freedom (6-DoF) pose of the workpiece was then solved. The experimental procedure is illustrated in Fig. 7:

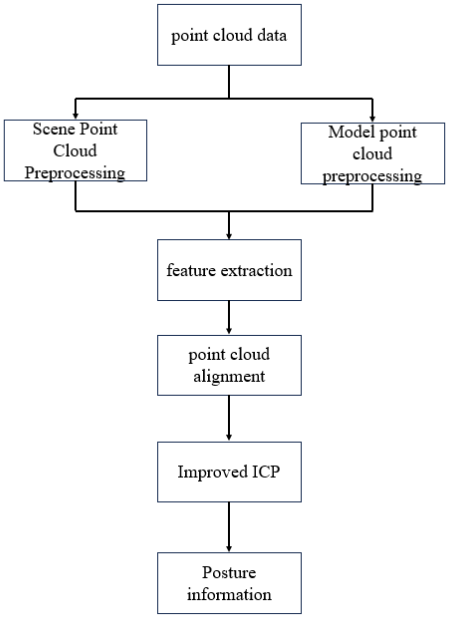
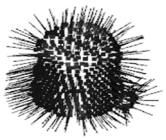


Fig 7. Flow chart of the experiment

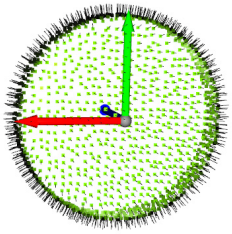
3.3. Pin Recognition Experiment

For the scene target recognition of pin disassembly, the experimental process acquires and preprocesses the scene point cloud and stencil point cloud of pin disassembly, as shown in Fig. 8 (a)-(c), and then extracts the target point cloud features and performs the alignment, and the final alignment result is shown in Fig. 8 (d).

(a) Pinfield attractions cloud



(b) Scene point cloud preprocessing



(c) Point cloud preprocessing for pin models

(d) Alignment plan

Fig 8. Flow of pin position recognition

Taking the center point of the circular hole on the upper surface of the pin as the origin, the true pin position is obtained by manually adjusting the lug plate to fit the pin. The pin position recognized by the algorithm is converted to the robot base coordinate system by hand-eye calibration, and the estimated pin position and the real pin position are shown in Table 1.

Table 1. Estimation of pin position

Posture	True pin position (P*x*/mm,P*y*/mm,P*z*/mm, R*x*/°,R*y*/°,R*z*/°)	Estimated pin position (P*x*/mm, P*y*/mm, P*z*/mm, R*x*/°,R*y*/°,R*z*/°)	Inaccuracies
Posture1	(-365.2,-1204.8,589.4, -78.6, -4.3,15.8)	(364.7,-1205.6,588.5, -79.2,-3.9,16.5)	(0.5,-0.8,-0.9, -0.6,0.4,0.7)
Posture2	(-412.5,-1003.3,573.1, -128.3,-6.0,-160.2)	(-413.1,-1002.4,572.0, -127.5,-6.6,-159.7)	(-0.6,0.9,-1.1, 0.8,-0.6,0.5)
Posture3	(-395.8,-890.2,564.7, -12.5,-7.4,28.6)	(-396.6,-889.5,566.0, -13.0,-8.1,27.9)	(-0.8,0.7,1.3, -0.5,-0.7,-0.7)

Using the Root Mean Square Error RMSE evaluation metric, the RMSE of Px, Py, Pz, Rx, Ry and Rz is calculated as in Table 2:

Table 2. Root mean square error of pin identification

	P*x*(mm)	P*y*(mm)	P*z*(mm)	R*x*(°)	R*y*(°)	R*z*(°)
RMSE	0.6458	0.8042	1.1121	0.6458	0.5809	0.6403

As shown in Tables 1 and 2, the root-mean-square (RMS) errors of Px and Py are within 1 mm, the RMS error of Pz is less than 1.5 mm, and the RMS errors of Rx, Ry, and Rz are within 1° of each other after the pin axis is aligned. The errors include the hand-eye calibration error and the robot's own error.

4. SUMMARY

This paper presents a machine vision–based method for pose recognition of pin shafts in derricks, aiming to enhance the accuracy of automated identification and localization of key components during the dismantling and assembly of drilling derricks. Initially, three-dimensional modeling techniques are employed to construct the geometric model and generate corresponding high-quality model point cloud data. For the raw scene point clouds acquired on-site, an effective preprocessing pipeline is designed. Subsequently, the model point clouds are registered with the processed scene point clouds to obtain the spatial pose information of the target components within the actual environment. This method achieves precise localization of critical parts in the complex and interference-rich derrick setting, providing essential visual perception support for intelligent derrick dismantling operations.

The proposed method demonstrated favorable performance in pose recognition of derrick components; however, certain limitations remain. The registration process is sensitive to point cloud quality, with accuracy prone to degradation under noise and occlusion. Additionally, the current approach is primarily designed for static scenes and has yet to address recognition requirements in

dynamic or complex operating conditions. Future work may focus on incorporating deep learning techniques to enhance point cloud feature representation and integrating real-time multi-sensor fusion technologies to improve system robustness and intelligence.

REFERENCES

[1] Hoque S, Arafat M Y, Xu S, et al. A Comprehensive Review on 3D Object Detection and 6D Pose Estimation with Deep Learning [J]. IEEE Access, 2021, 9: 143746-70.

[2] Arnold E, Al-jarrah O Y, Dianati M, et al. A Survey on 3D Object Detection Methods for Autonomous Driving Applications [J]. IEEE TRANSACtions on Intelligent Transportation Systems, 2019, 20(10): 3782-95.

[3] Chua C S, Jarvis R. Point signatures: A new representation for 3d object recognition[J]. International Journal of Computer Vision, 1997, 25: 63-85.

[4] Johnson A E, Hebert M. Surface matching for object recognition in complex three-dimensional scenes[J]. Image and Vision Computing, 1998, 16(9): 635-651.

[5] Johnson A E, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes[J]. IEEE Transactions on pattern analysis and machine intelligence, 1999, 21(5): 433-449.

[6] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms[C]. In Proceedings of International Conference on Intelligent Robots and Systems. IEEE, 2008: 3384-3391.

[7] Rusu R B, Marton Z C, Blodow N, et al. Learning informative point classes for the acquisition of object model maps[C]. International Conference on Control. IEEE, 2008.

[8] Song S, Shi X, Ma C, et al. MF-LIO: integrating multi-feature LiDAR inertial odometry with FPFH loop closure in SLAM[J]. Measurement Science and Technology, 2024, 35(8): 086308.

[9] Besl P J, McKay N D. Method for registration of 3-D shapes[C]. Sensor fusion IV: control paradigms and data structures. Spie, 1992, 1611: 586-606.

[10] Zhang Z . Iterative point matching for registration of free-form curves and surfaces[J]. International Journal of Computer Vision, 1994, 13(2).

[11] Trucco E, Fusiello A, Roberto V. Robust motion and correspondence of noisy 3-D point sets with missing data[J]. Pattern recognition letters, 1999, 20(9): 889-898.

[12] Bae K H, Lichti D D. A method for automated registration of unorganised point clouds[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2008, 63(1): 36-54.

[13] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[14] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.

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

[16] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]. In Proceedings of International Conference on Robotics and Automation. IEEE, 2009: 3212-3217.

[17] Zhang Z. Iterative point matching for registration of free-form curves and surfaces[J]. International Journal of Computer Vision, 1994, 13(2):119-152.