Computational Visual Media
https://doi.org/10.1007/s41095-022-0308-2 Vol. 10, No. 1, February 2024, 61–77

Research Article

6DOF pose estimation of a 3D rigid object based on edge-enhanced point pair features

Chenyi Liu[1], Fei Chen[2] (✉), Lu Deng[3], Renjiao Yi[1], Lintao Zheng[4], Chenyang Zhu[1], Jia Wang[5], and Kai Xu[1]

Abstract The point pair feature (PPF) is widely used for 6D pose estimation. In this paper, we propose an efficient 6D pose estimation method based on the PPF framework. We introduce a well-targeted down-sampling strategy that focuses on edge areas for efficient feature extraction for complex geometry. A pose hypothesis validation approach is proposed to resolve ambiguity due to symmetry by calculating the edge matching degree. We perform evaluations on two challenging datasets and one real-world collected dataset, demonstrating the superiority of our method for pose estimation for geometrically complex, occluded, symmetrical objects. We further validate our method by applying it to simulated punctures.

Keywords point pair feature (PPF); pose estimation; object recognition; 3D point cloud

1 Introduction

The goal of 6D pose estimation is to determine the position and orientation of a target object via a rigid transformation from the object's coordinate system to the camera coordinate system. Pose estimation is an important aspect of target recognition and scene understanding. Pose estimation has also been widely used in industrial and medical fields. In the medical field, with increased development of medical imaging, computer-assisted surgery, and 3D vision, robot-operated surgery based on 3D visual navigation has become a trend [1, 2]. In 3D visually navigated robot-operated surgery, registration of preoperative 3D models reconstructed by medical imaging and spine point clouds acquired by depth cameras during an operation is crucial.

In real surgical scenarios, the human spine has a complex geometry with high self-occlusion and symmetry [3], potentially leading to algorithmic errors. There is no satisfactory and universal solution to this problem. In this work, we propose a method of pose estimation taking into account the particular geometry of the spine. Given the complex shape of the spine, many spine feature points lie on edges. Therefore, an edge-focused sampling method is used to select stable and salient points to generate stable transformation hypotheses. To handle the ambiguities due to spinal symmetry, we consider that the difference in details between symmetric and highly occluded objects can be effectively distinguished by the degree of edge matching.

Overall, the contributions of our work may be summarized as follows:

- A well-target down-sampling strategy relying on edge information. It effectively retains edge points and points with large curvature variations. Robust hypothesis generation is achieved by sampling stable feature points.

1 College of Computing, National University of Defense Technology, Changsha 410073, China. E-mail: C. Liu, liuchenyi_1013@nudt.edu.cn; R. Yi, yirenjiao@nudt.edu.cn; C. Zhu, zhuchenyang07@nudt.edu.cn; K. Xu, kevin.kai.xu@gmail.com.

2 Department of Spine Surgery, the Second Xiangya Hospital, Central South University, Changsha 410011, China. E-mail: chenfei1972@csu.edu.cn (✉).

3 Clinical Nursing Teaching and Research Section, the Second Xiangya Hospital, Changsha 410011, China. E-mail: csdenglu1026@csu.edu.cn.

4 College of Meteorology and Oceanography, National University of Defense Technology, Changsha 410073, China. E-mail: zhenglintao13@nudt.edu.cn.

5 Beijing Institute of Tracking and Communication Technology, Beijing 100094, China. E-mail: kelexuebi2009@163.com.

Manuscript received: 2022-05-12; accepted: 2022-08-16

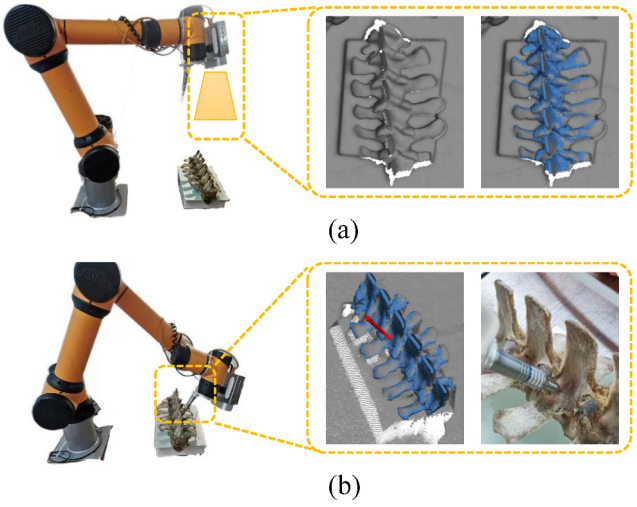

Fig. 1 Experiment on robot-operated positioning with vision-based navigation. (a) The depth camera scans the spine to perform template-based pose estimation. (b) After matching, the robotic arm drills the spine using a predetermined pose and position.

- A pose hypothesis verification method which considers the degree of matching for edge points. It has an early exit strategy to reduce time.
- An implementation of an experimental platform for robot-operated positioning based on this method. We use a position-based visual servoing scheme to control the robot arm to ensure the accuracy of the drilling position.

2 Related work

This section reviews relevant algorithms and their modifications for pose estimation for 3D point clouds, and point pair features.

2.1 Pose estimation methods

Many different methods have been proposed for 3D object detection and pose estimation. Existing research methods can be roughly divided into feature-based methods, template matching methods, point-based methods, and deep learning-based methods. Feature-based methods can be considered the broadest solution, and can be roughly divided into global feature-based and local feature-based algorithms. Algorithms based on global features [4–6] have good speed and memory consumption. However, their usefulness is limited in clinical applications due to their sensitivity to occlusion and noise, and the need to pre-isolate the region of interest from the background. Algorithms based on local features [7–10] are more robust to occlusion and clutter. However, they require additional computation time during the subsequent matching and hypothesis validation, so do not meet the requirements of a real-time surgical navigation system. Methods based on template matching [11] can detect texture-free targets but are sensitive to surgical instrument occlusion. The main point-based method is the iterative closest point algorithm (ICP) [12] and its variants [13, 14]. They depend on a good initial pose estimate and are usually used for pose refinement. Deep learning-based methods [15–19] work well on public 3D datasets. However, deep learning-based methods require significant computational power and time to label datasets. The difficulty of collecting medical samples and the small amount of data hinders the application of deep learning-based methods to surgical navigation.

2.2 Point pair features

In 2010, Drost et al. [20] proposed a rigid 6D pose estimation method based on point pair features (PPFs). It is a compromise between local feature and global feature methods, striking a good balance between accuracy and speed. PPFs describe the surface of an object through global modeling of four-dimensional features defined by directional point pairs. These features are used to find the corresponding relationships between scene and model point pairs, to generate numerous candidate hypotheses, and then to cluster and sequence the candidate poses to obtain the final hypotheses. PPFs are low-dimensional features based on oriented points and are suitable for objects with rich surface variation. Moreover, PPF descriptors, having global significance, show stronger discriminative power than most local features. They are suitable for the objects studied in this paper with complex structures and strong occlusion, so we choose the PPF framework as our basis.

Because of the advantages of PPF, many improved PPF methods have been proposed. Choi and Christensen [21] proposed a color point pair feature (CPPF), which uses color information to significantly improve the discrimination and accuracy of traditional point pair features. Drost and Ilic [22] proposed the concept of geometric and textured edges. Geometric edges are obtained using the intensity image and depth image to construct multimodal point pair features. Liu et al. [23] proposed a novel descriptor,

the boundary-to-boundary-using-tangent-line (B2B-TL) to estimate the poses of industrial parts. Vock et al. [24] utilized point pair features on edges for the quick generation of transformation guesses in a random sample consensus setting. Inspired by the above article, we propose a down-sampling method using a combination of edge points and geometric high curvature feature points for the spine. We also propose a pose hypothesis verification method based on edge matching to make it more competitive in detecting geometrically complex and symmetrical objects such as the spine.

The rest of this paper is organized as follows. Section 3 describes the original PPF method, and Section 4 describes our proposed method and the design of our robot-operated positioning experiments. Experimental results for the spine dataset and the public datasets are given in Section 5. Section 6 concludes the paper.

3 PPF method

Our approach is based on the original PPF method [20]. To better understand this article, we introduce the basic framework of that method in this section.

3.1 Point pair feature

A point pair feature is used to describe the relative distance and normals of a pair of oriented points, as shown in Fig. 2. Given a reference point p_r and a second point p_s with normals n_r and n_s respectively, the PPF is a four-dimensional vector defined as

$$\mathrm{PPF}\left(\boldsymbol{p}_\mathrm{r},\boldsymbol{p}_\mathrm{s}\right)=\left(\|\boldsymbol{d}\|_2,\angle\left(\boldsymbol{n}_\mathrm{r},\boldsymbol{d}\right),\angle\left(\boldsymbol{n}_\mathrm{s},\boldsymbol{d}\right),\angle\left(\boldsymbol{n}_\mathrm{r},\boldsymbol{n}_\mathrm{s}\right)\right) \tag{1}$$

where $\boldsymbol{d}=\boldsymbol{p}_\mathrm{r}-\boldsymbol{p}_\mathrm{s}$, and $\angle(\boldsymbol{a},\boldsymbol{b})$ denotes the angle between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

3.2 Drost's pipeline

The PPF method can be divided into offline global modeling and online local matching steps.

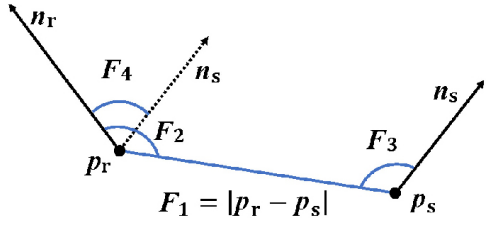

Fig. 2 Point pair feature definition.

In the offline global modeling phase, to create a description of the model, the model is down-sampled using uniform sampling. Then point pair features are computed and quantified for all possible model point pairs. The point pair features are used as hash keys in a hash table via a quantization function, while the value encodes the pose of the feature relative to the model. The pose of the model is encoded by storing the index of the reference point p_r and an angle α_m, the latter representing the angle between the projection of the model point pair concatenation and the positive direction of the y-axis.

The online local matching phase consists of two parts: (1) finding the correspondence between point pairs using four-dimensional point pair features, and (2) generating candidate poses from the correspondences and then clustering them to obtain the best object pose. In the first part, reference points are sampled from the scene. Uniform down-sampling of the scene point cloud is performed to obtain a set of scene points, and then the i-th (default $i=5$) scene point is used as the reference point. We use this reference point to calculate PPFs in conjunction with all other scene points. We also map it to the model reference point and angle α_m by matching using the previously constructed hash table. This process effectively solves the correspondence problem between point pairs by matching point pairs with the same quantized PPF. In the second part, α_s is calculated for the scene point pairs. α_s represents the angle between the connected projection of the scene point pairs and the positive direction of the y-axis. For each matched point pair feature, the angle $\alpha=\alpha_\mathrm{m}-\alpha_\mathrm{s}$ is found, and then voting is performed in the Hough space of $(\boldsymbol{p}_\mathrm{r},\alpha)$. The cell with the maximum number of votes in Hough space is extracted to form a pose hypothesis. After valid candidates have been generated for all reference points, we cluster similar poses, those with rotations and translations lower than thresholds. The group with the highest total number of votes is the resulting pose hypothesis.

4 Algorithm

4.1 Overview

We propose a new 6D pose estimation algorithm, the framework of which is shown in Fig. 3. Based on PPF, we mainly make the following improvements. Firstly,

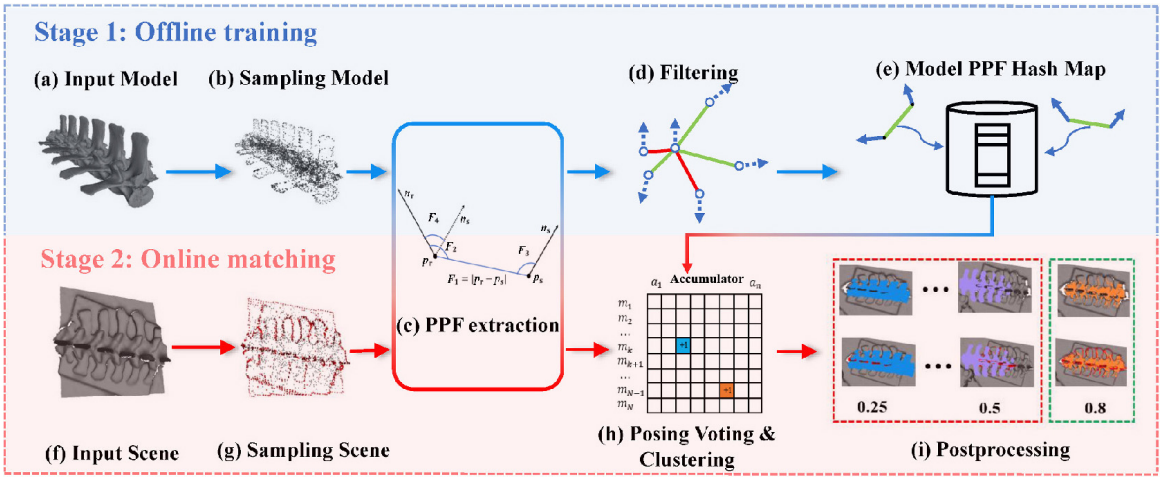

Fig. 3 Framework of the proposed method. It has two main stages: offline training and online matching. In the former, a CAD model is input (a). After downsampling (b), PPFs are extracted from the model (c). During filtering (d), we remove PPFs with angles higher than 175° or lower than 5° by considering normal vector angle differences. PPFs are extracted and stored in a hash table (e). In the online matching stage, the scene point cloud is input (f). It is pre-processed (g), using a clustering down-sampling method that takes into account the normal vector information; it focuses on edge points and points with large curvature. PPFs extracted from the scene point cloud (c) are matched to the hash table, and candidate poses are generated by voting and pose clustering (h). Each candidate pose is then post-processed (i). The pose with highest matching score is selected by an improved edge-based pose verification method. Finally, we use ICP to refine that pose.

when pre-processing the input model, we remove point pair features that interfere with the matching based on the normal vector angle for the input model. Secondly, when pre-processing the scene point cloud, we use a clustered down-sampling method that preserves edges in the point cloud. Finally, the pose verification operation is performed by checking the matching degree of edges to filter out wrong poses. These improvements are described further in the following sections.

4.2 Offline training

In the offline training phase, all point pair features of the model are extracted and stored in a hash table to create a global model description. However, due to self-occlusion, the global description contains some redundant point pair features that never appear in the input scene. The redundant point pair features not only increase the search time during the online matching phase but also increase the matching error. To mitigate the negative impact of redundant point pair features, we adopt a method based on Ref. [25] to determine the visibility of point pair features by using the normal vector angle between point pairs. If the angle between the normal vectors of two oriented points is higher than 175°, we consider the point pair to be almost invisible. Such point pair features are not stored.

Furthermore, it is common for the traditional PPF method to degrade when an object has many repetitive features, such as large planes. Therefore, we do not store pairs whose normal vector angle for the two oriented points differ by less than 5°, so that the algorithm focuses on geometrically-rich point pair features. As Fig. 4 shows, we mainly filter out the points that are self-obscured due to the viewpoint and points that lie on the same plane.

4.3 Online matching

4.3.1 Pre-processing

In order to accelerate the computation of object poses, the scene point cloud must be down-sampled.

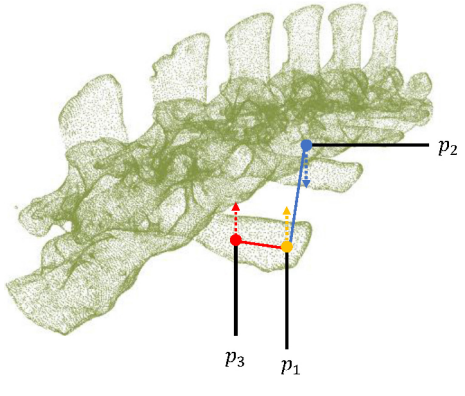

Fig. 4 Suppose p_1 is used as the reference point. p_2 has a normal vector angle of more than 175° to p_1, so does not appear in the same view due to the visibility constraint of the viewpoint. Because of the specificity of the plane structure, the points in the same plane such as p_3 are often mapped to the same hash bin in the hash table, which reduces the performance of the algorithm.

Unlike Drost's method [20], we use a clustering down-sampling method that takes into account normal, like Refs. [26, 27]. However, we also focus on edge points in the point cloud. Edge points can robustly describe the shape of the object, and for complex objects such as spinal bones, feature points have a higher probability of being present at edges. Our approach is shown in Fig. 5, where we first create a multi-resolution grid structure to discretize the scene point cloud according to the diameter of the model. Similar points with normal angle difference less than the threshold θ are then merged in a voxel grid. After this first fine-grained sampling, we extract the edge point clouds and continue with a fine-to-coarse multi-resolution sampling strategy for the non-edge points. To prevent some geometric features from being filtered out in the coarse-grained grid, the threshold θ is gradually reduced proportionally. The above operation can effectively preserve edge points and points with large curvature.

4.3.2 Feature extraction

For scene point clouds, we follow the solution proposed in Ref. [20], choosing 1/5th of the points in the scene as reference points and other points as the second point of the point pair feature. To improve the efficiency of the matching part, we use a kd-tree structure and adopt the intelligent sampling strategy of Hinterstoisser et al. [28] to select other points within the model diameter d from the model to construct as point pairs.

4.3.3 Pose clustering

To merge similar candidates, we use a hierarchical clustering method [26]. If the rotation and transformation between the two candidate poses are lower than the predefined threshold, the two candidate poses are grouped. All poses within each cluster follow the same conditions based on the two thresholds of rotation and transformation. Finally, the quaternion average for each cluster is used to calculate a new candidate pose, and the score of each pose is added up to the score of the new candidate pose.

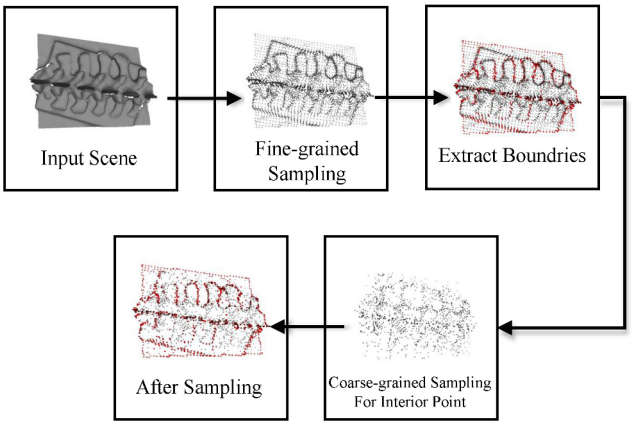

Fig. 5 Flowchart for clustered down-sampling method considering edge information.

4.3.4 Post-processing

The score of each pose is obtained by adding the votes of the candidates in the cluster. In the presence of sensor noise and background clutter, the score of the poses may not correctly represent the degree of matching. Therefore, we recommend that a more reliable score be calculated through an additional re-scoring process. We observed that in most approaches [26–29], most of the computational time is spent on pose verification. So, for speed of pose estimation, we propose an edge-based pose hypothesis verification method with an early exit strategy.

Edges are distinctive features of an object and can well represent the shape characteristics and contours of the object. Using the edge information from the point cloud, it is possible to select the correct pose from a set of candidate poses with high probability. In our pose hypothesis verification method, for the input candidate pose, the axis-aligned bounding box (AABB) of the computed candidate pose is used as the region of interest (ROI). Edge points within the ROI are clustered, and the distance between the edge clustering center and the center of the candidate pose is computed to remove remote and divergent edge points. The reason for filtering based on distance to the centroid of edge clustering is that often cluttered edges that are not in the object are discontinuous and distant. The final degree of edge matching score for this candidate pose is given by

$$S = \frac{N_{\text{Matching}}}{N_{\text{ROI}}} \tag{2}$$

N_{ROI} denotes the number of edge points in the ROI (red and blue in Fig. 6) after filtering out outliers

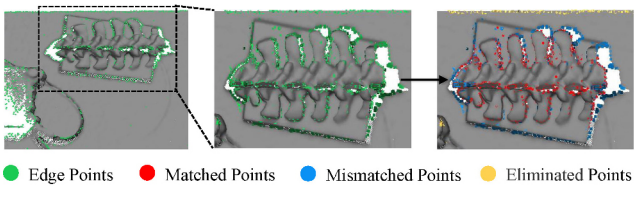

Fig. 6 Classification of edge points in the ROI.

(yellow in Fig. 6). N_{Matching} is the number of edge points close to the candidate poses (red in Fig. 6).

Detailed steps in the pose verification process are as follows:

- The input candidate poses are sorted according to the number of votes; the maximum number of votes for the candidate poses is $V_{\max}$. The candidate poses are divided into two categories according to $V_{\max}$. The first contains the candidate poses whose number of votes greater than $V_{\max}/2$, so are more likely to be the correct pose. The second category contains the remaining candidates. The number of candidates in this category is much larger.
- For the first category of candidate poses, we use a kd-tree to quickly see how well each pose matches the edges of the scene. Edge points close to the model indicate support for the pose hypothesis. The N candidate poses with the highest scores (the value of N is given in Section 5.4) are selected for more detailed filtering using Eq. (2). We do not directly use the edge match to all points in the scene point as correctness of the match is greatly reduced when the scene is prone to clutter. If the pose score computed by Eq. (2) is higher than 0.7, it is directly considered to be the correct pose and subsequent computation is stopped. If the score is lower than 0.7 but higher than 0.6, the pose with the highest score among the N poses is selected.
- If the score is higher than 0.6, poses in the second category are processed in the same way as for poses in the first category. If the score for the N poses of the second category is also under 0.6, the pose with the highest score from the $2N$ candidate poses is selected as the final pose.

After selecting the final pose, ICP [13] is used to further refine the pose to improve the accuracy of the match.

4.4 Design of robot-operated positioning experiment

4.4.1 Hardware

The hardware used in our experiment is shown in Fig. 7. The 3D camera used is the Azure Kinect DK depth camera. The robotic arm is the AUBO collaborative robot with six joints for flexible operation, and is used to perform fixed-point movements to complete operations on the spine. The medical drill is fixed at the end of the robotic arm and is equipped with various drill holes, adjusted for different speeds, pointing at the spine. We simulate the platform as well as using real equipment.

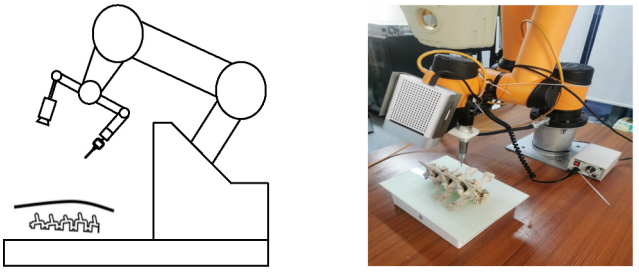

Fig. 7 Hardware used in our experiments. Left: schematic diagram. Right: photo.

4.4.2 Position-based visual servoing scheme

Visual servoing uses visual information extracted from images or point clouds captured by one or more cameras to control the motion of a robot. Visual servoing is a closed-loop system in which vision analysis provides guidance for the robot and robot motion provides new vision analysis for the camera. Closed-loop design can effectively improve success and reduce deviation.

We use a position-based visual servoing scheme, as shown in Fig. 8. The input is the difference between the detected actual pose of the spine and the desired spine pose. The output controls the robot velocity, to make the robot move quickly to the target pose state. After the instruction is completed, the camera continues to receive feedback of the robot state, forming a closed-loop control system. The closer the real pose is to the desired pose, the smaller the speed of the robot arm will be. When the difference is less than the threshold we set, the speed of the robot arm is 0, and the servo stops.

4.4.3 Transformation relationship analysis

In order to control the drill mounted on the robotic arm to drill in the attitude we specify, we perform coordinate transformation. Transformations relate

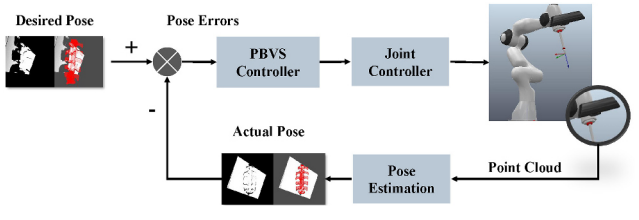

Fig. 8 Position-based visual servoing scheme.

the model of the spine, the fixed drill, the depth camera, and the end of the robotic arm.

First, based on the preoperative surgeon's design, we obtain the target drill pose and position in the spine model coordinate system in advance; we denote it $T_{\mathrm{s}}^{\mathrm{thope}}$. Next, a hand-eye calibration process gets the matrix $T_{\mathrm{c}}^{\mathrm{e}}$ that converts the coordinate system of the camera to the coordinate system of the end of the robotic arm. Then the tool calibration process gets the matrix $T_{\mathrm{t}}^{\mathrm{e}}$ that converts the coordinate system of the fixed drill to the coordinate system of the end of the robotic arm. Since both the camera and the drill are fixed to the robot arm, the conversion relationship is fixed, and $T_{\mathrm{t}}^{\mathrm{c}}$ means that the coordinate system of the fixed drill is converted to the coordinate system of the camera. Thus, in Section 4.4.2, the desired pose of the camera with respect to the spine is

$$T_{\mathrm{s}}^{\mathrm{Chope}} = T_{\mathrm{t}}^{\mathrm{c}} \cdot T_{\mathrm{s}}^{\mathrm{thope}} \tag{3}$$

The transformation matrix $T_{\mathrm{s}}^{\mathrm{c}}$ from the spine model coordinate system to the camera coordinate system is obtained from the above pose estimation algorithm. $T_{\mathrm{s}}^{\mathrm{c}}$ is the current pose in Section 4.4.2. The gap between $T_{\mathrm{s}}^{\mathrm{c}}$ and $T_{\mathrm{s}}^{\mathrm{Chope}}$ is reduced by the position-based visual servoing scheme.

5 Experiments

In this section, after describing the datasets required for the experiments, the evaluation criteria, and the open-source methods used for comparison, we first evaluate the impact of different parameters on the real spine dataset. Then, in Sections 5.5 and 5.6, a real spine dataset and a publicly available dataset are tested to investigate the robustness of the algorithm and to validate the algorithm design. In Section 5.7, we evaluate our method quantitatively and qualitatively on the real spine dataset and show the result of the robot-operated positioning experiment. Finally, to demonstrate the effectiveness of our pose estimation method for complex, symmetric objects, and its generality for objects of different shapes, we perform a comprehensive comparison of recognition rates and efficiency with state-of-the-art methods on two well-known publicly available datasets in Section 5.8.

The algorithm proposed in this paper was implemented using the Point Cloud Library (PCL) and tested on a PC with a 3.6 GHz Intel i9-10850K CPU and 16 GB of RAM. The algorithm uses OpenMP technology to improve matching speed.

5.1 Datasets

5.1.1 Pubic datasets

The public datasets contain both UWA dataset [30] and DTU dataset [31]. The UWA dataset contains 5 complete 3D models as well as 50 2.5D scenes, where the rhino models are mainly used for interference. Each 2.5D scene contains 4–5 models, and the degree of model occlusion ranges from 65% to 95%. 5 models and some scenes are shown in Fig. 9(a). The DTU is a large dataset consisting of 45 objects and 3204 scenarios captured by a structured light scanner, each of which contains 10 objects. These objects belong to three different types: geometrically complex models, cylindrical and planar models. Because some objects are highly occluded. We do not consider objects with more than 98% occlusion. The DTU dataset is challenging because of the high occlusion, high similarity, and diversity of models. Some of the models and scenes are shown in Fig. 9(b).

5.1.2 Spine dataset

To validate the effectiveness of our algorithm for spinal bone pose estimation, we constructed a real dataset using a pig spine. The spine model point cloud used CT scanning of the spine for accurate reconstruction. The professional medical software Mimics Research was used to convert medical data in DICOM format into 3D models. Our experimental platform was used to collect three types of spine datasets with an Azure Kinect DK depth camera:

- less occluded far-field spine scenes (S1, Fig. 10(b));
- more occluded near-field spine scenes (S2, Fig. 10(c));
- cluttered randomly placed spine scenes (S3, Fig. 10(d)).

S1–S3 each have 40 scenes, with a total of 120 scenes.

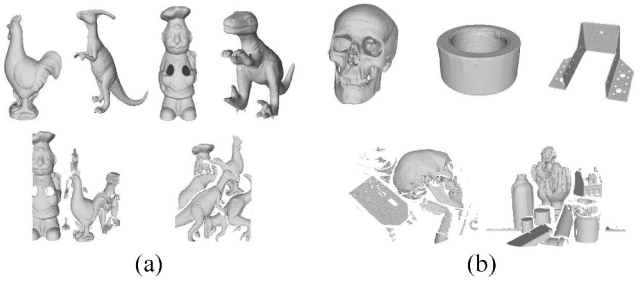

Fig. 9 Various object models and two random scenes in (a) UWA dataset and (b) DTU dataset.

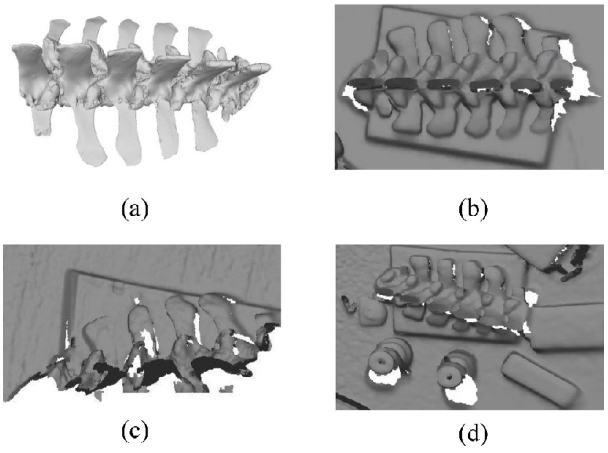

Fig. 10 Spine dataset: (a) spine model, and (b–d) S1–S3 respectively.

5.2 Evaluation criteria

To determine pose accuracy, we adopt the average distance metric (ADM) [32] as the pose error metric. It considers both the visible and invisible parts of the 3D model surface. ADM measures the mean Euclidean distance between the model points in the estimated pose $\hat{\boldsymbol{T}}$ and in the true pose $\bar{\boldsymbol{T}}$. In Ref. [27], two alternatives to ADM (ADD and ADI) are used for objects that do not have symmetry and those that do. We also use these evaluation criteria. We accept the pose estimation as positive if the pose error is less than ζ_{e}, where ζ_{e} is related to the object diameter d. The ADD and ADI pose error metrics are given by

$$e_{\mathrm{ADD}} = \operatorname*{avg}_{\boldsymbol{x}\in\mathcal{M}} \|\overline{\boldsymbol{T}}\boldsymbol{x} - \hat{\boldsymbol{T}}\boldsymbol{x}\|_2 \tag{4}$$

$$e'_{\mathrm{ADI}} = \max\left(\operatorname*{avg}_{\boldsymbol{x}_1\in\mathcal{M}} \min_{\boldsymbol{x}_2\in\mathcal{M}} \left\|\overline{\boldsymbol{T}}\boldsymbol{x}_1 - \hat{\boldsymbol{T}}\boldsymbol{x}_2\right\|_2, \left\|\overline{\boldsymbol{T}}\boldsymbol{c}_{\mathrm{o}} - \hat{\boldsymbol{T}}\boldsymbol{c}_{\mathrm{o}}\right\|_2\right) \tag{5}$$

where $\mathcal{M}$ is the point cloud of model and $\boldsymbol{c}_{\mathrm{o}}$ is the object center. e_{ADD} computes the average Euclidean distance of the same points after transformation, while e_{ADI} computes the average Euclidean distance of the two closest points after transformation and also takes into account the distance to the object center.

We use two evaluation criteria, recognition rate (RR) and mean recall (MR) to evaluate the accuracy of the algorithm. RR is the ratio of correct poses to all detected poses. MR is the average recognition rate for all objects and is used to measure the quality of the algorithm over the entire dataset:

$$\mathrm{MR} = \operatorname*{avg}_{o\in O}\left(\sum_{s\in S}|P(o,s)|/\sum_{s\in S}|G(o,s)|\right) \tag{6}$$

where O and S are the sets of all template objects and scenes respectively, $P(o,s)$ is the set of correctly detected poses, and $G(o,s)$ is the set of ground-truth poses of object o in scene s.

5.3 Baselines

We compare our method with several baseline algorithms using only depth images as input: these include Drost-PPF [20] and Buch-17 [33]; we choose the commercial machine vision software MVTec HALCON to implement the original PPF and the optimization algorithm. The open-source method Buch-17 [33] is a 3D object recognition method. It uses various three-dimensional local feature descriptors to find point pair correspondences that are constrained to vote in a 1-DOF rotation subgroup of the entire pose, SE(3). Kernel density estimation allows for an efficient combination of voting to determine the resulting pose. The method relies on three-dimensional local feature descriptors, and is evaluated using several descriptors: ECSAD [34], NDHist [35], SI [7], SHOT [36], FPFH [8], and PPF [20].

5.4 Parameter settings

In this section, we use the spine dataset to determine parameter settings. To do so, we use the variable control method for parameter validation. If the parameter does not have a determined value, we use the default value for the assignment. We analyze the following four parameters: the distance quantization step of Δdist, the angle quantization step Δangle, the number of poses in the pose verification function N, and the size of AABB box s. Δdist is related to the diameter of the model. As Fig. 11 shows, the best results are obtained with $\Delta\mathrm{angle} = 5$ and $\Delta\mathrm{dist} = 0.02$. An axis-aligned bounding box (AABB) is used as the ROI in the pose verification function. The larger the AABB, the more points around the pose are considered, so it is easy to filter out some poses that only partially match the spine. We hope to determine the correctness of the poses by considering the matching degree of the points in the AABB box, but when the AABB box is larger than a certain degree, the pose accuracy is susceptible to the influence of outliers and tends to decrease, so we set the AABB box size to 140%. The more the selected poses, the better the results, but taking into account the time consumption, we set $N = 9$.

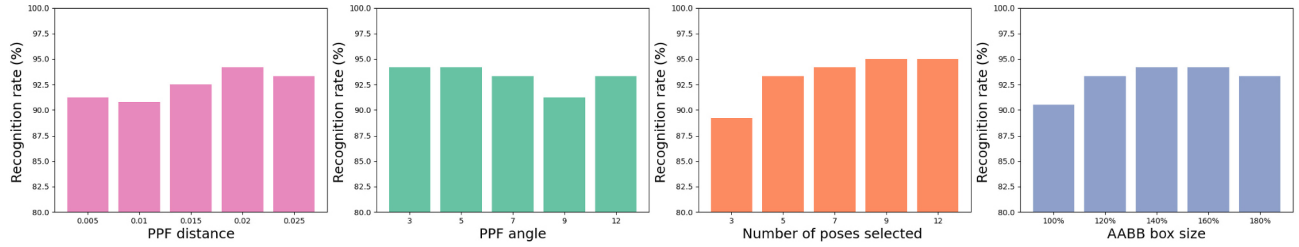

Fig. 11 Parameter analysis for spine dataset. Default values of these parameters are: quantization step for distance $\Delta\text{dist} = 0.025$, quantization step for angle $\Delta\text{angle} = 5$, number of poses in pose verification function $N = 10$, and size of AABB box $s = 150\%$.

5.5 Quality and robustness

We next test the robustness of our method in the presence of Gaussian noise using the real bone dataset and the open dataset UWA. We randomly added Gaussian noise with different standard deviations to the point coordinates using standard deviations of 0.0, 0.5, 1.0, 1.5, and 2.0 mm. Table 1 shows the results. Quality decreases slightly as the noise level increases, but we still perform well on the noisy data.

5.6 Ablation study

5.6.1 Effectiveness of sampling

To validate the contribution of sampling in our method, we compare it to a sampling method [27] that does not emphasize edge points. In order to make the number of points sampled by the method focusing on edge points smaller or equal to the compared method, we perform an additional sampling step for non-edge points. As Table 2 shows, higher recall is achieved for sampling more focused on edge points, which we attribute to the fact that stable features are more present on the contours of the object. Increasing the number of edge points sampled can improve matching results.

5.6.2 Effectiveness of our pose verification

We next compared our edge-based post-processing method and the pose verification method in Ref. [29]. In Ref. [29], scoring is based on the overlap of surfaces, and those model points that are close to the scene vote to indicate support for the pose hypothesis. As shown in Table 3, our edge-based post-processing approach is more discriminative. Edge information can robustly describe the geometric contours of the object. When in the ROI region, the better the matching of edge points, the higher the probability that this is the correct pose.

Table 1 Results of our algorithm with added noise

Dataset	ζ_e	Noise=0	Noise=0.5	Noise=1.0	Noise=1.5	Noise=2.0
Spine dataset	$0.05d$	93.33	89.17	85.83	80.83	78.33
	$0.1d$	95.00	90.00	87.50	83.33	80.83
UWA dataset	$0.05d$	99.47	98.94	94.15	89.36	86.17
	$0.1d$	100.00	98.94	94.15	90.43	86.70

Table 2 Validation of edge-based sampling method

Sampling method	ζ_e	RR_{S1}	RR_{S2}	RR_{S3}	$RR_{chicken}$	RR_{para}	RR_{cheff}	RR_{Trex}
Ref. [27]	$0.05d$	**92.5**	77.5	90.0	**97.9**	97.8	**98.0**	97.8
	$0.1d$	**92.5**	80.0	90.0	97.9	97.8	98.0	**100.0**
Our method	$0.05d$	**92.5**	**80.0**	**92.5**	**97.9**	**100.0**	**98.0**	**100.0**
	$0.1d$	**92.5**	**82.5**	**92.5**	**100.0**	**100.0**	**100.0**	**100.0**

Table 3 Validation of our pose verification method

Sampling method	ζ_e	RR_{S1}	RR_{S2}	RR_{S3}	$RR_{chicken}$	RR_{para}	RR_{cheff}	RR_{Trex}
Ref. [29]	$0.05d$	90.0	82.5	85.0	91.7	91.1	98.0	95.6
	$0.1d$	92.5	82.5	85.0	93.8	91.1	98.0	95.6
Ours	$0.05d$	**95.0**	**85.0**	**100.0**	**97.9**	**100.0**	**100.0**	**100.0**
	$0.1d$	**97.5**	**87.5**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

5.6.3 *Effectiveness of early exit strategy on performance*

We next considered the effect on speed of our pose verification function, and compare three ways of using the pose verification function. In the first, poses are not classified and then post-processed. The second approach is as described in Section 4.3.4, but without using an early exit strategy. The third is our full method, using the early exit strategy when the threshold is exceeded. As Fig. 12 shows, the third is the fastest. Our pose classification and early-exit strategy improve efficiency. Pose classification increases speed because poses with larger scores are more likely to be the correct pose. Processing the few such poses can reduce the time significantly.

5.7 Effectiveness of the prototype system in operation

5.7.1 *Recognition results on the spine dataset*

As Table 4 shows, our algorithm achieves excellent results in terms of correctness, and outperforms the other competitors. In terms of speed, the commercial HALCON software is the fastest because it makes full use of the hardware and is also fully optimized at each step. Compared to Ref. [33], our method is faster than most 3D descriptor algorithms. Our algorithm has potential to be further accelerated at each step on the GPU for surgical navigation applications. Figure 13 shows a qualitative comparison of these methods for several scenes.

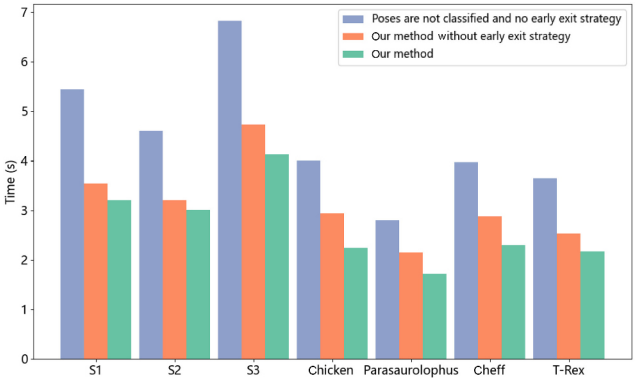

Fig. 12 Comparison of time taken for three ways of using the pose verification functions for data from UWA and spine datasets.

5.7.2 *Results of navigation and positioning*

In order to verify the effectiveness of the robot control method, we modeled the scheme in the simulation environment. Figure 14 shows the visualization interface, simulated in CoppeliaSim. In the simulation environment, camera intrinsics, hand-eye calibration

Table 4 Comparison of eight algorithms on the spine dataset

Method	ζ_e	RR_{S1}	RR_{S2}	RR_{S3}	MR	Time (s)
Buch-17-ECSAD	0.05d	92.5	77.5	87.5	85.8	2.83
	0.1d	95.0	77.5	87.5	86.7	
Buch-17-FPFH	0.05d	92.5	70.0	77.5	80.0	8.76
	0.1d	92.5	70.0	77.5	80.0	
Buch-17-NDHist	0.05d	95.0	77.5	92.5	88.3	3.28
	0.1d	95.0	80.0	92.5	89.2	
Buch-17-PPF	0.05d	**97.5**	80.0	90.0	89.2	3.62
	0.1d	**97.5**	80.0	92.5	90.0	
Buch-17-SHOT	0.05d	90.0	65.0	80.0	78.3	5.96
	0.1d	90.0	65.0	80.0	78.3	
Buch-17-SI	0.05d	92.5	72.5	87.5	84.2	3.11
	0.1d	92.5	72.5	87.5	84.2	
HALCON	0.05d	**97.5**	82.5	**100.0**	**93.3**	**1.78**
	0.1d	**97.5**	82.5	**100.0**	93.3	
Ours	0.05d	95.0	**85.0**	**100.0**	**93.3**	3.45
	0.1d	**97.5**	**87.5**	**100.0**	**95.0**	

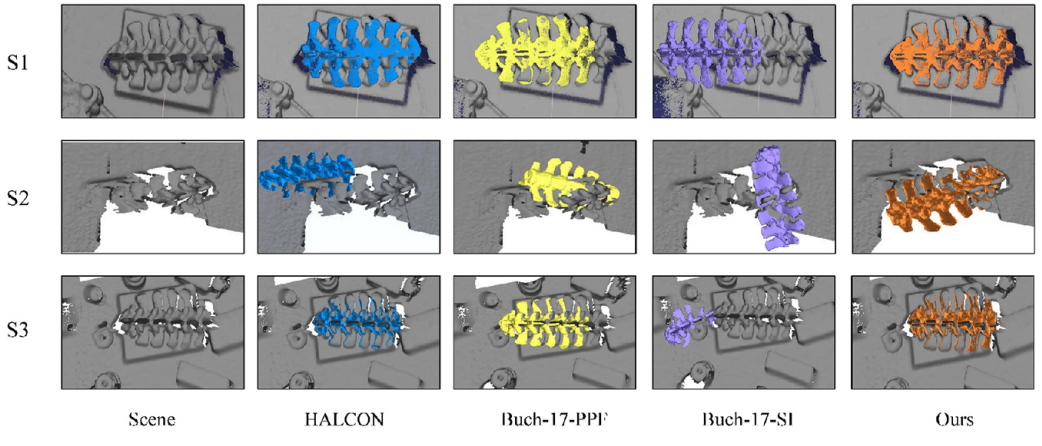

Fig. 13 Qualitative comparison of results from various methods for scenes S1–S3 from the spine dataset.

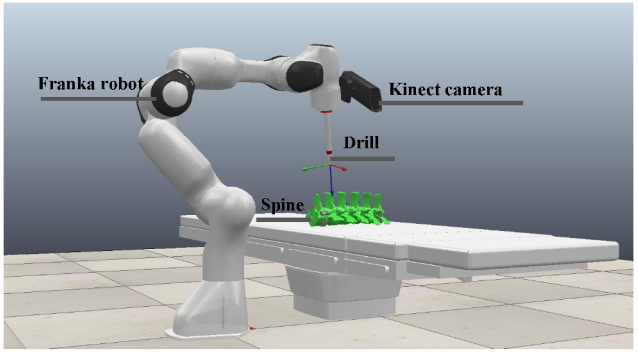

Fig. 14 Simulation environment.

parameters, and tool calibration parameters can be directly calculated. However, in the real scene, these parameters can only be obtained by calibration, and there are errors in the calibration process, which can not be accurately determined. To simulate the real situation, we added noise to these parameters. Based on our experience of real scenarios, we added Gaussian noise with $\sigma = 5$ for f_x, f_y and $\sigma = 1$ for c_x, c_y in the camera intrinsics. Gaussian noise with $\sigma = 0.01$ was added for the rotation and translation vectors of the calibration parameters.

In this setting, the robot arm performs a movement of two seconds at a time. During the simulation, the motion trajectory of the camera's optical center (Fig. 15(a)), camera velocities (Fig. 15(b)), and visual features error (Fig. 15(c)) were recorded. It can be seen from the change of camera speed and feature errors that the closer the drill is to the target pose, the lower the speed of the robot arm. The calculated

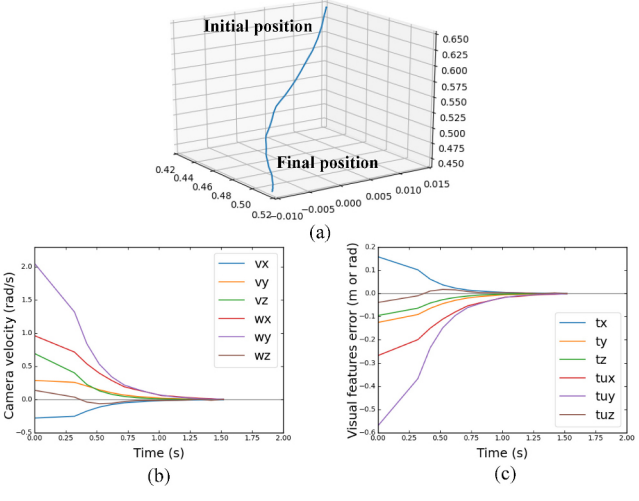

Fig. 15 Experimental results for the simulation. (a) Motion trajectory of the camera's optical center in Cartesian space. (b) Camera velocities. (c) Visual feature error.

tip distance error is within 1 mm and the orientation error is within 1°.

Figure 16 shows the qualitative experimental results in the real environment. The left is a pose diagram of the prescribed drill, and the right is the robotic arm's effect.

5.8 Recognition results on public datasets

To demonstrate not only the high recognition rate of our algorithm for complex and symmetric objects like the spine, but also the effectiveness of our algorithm for objects of other shapes, we tested it using the public UWA and DTU datasets.

Table 5 shows the recognition results of our algorithm and the other seven algorithms on the UWA dataset. In terms of speed, our algorithm is superior to the others apart from the commercial HALCON software. In terms of recognition accuracy, we achieve a 100% recognition rate for most objects, surpassing the other algorithms even in highly occluded cases. Figure 17 shows that for the UWA dataset, our algorithm still gives stable and correct results in the case of strong occlusion.

The DTU dataset contains many different types of geometric models. In order to more clearly show the effect of our algorithm on different geometric structures, we artificially divided the DTU dataset into geometrically complex, planar, and cylindrical models (see the Appendix).

We selected some complex and symmetric objects with bone properties from the DTU dataset. A quantitative comparison of results of these eight algorithms is given in Table 6, which shows the clear advantage of our algorithm for this type of object.

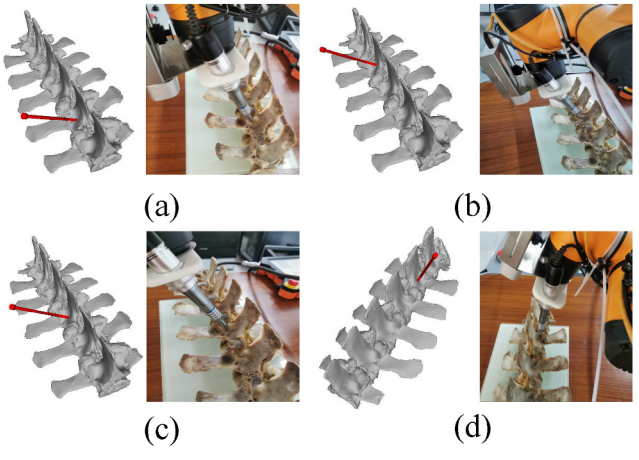

Fig. 16 Predetermined pose and actual effect.

Table 5 Comparison of eight algorithms on the UWA dataset

Method	ζ_e	$RR_{chicken}$	RR_{para}	RR_{cheff}	RR_{Trex}	MR	Time (s)
Buch-17-ECSAD	0.05d	93.75	80.00	**100.00**	80.00	88.83	3.08
	0.1d	93.75	80.00	**100.00**	80.00	88.83	
Buch-17-FPFH	0.05d	91.67	84.44	88.00	86.67	87.77	6.12
	0.1d	91.67	84.44	92.00	88.89	89.36	
Buch-17-NDHist	0.05d	93.75	95.56	**100.00**	**100.00**	97.34	3.61
	0.1d	93.75	95.56	**100.00**	**100.00**	97.34	
Buch-17-PPF	0.05d	93.75	95.56	**100.00**	**100.00**	97.34	3.77
	0.1d	93.75	95.56	**100.00**	**100.00**	97.34	
Buch-17-SHOT	0.05d	89.58	75.56	98.00	71.11	84.04	3.95
	0.1d	89.58	75.56	98.00	71.11	84.04	
Buch-17-SI	0.05d	93.75	91.11	**100.00**	97.78	95.74	4.12
	0.1d	93.75	91.11	**100.00**	97.78	95.74	
HALCON	0.05d	91.67	91.11	98.00	95.56	94.15	**0.4**
	0.1d	91.67	91.11	98.00	97.88	94.68	
Ours	0.05d	**97.92**	**100.00**	**100.00**	**100.00**	**99.47**	2.11
	0.1d	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	

Table 6 Quantitative comparison of results for various complex, symmetric DTU models

Method	ζ_e	RR_1	RR_2	RR_3	RR_4	RR_5	RR_7	RR_{26}	RR_{36}	RR_{37}	RR_{38}	RR_{39}	RR_{46}	MR
Buch-17-ECSAD	0.05d	41.61	68.57	31.25	66.30	68.29	91.59	5.33	4.00	5.08	0.00	0.00	4.44	42.45
	0.1d	43.07	68.57	31.25	67.93	69.51	91.59	5.33	8.00	7.61	0.00	0.00	6.67	43.04
Buch-17-FPFH	0.05d	33.58	40.71	12.50	54.35	69.51	89.72	4.00	8.00	4.06	0.00	0.00	0.00	34.28
	0.1d	33.58	40.71	12.50	55.43	69.51	90.65	5.33	16.00	5.58	0.00	0.00	0.00	34.68
Buch-17-NDHist	0.05d	53.28	49.29	26.56	67.93	89.02	**93.46**	20.00	8.00	8.12	0.00	10.53	35.56	45.96
	0.1d	53.28	50.00	26.56	67.93	89.02	**93.46**	20.00	12.00	10.66	0.00	10.53	35.56	46.27
Buch-17-PPF	0.05d	**66.42**	72.86	39.06	77.72	93.90	**93.46**	34.67	16.00	14.21	2.63	15.79	60.00	57.05
	0.1d	**67.15**	75.00	40.63	79.89	93.90	**93.46**	36.00	20.00	16.24	2.63	15.79	62.22	57.77
Buch-17-SHOT	0.05d	33.58	50.71	18.75	57.07	57.76	76.64	5.33	4.00	6.09	2.63	0.00	4.44	35.35
	0.1d	34.31	52.14	20.31	58.15	57.76	76.64	9.33	12.00	10.15	2.63	0.00	8.89	36.48
Buch-17-SI	0.05d	64.23	63.57	37.50	72.83	86.59	91.59	20.00	16.00	10.15	0.00	0.00	15.56	50.49
	0.1d	64.23	65.71	40.63	72.83	86.59	92.52	24.00	20.00	16.75	2.63	0.00	15.56	51.57
HALCON	0.05d	48.91	67.86	40.63	68.48	81.71	90.65	61.33	20.00	**22.34**	42.11	10.53	51.11	56.24
	0.1d	48.91	67.86	42.19	69.48	81.71	91.59	61.33	**24.00**	**26.90**	47.37	10.53	**73.33**	57.32
Ours	0.05d	60.58	**79.29**	**64.06**	**89.67**	**95.12**	91.59	**70.67**	**24.00**	20.30	**60.53**	**36.84**	**66.67**	**66.04**
	0.1d	60.58	**79.29**	**64.06**	**90.76**	**95.12**	92.52	**78.67**	**24.00**	20.30	**60.53**	**42.11**	**73.33**	**67.21**

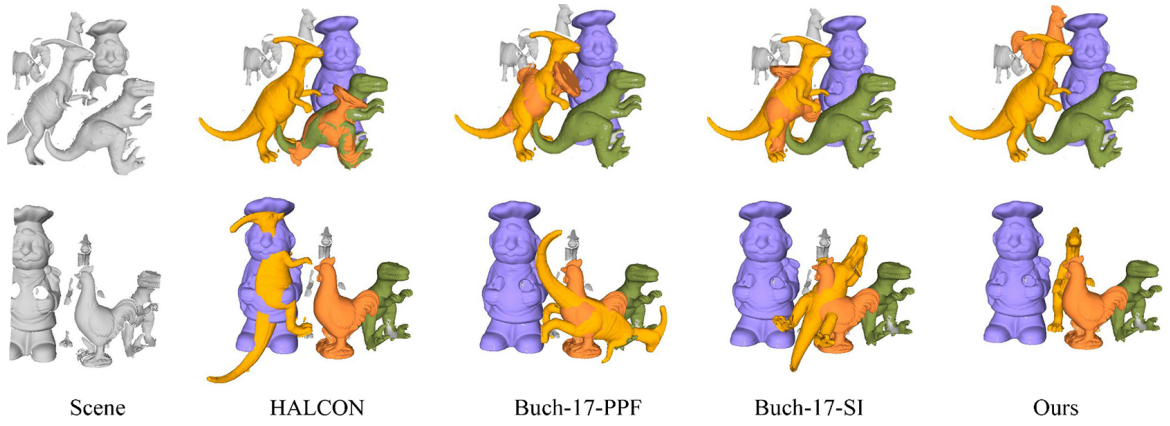

Fig. 17 Qualitative comparison of results for scenes from the UWA dataset.

We compared our algorithm to other algorithms for different geometric structures in the DTU dataset; the results are given in Table 7. It can be seen that our algorithm outperforms other matching algorithms for various types of 3D models in the DTU dataset. Thus, our algorithm has general applicability. A qualitative comparison for the DTU dataset is shown in Fig. 18.

Table 7 Comparison of eight algorithms on the DTU dataset

Method	ζ_e	Geometrically complex	Cylindrical	Planar	MR
Buch-17-ECSAD	0.05d	57.12	33.35	35.69	40.63
	0.1d	57.83	36.24	38.23	42.83
Buch-17-FPFH	0.05d	45.25	34.62	15.76	33.98
	0.1d	45.88	38.26	17.61	36.40
Buch-17-NDHist	0.05d	61.71	41.11	47.97	48.37
	0.1d	61.86	43.61	50.28	50.16
Buch-17-PPF	0.05d	71.60	51.95	61.88	59.53
	0.1d	72.31	54.05	64.58	61.37
Buch-17-SHOT	0.05d	48.50	27.07	36.85	35.13
	0.1d	49.36	30.23	39.04	37.45
Buch-17-SI	0.05d	61.95	38.96	27.00	43.21
	0.1d	62.73	42.23	28.89	45.73
HALCON	0.05d	69.38	45.90	55.62	54.54
	0.1d	70.49	49.45	56.89	56.95
Ours	0.05d	**82.20**	**57.13**	**71.73**	**67.18**
	0.1d	**82.67**	**60.02**	**73.23**	**69.11**

6 Conclusions

Considering the structural characteristics of the human spine, we have proposed a pose estimation method based on edge-enhanced point pair features. Its main features are an edge-based sampling method and an edge-matching-based pose verification method. We have performed extensive tests on the pig spine dataset and open datasets; they demonstrate that our method is suitable for automatic surgical navigation systems, having high accuracy, robustness, and good speed. Moreover, our algorithm is completely based on depth information for point cloud registration, and can serve as an excellent solution to light shading in surgical scenes.

Appendix

Figures 19 and 20 show our classification of shapes in the DTU dataset [31] according to symmetry and geometric properties.

Availability of data and materials

Open datasets in the paper are from public repositories (`https://roboimagedata.compute.dtu.dk/` and `http://vision.deis.unibo.it/keypoints3d/ds/UWA.7z`). The open-source algorithms used for comparisons are from public repositories (`https://www.mvtec.com/products/halcon` and `https://gitlab.com/caro-sdu/covis`).

Funding

This work was supported in part by the National Key R&D Program of China (2018AAA0102200), National Natural Science Foundation of China (62132021, 62102435, 61902419, 62002375, 62002376), Natural Science Foundation of Hunan Province of China (2021JJ40696), Huxiang Youth Talent Support Program (2021RC3071), and NUDT Research Grants (ZK19-30, ZK22-52).

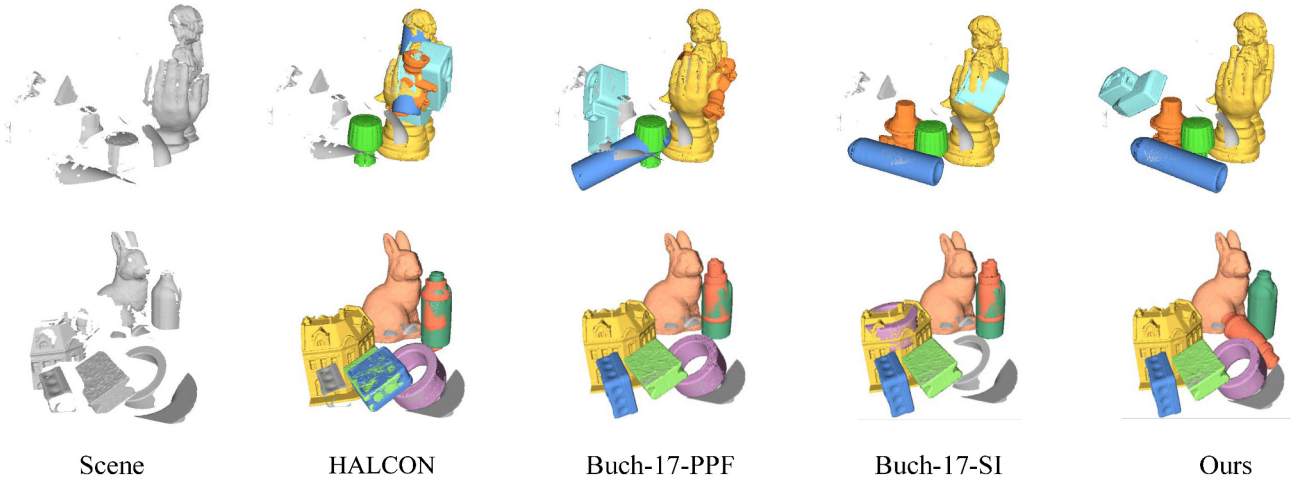

Fig. 18 Qualitative comparison of results for scenes from the DTU dataset.

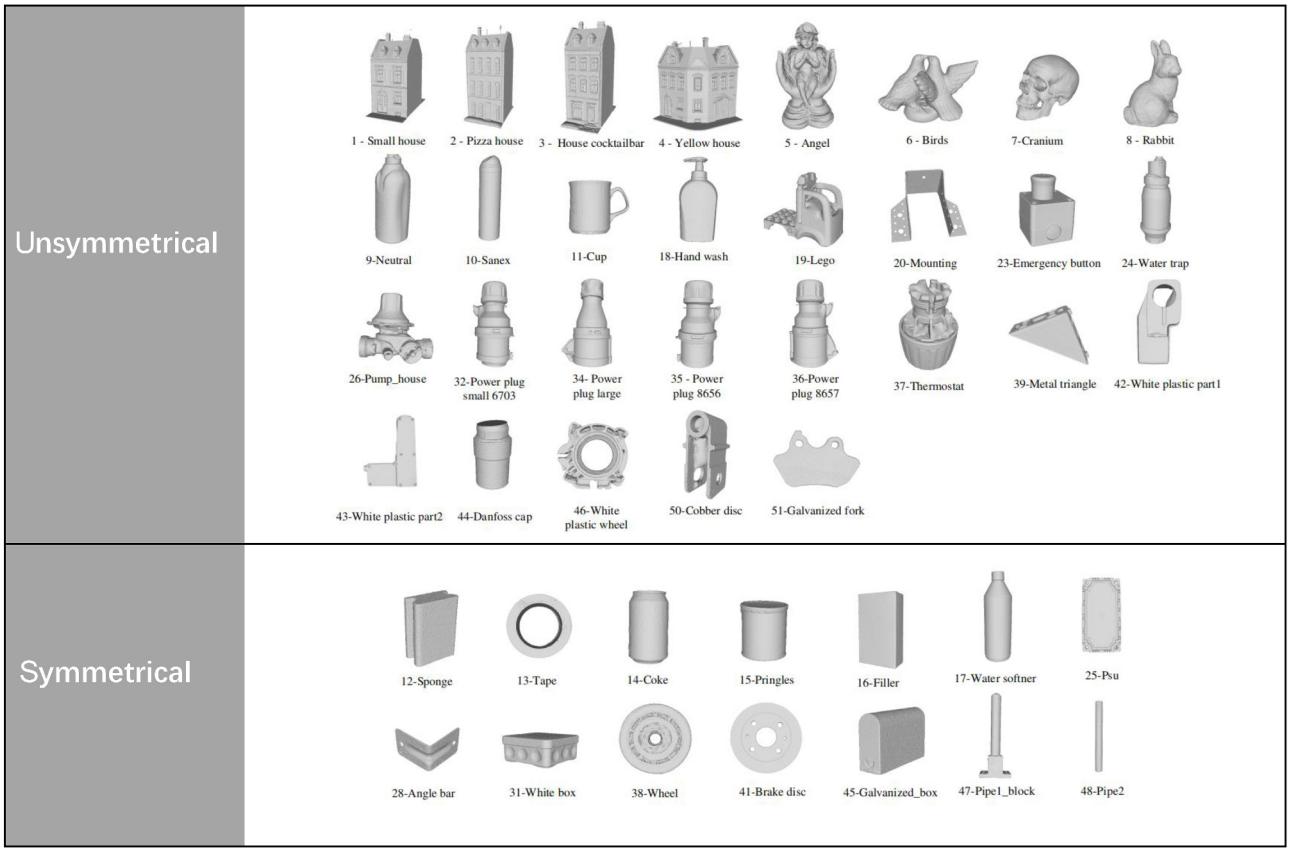

Fig. 19 Symmetry classification of the DTU dataset.

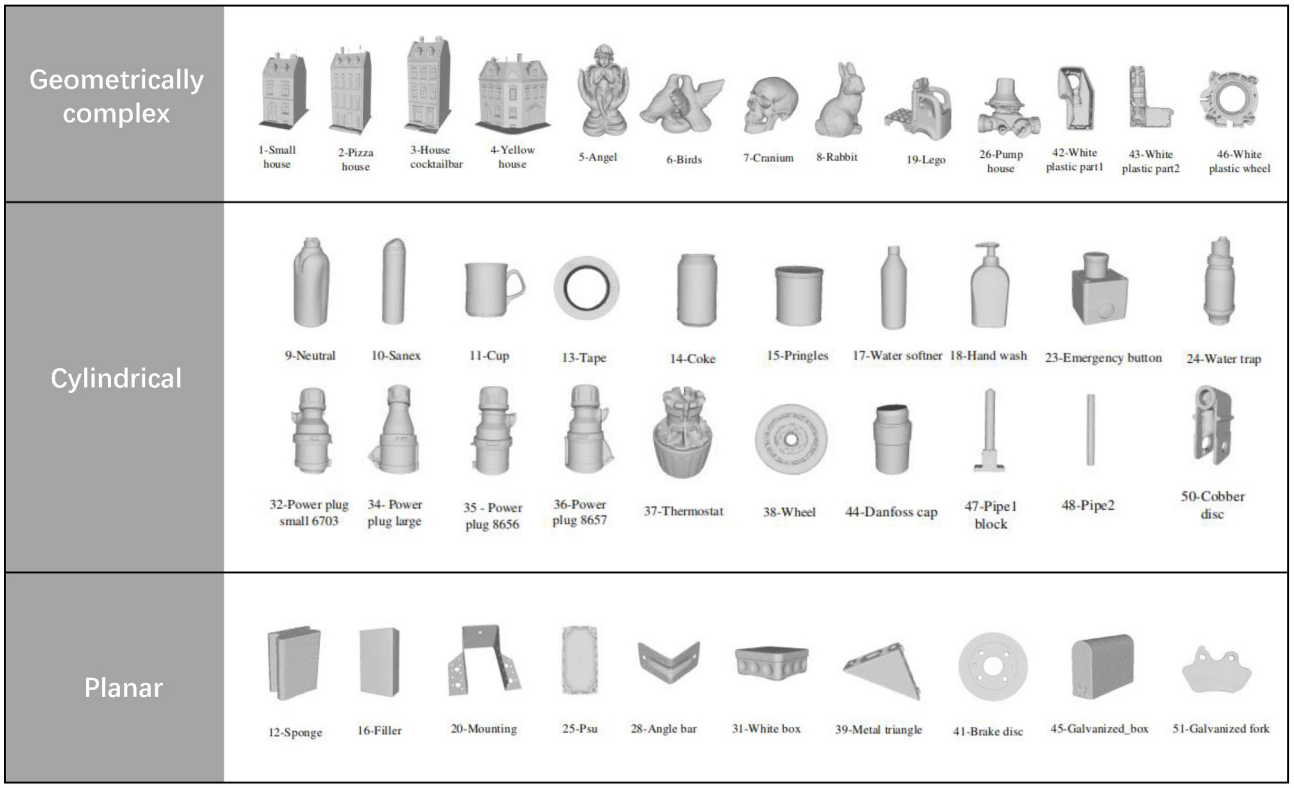

Fig. 20 Geometric classification of the DTU dataset.

Author contributions

Chenyi Liu: Methodology, Writing Draft, Visualization, Results Analysis; Fei Chen: Methodology, Supervision; Lu Deng: Supervision; Renjiao Yi: Supervision, Results Analysis; Lintao Zheng: Supervision; Chenyang Zhu: Supervision, Results Analysis; Jia Wang: Supervision; Kai Xu: Methodology, Supervision.

Acknowledgements

We thank Jiazhao Zhang and Yuqin Lan for helpful discussions.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Kai Xu is the Area Executive Editor of this journal.

References

[1] Li, R. T.; Si, W. X.; Liao, X. Y.; Wang, Q.; Klein, R.; Heng, P. A. Mixed reality based respiratory liver tumor puncture navigation. *Computational Visual Media* Vol. 5, No. 4, 363–374, 2019.

[2] Wang, Y.; Cao, D.; Chen, S. L.; Li, Y. M.; Zheng, Y. W.; Ohkohchi, N. Current trends in three-dimensional visualization and real-time navigation as well as robot-assisted technologies in hepatobiliary surgery. *World Journal of Gastrointestinal Surgery* Vol. 13, No. 9, 904–922, 2021.

[3] Kim, K.; Lee, S. Vertebrae localization in CT using both local and global symmetry features. *Computerized Medical Imaging and Graphics* Vol. 58, 45–55, 2017.

[4] Rusu, R. B.; Bradski, G.; Thibaux, R.; Hsu, J. Fast 3D recognition and pose using the Viewpoint Feature Histogram. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2155–2162, 2010.

[5] Marton, Z. C.; Pangercic, D.; Blodow, N.; Beetz, M. Combined 2D–3D categorization and classification for multimodal perception systems. *The International Journal of Robotics Research* Vol. 30, No. 11, 1378–1402, 2011.

[6] Madry, M.; Ek, C. H.; Detry, R.; Hang, K. Y.; Kragic, D. Improving generalization for 3D object categorization with Global Structure Histograms. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1379–1386, 2012.

[7] Johnson, A. E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21, No. 5, 433–449, 1999.

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

[9] Tombari, F.; Salti, S.; Di Stefano, L. Unique signatures of histograms for local surface description. In: *Computer Vision – ECCV 2010. Lecture Notes in Computer Science, Vol. 6313.* Daniilidis, K.; Maragos, P.; Paragios, N. Eds. Springer Berlin Heidelberg, 356–369, 2010.

[10] Rusu, R. B.; Holzbach, A.; Beetz, M.; Bradski, G. Detecting and segmenting objects for mobile manipulation. In: Proceedings of the IEEE 12th International Conference on Computer Vision Workshops, 47–54, 2009.

[11] Hinterstoisser, S.; Holzer, S.; Cagniart, C.; Ilic, S.; Konolige, K.; Navab, N.; Lepetit, V. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: Proceedings of the International Conference on Computer Vision, 858–865, 2011.

[12] Besl, P. J.; McKay, N. D. Method for registration of 3-D shapes. In: Proceedings of the SPIE Volume 1611, Sensor Fusion IV: Control Paradigms and Data Structures, 586–606, 1992.

[13] Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image and Vision Computing* Vol. 10, No. 3, 145–155, 1992.

[14] Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In: Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling, 145–152, 2001.

[15] Park, K.; Patten, T.; Vincze, M. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7667–7676, 2019.

[16] Hodaň T.; Baráth, D.; Matas, J. EPOS: Estimating 6D pose of objects with symmetries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11700–11709, 2020.

[17] Liang, H. Z.; Ma, X. J.; Li, S.; Görner, M.; Tang, S.; Fang, B.; Sun, F. C.; Zhang, J. W. PointNetGPD: Detecting grasp configurations from point sets. In: Proceedings of the International Conference on Robotics and Automation, 3629–3635, 2019.

[18] Lan, Y. Q.; Duan, Y.; Liu, C. Y.; Zhu, C. Y.; Xiong, Y. S.; Huang, H.; Xu, K. ARM3D: Attention-based relation module for indoor 3D object detection. *Computational Visual Media* Vol. 8, No. 3, 395–414, 2022.

[19] Zeng, L.; Lv, W. J.; Dong, Z. K.; Liu, Y. J. PPR-net: Accurate 6-D pose estimation in stacked scenarios. *IEEE Transactions on Automation Science and Engineering* Vol. 19, No. 4, 3139–3151, 2022.

[20] Drost, B.; Ulrich, M.; Navab, N.; Ilic, S. Model globally, match locally: Efficient and robust 3D object recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 998–1005, 2010.

[21] Choi, C.; Christensen, H. I. RGB-D object pose estimation in unstructured environments. *Robotics and Autonomous Systems* Vol. 75, 595–613, 2016.

[22] Drost, B.; Ilic, S. 3D object detection and localization using multimodal point pair features. In: Proceedings of the 2nd International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 9–16, 2012.

[23] Liu, D. Y.; Arai, S.; Miao, J. Q.; Kinugawa, J.; Wang, Z.; Kosuge, K. Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic Bin picking. *Sensors* Vol. 18, No. 8, 2719, 2018.

[24] Vock, R.; Dieckmann, A.; Ochmann, S.; Klein, R. Fast template matching and pose estimation in 3D point clouds. *Computers & Graphics* Vol. 79, 36–45, 2019.

[25] Lu, R. R.; Zhu, F.; Wu, Q. X.; Chen, F. J.; Cui, Y. G.; Kong, Y. Z. Three-dimensional object recognition based on enhanced point pair features. *Acta Optica Sinica* Vol. 39, No. 8, 0815006, 2019.

[26] Vidal, J.; Lin, C. Y.; Lladó, X.; Martí, R. A method for 6D pose estimation of free-form rigid objects using point pair features on range data. *Sensors* Vol. 18, No. 8, 2678, 2018.

[27] Guo, J. W.; Xing, X. J.; Quan, W. Z.; Yan, D. M.; Gu, Q. Y.; Liu, Y.; Zhang, X. P. Efficient center voting for object detection and 6D pose estimation in 3D point cloud. *IEEE Transactions on Image Processing* Vol. 30, 5072–5084, 2021.

[28] Hinterstoisser, S.; Lepetit, V.; Rajkumar, N.; Konolige, K. Going further with point pair features. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9907*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 834–848, 2016.

[29] Papazov, C.; Burschka, D. An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: *Computer Vision – ACCV 2010. Lecture Notes in Computer Science, Vol. 6492*. Kimmel, R.; Klette, R.; Sugimoto, A. Eds. Springer Berlin Heidelberg, 135–148, 2011.

[30] Mian, A. S.; Bennamoun, M.; Owens, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 28, No. 10, 1584–1601, 2006.

[31] Sølund, T.; Buch, A. G.; Krüger, N.; Aanæs, H. A large-scale 3D object recognition dataset. In: Proceedings of the 4th International Conference on 3D Vision, 73–82, 2016.

[32] Hodaň T.; Matas, J.; Obdržálek, Š. On evaluation of 6D object pose estimation. In: *Computer Vision – ECCV 2016 Workshops. Lecture Notes in Computer Science, Vol. 9915*. Hua, G.; Jégou, H. Eds. Springer Cham, 606–619, 2016.

[33] Buch, A. G.; Kiforenko, L.; Kraft, D. Rotational subgroup voting and pose clustering for robust 3D object recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 4137–4145, 2017.

[34] Jørgensen, T. B.; Buch, A. G.; Kraft, D. Geometric edge description and classification in point cloud data with application to 3D object recognition. In: Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Vol. 2, 333–340, 2015.

[35] Buch, A. G.; Petersen, H. G.; Krüger, N. Local shape feature fusion for improved matching, pose estimation and 3D object recognition. *SpringerPlus* Vol. 5, No. 1, 1–33, 2016.

[36] Salti, S.; Tombari, F.; Di Stefano, L. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding* Vol. 125, 251–264, 2014.

Chenyi Liu received her B.E. degree in software engineering from Tianjin Normal University, China, in 2020. She is now a master student in the National University of Defense Technology (NUDT), China. Her research interests cover 3D point cloud registration.

Fei Chen is a professor of spinal surgery in the Second Xiangya Hospital. His current interests lie in surgical robot perception and automatic navigation.

Lu Deng is a professor in the Surgery Department of the Second Xiangya Hospital. Her current interest is in automatic surgical navigation.

Renjiao Yi is an assistant professor in the School of Computing, NUDT. She received her Ph.D. degree from Simon Fraser University in 2019. She is interested in 3D vision problems such as inverse rendering and image-based relighting.

Lintao Zheng is an assistant professor in the College of Meteorology and Oceanography, NUDT. He earned his Ph.D. degree in computer science from NUDT. His research interests focus on 3D vision and robot perception.

Chenyang Zhu is an assistant professor in the School of Computing, NUDT. He received his Ph.D. degree from Simon Fraser University in 2019. His current directions of interest include 3D vision, and robot perception and navigation.

Jia Wang received her B.E. and M.E. degrees from NUDT. She is currently an assistant research fellow at Beijing Institute of Tracking and Communication Technology. Her research interests focus on launch informatics.

Kai Xu is a professor in the School of Computing, NUDT, where he received his Ph.D. degree in 2011. He serves on the editorial boards of *ACM Transactions on Graphics*, *Computer Graphics Forum*, *Computers & Graphics*, etc.