IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2018, accepted February 9, 2018, date of publication February 20, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808225

Simultaneous 3D Object Recognition and Pose Estimation Based on RGB-D Images

CHI-YI TSAI, (Member, IEEE), AND SHU-HSIANG TSAI
Department of Electrical and Computer Engineering, Tamkang University, New Taipei City 251, Taiwan
Corresponding author: Chi-Yi Tsai (chiyi_tsai@mail.tku.edu.tw)

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2221-E-032-006.

ABSTRACT Object recognition and pose estimation are essential functions in applications of computer vision, and they also are fundamental modules in robotic vision systems. In recent years, RGB-D cameras become more and more popular, and the 3D object recognition technology has got more and more attention. In this paper, a novel design of simultaneous 3D object recognition and pose estimation algorithm is proposed based on RGB-D images. The proposed system converts the input RGB-D image to colored point cloud data and extracts features of the scene from the colored point cloud. Then, the existing color signature of histograms of orientations (CSHOT) description algorithm is employed to build descriptors of the detected features based on local texture and shape information. Given the extracted feature descriptors, a two-stage matching process is performed to find correspondences between the scene and a colored point cloud model of an object. Next, a Hough voting algorithm is used to filter out matching errors in the correspondence set and estimate the initial 3D pose of the object. Finally, the pose estimation stage employs RANdom SAmple Consensus (RANSAC) and hypothesis verification algorithms to refine the initial pose and filter out poor estimation results with error hypotheses. Experimental results show that the proposed system not only successfully recognizes the object in a complex scene but also accurately estimates the 3D pose information of the object with respect to the camera.

INDEX TERMS 3D keypoint matching, 3D object detection, 3D object recognition, 3D pose estimation.

I. INTRODUCTION

Due to the widespread use of RGB-D cameras in recent years, 3D object recognition technology has become popular in many practical applications because it not only has a higher object recognition rate in a complex environment, but also can accurately estimate 3D pose information of the object with respect to the camera. The main difference between the 2D and 3D feature-based recognition is that the former is calculated using local image texture information, while the latter is based on 3D geometric information, such as point clouds, triangle meshes, etc. In the aspect of 3D object recognition, the current methods can be divided into global and local approaches according to the way of constructing feature descriptors. The global approaches extract feature descriptors based on the surface geometry of the entire object cluster, and the local approaches extract and describe a feature point in a local region of the object.

In the study of global approaches, Marton *et al.* [1] proposed a comprehensive object categorization and classification system, which adopts a global radius-based surface description (GRSD) approach to categorize point clusters of an object based on geometric labels (e.g. plane, surface, edge, and sphere, etc) at each voxel cell. When an object point cloud cluster is segmented, the GRSD approach can produce a corresponding global cluster annotation, which represents a unique signature for the object cluster. Hence, a conventional support vector machine (SVM) model can be used to categorize object clusters based on the GRSD descriptor efficiently. Rusu *et al.* [2] proposed a 3D feature descriptor called the viewpoint feature histogram (VFH), which encodes geometry and viewpoint cues for applications of object recognition and pose identification. However, the VFH descriptor requires that objects are in light clutter and thus cannot perform well in complex real-world environments.

A variety of local approaches have been proposed in the literature. Compared with 3D global descriptors, 3D local descriptors are more robust in cluttered scenes because they can provide more geometric information regarding 3D objects or scenes. Frome *et al.* [3] proposed two regional shape descriptors: 3D shape contexts (3DSC) and

harmonic shape contexts. They showed that the 3DSC descriptor has a higher recognition rate on noisy scenes. However, the 3DSC descriptor is not rotation invariant. To deal with this issue, Tombari *et al.* [4] proposed a unique shape context (USC) descriptor, which improves the accuracy and robustness of the 3DSC descriptor by deploying a unique local reference frame (RF). However, the USC description method is computationally expensive because the dimension of the USC descriptor is of size 1960. Rusu *et al.* [5]–[7] proposed the point feature histogram (PFH) descriptor and further improved it to yield the fast point feature histogram (FPFH) descriptor to handle point registration problems. The PFH descriptor accurately captures geometric information surrounding the feature point according to the difference between the directions of the normal vectors in a local region. However, the PFH description method is also computationally expensive and is difficult to perform in real time. The FPFH description method greatly reduces the computation load of PFH by removing additional links between the feature point and its neighbors. Tombari *et al.* [8] proposed the signature of histograms of orientations (SHOT) descriptor, which encodes surface information within a spherical region surrounding the feature point. This sphere is divided into several volumes; each of them produces a local histogram of angles between the normal vectors of the feature point and its neighbors within that volume. The SHOT descriptor is then obtained by stitching all local histograms together. Because the SHOT descriptor is also computed based on a local RF, it is rotation invariant and robust to noisy scenes. Tang *et al.* [9] proposed a robust local shape descriptor called structure of geometric centroids (SGC), which is computed by voxelizing the local shape within a uniquely defined local RF and concatenating geometric centroid and point density features extracted from each voxel. The SGC descriptor is robust to occlusion and noise and supports matching keypoints near scan boundary.

As 3D keypoint matching is a key step in local approaches, some recent works on this topic have been proposed to address this problem. Ma *et al.* [10] proposed a robust point matching algorithm called vector field consensus (VFC), which solves for correspondence by interpolating a vector field between two sets of points to estimate a consensus of inlier points whose matching satisfies a nonparametric geometrical constraint. Ma *et al.* [11] proposed a robust L2-minimizing based transformation estimation algorithm and applied it to non-rigid registration problem for building sparse and dense correspondences. Tsai et al. [12] proposed an L1-norm based multi-resolution exhaustive search (MRES) algorithm to match high-dimensional image keypoint descriptors efficiently. One merit of the MRES algorithm is that it is suitable for parallel implementation on the graphics processing unit to achieve better real-time performance.

In the literature of object pose estimation, Aldoma *et al.* [13] proposed an approach that uses the CAD model to create clustered VFH (CVFH) descriptors of the object combined with cameras roll histogram to assist 3D pose computation of the object in real environments. Zhu *et al.* [14] proposed a deformable part-based model, which is trained on clusters of the model silhouettes with respect to some possible poses. A set of hypotheses about possible object locations, which can be used to segment and verify the object in the scene simultaneously, is then produced according to the deformable part-based model. The final object pose is iteratively calculated by fitting the projection of the 3D model to the object contour in the image. Drouard *et al.* [15] proposed a robust head-pose estimation method based on a partially-latent mixture of linear regressions, which directly maps high-dimensional feature vectors onto the joint space of head-pose angles and bounding-box shifts. Recently, Zhang *et al.* [16] proposed a multistream multitask deep network, which uses depth, RGB, and optical flow data to jointly detect human and estimate head pose in RGB-D videos.

From the above discussion, there are several studies related to 3D feature description, 3D keypoint matching, 3D object recognition, and 3D pose estimation. However, only a few integrated systems have been proposed to deal with these tasks efficiently. Therefore, this paper presents the design, implementation, and verification of a 3D object recognition and pose estimation system based on an RGB-D camera to simultaneously handle object recognition and pose estimation tasks in real-world environments. In the object recognition process, a point-cloud segmentation method [17] is used to obtain possible object clusters before starting the calculation of feature description. Then, a keypoint-based two-stage matching process is performed to speedup the computation of finding correspondences between the object clusters of the current scene and a colored point cloud model of an object. Next, a Hough voting algorithm [18] is employed to filter out matching errors in the correspondence set and estimate the initial 3D pose of the object. In the pose estimation process, we utilize RANSAC and hypothesis verification algorithms to refine the initial pose and filter out poor estimation results with error hypothesis. Experimental results validate the object recognition performance and pose estimation accuracy of the proposed system in a complex real-world scene.

The remainder of this paper is organized as follows. Section II introduces the system architecture of the proposed object recognition and pose estimation algorithm. Section III and Section IV present technical details of the proposed 3D object recognition and 3D pose estimation modules, respectively. Experimental results are reported in Section V to evaluate the effectiveness and efficiency of the proposed object recognition and pose estimation method. Section VI concludes the contributions of this paper.

II. SYSTEM ARCHITECTURE

Figure 1 shows system architecture of the proposed object recognition and pose estimation algorithm based on colored point clouds. The proposed system consists of an object

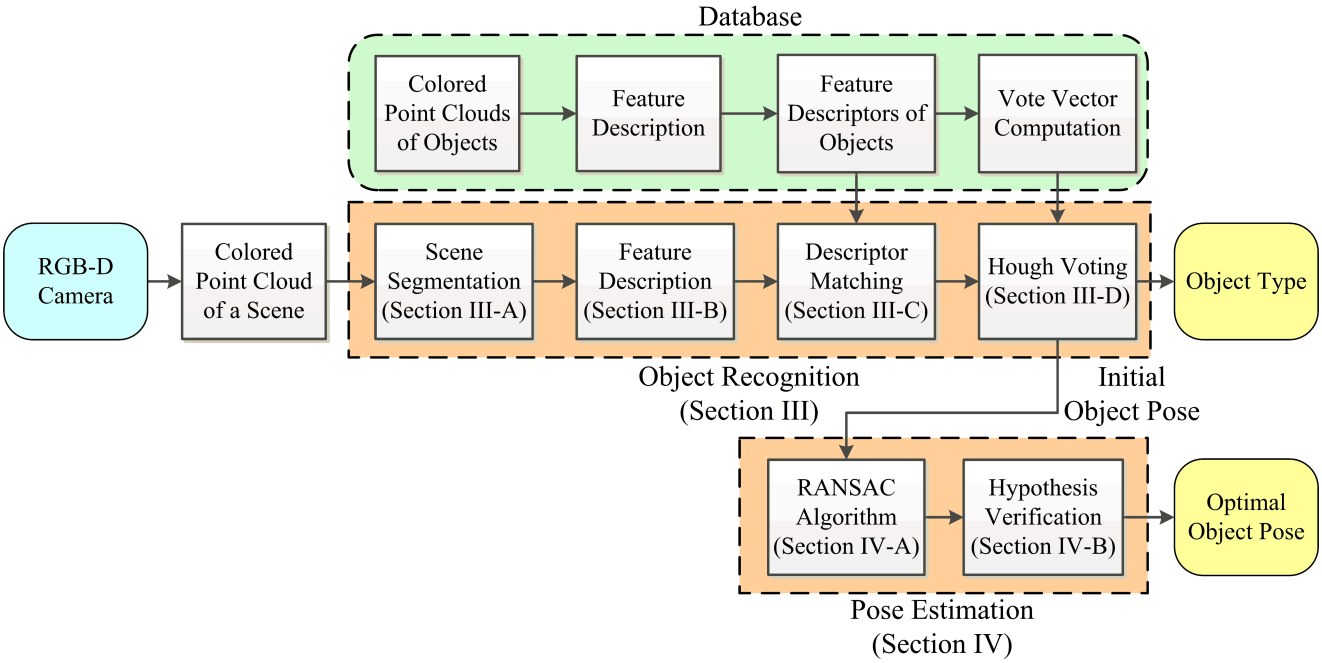

FIGURE 1. System architecture of the proposed object recognition and pose estimation algorithm.

recognition module and a pose estimation module. First, the color and depth images of a scene are captured from an RGB-D camera to produce a colored point cloud of the scene. The object recognition module then detects the foreground objects in the scene and identifies the object-of-interest (OOI) by matching feature descriptors between the foreground objects and the OOI model recorded in the database. Finally, the type of the OOI and its initial pose in the scene are simultaneously obtained from a Hough voting process.

To improve the accuracy of pose estimation, the initial pose of the OOI is further refined via the pose estimation module, which implements a RANSAC algorithm to optimize the relative posture of the object in camera frame and employs a hypothesis validation algorithm to obtain the best object pose estimation result under some predefined pose hypotheses. The following sections present the technical details of both object recognition and pose estimation modules.

III. OBJECT RECOGNITION

This section introduces the processing steps of the proposed 3D object recognition module, which consists of four units to perform scene segmentation, feature description, descriptor matching and Hough voting processes, respectively. When the colored point cloud of the scene is captured from the RGB-D camera, the scene segmentation unit is used to remove background points in the point cloud. The feature description unit is then applied to construct feature descriptors of all foreground objects in the scene. Finally, the descriptor matching and Hough voting units identify the OOI and its initial pose information.

A. SCENE SEGMENTATION

Scene segmentation is an important task for object detection and recognition. The purpose of scene segmentation is to separate foreground object and planar background regions of the scene for increasing the computational efficiency in the following feature extraction and description process. First, the point cloud of the scene is divided into multiple planar regions, which are supposed to be the background regions of the scene. Next, the foreground regions of the scene are clustered by a Euclidean distance criterion to extract objects on each background plane. Finally, the clustered object points are merged to form a foreground point cloud to be used as the input of the feature description unit. In a complex scene, there may be more than one plane to be segmented. Thus, the proposed scene segmentation unit employs a multi-plane segmentation method based on connected components [17] to handle this situation.

The connected component algorithm is one of the conventional techniques for image segmentation. The point-cloud segmentation method proposed in [17] divides a 3D point cloud data sampled from a regular 2D grid, called an organized point cloud, based on labels of the connectivity components. This method is more efficient than others existing methods that usually require a nearest neighborhood search (NNS) process to determine adjacent neighbors of each data point. After obtaining the labeled image of the connected components, the background planes in the scene are detected by the larger connected regions in the labeled image. These regions usually correspond to the wall plane, the ground plane or the desktop plane in the scene and can be expressed by the planar equation

$n_x x + n_y y + n_z z + n_d = 0$, where $\mathbf{n} = [n_x, n_y, n_z]^T$ is the surface normal vector of the plane, and n_d is the projection distance between the normal vector $\mathbf{n}$ and a point $\mathbf{d} = [x, y, z]^T$ on the plane such that

$$n_d = \mathbf{d}^T \mathbf{n}. \tag{1}$$

Here, we employ a real-time surface normal vector estimation algorithm proposed in [19] to efficiently estimate the normal vector of each plane in the scene. Finally, a data point $\mathbf{p}$ in the organized point cloud is represented by $\mathbf{p} = [\mathbf{d}^T, \mathbf{n}^T, n_d]^T$, where the projection distance n_d is calculated by the inner product of expression (1).

Let $\mathbf{p}_1$ and $\mathbf{p}_2$ denote two adjacent data points of the organized point cloud. Then, two distance metrics are used to measure the similarity between $\mathbf{p}_1$ and $\mathbf{p}_2$ such that [17]

$$d_\theta(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{n}_1^T \mathbf{n}_2, \tag{2}$$

$$d_d(\mathbf{p}_1, \mathbf{p}_2) = \left| n_d^1 - n_d^2 \right|. \tag{3}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ denote the normal vector of $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively; n_d^1 and n_d^2 denote the vertical distance of $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively. The metrics d_θ and d_d measure the difference in between $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively. Based on these two metrics, the connected components of a plane in the organized point cloud can be decided by

$$C_p(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} true, & \text{if } (d_\theta < t_\theta \&\& d_d < t_d), \\ false, & \text{otherwise}, \end{cases} \tag{4}$$

where t_θ and t_d are thresholds to evaluate the similarity of orientation angle and projection distance between $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively. Finally, a labeled point cloud L that indicates background planes in the scene is generated by substituting the connected-component result obtained from Eq. (4) into a 3D connected-component labeling algorithm [20].

After separating the background planes, a Euclidean clustering process is applied to the rest of data points to classify object clusters in the scene. To achieve this purpose, the Euclidean distance between $\mathbf{p}_1$ and $\mathbf{p}_2$, denoted by $d_e(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\|_2$, is employed. Let Ω_p be a label set of the detected planes in the organized point cloud. Then, the connected components of an object in the scene can be determined according to the labeled point cloud L such that

$$C_{obj}(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} false, & if\ (L(\mathbf{p}_1) \in \Omega_p || L(\mathbf{p}_2) \in \Omega_p || d_e(\mathbf{p}_1, \mathbf{p}_2) > t_e), \\ true, & \text{otherwise}, \end{cases} \tag{5}$$

where t_e represents the threshold of the Euclidean distance between $\mathbf{p}_1$ and $\mathbf{p}_2$. If the Euclidean distance between two data points is greater than t_e, or one of the points belongs to the label set Ω_p, then the two data points do not belong to the same object cluster. Similarly, we apply the 3D connected-component labeling algorithm to the connected-component result of Eq. (5) to obtain the labeled object clusters, which are treated as the detected objects in the scene.

B. FEATURE EXTRACTION AND DESCRIPTION

The feature extraction operation only applies to the data points of the detected object clusters obtained from the previous scene segmentation process. There are several ways to extract 3D local features in a point cloud such as intrinsic shape signatures (ISS) [21], normal aligned radial feature (NARF) [22], and uniform sampling (US), etc. In this work, the US method is used to uniformly down-sample the data points of each object cluster, and the remained points are considered as the feature points of each detected object. More specifically, the US method divides the point cloud of an object into multiple cube regions and takes the centroid point of each cube as a feature point of the object. Empirically, setting the size of a cube region as $1 \times 1 \times 1$ cm^3 works well in our testing. The main advantage of the US method is that it can greatly reduce the computational cost of the process.

Next, the feature description operation is performed on each feature point of the detected object. In this step, we employ the existing CSHOT feature description algorithm [23], which characterizes a feature point with its neighboring points to produce a combined texture-shape 3D descriptor as shown in Fig. 2. Let $\mathbf{p}_f$ denote a feature point of the detected object. Then, the CSHOT descriptor for the feature point $\mathbf{p}_f$ is established by multiple signatures of histograms such that

$$D(\mathbf{p}_f) = \bigcup_{i=1}^{m} SH_{f_i}^{G_i}(\mathbf{p}_f), \tag{6}$$

where m denotes the number of signatures of histograms, and $SH_{f_i}^{G_i}(\mathbf{p})$ denotes the signature of histograms of a given feature point $\mathbf{p}$ relative to the ith property function G_i and the ith metric function f_i. Here, the CSHOT descriptor incorporates two signatures of histograms ($m = 2$). The first one is a signature of histograms of shape-related measurements, which defines the G_1 and f_1 functions as the normal vector of the feature point and the inner product of normal vectors, respectively, such that

$$f_1(G_1(\mathbf{p}), G_1(\mathbf{q})) = \mathbf{n}_\mathbf{p}^T \mathbf{n}_\mathbf{q}, \tag{7}$$

where $\mathbf{q}$ is an adjacent point of the feature point $\mathbf{p}$. The second one is a signature of texture-related measurements, and here

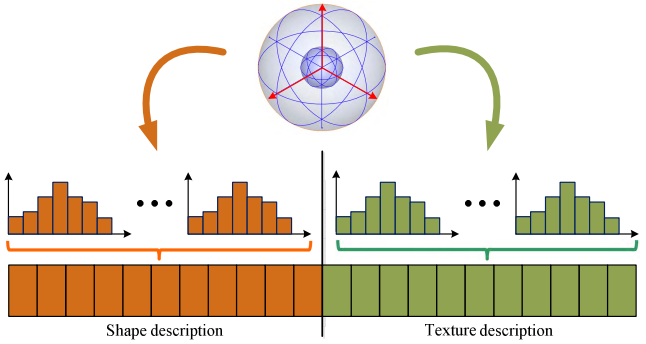

FIGURE 2. Concept of the CSHOT descriptor.

we define the G_2 and f_2 functions as the HSV color texture of the feature point and the one-norm value of color differences, respectively, such that

$$f_2(G_2(\mathbf{p}), G_2(\mathbf{q})) = \sum_{j=1}^{3} \left|\mathbf{h}_j(\mathbf{p}) - \mathbf{h}_j(\mathbf{q})\right|, \tag{8}$$

where $\mathbf{h}(\mathbf{p})$ represents the HSV color triplets associated to the feature point $\mathbf{p}$, and j is the index of the color channel. In general, the CSHOT descriptor can improve the accuracy of 3D object recognition due to incorporating two different types of signatures of histograms to construct a 3D feature descriptor.

C. DESCRIPTOR MATCHING

After extracting the CSHOT descriptors of each feature point in the object clusters, a descriptor matching process is performed to find 3D correspondences between the detected object point cloud and the recorded model point cloud. Traditionally, this problem is resolved by a NNS algorithm based on k-d tree techniques. However, the CSHOT descriptor used in this paper has a dimension of 1344, which greatly degrades matching performance of the k-d tree based NNS algorithm due to the curse-of-dimensionality issue [24], [25]. To deal with this issue, we add two designs to the k-d tree based NNS algorithm. The first one is to use multiple randomized k-d trees instead of the traditional k-d tree method. This design helps to speed up the NNS process and can be easily implemented by the FLANN (Fast Library for Approximate Nearest Neighbors) library [26], which provides an efficient way to resolve the NNS problem in a large dataset of high-dimensional feature descriptors.

The second one is to reduce the computational complexity of the NNS process by adopting a two-stage CSHOT descriptor matching algorithm, which divides the NNS process into a preliminary candidate search (PCS) stage and a best match determination (BMD) stage. In the PCS stage, we only take the shape description of the CSHOT descriptor to search the preliminary matching candidates for each feature point. In the BMD stage, the best match of each feature point is selected from the preliminary matching candidates by evaluating the similarity of the full CSHOT descriptor. The main idea of the proposed two-stage matching algorithm is that it can efficiently search candidate feature matches by less complex matching operations and then select the best match from the candidate matches in a more rigorous way. Therefore, the processing speed of the NNS process can be greatly improved when the number of feature points is increased rapidly.

D. HOUGH VOTING

After obtaining the 3D correspondences between the object model and the current scene, the Hough voting algorithm [18] is used to recognize the OOI while estimating its initial pose in the scene. Moreover, the matching outliers can be filtered out through the voting process. The Hough voting algorithm is divided into offline and online processes. The former produces voting vectors of the feature points of the OOI model, and the latter produces voting vectors of the correspondences between the OOI model and the current scene. Let $\mathbf{d}_c^M$ and $\mathbf{d}_i^M$ denote the center point and the ith feature point of the OOI model in a global RF, respectively. In the offline process, for each feature point of the OOI model, a rotation and translation invariant voting vector represented in a local RF is computed by

$$\mathbf{v}_{L_i}^M = \mathbf{R}_{GL_i}^M(\mathbf{d}_c^M - \mathbf{d}_i^M), \tag{9}$$

where $\mathbf{R}_{GL_i}^M$ is the transformation matrix from the global RF to the local RF associated with the local RF of the feature point $\mathbf{d}_i^M$, which is obtained from an invariant local RF estimation algorithm [27].

When the correspondences between the OOI model and the current scene are obtained from the descriptor matching operation, the online process is activated to determine the voting vector of the ith correspondence in the global RF of the scene such that

$$\mathbf{v}_{G_i}^S = \mathbf{R}_{L_iG}^S \mathbf{v}_{L_i}^S + \mathbf{d}_i^S, \tag{10}$$

where $(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)$ denotes the ith correspondence between the feature point of the scene and the feature point of the OOI model, $\mathbf{v}_{L_i}^S = \mathbf{v}_{L_i}^M$ is the rotation and translation invariant voting vector of the feature point $\mathbf{d}_i^S$ in the local RF, and $\mathbf{R}_{L_iG}^S$ is the transformation matrix from the local RF to the global RF associated with the local RF of the feature vector $\mathbf{d}_i^S$. Figure 3 shows the concept of voting vector computation based on Eq. (9) and Eq. (10). Because a voting vector $\mathbf{v}_{G_i}^S$ can cast a vote in the global RF of the scene, the center point $\mathbf{d}_c^S$ of the OOI in the scene can be efficiently identified by the cell having the maximum number of votes in 3D Hough space. The matched points whose voting vectors do not point to the same cell are regarded as the matching outliers, as indicated by the red lines in Fig. 4. Finally, the initial pose of the OOI is estimated according to the identified center point $\mathbf{d}_c^S$ of the OOI such that

$$\mathbf{R}_0^S = \mathbf{R}_{L_iG}^S \mathbf{R}_{GL_i}^M \quad \text{and} \quad \mathbf{t}_0^S = \mathbf{d}_c^S - \mathbf{d}_c^M, \tag{11}$$

where $\mathbf{R}_0^S$ and $\mathbf{t}_0^S$ are the initial rotation matrix and translation vector of the OOI in the global RF of the scene.

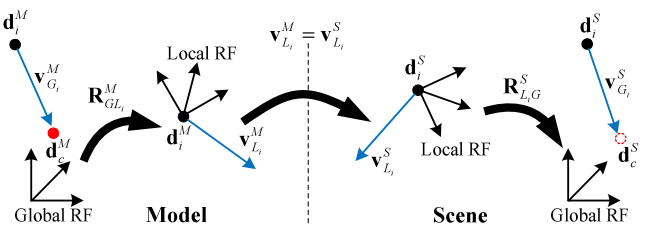

FIGURE 3. Voting vector computation based on the ith correspondence between the feature point of the scene and the feature point of the OOI model.

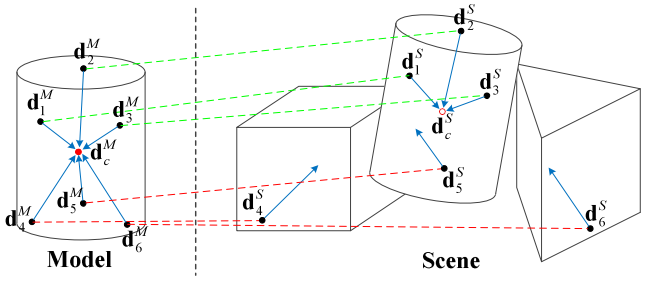

FIGURE 4. Example of 3D object recognition based on the Hough voting scheme. The center point of the OOI can be identified by the cell having the maximum number of votes in the Hough space.

IV. POSE ESTIMATION

In this section, we introduce the processing steps of the proposed pose estimation scheme based on the RANSAC algorithm [28] to refine the initial pose of the OOI obtained from the Hough voting approach. Next, a hypothesis verification algorithm is used to get the best object pose estimation result.

A. MODIFIED RANSAC ALGORITHM

Although the Hough vote algorithm can provide a preliminary pose estimation of the OOI, a post-optimization process is still required to improve the robustness of pose estimation process against matching outliers. Given N correspondences between the OOI model and the current scene, previously defined as $(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)$, for $i = 1 \sim N$. Let $\mathbf{T} = [\mathbf{R}|\mathbf{t}] \in \Re^{3\times 4}$ denote a transformation matrix associated with a 3-by-3 rotation matrix $\mathbf{R}$ and a 3-by-1 translation vector $\mathbf{t}$. Define a metric based on the sum of squared distances between the correspondences $(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)$ associated with a transformation matrix $\mathbf{T}$ such that

$$\varepsilon(\mathbf{T})|^N_{(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)} = \sum_{i=1}^{N} \left\| \Pi(\mathbf{T}, \mathbf{d}_i^M) - \mathbf{d}_i^S \right\|_2^2, \tag{12}$$

where $\Pi(\mathbf{T}, \mathbf{d}) = \mathbf{Rd} + \mathbf{t}$ is a rigid transformation of a 3D point $\mathbf{d}$ associated with a transformation matrix $\mathbf{T}$. Then, the optimal transformation matrix $\hat{\mathbf{T}}$ between the correspondences $(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)$ is computed by minimizing the metric (12) such that

$$\hat{\mathbf{T}} = \underset{\mathbf{T}\in\Re^{3\times 4}}{\operatorname{argmin}}\ \varepsilon(\mathbf{T})|^N_{(\mathbf{d}_i^S \leftrightarrow \mathbf{d}_i^M)}, \tag{13}$$

which can be solved by the Levenberg-Marquardt algorithm. However, the pose estimation method using Eq. (13) is very sensitive to the matching outliers. To deal with this problem, a RANSAC algorithm is employed to collaborate with the pose estimation method (13).

Let $\bar{\mathbf{d}}_i^M = \Pi(\mathbf{T}_0^s, \mathbf{d}_i^M)$ denote the ith transformed feature point of the OOI model associated with the initial transformation matrix $\mathbf{T}_0^s = [\mathbf{R}_0^s|\mathbf{t}_0^s]$. The processing steps of the proposed RANSAC pose estimation algorithm are listed below.

Initialization: Set a positive threshold t_{poly} to evaluate the similarity of polygon edge lengths. Clear an iterative counter $k = 0$ and a best inlier number $N_{best} = 0$. Set a maximum number of iterative counter $k_{\max}$.

Step 1: Randomly select $n \geq 3$ correspondences between the transformed OOI model and the current scene, $(\mathbf{d}_i^S \leftrightarrow \bar{\mathbf{d}}_i^M)$, for $i = 1 \sim n$.

Step 2: Calculate a dissimilarity vector $\boldsymbol{\delta}$ between the n sampled polygon edge lengths [29]

$$\boldsymbol{\delta} = \left[\frac{|l_1^S - l_1^M|}{\max(l_1^S, l_1^M)} \cdots \frac{|l_n^S - l_n^M|}{\max(l_n^S, l_n^M)} \right]^T \in \Re^{n\times 1}, \tag{14}$$

where $l_i^S = \left\| \mathbf{d}_j^S - \mathbf{d}_i^S \right\|_2$ and $l_i^M = \left\| \bar{\mathbf{d}}_j^M - \bar{\mathbf{d}}_i^M \right\|_2$ for $j = i+1 \bmod n$ denote the edge length of the scene polygon and of the transformed model polygon, respectively. If $\|\boldsymbol{\delta}\|_2 > t_{poly}$, then go back to Step 1.

Step 3: Estimate a suboptimal hypothesis transformation $\mathbf{T}_1^S$ using the n correspondences $(\mathbf{d}_i^S \leftrightarrow \bar{\mathbf{d}}_i^M)$ such that

$$\mathbf{T}_1^S = \underset{\mathbf{T}\in\Re^{3\times 4}}{\operatorname{argmin}}\ \varepsilon(\mathbf{T})|^n_{(\mathbf{d}_i^S \leftrightarrow \bar{\mathbf{d}}_i^M)}. \tag{15}$$

Step 4: Apply the suboptimal hypothesis transformation to the transformed OOI model to obtain hypothesis OOI model points $\hat{\mathbf{d}}_i^M = \Pi(\mathbf{T}_1^S, \bar{\mathbf{d}}_i^M)$.

Step 5: Find the matching inliers by the NNS process between the feature points of $\mathbf{d}_i^S$ and of $\hat{\mathbf{d}}_i^M$. If the number of the current inliers N_{in} is lower than the best inlier number N_{best}, then increase the iterative counter $k = k + 1$ and go back to Step 1.

Step 6: Record the current matching inliers as the best correspondences and set the best inlier number as $N_{best} = N_{in}$.

Step 7: Update the maximum number of iterative counter as

$$k_{\max} = \frac{\ln(1-p)}{\ln(1-w^n)}, \tag{16}$$

where $w = N_{best}/N$ is the current inlier probability, p is the desired inlier probability, and n is the sampling number used in Step 1. Increase the iterative counter $k = k+1$. If $k < k_{max}$, then go back to Step 1.

Step 8: Estimate a hypothesis transformation $\mathbf{T}_2^S$ using the best correspondences $(\mathbf{d}_i^S \leftrightarrow \hat{\mathbf{d}}_i^M)$ such that

$$\mathbf{T}_2^S = \underset{\mathbf{T}\in\Re^{3\times 4}}{\operatorname{argmin}}\ \varepsilon(\mathbf{T})|^{N_{best}}_{(\mathbf{d}_i^S \leftrightarrow \hat{\mathbf{d}}_i^M)}. \tag{17}$$

Step 9: Recovery the optimal transformation of the OOI using transformations $\mathbf{T}_0^S$, $\mathbf{T}_1^S$, and $\mathbf{T}_2^S$ such that

$$\mathbf{T}_{OOI}^S = \mathbf{T}_2^S \circ (\mathbf{T}_1^S \circ \mathbf{T}_0^S), \tag{18}$$

where the operation $\mathbf{T}_2 \circ \mathbf{T}_1 = [\mathbf{R}_2\mathbf{R}_1|\mathbf{t}_2 + \mathbf{R}_2\mathbf{t}_1]$ denotes a composition computation between two transformation matrices. Finally, the resulting transformation $\mathbf{T}_{OOI}^S$ represents the refined pose of the OOI in the current scene.

In general, we also can define a convergence threshold to terminate the RANSAC algorithm when the metric $\varepsilon(\mathbf{T})|^{N_{best}}_{(\mathbf{d}_i^S \leftrightarrow \hat{\mathbf{d}}_i^M)}$ falls below the threshold.

B. HYPOTHESIS VERIFICATION

In the Hough voting process, there may be more than one object hypothesis being detected in the scene. However, not every hypothesis is corresponding to one valid OOI detection result. Therefore, the last step of the proposed pose estimation algorithm is a global hypothesis verification process that evaluates each object hypothesis according to geometrical cues of the OOI model and the current scene. Suppose that the Hough voting process provides m recognition hypotheses related to an OOI model. Let $\mathbf{T}_O^S\big|_j, j = 1 \sim m$, denote the jth optimal transformation matrix corresponding to the jth object hypothesis and $\Omega_O^S\big|_j = \{\mathbf{d} : \mathbf{d} \in \Pi(\mathbf{T}_O^S\big|_j, \mathbf{d}_i^M)\}$ a point set of the jth object hypothesis mapping into the current scene. Let Ω^S be the point set of the current scene. Then, the local fitness between a scene point $\mathbf{p} \in \Omega^S$ and its nearest neighbor $\mathbf{q} \in \Omega_O^S\big|_j$ can be measured by a weight function

$$w_\rho(\mathbf{p}, \mathbf{q})\big|_j = \left\lfloor 1 - \frac{\|\mathbf{p} - \mathbf{q}\|_2}{\rho} \right\rfloor_0 \mathbf{n}_\mathbf{p}^T \mathbf{n}_\mathbf{q}, \tag{19}$$

where $\lfloor x \rfloor_0$ is a clipping function that sets $x = 0$ if $x < 0$, ρ is a positive threshold to evaluate the distance between $\mathbf{p}$ and $\mathbf{q}$, and $\mathbf{n}_\mathbf{p}$ and $\mathbf{n}_\mathbf{q}$ are the normal vectors of $\mathbf{p}$ and $\mathbf{q}$, respectively.

Define a set of Boolean variables $\chi_b = \{b_1, b_2, \ldots, b_m\}$ with each $b_j \in \{0, 1\}$ indicating that the jth recognition hypothesis is invalidated/validated. To find the optimal hypotheses, a global hypothesis verification function $\Im(\chi_b) : B^m \to \Re$ that summarizes geometrical cues of the OOI model and the current scene [30] is employed such that

$$\Im(\chi_b) = f_S(\chi_b) + \lambda \cdot f_M(\chi_b), \tag{20}$$

where λ is a regularization coefficient, and f_M, f_S are scalar functions associated with the geometrical cues of the OOI model and the current scene given by

$$f_S(\chi_b) = \sum_{\mathbf{p} \in \Omega^S} \left(\Lambda_{\chi_b}(\mathbf{p}) + \Gamma_{\chi_b}(\mathbf{p}) - E_{\chi_b}(\mathbf{p}) \right), \tag{21}$$

$$f_M(\chi_b) = \sum_{j=1}^{m} b_j \Phi_j^M, \tag{22}$$

where $E_{\chi_b}(\mathbf{p}) = \sum_{j=1}^{m} b_j \, w_\rho(\mathbf{p}, \mathbf{q})\big|_j$ evaluates the geometrical cue of fitness between Ω^S and $\Omega_O^S\big|_j$. $\Lambda_{\chi_b}(\mathbf{p}) = \left\lfloor \sum_{j=1}^{m} sgn(w_\rho(\mathbf{p}, \mathbf{q})\big|_j) \right\rfloor_0$ counts the number of multiple assignment of each scene point between all hypotheses, where $sgn(x)$ is a sign function of a real number x. $\Gamma_{\chi_b}(\mathbf{p}) = \sum_{j=1}^{m} b_j \, r_\eta^\kappa(\mathbf{p}, \mathbf{q})\Big|_j$ measures the effect of an unexplained scene clutter set Ω_c^s nearby the jth hypothesis set $\Omega_O^S\big|_j$, in which

$$r_\eta^\kappa(\mathbf{p}, \mathbf{q})\Big|_j = \begin{cases} \kappa, & \|\mathbf{p} - \mathbf{q}\|_2 \le \eta \text{ and } \mathbf{p} \in \Omega_c^s, \\ w_\eta(\mathbf{p}, \mathbf{q})\big|_j, & otherwise, \end{cases}$$

is a clutter weight function, where κ is a positive constant to penalize unexplained scene points nearby the set $\Omega_O^S\big|_j$, and η is a positive threshold to define the range of the clutter set Ω_c^s. Φ_j^M is the number of outliers for the jth object hypothesis mapping into the current scene. Finally, the optimal hypotheses are determined via an constrained optimization process such that

$$\hat{\chi}_b = \underset{\chi_b \in B^n}{\operatorname{argmin}} \, \Im(\chi_b) \quad \text{subject to } \|\chi_b\|_\infty > 0, \tag{23}$$

which can be resolved by a classical simulated annealing algorithm [31]. The initial Boolean set for the iterative update process is set as $\chi_b^{(0)} = \{1, 1, \ldots, 1\}$, which means all hypotheses to be active at the beginning.

V. EXPERIMENTAL RESULTS

The proposed object recognition and pose estimation algorithm was implemented with Point Cloud Library (PCL) [32] running on a Windows 7 platform equipped with a 3.2 GHz Intel®Core(TM) i5-4460 CPU and 8GB system memory. The RGB-D camera used in the experiments was a Microsoft Kinect sensor. To evaluate the performance of the proposed algorithm, the following experiments consist of three parts: object recognition, pose estimation and computational efficiency of the proposed algorithm. Figure 5(a) and 5(b) illustrate two point cloud models used in the experiment of object recognition and pose estimation, respectively. Moreover, the bottle object was mounted on the end-effector of UR5 robot manipulator [Fig. 5(c)] to provide the ground truth of the OOI poses for evaluation of 3D pose estimation results.

(a) (b) (c)

FIGURE 5. Experiment setup: (a) a bear point cloud model used in the object recognition testing, (b) a bottle point cloud model used in the pose estimation testing, and (c) a UR5 robotic manipulator used to grasp the bottle object (b) to provide the ground truth of the OOI poses.

A. OBJECT RECOGNITION RESULTS

Figure 6 shows the experimental results of 3D object recognition obtained from the proposed algorithm. In this experiment, the OOI was surrounded by many other objects as shown in Figs. 6(a1)-(a3) to increase the difficulty of the object recognition task. Figures 6(b1)-(b3) show the corresponding object recognition results, in which the green data points indicate the object recognition results obtained from the proposed algorithm. It is clear from the experimental results that the proposed algorithm succeeds to recognize the

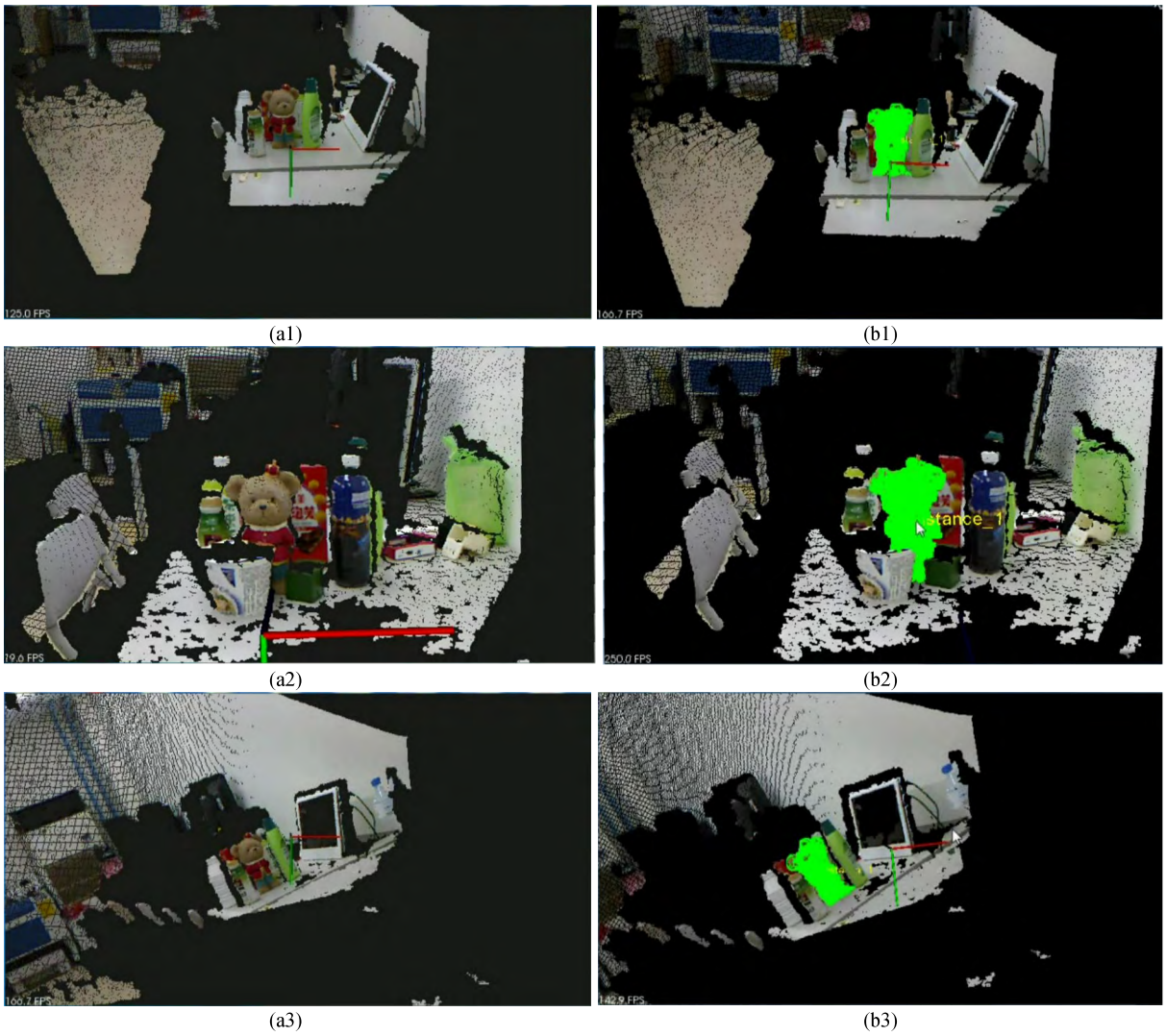

FIGURE 6. Experimental results of the 3D object recognition: (a1-a3) the scene point clouds, (b1-b3) the object recognition results, in which the green data points indicate the object recognition results obtained from the proposed algorithm.

OOI in the complex environment; even the OOI is occluded [Fig. 6(b2)] or the camera is rotated [Fig. 6(b3)]. Therefore, the object recognition performance of the proposed algorithm is validated. More experimental results can refer to the web-page of [33] and [34].

B. POSE ESTIMATION RESULTS

Figure 7 illustrates the experimental results of 3D pose estimation obtained from the proposed algorithm, in which the green data points also indicate the object recognition results of the proposed algorithm. From Fig. 7, one can see that the proposed algorithm successes to detect the OOI mounted on the end-effector of the robot manipulator. In this experiment, the actual 3D poses of the OOI were recorded according to the pose information of robot end-effector, which has a precise repeatability accuracy of 0.1 mm. Table 1 tabulates the actual 3D poses of the OOI and the corresponding estimation results obtained by the proposed algorithm, in which t_x, t_y, and t_z denote the translation quantity of the OOI on x-, y-, and z-axis, respectively; r_x, r_y, and r_z denote the Euler angle of the OOI about x-, y-, and z-axis, respectively; (t_x^*, t_y^*, t_z^*) and (r_x^*, r_y^*, r_z^*) are the corresponding translation and rotation estimates obtained from the proposed algorithm. To analyze pose estimation errors of the proposed algorithm, the following absolute estimation errors are used to evaluate the performance of our system such that

$$e_\Omega^t = \left|t_\Omega - t_\Omega^*\right| \quad \text{and } e_\Omega^r = \left|r_\Omega - r_\Omega^*\right|, \tag{24}$$

where $\Omega = \{x, y, z\}$ denotes one of the three axes of the 3D Cartesian coordinate system. Table 2 records the absolute estimation errors of the proposed algorithm in the pose estimation experiment. From Table 2, one can see that the average absolute translation errors on x-, y-, and z-axis are about 0.49 cm, 0.92 cm, and 0.55 cm, respectively. Moreover, the maximum absolute translation error on each axis is all smaller than 2.0 cm. Therefore, the accuracy of object

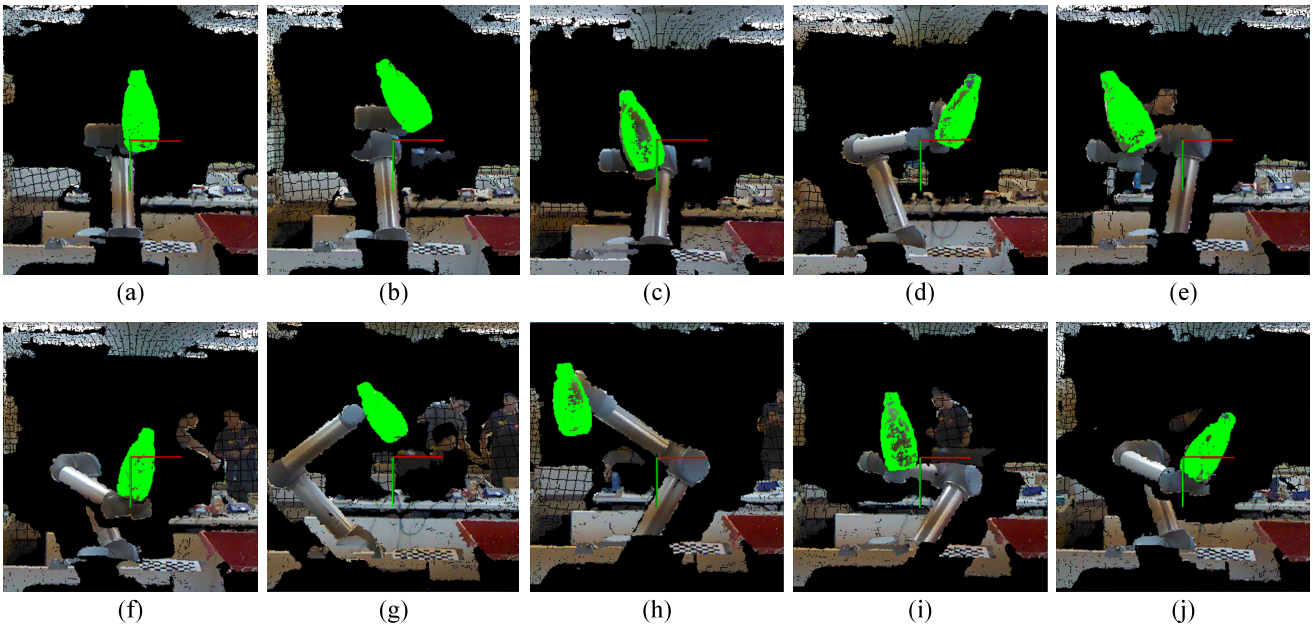

FIGURE 7. Experimental results of the 3D pose estimation of the bottle object mounted on the end-effector of the UR5 robotic manipulator.

TABLE 1. Actual target poses and the corresponding estimation results of the 3D pose estimation experiment.

	Actual 3D Poses						Estimation Results					
Unit	cm			degree			cm			degree		
Figure	t_x	t_y	t_z	r_x	r_y	r_z	t_x^*	t_y^*	t_z^*	r_x^*	r_y^*	r_z^*
6(a)	5	0	55	10	−10	−10	4.648	0.313	54.701	12.849	−6.411	−9.868
6(b)	5	−5	60	15	0	−30	3.449	−5.996	59.16	14.677	7.293	−34.556
6(c)	−5	5	60	−5	15	−15	−5.821	3.082	60.303	−4.827	14.265	−12.258
6(d)	15	0	70	0	−20	20	14.982	0.572	70.479	0.434	−12.086	20.089
6(e)	−15	−5	50	−20	5	−25	−14.856	−3.632	49.964	−17.561	10.453	−24.855
6(f)	5	15	75	−20	0	15	5.434	13.498	76.271	−19.708	−3.269	15.574
6(g)	0	0	85	20	10	−30	0.401	−0.558	85.245	18.499	11.346	−25.856
6(h)	−25	−5	60	25	0	0	−24.472	−5.756	59.817	27.232	−3.938	−6.76
6(i)	−5	5	65	−10	0	−10	−5.565	3.966	65.556	−10.399	8.818	−10.085
6(j)	10	10	70	−15	−10	30	9.941	10.171	71.268	−12.646	−8.534	23.301

translation estimation of the proposed algorithm is validated. On the other hand, the rotation estimation on y-axis has the maximum estimation error about 4.38 degrees in average. By contrast, the average absolute rotation errors on both x- and z-axis are about 1.30 degrees and 2.59 degrees, respectively. Therefore, the above experimental results evaluate that the proposed algorithm can provide accurate 3D pose estimation results of the OOI based on a given point cloud model.

C. COMPUTATIONAL EFFICIENCY

Table 3 tabulates the average processing time in each step of the proposed object recognition and pose estimation algorithm. It is clear from Table 3 that the total processing time of the proposed algorithm depends on the number of features detected in the scene point cloud, i.e., it increases from 2.16 to 3.49 seconds when the number of features increases to 4,000 keypoints. Moreover, one can see that the stage of descriptor matching costs the most processing time of the proposed algorithm, especially when the number of features becomes large. This observation highlights the importance of the proposed descriptor matching method. Table 4 records

TABLE 2. Absolute estimation errors of the 3D pose estimation experiment.

Unit	cm			degree		
Figure	e_x^t	e_y^t	e_z^t	e_x^r	e_y^r	e_z^r
6(a)	0.352	0.313	0.299	2.849	3.589	0.132
6(b)	1.551	0.996	0.840	0.323	7.293	4.556
6(c)	0.821	1.918	0.303	0.173	0.735	2.742
6(d)	0.018	0.572	0.479	0.434	7.914	0.089
6(e)	0.144	1.368	0.036	2.439	5.453	0.145
6(f)	0.434	1.502	1.271	0.292	3.269	0.574
6(g)	0.401	0.558	0.245	1.501	1.346	4.144
6(h)	0.528	0.756	0.183	2.232	3.938	6.760
6(i)	0.565	1.034	0.556	0.399	8.818	0.085
6(j)	0.059	0.171	1.268	2.354	1.466	6.699
Avg.	**0.4873**	**0.9188**	**0.5480**	**1.2996**	**4.3821**	**2.5926**

TABLE 3. Average processing time (in seconds) in each step of the proposed algorithm.

Feature Number	Scene Segmentation	Feature Description	Descriptor Matching	Hough Voting	RANSAC Algorithm	Hypothesis Verification	Total Processing Time
1000	0.185	0.358	0.863	0.641	0.052	0.062	2.161
3000	0.201	0.524	1.857	0.625	0.190	0.093	3.490

TABLE 4. Comparisons of average descriptor matching time (in seconds) between the k-d tree NNS method and the proposed two-stage matching method.

Feature Number	k-d tree NNS	Proposed Method	Percentage Change
<1000	1.2	0.8	−33.3%
>4000	6.2	3.4	−45.2%

the processing time comparisons between the k-d tree based NNS method and the proposed two-stage matching method. When the number of features is less than 1,000, the proposed method can reduce overall descriptor matching time of the k-d tree method about 33.3%. However, when the number of features is more than 4,000, the proposed method significantly reduces the descriptor matching time of the k-d tree method up to 45.2% in average. Therefore, the proposed algorithm can perform more efficiently when the number of features increases rapidly.

VI. CONCLUSION AND FUTURE WORK

In this paper, a novel object recognition and pose estimation algorithm has been proposed based on RGB-D cameras. Regarding descriptor matching, a two-stage matching algorithm is proposed to greatly speed up feature matching process, especially when the number of features increases rapidly. Regarding object recognition, the geometric shape features are combined with color textures to achieve robust 3D object detection and recognition efficiently. Moreover, the proposed method can recognize a partially occluded irregular OOI in a crowded environment while providing an initial pose estimation of the OOI. Regarding pose estimation, a robust RANSAC algorithm is proposed to estimate the optimal pose of the OOI against the matching outliers. Finally, a global hypothesis verification method is employed to evaluate each object hypothesis according to geometrical cues of the OOI model and the current scene. Experimental results show that the proposed method not only successes to recognize an OOI in a crowded and complex environment, but also provides accurate pose estimation results. The average translation and rotation estimation errors in the three axes are all smaller than 1.0 cm and 5.0 degrees, respectively. Therefore, the experimental results validate the performance of the proposed algorithm.

In the future, the design of GPU acceleration for the proposed algorithm will be further investigated to improve the overall computational efficiency of the object recognition and pose estimation system. By doing so, the proposed algorithm can be used in many practical robotics and computer vision applications.

REFERENCES

[1] Z. C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, Dec. 2010, pp. 365–370.

[2] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2155–2162.

[3] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comput. Vis.*, vol. 3023. 2004, pp. 224–237.

[4] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. ACM Workshop 3D Object Retr.*, Firenze, Italy, 2010, pp. 57–62.

[5] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *Proc. Int. Conf. Control, Autom., Robot. Vis.*, Dec. 2008, pp. 17–20.

[6] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetx, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.

[7] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[8] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 356–369.

[9] K. Tang, P. Song, and X. Chen, "3D object recognition in cluttered scenes with robust shape description and correspondence selection," *IEEE Access*, vol. 5, pp. 1833–1845, 2017.

[10] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1706–1721, Apr. 2014.

[11] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust L2E estimation of transformation for non-rigid registration," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1115–1129, Mar. 2015.

[12] C.-Y. Tsai, C.-H. Huang, and A.-H. Tsao, "Graphics processing unit-accelerated multi-resolution exhaustive search algorithm for real-time keypoint descriptor matching in high-dimensional spaces," *IET Comput. Vis.*, vol. 10, no. 3, pp. 212–219, 2016.

[13] A. Aldoma *et al.*, "CAD-model recognition and 6DOF pose estimation using 3D cues," in *Proc. Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 585–592.

[14] M. Zhu *et al.*, "Single image 3D object detection and pose estimation for grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 3936–3943.

[15] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1428–1440, Mar. 2017.

[16] G. Zhang, J. Liu, H. Li, Y. Q. Chen, and L. S. Davis, "Joint human detection and head pose estimation via multistream networks for RGB-D videos," *IEEE Signal Process. Lett.*, vol. 24, no. 11, pp. 1666–1670, Nov. 2017.

[17] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proc. 3rd Workshop Semantic Perception Mapping Exploration*, 2013, pp. 1–6.

[18] F. Tombari and L. Di Stefano, "Hough voting for 3D object recognition under occlusion and clutter," *IPSJ Trans. Comput. Vis. Appl.*, vol. 4, pp. 20–29, Mar. 2012.

[19] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2684–2689.

[20] Q. Hu, G. Qian, and W. L. Nowinski, "Fast connected-component labelling in three-dimensional binary images based on iterative recursion," *Comput. Vis. Image Understand.*, vol. 99, no. 3, pp. 414–434, 2005.
[21] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, Sep. 2009, pp. 689–696.
[22] B. Steder, R. Rusu, K. Konolige, and W. Burgard, "NARF: 3D range image features for object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, Oct. 2010, pp. 1–2.
[23] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proc. IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 809–812.
[24] S. A. Nene and S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 9, pp. 989–1003, Sep. 1997.
[25] R. Weber, H. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proc. 24th Very Large Data Bases*, New York, NY, USA, 1998, pp. 194–205.
[26] *FLANN (Fast Library for Approximate Nearest Neighbors)*. Accessed: May 15, 2018. [Online]. Available: http://www.cs.ubc.ca/research/flann/
[27] A. Petrelli and L. Di Stefano, "On the repeatability of the local reference frame for partial shape matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2244–2251.
[28] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[29] A. G. Buch, D. Kraft, J. K. Kamarainen, H. G. Petersen, and N. Kruger, "Pose estimation using local structure-specific shape and appearance context," in *Proc. Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 2080–2087.
[30] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypotheses verification method for 3D object recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 511–524.
[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[32] *Point Cloud Library (PCL)*. Accessed: May 15, 2018. [Online]. Available: http://pointclouds.org/
[33] *Experimental Result of 3D Object Recognition With Partial Occlusion*. Accessed: May 15, 2018. [Online]. Available: https://www.youtube.com/watch?v=B-_T4a24BfQ
[34] *Experimental Result of 3D Object Recognition With Camera Motion and Rotation*. Accessed: May 15, 2018. [Online]. Available: https://www.youtube.com/watch?v=V_VhgEwPtVU

CHI-YI TSAI received the B.S. and M.S. degrees in electrical engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2008.

In 2010, he joined the Department of Electrical Engineering, Tamkang University, New Taipei City, Taiwan, where he is currently an Associate Professor. His research interests include image processing, color image enhancement processing, visual tracking, visual servoing, computer vision, and deep learning.

SHU-HSIANG TSAI was born in Taipei, Taiwan, in 1992. He received the B.S. and M.S. degrees in electrical engineering from Tamkang University, New Taipei City, Taiwan, in 2014 and 2016, respectively.

He is currently an Engineer with the Research and Development Division, System Integration Department, Gallant Micro Machining Corporation. His research interests include point cloud processing and computer vision.