

Recognition and Pose Estimation of Auto Parts for an Autonomous Spray Painting Robot

Weiyang Lin, *Member, IEEE*, Ali Anwar , Zhan Li , Mingsi Tong, Jianbin Qiu , *Senior Member, IEEE*, and Huijun Gao , *Fellow, IEEE*

Abstract—The autonomous operation of industrial robots with minimal human supervision has always been in high demand. To prepare the autonomous operation of a car part spray painting robot, novel object detection, and pose estimation algorithms have been developed in this paper. The object detection part used principal components analysis (PCA) to reduce the dimension of three-dimensional (3-D) point cloud to 2-D binary image. Distance measure between the auto and cross correlation of the binary features was established to find out the similarity between them. Resultantly, the type of auto part was successfully obtained. Furthermore, iterative closest point (ICP) algorithm was used to estimate the pose difference of the auto part with respect to the camera reference frame, which was mounted on the robot. An issue with ICP's lack of robustness to local minimum was solved by the combination of ICP and genetic algorithm (GA). This allowed the optimization of pose error and addressed the problem of local minimum entrapment in ICP. For experimental validation: the proposed object recognition pipeline was implemented in both serial and parallel programming paradigms. The results were obtained for the acquired point clouds of side body car parts and compared with the major 3-D object detection systems in terms of computational cost. Pose estimation error was calculated with both ICP and the modified point set registration schemes, and it was shown to be decreasing in the case of later. All shown results supported the research claims.

Index Terms—Autonomous robotics, principal components analysis, spray painting robot, three-dimensional object recognition, visual servoing.

I. INTRODUCTION

DUE to the ever increasing demand of automobiles, their production is expected to grow steadily during the next

Manuscript received June 21, 2018; revised August 20, 2018 and September 27, 2018; accepted November 13, 2018. Date of publication November 20, 2018; date of current version March 1, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61627901, Grant 61873311, and Grant 61503101, in part by the Key Laboratory of Micro-Systems and Micro-Structures Manufacturing of Ministry of Education (HIT: NO. 2017 KM008), in part by the Self-Planned Task of State Key Laboratory of Robotics and System (HIT: NO. SKLRS201809B), and in part by the 111 Project (B16014). Paper no. TII-18-1619. (Corresponding author: Huijun Gao.)

The authors are with the State Key Laboratory of Robotics and System, the Key Laboratory of Micro-Systems and Micro-Structures Manufacturing of Ministry of Education, and the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: wylin@hit.edu.cn; ali.anwar@hit.edu.cn; zhanli@hit.edu.cn; tongms@hit.edu.cn; jbqiu@hit.edu.cn; hjgao@hit.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2018.2882446



Fig. 1. Spray painting robot.

ten years. An estimate shows that worldwide production of vehicles will increase to 111.7 million units in 2023 from 92.2 million units in 2016 [1]. With a huge number of vehicles; small and medium enterprises are expected to provide services for their maintenance. A vital part of vehicle maintenance is the repainting of its parts upon the damage or wearing of their old paint. Presently, it is carried out by the human labor, which has serious consequences on their health. The paint fumes include hazardous chemicals, thus their prolonged exposure to the labor is not recommended. Industrial standards also do not recommend delays in repainting of parts due to limited work timing of human labor or increase in the number of workers, which consequently increases the financial cost. In order to address this situation, it is highly desired to design autonomous robots capable of painting the car parts at a higher throughput and replacing the human labor from hazardous conditions.

In this regard, a spray painting robot has been designed, which is able to paint the car parts within its operational area and can be operated by the worker. It is shown in Fig. 1. Contrary to the manual operation [2], [3], in order to make it autonomous; a PBVS control technique can be used [4]. PBVS employs computer vision to estimate the pose of the manipulated object with respect to the pose of the robot. However, a major challenge in implementing PBVS is the design of computer vision algorithm which differentiates the various car parts with respect to each other and calculates their pose with respect to the robot's frame of reference. Since the robot needs to move in six degrees of freedom, therefore, three-dimensional (3-D) point clouds are

used to have a precise measurement of depth. To prepare the robot for the autonomous operation with PBVS: in this paper, the recognition pipeline for the identification of car parts has been designed. Moreover, a pose estimation pipeline has been developed that decreases the pose error.

A. Object Recognition

Object recognition is a hot research area in computer vision [5] and robotics engineering [6]. Contrary to 2-D object detection, 3-D object recognition is relatively new and mainly owes its progress to the advancement in computational capability of the modern computing systems. Furthermore, with the advent of cheap 3-D cameras such as Microsoft Kinect, 3-D point cloud processing gained more traction. It is mainly due to its superiority in describing 3-D information for applications in SLAM [7], environmental surveillance [8], and remote sensing [9], etc. Two main types of features exist for the object recognition in 3-D point cloud processing; 1) global features and 2) local features. Global features describe the entire point cloud, while local features describe the local neighborhoods around the selected key points. For precise recognition, global features play a vital role in identifying the exact object within the scene, while local features might mix up neighborhoods having similar geometric signature. However, local features are robust against occlusions and clutter [10]. Some famous examples of global features include viewpoint feature histogram [11], geometric scale space [12], oriented point features [13], shape distributions [14], etc., while a comprehensive survey of local features is provided in [5]. Furthermore, recently deep learning based methods have also been used to identify objects in ordered [15]–[17] and unordered point sets [18]. Despite the success of these methods as robust object classification algorithms, their training stage requires a large amount of hand labeled data that are not trivial to construct. Moreover, it also requires considerable amount of time based on the availability of computational resources.

Irrespective of the work mentioned above, point cloud processing algorithms are usually costly in terms of their computational performance [19]. In this regard, the algorithms from 2-D object recognition come handy. Due to their extensive research depth and implementable maturity; they are simpler and far more efficient. Dimensionality reduction [20] is a key bridge in this regard, it can be used in reducing the dimension of 3-D point clouds to 2-D images without compromising upon the significant feature information. PCA [21] is one of the most famous dimensionality reduction techniques that uses covariance and singular value decomposition to obtain the lower dimensional subspaces from the higher dimensional datasets. By using PCA, the significant information of auto part point clouds can be preserved in 2-D binary images, this greatly streamlines the recognition process with its 2-D counterpart. After dimensionality reduction, a plethora of techniques can be used from 2-D object detection that might include [22]–[25], etc.

In this paper, due to computational cost and implementation complexity, the object recognition within dense point clouds is avoided, instead the solution is simplified. After the application of PCA, the resulting binary image is used to recognize the

object within the point cloud in real time. Motivated by these results, a novel object detection technique is developed that utilizes the knowledge of statistical mathematics. By combining the geometric features obtained from Fan shape model (FSM) [26] and the ideas of cross and autocorrelation, a recognition pipeline is constructed that differentiates auto parts based on their obtained point clouds. The recognition pipeline has been implemented with both serial and parallel programming paradigms, and experimental results have been presented. The proposed method utilizes the referential approach, which means that it operates by comparing the real-time images with the reference images. Due to this, it can easily be used in other applications.

B. Pose Estimation

Another key part of preparing the spray painting robot for PBVS is pose estimation of the object. It involves the calculation of camera position with respect to the car part's position and decrease in its error by moving the robot's end effector. In order to achieve this task, point set registration techniques can be employed. They use point to point correspondences to obtain the transformation between the point cloud in its standard and arbitrary positions. One of the most famous algorithms for this task is iterative closest point (ICP) [27], which decreases the euclidean distance between the closest points of the source and target clouds. In this regard, other important schemes include [28]–[30], however, ICP is known to provide superior performance in terms of computational cost and its ease of implementation. It consists of two main steps that include the establishment of correspondences and distance minimization. For distance minimization, it has been noticed that ICP suffers from the problem of local minima entrapment [31]. In order to enhance its search space, genetic algorithm (GA) [32] is combined with ICP in various works [33]–[35]. GA is a global optimization scheme that takes inspiration from the idea of natural selection in evolutionary biology. It enhances ICP's search space and improves the pose estimation error by finding out the global minimum.

In this paper, the combination of ICP and GA was defined independently from the works mentioned above. Compared to [33]–[35], in this paper, the error function of ICP was minimized furthermore without introducing any modifications in it. The error function was treated as a fitness function for the GA algorithm. This solved the problem of local optimum in ICP and decreased the pose estimation error. In addition to this, GA was applied after the application of ICP to search for better options in the neighborhood of ICP's optimum. During experimental observation, it had been noticed that proposed pose estimation algorithm performed better than the traditional ICP algorithm in terms of achieving the optimal paint coat thickness.

C. Contribution Summary

Key contributions of this paper can be summarized in the following.

- 1) For autonomous operation of a spray painting robot, a novel car parts' recognition pipeline has been designed and implemented using dimensionality reduction and statistical matching. The pipeline detects all side body parts

of the car in real time. In addition to this, the proposed algorithm has been compared with major 3-D object recognition algorithms in terms of their computational cost.

- 2) For pose estimation of the auto parts, ICP has been used. Its shortcoming with the lack of robustness to local minimum has been addressed by combining it with GA. The modified point set registration pipeline is shown to decrease the pose estimation error as compared to the traditional ICP implementation.

Remaining part of this paper is organized as follows. Section II provides a brief introduction to the basic concepts of object recognition pipeline. Section III presents the binary image features used in recognition procedure. The recognition pipeline is formally presented in Section IV. Section V contains the proposed registration pipeline. Experimental Results have been presented in Section VI. Finally the paper is concluded in Section VII.

II. OVERVIEW OF CAR PART RECOGNITION

The key ideas used in recognition pipeline are briefly overviewed in the following sections.

A. Segmentation of Point Cloud

Planar segmentation was used to remove the scene noise from the point clouds. Experimental observations showed that the background mostly consisted of the wall and floor; therefore, random sample consensus (RANSAC) based segmentation algorithm was used to single out the object point set from that of the scene [36]. The key steps of segmentation algorithm can be described in the following.

- 1) Since three noncollinear points can be used to define a plane, therefore, they are randomly selected as $\{p_i, p_j, p_k\}$ from a point set P .
- 2) The model coefficients $\{a, b, c\}$ are computed in the plane equation $ax + by + cz + d = 0$ from three points in the previous step.
- 3) The distance from all points $p_i \in P$ where $i = 1, 2, \dots, n$ is computed from the plane model defined by coefficients in previous step.
- 4) Select and store the number of points $p^* \in P$ whose distance d to the plane model falls under a user specified threshold defined as $0 \leq d \leq d_t$.

Steps (1)–(4) defined above are repeated with new randomly selected points for k iterations. Here, k is calculated by the following relation:

$$k = \frac{\log(1 - \mathbf{p})}{\log(1 - (1 - \epsilon)^s)} \quad (1)$$

where ϵ is the probability of selecting an *outlier*, $1 - \epsilon$ is the probability of selecting an *inlier*. Therefore, the probability of selecting s good samples is $(1 - \epsilon)^s$. Hence, for k iterations the probability of bad sampling becomes $(1 - (1 - \epsilon)^s)^k$, and \mathbf{p} is the desired probability of good samples. Taking \mathbf{p} as the probability of success gives following equation:

$$1 - \mathbf{p} = (1 - (1 - \epsilon)^s)^k \quad (2)$$

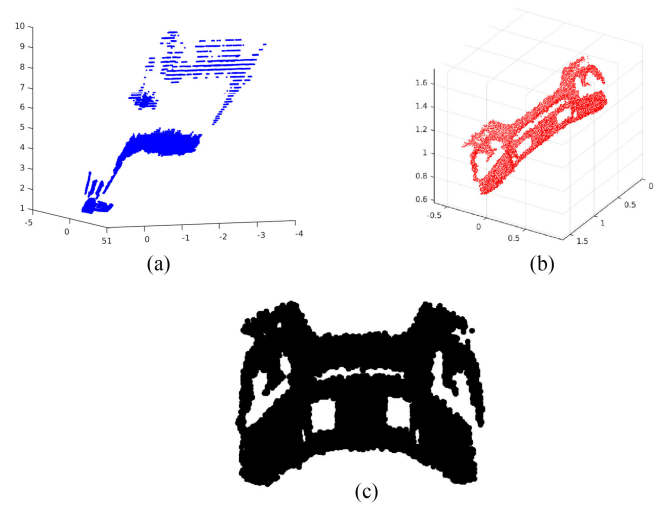


Fig. 2. Preprocessing and application of PCA on front bumper of the Car. (a) Before preprocessing. (b) Before PCA. (c) After PCA.

which can be used to produce (1). Fig. 2 shows the application of preprocessing upon the point set of front bumper of the car. Fig. 2(b) shows the extracted point set from scene, which is shown in Fig. 2(a). This algorithm has been implemented in open source *point cloud library (PCL)* [37].

B. Principal Component Analysis

PCA is commonly applied to find out the lower order linear subspaces in the higher dimensional datasets. It is widely used in social as well as natural sciences, which particularly require data interpretation [21]. In addition to analysis, it has also been used in visualizing the higher dimensional data, which is otherwise difficult to comprehend [38]. It primarily solves a singular value decomposition problem, in which the covariance matrix of input data is decomposed into its eigenvalues. The largest eigenvalues are retained since they represent the highest covariance. Linear transformation of covariance matrix with its eigenvalues gives the corresponding eigenvectors that are known as the principal components of the data. The number of these components correspond to the number of reduced dimensions. A brief mathematical introduction to PCA with reference to the analysis of 3-D point sets is given in the following.

Suppose an input point set $S = \{s_i\}$ where $i = 1, 2, 3$ consists of three vectors representing the data in 3-D euclidean space coordinates x, y, z ; therefore, $s_i \in \mathbb{R}^{3 \times N}$, where N denotes the number of points in the point set. The objective is to project this data onto a subspace of dimension 2, which can be easily interpreted as a 2-D binary image. Thus, the input dataset needs to be transformed from input space $S \in \mathbb{R}^{3 \times N}$ onto a subspace $\mathbb{R}^{2 \times N}$. Assuming that we have a transformed dataset in subspace as $Y = \{y_i\}$ where $Y \in \mathbb{R}^{2 \times N}$, we can define the transformation as

$$Y = AS \quad (3)$$

where $A = \{u_1^T, \dots, u_j^T\}$ and $u_j^T u_j = 1 \forall j = 1, 2$, the length of $u_j^T = \dim(S)$ is 3. Determination of Y can be carried out by

using the covariance matrix of \hat{S} , where $\hat{S} = S - \bar{S}$ and $\bar{S} = \text{mean}(S)$. By using \hat{S} , the covariance matrix C is constructed as

$$C = \text{cov}(\hat{S}) = \begin{bmatrix} \hat{s}_{xx} & \hat{s}_{xy} & \hat{s}_{xz} \\ \hat{s}_{yx} & \hat{s}_{yy} & \hat{s}_{yz} \\ \hat{s}_{zx} & \hat{s}_{zy} & \hat{s}_{zz} \end{bmatrix} \quad (4)$$

where \hat{s}_{ij} is the covariance between vectors \hat{s}_i and \hat{s}_j . This can be written as

$$\begin{aligned} \hat{s}_{xx} &= \frac{\sum_n (\hat{s}_x)(\hat{s}_x)}{n-1} \\ \hat{s}_{xy} &= \frac{\sum_n (\hat{s}_x)(\hat{s}_y)}{n-1} \\ &\vdots \\ \hat{s}_{zz} &= \frac{\sum_n (\hat{s}_z)(\hat{s}_z)}{n-1} \end{aligned} \quad (5)$$

the length of vectors \hat{s}_x , \hat{s}_y , and \hat{s}_z in (5) is n . Taking singular value decomposition of covariance matrix in (4) gives its eigenvalues denoted as $\lambda_1, \lambda_2, \lambda_3$, the number of eigenvalues is equal to the dimension of input dataset, which in this case is 3. Assume that the order of eigenvalues after rearranging them in descending order is $\lambda_1 > \lambda_2 > \lambda_3$, after applying the eigenvector analysis one can obtain the eigenvectors in descending order as $u_1 > u_2 > u_3$, where the first two eigenvectors denote the principle components of data. Revisiting (3) shows that matrix A is given as

$$A = [u_1 u_2]^t \quad (6)$$

and, thus, (3) projects the data with maximum covariance from 3-D euclidean space to 2-D subspace. The data obtained in the lower dimension can be plotted as a binary contour, which can be shown as black dots on white background. Fig. 2(c) shows the point set after the application of PCA.

C. Contour Searching

Contours are a robust set of image features that upon segmentation provide a topological structure of the object under analysis. They can intuitively be conceptualized as the boundaries surrounding the objects' geometries. Contour detection has its roots in the algorithms related to edge detection. Some of the earliest examples of edge detectors include: Canny's edge detector [39], Suzuki and Abe's boundary detector [40] and Hough's line detector [41], etc. Generally, edge detection techniques employ gradient analysis upon images. The gradient is taken across both x and y coordinates, while the locations corresponding to its sharp change are selected as edge points. Another idea that also serves as the basis of contour detection is connected component analysis. In this paper, contour searching has been used for shape analysis of the binary image obtained after PCA. An implementation of Suzuki and Abe's boundary detector [40] in *Open Source Computer Vision* library [42] has been utilized in this regard. It uses the idea of connected components to obtain the sequence of points representing the outline of the object. The pixels $p_i(x, y)$ for $i = 1, 2, \dots, n$ (n is total number of pixels) are scanned in the image for connectedness properties and

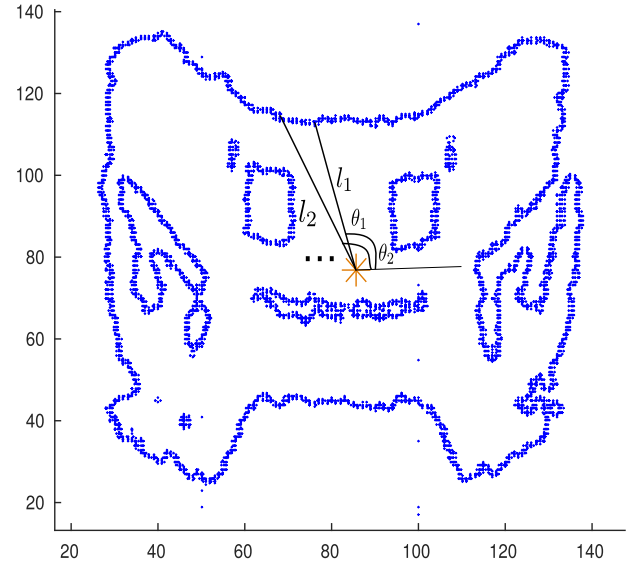


Fig. 3. Contour with features.

classified as contour sequences. Furthermore, in order to get the desired contour from the set of available contours, the largest one is selected as preprocessing of point cloud ensures the removal of noise and accurate segmentation. Fig. 3 shows the contour obtained from the binary image in Fig. 2(c).

D. Cross Correlation

In this paper, cross correlation has been used to obtain the similarity between two feature vectors. Similar to convolution it is a displacement measure of one vector with respect to another. Mathematically, it can be written for two discrete vectors as

$$f \otimes g = \sum_{m=-\infty}^{\infty} f^*[m]g[m+n] \quad (7)$$

where f and g denotes the feature vectors. f^* denotes the complex conjugate of the vector f , m denotes the length of vectors f and g , and n denotes any integer such that $-\infty \leq n \leq \infty$. It should be noted that in the case that lengths of f and g are not equal, appropriate number of zeros are appended to the vector with smaller length.

III. FEATURE SELECTION

Shape modeling is a key issue in image contour analysis [43]. Since contours are a list of pixels that do not represent any qualitative feature in themselves; therefore, it is necessary to formalize a framework that can use this information and produce useful results. In this paper, FSM [26] has been utilized, which creates a contour signature for uniquely identifying the deformed shapes of car parts and differentiating them from each other.

FSM introduced three basic features to qualitatively define the shape represented by the contour. In order to obtain these features, a reference point must be chosen priorly. In this paper, it is chosen as the centre of gravity of the contour. The features defined by FSM include: lengths of the rays from reference

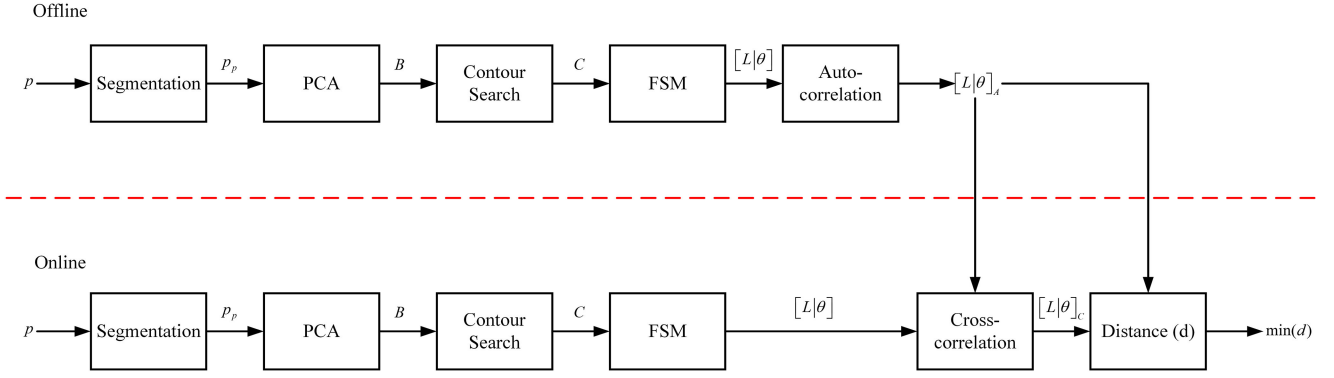


Fig. 4. Flowchart of object recognition pipeline.

point to each of the contour points (ray length), the angles of rays with the horizontal axis passing through the reference point (ray angle), and angles between the rays and the tangents of each of the contour points (tangent angle). In this paper, tangent angle has been dropped and only first two features have been used. Since experimental validation shows that ray lengths and angles are successfully able to differentiate the binary images, therefore, redundant computational cost due to tangent angle has been avoided.

Mathematically, the ray length can be defined as l and the ray angle as θ . Assuming that the contour C is obtained as a list of points defined as $\{p_1, p_2, p_3, \dots, p_n\} \in C$, then the sets defining ray lengths and angles can be written as

$$L \triangleq \begin{cases} l_1, l_2, \dots, l_n \\ \exists \quad l_i = \sqrt{\overrightarrow{o_x p_{xi}}^2 + \overrightarrow{o_y p_{yi}}^2} \\ \forall \quad i = 1, 2, \dots, n \end{cases} \quad (8)$$

where $\overrightarrow{o_x p_{xi}}$ is the length of the ray projected from reference point o to point p_i in x -coordinate of the image plane, similarly $\overrightarrow{o_y p_{yi}}$ is the length of projected ray in y -coordinate of the image plane. Furthermore, ray angle θ is defined as

$$\Theta \triangleq \begin{cases} \theta_1, \theta_2, \dots, \theta_n \\ \exists \quad \theta_i = \tan^{-1} \frac{\|\overrightarrow{o_y p_{yi}}\|}{\|\overrightarrow{o_x p_{xi}}\|} \\ \forall \quad i = 1, 2, \dots, n. \end{cases} \quad (9)$$

The selected features are significant in a sense that they have a strong discriminating power and can successfully describe the shapes of auto parts for recognition purposes. They have tolerance for significant scale and rotation variance, due to the reason that ray lengths preserve the feature information upon the change in rotation and ray angles preserve the feature information upon the change in scale. Another significant property of FSM includes its low computational cost and its characteristics of handling large shape deformation, scale and rotation variance.

Fig. 3 shows the features acquired from the contour of front bumper of the car. The reference point is denoted by the star at the center of the contour, which corresponds to its center of

gravity as well. It can be obtained as

$$c = \frac{\sum_{i=1}^n p_i}{N} \quad (10)$$

where c denotes the center of gravity of the contour that comprises of points $\{p_1, p_2, p_3, \dots, p_n\}$ and N denotes the total number of contour points.

IV. OBJECT RECOGNITION PIPELINE

For accurate trajectory generation of the spray painting nozzle, the precise information of the car part's shape is helpful. Before performing shape analysis, the recognition of the part is necessary. The approach used in this paper is to first identify the object within the image and subsequently use this information to obtain the shape related parameters stored offline. Since car parts have fixed geometric outline, which mostly varies in scale across the different models of the cars; therefore, the solution boils down to distinction between a handful of parts that should be painted automatically. In this paper, side body parts of the car have been accurately identified. However, this algorithm can be extended to any type of objects that undergo substantial variation in their geometries. A novel car part recognition procedure has been designed, which consists of *offline* and *online* stages. A flowchart of this approach can be viewed in Fig. 4.

During offline stage: the reference data for later comparison was prepared and saved. RANSAC-based segmentation was performed to extract out the car part from the scene point cloud. Subsequently, PCA was performed upon the extracted point cloud to obtain the projection of 3-D point set in 2-D. This information was converted into a binary image for shape identification using 2-D machine vision techniques. Image contours were searched on the binary image by using the boundary search method of Suzuki and Abe [40]. The FSM features described in the preceding section were obtained from the contours. The ray lengths (L) and ray angles (θ) features obtained through FSM were autocorrelated with each other to create offline feature matrices $[L|\theta]_A$. They were saved as reference feature matrices for later manipulation. Algorithm 1 showcases the offline stage of the recognition pipeline.

For online stage most of the steps mentioned in preceding passage were repeated, which included segmentation, PCA, contour

Algorithm 1: Recognition of Car Parts: Offline Stage.**Input:** Input Point Set (P)**Output:** Feature Matrix $[L|\Theta]_A$ **Procedure** Preparation of reference features:

- 1: *Obj:* Obtain $[L|\Theta]_A$ for P
- 2: **for** $count = 1$ to 8 **do**
- 3: $P_p \leftarrow \text{pre_process}(P)$
- 4: $B \leftarrow \text{PCA}(P_p)$
- 5: $C \leftarrow \text{Contour}(B)$
- 6: $[L|\Theta] \leftarrow \text{FSM}(C)$
- 7: $[L|\Theta]_A \leftarrow \text{auto_correlation}([L|\Theta])$
- 8: **end for**
- 9: **return** save $[L|\Theta]_A$ for later manipulation

Algorithm 2: Recognition of Car Parts: Online Stage.**Input:** Input Point Set (P)**Output:** Identification of car part**Procedure** Object Recognition:

- 1: *Obj:* Identify the Object within Point Set
- 2: **for** $count = 1$ to 8 **do**
- 3: $P_p \leftarrow \text{pre_process}(P)$
- 4: $B \leftarrow \text{PCA}(P_p)$
- 5: $C \leftarrow \text{Contour}(B)$
- 6: $[L|\Theta] \leftarrow \text{FSM}(C)$
- 7: $[L|\Theta]_C \leftarrow \text{cross_correlation}([L|\Theta]_A, [L|\Theta])$
- 8: $k \leftarrow \min(\| [L|\Theta]_A - [L|\Theta]_C \|_{l1})$
- 9: **end for**
- 10: **return** object corresponding to $d(\min(d))$

searching and features selection. In addition to these, the cross correlation was taken between the feature matrix $[L|\Theta]$ obtained in this stage and the reference feature matrix $[L|\Theta]_A$ obtained in the offline stage, for each car part. The similarity of features was established by using the l_1 -norm of the euclidean distance between autocorrelated and cross-correlated matrices. The minimum l_1 -norm of the euclidean distance between the resultant feature matrix $[L|\Theta]_C$ and $[L|\Theta]_A$ provided the best match. The algorithm was repeated eight times for each part and best match was used to identify the input point cloud. The side body parts recognized in this paper are mentioned in Table I. Algorithm 2 outlines the online stage of recognition pipeline.

Mathematically, similarity measure between the offline and online frameworks can be derived. Supposing that the auto-correlated feature matrices in the offline framework are represented as $\{[L|\Theta]_{A1}, [L|\Theta]_{A2}, \dots, [L|\Theta]_{A8}\} \in \mathbf{A}$, similarly the cross-correlated feature matrices in the online framework can be defined as $\{[L|\Theta]_{C1}, [L|\Theta]_{C2}, \dots, [L|\Theta]_{C8}\} \in \mathbf{C}$. Then, the minimum l_1 -norm of the distance between \mathbf{A} and \mathbf{C} can be written as

$$k = \min\|\mathbf{A} - \mathbf{C}\|_{l_1} \quad (11)$$

where k denotes the best match and, thus, identifies the type of car part in the offline training framework.

Note 1: In the case that the lengths of vectors \mathbf{A} and \mathbf{C} are not equal, the appropriate number of zeros can be appended on either of the short length vectors.

V. POINT SET REGISTRATION PIPELINE

Registration plays a fundamental part in pose estimation of point sets. It involves determination of point to point correspondences between the source and target point sets, and recovering the transformation that maps source to target. In this paper, the famous ICP [27] algorithm has been utilized for the pose estimation of car part with respect to its standard position in the robot's operational space. ICP calculates the error in the position of the part in the case of misplacement and reorients the robot's end effector for required painting task. It is an iterative method and works by minimizing the distance between the closest points of source and target sets upon each iteration. Moreover, its approach can be categorized under the local optimization approaches. Due to this, it has a tendency to be trapped in local minima and, therefore, needs multiple initial positions for better results. Supposing that P is the reference point cloud in its standard position and Q is the source point cloud at input. Then, their euclidean mean squared error can be mentioned as the cost function for local minimization as

$$E(R, t) = \frac{1}{N_P} \sum_{i=1}^{N_P} \|q_i - Rp_i - t\|^2 \quad (12)$$

where $q_i \in \mathbb{R}^Q$ and $p_i \in \mathbb{R}^P$ are the i th points. N_P is the total number of points in P , and R and t are the rotation and translation matrices, respectively, that constitute the so called euclidean transformation matrix $T = [R|t]$ between the source and target clouds. Mathematically, this transformation between P and Q can be mentioned as

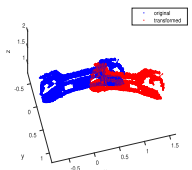
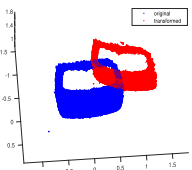
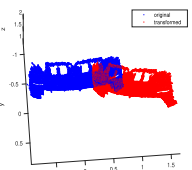
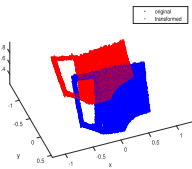
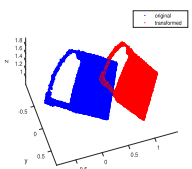
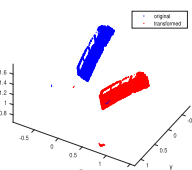
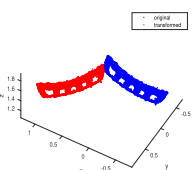
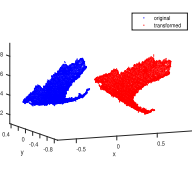
$$P = R \times Q + t \quad (13)$$

where $T = [R|t] \in \mathbb{R}^{3 \times 4}$ is a concatenated matrix of R and t . To solve the problem of local minimum entrapment, ICP must be used in conjunction with a global optimization technique. In this paper, GA [32] has been used for this purpose. It is a search-based global optimization technique based on the idea of genetics and natural selection. It uses multiple random starting points as initial population and then applies the ideas of crossover, mutation, and selection from evolutionary biology to find the optimal solution. In addition to that, it is known to work with arbitrary fitness functions unlike the conventional derivative based optimization techniques.

In order to achieve this, ICP is used to find an initial solution. Furthermore, GA is used to search the vicinity of ICP's solution and reduce the value of fitness function in (12). The flowchart of this scheme is shown in Fig. 5. For the purpose of GA-based optimization, the same fitness function mentioned in (12) has been minimized, so that both ICP and GA cooperatively search for the best optimum of (12). The algorithmic flow of pose estimation pipeline is mentioned in Algorithm 3.

It can be seen in Algorithm 3, that ICP and GA work in cooperation with each other. Both of these techniques minimize the fitness function defined in (12) and find an ideal solution for

TABLE I
EXPERIMENTAL VALIDATION

Serial no.	Name	Image	Time Complexity Analysis (seconds)					Minima of $E(R, t)$		
			[22]	[44]	[45]	[46]	Proposed	T	ICP	ICP-GA
1	Front Bumper		45.4	159.9	128.1	26.7	0.084	T_1	0.1032	$0.9839e^{-6}$
								T_2	0.0098	$0.6023e^{-6}$
								T_3	0.0092	$0.1824e^{-6}$
								T_4	0.0443	$0.4610e^{-6}$
								T_5	0.1036	$0.6680e^{-6}$
2	Trunk Lid		57.9	154.4	136.5	32.1	0.0554	T_1	0.101	$0.2600e^{-6}$
								T_2	0.0105	$0.0400e^{-6}$
								T_3	0.0752	$0.8780e^{-6}$
								T_4	0.0735	$0.7349e^{-6}$
								T_5	0.0885	$0.5431e^{-6}$
3	Back Bumper		53.1	168	135.5	32	0.0514	T_1	0.0068	$0.6907e^{-6}$
								T_2	0.0093	$0.8624e^{-6}$
								T_3	0.0092	$0.7856e^{-6}$
								T_4	0.0370	$0.0868e^{-6}$
								T_5	0.0889	$0.1774e^{-6}$
4	Back Door		54.6	144.6	138.3	34.6	0.05113	T_1	0.0068	$0.5967e^{-6}$
								T_2	0.0069	$0.3431e^{-6}$
								T_3	0.083	$0.9387e^{-6}$
								T_4	0.085	$0.2153e^{-6}$
								T_5	0.084	$0.3959e^{-6}$
5	Front Door		54	158.9	142.6	32.5	0.0515	T_1	0.0097	$0.2793e^{-6}$
								T_2	0.0075	$0.1122e^{-6}$
								T_3	0.074	$0.30e^{-6}$
								T_4	0.0741	$0.5706e^{-6}$
								T_5	0.08	$0.5e^{-6}$
6	Radiator Support		33.4	87.8	120.7	19.7	0.0598	T_1	0.0067	$0.2792e^{-6}$
								T_2	0.026	$0.6e^{-6}$
								T_3	0.14	$0.86e^{-6}$
								T_4	0.248	$0.3236e^{-6}$
								T_5	0.248	$0.44e^{-6}$
7	Grilles		34	92.7	121	20	0.0601	T_1	0.0109	$0.6838e^{-6}$
								T_2	0.0243	$0.63e^{-6}$
								T_3	0.024	$0.82e^{-6}$
								T_4	0.024	$0.95e^{-6}$
								T_5	0.038	$0.635e^{-6}$
8	Fenders		35.3	96	121	23	0.05101	T_1	0.0051	$0.81e^{-6}$
								T_2	0.069	$0.78e^{-6}$
								T_3	0.069	$0.83e^{-6}$
								T_4	0.053	$0.6e^{-6}$
								T_5	0.092	$0.15e^{-6}$

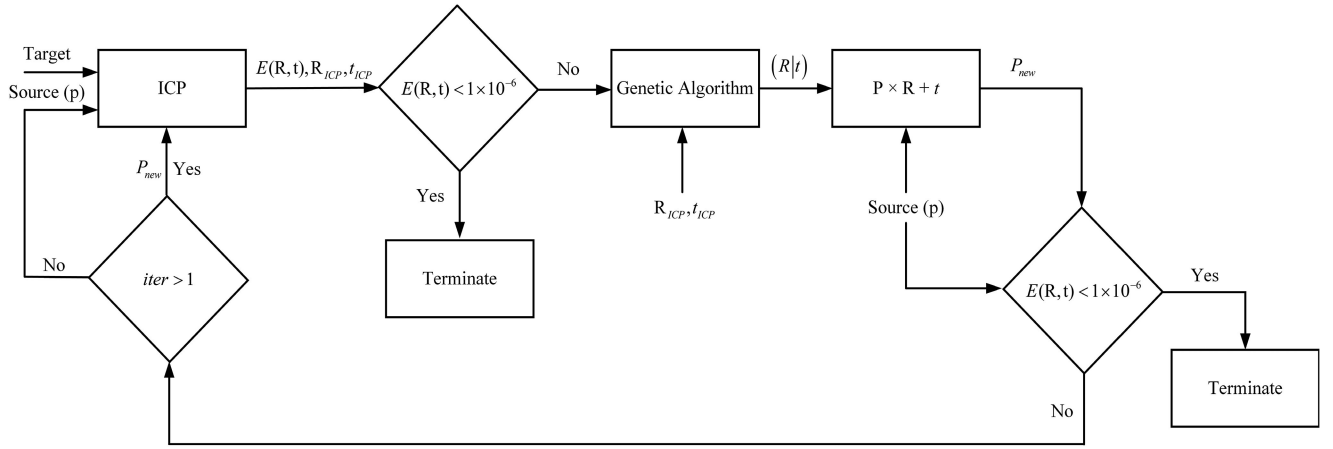


Fig. 5. Flowchart of pose estimation pipeline.

Algorithm 3: Pose Estimation of Auto Parts.

Input: Input Point Set (P), fitness function $E(R, t)$

Output: Estimated Pose of the Object ($T = [R|t]$)

Procedure Point Set Registration:

1: *Obj: Obtain Pose* $T = [R|t]$

2: Let P = source point set, P_{new} = source point set after GA, P_T = target point set

3: **for true do**

4: $E(R, t) \leftarrow \text{ICP}(P_T, P|P_{new})$

5: **if** $E(R, t) < 1e^{-6}$ **then**

6: **return** $T = T_{ICP} = [R_{ICP}|t_{ICP}]$

7: **break;**

8: **else**

9: **minimize** $E(R, t)$ with GA

10: $T = [R|t] \leftarrow \text{GA}(P_T, P, R_{ICP}, t_{ICP})$

11: $P_{new} = P \times R + t$

12: **if** $E(R, t) < 1e^{-6}$ **then**

13: **return** $T = [R|t]$

14: **break;**

15: **else**

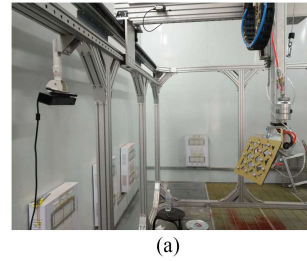
16: **continue;**

17: **end if**

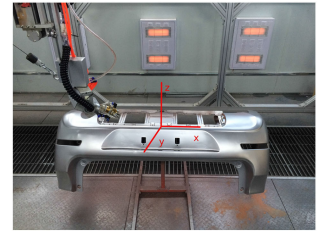
18: **end if**

19: **end for**

20: **return** $T = [R|t]$

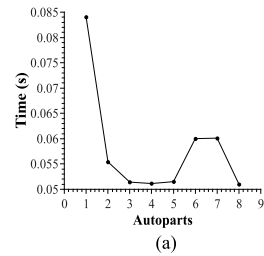


(a)

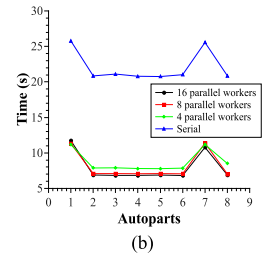


(b)

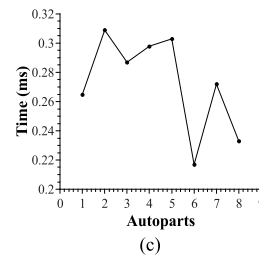
Fig. 6. (a) Microsoft Kinect mounted on the robot. (b) Object reference frame.



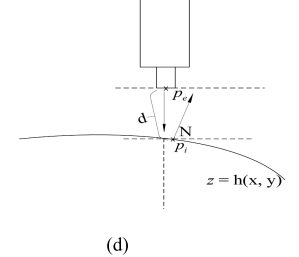
(a)



(b)



(c)



(d)

Fig. 7. Time Consumption. (a) Object recognition pipeline. (b) Object recognition pipeline with 1000 iterations. (c) Application of PCA. (d) Spray Surface with nozzle.

the pose estimation problem. It has been noticed experimentally that this solution provides a better result in comparison to the standalone application of ICP and estimated pose is acceptable for the appropriate paint coat thickness.

VI. EXPERIMENTAL RESULTS

In this section, the experimental analysis has been presented, which demonstrates the effectiveness of the proposed algorithms. All the auto parts mentioned in Table I have been successfully recognized and their poses estimated for the spray painting Cartesian robot shown in Fig. 1. Microsoft Kinect has

been used as a 3-D sensor for obtaining point clouds of the car parts, it was mounted on the robot as shown in Fig. 6(a).

A. Time Complexity Analysis

In this section, the time complexity of the proposed car part recognition pipeline has been discussed. Fig. 7 details the time

consumption of the proposed pipeline in *seconds*. It can be seen in Fig. 7(a) that the proposed recognition pipeline is able to differentiate the car parts in 80 ms on average. The parallel processing of this algorithm took more time than the serial implementation in this experiment. The reason behind this anomaly is that the *parallel overhead* of the parallelized version of this algorithm took more time than the actual processing of the algorithm. Since the number of samples to be found are eight, therefore, it was not feasible to use parallel version in this case. It was hypothesized that increasing the number of samples will improve the result in the favor of parallel version and it can be noticed for this case in Fig. 7(b). For a larger number of iterations, it can be seen that the parallel version performs much better. The code was tested on 4, 8, and 16 parallel workers, and slight improvement in terms of time consumption was observed as the number of parallel workers was increased. Fig. 7(c) shows the cost of application of dimensionality reduction upon the 3-D point sets. It can be noticed that PCA does not take enormous time in reducing the dimension of all of the auto parts. It was able to convert the point cloud into the 2-D binary image in less than 1 ms.

Furthermore, the time complexity of the car part recognition pipeline was also compared with other major object detection frameworks in the field of 3-D point cloud processing. The works presented in [22] and [44]–[46] were used as benchmarks and reimplemented for the current application. The time for the identification of individual parts was recorded for each one of them and is presented in Table I. It can be seen that proposed pipeline gives much better results in terms of computational time as compared to other major implementations. The average time complexity of the algorithms presented in [22] and [44]–[46] was found to be 84.2161 s, which was much higher than the average time complexity of proposed algorithm that was found to be 0.0890 s. This proves the research claim of this paper that dimensionality reduction can be used to accelerate the object detection algorithms in 3-D point clouds.

B. Comparative Analysis for Pose Estimation

In order to ensure the accuracy of pose estimation algorithms, an experiment was designed to evaluate optima of the fitness function after successful alignment of point clouds with randomly introduced geometric transformations. First, the point clouds were captured at their standard positions, later on their positions were changed and the point clouds were recaptured. In this paper, the requirement for pose error between end effector and the spray surface [as shown in Fig. 7(d)] is that the normal distance (d) between the spray nozzle and spray surface should be kept within the range of 10–20 cm at the start of spray painting trajectory. This requirement corresponds to the required paint coat thickness, and is kept in mind while tuning the parameters of registration algorithm utilized in this paper. The units used for pose measurement were selected as *radians* for rotation and *meters* for translation. This allowed the retrieval of information for calculating the error metric that is defined below. Fig. 7(d) shows the spray nozzle and the spray surface. Industrial requirements dictate that the ideal paint coat thickness can be achieved when the normal of the spray surface points in the direction of

spray nozzle. In this application, the nozzle design also dictates that the distance between surface and the nozzle is kept constant. The parameter d can be shown as the directed distance between the points p_e and p_i . Mathematically, it can be written as

$$d = \|p_e - p_i\|. \quad (14)$$

Assuming that an initial euclidean transformation T exists between the target and source clouds such that $T = [\theta_x \ \theta_y \ \theta_z \ x \ y \ z]$, where $[\theta_x \ \theta_y \ \theta_z]$ denotes the so called Euler's angles about xyz-axes (measured in *radians*) and $[x \ y \ z]$ denote the position in Cartesian coordinate frame (measured in *meters*). In this experiment, five random initial transformations are chosen. They were applied on each of the car parts, in order to obtain the source point clouds. First, ICP was applied and the minimum value was obtained for the fitness function mentioned in (12). Later on, combined ICP-GA algorithm was applied for further minimization of (12). The transformations used in this experiment are mentioned as

$$\begin{aligned} T_1 &= [0.5 \times \pi \quad 0 \quad 0 \quad 0.8 \quad 0 \quad 0.5] \\ T_2 &= [0.5 \times \pi \quad 0 \quad 0.1 \times \pi \quad 0.8 \quad 0.1 \quad 0.5] \\ T_3 &= [0.5 \times \pi \quad 0.2 \times \pi \quad 0.3 \times \pi \quad 0.1 \quad 0.3 \quad 0.1] \\ T_4 &= [0.6 \times \pi \quad 0.3 \times \pi \quad 0.4 \times \pi \quad 0.3 \quad 0.2 \quad 0.4] \\ T_5 &= [0.3 \times \pi \quad 0.1 \times \pi \quad 0.6 \times \pi \quad 0.4 \quad 0.6 \quad 0.3]. \end{aligned} \quad (15)$$

The results for this experiment are presented in Table I. As mentioned in Algorithm 3, the proposed method minimizes the fitness function mentioned in (12) until its value is less than e^{-6} . It can be seen in Table I that in comparison to the standalone application of ICP, the proposed method is superior in terms of accuracy. It has been observed in our initial tests that the smaller pose error is significant in defining an accurate trajectory for the spray painting process and ensuring the ideal range of distance d as mentioned in preceding passages.

VII. CONCLUSION

In this paper, the problem of recognition and pose estimation of auto part point clouds has been solved for automated operation of spray painting robot. First, the recognition pipeline was designed, which incorporated PCA as a method of dimensionality reduction. Application of PCA reduced the problem complexity of the recognition to the feature matching of 2-D binary images. Novel statistical techniques were designed to measure the similarity of these images using the shape features from FSM. Finally, the recognition algorithm was implemented in real time and desired results were presented. Furthermore, the pose estimation was carried out using ICP algorithm. It was noticed that ICP has a potential to get trapped in the local minimum. To solve this issue, it was combined with the GA. This ensured the search of global minimum and decreased the pose estimation error furthermore. The time complexity analysis for the proposed object detection pipeline was presented and compared with other major object detection systems in 3-D point cloud processing. In addition to this, the error analysis for the proposed pose estimation pipeline was presented. All the results supported the research claims and proved the efficacy of

proposed algorithms. The methods presented in this paper prepare the spray painting robot to recognize and evaluate the pose of auto parts autonomous painting.

REFERENCES

- [1] KPMG International Cooperative, "Global automotive executive survey," Ganghoferstrasse 29, Munich, Germany, Tech. Rep. gaes2017. [Online]. Available: <https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2017/01/global-automotive-executive-survey-2017.pdf>
- [2] B. Huang, M. Ye, Y. Hu, S.-L. Lee, A. Vandini, and G. Yang, "A multi-robot cooperation framework for sewing personalised stent grafts," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1776–1785, Apr. 2018.
- [3] K.-S. Hwang, W.-C. Jiang, Y.-J. Chen, and H. Shi, "Motion segmentation and balancing for a biped robot's imitation learning," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1099–1108, Jun. 2017.
- [4] G. Dong and Z. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica*, vol. 115, pp. 291–302, Oct. 2015.
- [5] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3d object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.
- [6] H. Liu, "Exploring human hand capabilities into embedded multifingered object manipulation," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 389–398, Aug. 2011.
- [7] H. Zhang and C. Ye, "An indoor wayfinding system based on geometric features aided graph SLAM for the visually impaired," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1592–1604, Sep. 2017.
- [8] A. W. Vieira, P. L. J. Drews, and M. F. M. Campos, "Spatial density patterns for efficient change detection in 3d environment for autonomous surveillance robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 3, pp. 766–774, Jul. 2014.
- [9] Y. Luo, H. Ma, and L. Zhou, "DEM retrieval from airborne LiDAR point clouds in mountain areas via deep neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1770–1774, Oct. 2017.
- [10] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1383–1396, Jul. 2016.
- [11] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2155–2162.
- [12] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino, "3d geometric scale variability in range images: Features and descriptors," *Int. J. Comput. Vis.*, vol. 99, no. 2, pp. 232–255, Apr. 2012.
- [13] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.
- [14] J. Xie, F. Zhu, G. Dai, L. Shao, and Y. Fang, "Progressive shape-distribution-encoder for learning 3d shape representation," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1231–1242, Mar. 2017.
- [15] K. Guo, D. Zou, and X. Chen, "3d mesh labeling via deep convolutional neural networks," *ACM Trans. Graph.*, vol. 35, no. 1, pp. 1–12, Dec. 2015.
- [16] Z. Wu *et al.*, "3d ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.
- [17] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3d data," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.
- [18] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 77–85, 2017.
- [19] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3d object retrieval with deep embedding network," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5526–5537, Dec. 2016.
- [20] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, pp. 2859–2900, 2015. [Online]. Available: <http://jmlr.org/papers/v16/cunningham15a.html>
- [21] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2006.
- [22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [23] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "Fast-match: Fast affine template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2331–2338.
- [24] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
- [25] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jun. 2010.
- [26] X. Wang, X. Bai, T. Ma, W. Liu, and L. J. Latecki, "Fan shape model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 151–158.
- [27] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [28] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.
- [29] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.
- [30] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1459–1473, Aug. 2010.
- [31] L. Li, M. Yang, C. Wang, and B. Wang, "Cubature split covariance intersection filter-based point set registration," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3942–3953, Aug. 2018.
- [32] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
- [33] S. Ji, Y. Ren, Z. Ji, X. Liu, and G. Hong, "An improved method for registration of point cloud," *Optik Int. J. Light Electron Opt.*, vol. 140, pp. 451–458, Jul. 2017.
- [34] L. Yan, J. Tan, H. Liu, H. Xie, and C. Chen, "Automatic registration of TLS-TLS and TLS-MLS point clouds using a genetic algorithm," *Sensors*, vol. 17, no. 9, pp. 1979–1997, 2017.
- [35] Q. Muhlbaier, K. Kuhnlenz, and M. Buss, "Fusing laser and vision data with a genetic ICP algorithm," in *Proc. 10th Int. Conf. Control, Automat. Robot. Vis.*, Dec. 2008, pp. 1844–1849.
- [36] R. B. Rusu, *Semantic 3D Object Maps for Everyday Robot Manipulation*. Berlin, Germany: Springer, 2013.
- [37] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, Shanghai, China, 2011, May 9–13.
- [38] M. Gallagher and T. Downs, "Visualization of learning in multilayer perceptron networks using principal component analysis," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 33, no. 1, pp. 28–34, Feb. 2003.
- [39] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [40] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985.
- [41] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.
- [42] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O'Reilly media, 2008.
- [43] H. Gao, C. Ding, C. Song, and J. Mei, "Automated inspection of e-shaped magnetic core elements using k-TSL-center clustering and active shape models," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1782–1789, Aug. 2013.
- [44] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," in *Comput. Vision*. Berlin, Germany: Springer, 2012, pp. 511–524.
- [45] I. Sipiran and B. Bustos, "Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes," *Vis. Comput.*, vol. 27, no. 11, pp. 963–976, Jul. 2011.
- [46] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, Sep. 2009, pp. 689–696.



Weiyang Lin (M'18) received the bachelor's and M.Sc. degree in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 2006 and 2008, respectively, and the Ph.D. degree in mechatronics engineering from Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China, in 2014.

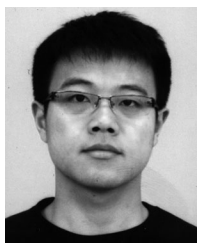
He is currently an Assistant Professor with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology. His research interests include parallel manipulators, robotic motion control, and visual servoing.



Ali Anwar received the B.Eng. degree in electronics engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2012 and the M.Eng. degree in control science and engineering in 2015 from Harbin Institute of Technology, Harbin, China, where he is currently working toward the Ph.D. degree in control science and engineering.

His research interests include visual servo control, industrial robotics, and computer vision.

Mr. Anwar is currently a Reviewer of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.



Zhan Li received the Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2015, and the B.Sc. degree in automation, and the master's degree in pattern recognition and intelligent system from Harbin Engineering University, Harbin, China, in 2008 and 2011, respectively.

From 2013 to 2014, he was visiting University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Canada, as a joint Ph.D. student supported by China Scholarship Council (CSC), working with coadvisor Prof. Hugh Liu. He is currently a Lecturer with the School of Astronautics, Harbin Institute of Technology. His current research interests include motion control, industrial robot control, robust control of small UAVs, and cooperative control of multivehicle systems.



Mingsi Tong received the Ph.D. degree in mechatronic engineering from Harbin Institute of Technology, Harbin, China, in 2016.

During graduate studies, he also concurrently studied in the National Institute of Standards and Technology, DC, USA, as a Guest Researcher. He is currently an Assistant Professor with the School of Mechatronics Engineering, Harbin Institute of Technology. His research interests include forensic science, pattern recognition, and surface metrology.



Jianbin Qiu (M'10–SM'15) received the B.Eng. and Ph.D. degrees in mechanical and electrical engineering from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in mechatronics engineering from the City University of Hong Kong, Hong Kong.

He is currently a Full Professor with the School of Astronautics, Harbin Institute of Technology, Harbin, China. He was an Alexander von Humboldt Research Fellow with the Institute for Automatic Control and Complex Systems, Uni-

versity of Duisburg-Essen, Duisburg, Germany. His current research interests include intelligent and hybrid control systems, signal processing, and robotics.

Prof. Qiu is the Chairman of the IEEE Industrial Electronics Society Harbin Chapter, China. He is an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS.



Huijun Gao (SM'09–F'13) received the Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China.

He carried out his Postdoctoral Research with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Since 2004, he has been with Harbin Institute of Technology, where he is currently a Full Professor, the Dean of the School of Science, and the Director of the Research Institute of Intelligent Control and Systems. His current

research interests include intelligent and robust control, robotics, mechatronics, and their engineering applications.

Prof. Gao is the Co-Editor-in-Chief of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and an Associate Editor for *Automatica*, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE/ASME TRANSACTIONS ON MECHATRONICS. He is an IEEE Industrial Electronics Society (IES) Administration Committee (AdCom) member, and a Council Member of IFAC.