

# A Robotic Defect Inspection System for Free-form Specular Surfaces

Shengzeng Huo, David Navarro-Alarcon and David TW Chik

**Abstract**—In this paper, we present a robotic system to automatically perform defect inspection tasks over free-form specular surfaces, which the image acquisition sub-system is equipped with a 6-DOF robot manipulator to achieve flexible scanning. Given the mesh model of the workpiece, we implement K-means based region segmentation algorithm on the point cloud after preprocessing. Then, we take the smooth regions as input to plan the scanning path. A projection registration method that robustly localizes the object in the robot’s frame is proposed for real-time workpiece localization. According to the optical features of the high-resolution line scan, we design an image processing pipeline to detect defects from the captured images. We report a detailed experimental study to validate the proposed methodology.

## I. INTRODUCTION

Surface inspection is important in modern manufacturing because defects severely affect the value and quality of products. The common practice of many factories is still to hire numerous workers to perform the inspection task manually, which is costly and time-consuming, and the performance can be very subjective. Based on machine vision [1], several automatic inspection systems in various fields have been proposed, such as railway surfaces [2], automotive exterior body parts [3], and steel surfaces [4].

A typical setup of an automatic vision inspection system in a production line is shown in Fig. 1(a), where a camera is mounted on top of the workspace to scan objects continuously. [5] proposed a fast fabric inspection system focusing on objects with a continuous stream, such as paper and plywood. [6] used matrix cameras instead of linear ones with  $1\text{ mm}^2$  resolution to deal with continuous flat metallic products in the production line. [7] adopted a diffusion-lighting system to avoid the strong reflections from the surface of car’s body. Using a rotating table with a fixed scan camera is another common setup, as shown in Fig. 1(b). The camera can capture multiple images of parts from different angles with this setting. [8] used an industrial camera and coaxial area lighting to scan smooth products on an electronically controlled rotary table. [9] used a high-resolution line camera to scan cultural heritage objects, with a table that rotates 5 degrees after each scan.

The common feature of the above two methods is a fixed camera with a moving object, making it only suitable for

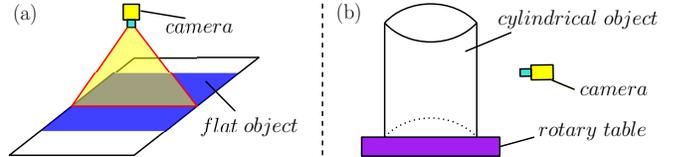


Fig. 1: Two typical inspection setups in production line: (a) Mount the camera on top and capture continuous flat plane, (b) Capture images of different angles with a object on a rotary table.

certain standard shaped objects. However, free-form parts (a surface normal that is continuous almost everywhere except at vertices, edges, and cusps [10]) are widespread in industrial production, while neither of them can deal with this issue. Considering the camera settings and real inspection conditions, we concentrate on convex free-form objects [11] in this paper. To enable a flexible scanning path for general free-form surfaces, we try to combine the robotic manipulator with the image acquisition module of machine vision.

Focusing on automatic defect inspection issues over free-form specular surfaces, high reflection ratio and continuous variation of curvature are two main challenges in our task. The main original contributions with respect to these challenges of this paper are as follows:

- Provide a complete pipeline to plan the path of the line scanner with respect to the given CAD (Computer-aided design) mesh model.
- Design an ICP (Iterative Closest Point)-based registration method for localization of specular objects.
- Validate the developed automatic inspection system with experiments using free-form specular objects.

The rest of this paper is organized as follows: Sec. 2 introduces the architecture of the system. Sec. 3 describes the planning and control methods. Sec. 4 presents experimental results. Sec. 5 gives final conclusion.

## II. ROBOTIC SYSTEM

For an outstanding machine vision system for defect inspection, highlighting defects and sufficient detection resolution are two fundamental requirements. Applying area scan camera into inspection applications has several disadvantages, such as blurring, inspection speed, and restriction of inspection range [5]. By contrast, the high-resolution, high-frequency characteristics of the line scan camera make it ideal for scanning surfaces with continuously varying curvature [12]. Thus, a line scan camera, Basler raL2048-48gm GigE camera, is used in our system. Generally, lighting is always needed in the field of machine vision to enhance the

S. Huo and D Navarro-Alarcon are with The Hong Kong Polytechnic University, KLN, Hong Kong. Corresponding author: dna@ieee.org.

D. Chik is with the Hong Kong Applied Science and Technology Research Institute, NT, Hong Kong.

This work is supported by ASTRI (grant PO2019/399), by the Research Grants Council (grant 14203917), by the Jiangsu Industrial Technology Research Institute Collaborative Research Program Scheme (grant ZG9V), by the Key-Area Research and Development Program of Guangdong Province 2020 (project 76), and by PolyU (grants G-YBYT and 4-ZZHJ).

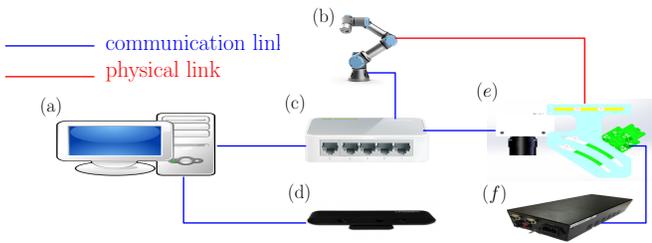


Fig. 2: Architecture of the proposed robotic inspection system. (a) Linux PC. (b) UR3 robot manipulator. (c) Network router. (d) Orbbec Astra S depth camera. (e) Image acquisition system, including line scan camera, line light and a holder. (f) Analog control box high power LED strobe.

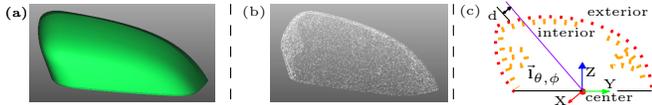


Fig. 3: Preprocessing for input mesh model: (a) Typical free-form CAD mesh model. (b) Sampled point cloud. (c) Conceptual explanation of irrelevant points filtering.

difference between the defects and the background. Focusing on specular surfaces, dark-field illumination configuration is always adopted [13], in which only the light encountering the defects passes into the camera. Thus, the image acquisition system also includes a line lighting to cooperate with the line scan camera.

Fig. 2(a) illustrates the architecture of our proposed robotic prototype. A network router and ROS [14] are used to establish a local network and communicate different components in the system.

### III. METHODOLOGY

#### A. Preprocessing

We firstly obtain a point cloud from meshes by randomly selecting a given number of points on each triangle via [2]. An example about a mesh model and sampled point cloud  $M$  are shown in Fig. 3(a) and (b), respectively. Each point in  $M$  is a vector  $\vec{\mathbf{m}} = [x \ y \ z]^T$ . Generally, only the exterior surface of the object is emphasized in defect inspection of quality control, while the sampled  $M$  includes points from all faces of the object. The irrelevant points are not only useless for our scanning, but also affect the following feature extraction procedure. Hence, we try to filter them out in this preprocessing section.

In order to define irrelevant points mathematically, we consider them as the closer one along a search direction from the center  $\vec{\mathbf{c}}$  to the space. At first, the center  $\vec{\mathbf{c}}$  is defined as

$$\vec{\mathbf{c}} = \frac{\sum_{i=0}^{N_M} \vec{\mathbf{m}}_i}{N_M} \quad (1)$$

where  $\vec{\mathbf{m}}_i$  is  $i$ -th point of  $M$ , and  $N_M$  is the size of  $M$ .

The next step is to eject multiple search lines from  $\vec{\mathbf{c}}$  to the surface in different directions. We utilize a sphere coordinate to distribute the lines evenly [15].

$$\vec{\mathbf{l}}_{\theta, \phi} = [r \sin \theta \cos \phi \ r \sin \theta \sin \phi \ r \cos \theta]^T \quad (2)$$

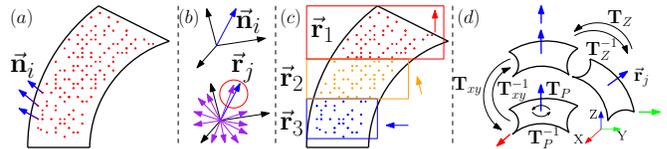


Fig. 4: Conceptual representation of K-means based region segmentation. (a) Normal estimation. (b) Random  $K$  feature vectors initialization and classification in terms of each normal vector. (c) Feature vectors update. (d) Region pose adjustment.

where  $r, \theta, \phi$  represent radius, polar angle, and azimuthal angle, respectively. Illustrated in Fig. 3(c), we ought to look for the points on each search line and select the one whose Euclidean distance to the center is biggest. Due to the sampling discreteness, we consider the points whose distance to the search line is smaller than a threshold as intersection points. The distance between  $i$ -th point  $\vec{\mathbf{m}}_i$  and line  $\vec{\mathbf{l}}_{\theta, \phi}$  can be computed by

$$d = \frac{|\vec{\mathbf{q}}_i \cdot \vec{\mathbf{l}}_{\theta, \phi}|}{|\vec{\mathbf{l}}_{\theta, \phi}|} \quad (3)$$

where  $\vec{\mathbf{q}}_i = \vec{\mathbf{m}}_i - \vec{\mathbf{c}}$ , and each line  $\vec{\mathbf{l}}_{\theta, \phi}$  is calculated by (2). Among those points on the search line, only the point farthest from  $\vec{\mathbf{c}}$  with Euclidean distance will be saved. At last, we implement an alternative iteration search to filter out repeated points.

#### B. Region Segmentation

Although there are different algorithms in the field of point cloud segmentation [16], most of them focus on absorbing points on the edge of the existing clusters instead of global similarity, which is not applicable to our issue. Here, we formulate the region segmentation issue as a clustering analysis problem [17]. In our method, we build on K-means clustering [18], an unsupervised classification algorithm to partition the free-form surface into regions.

It is intuitive that the normal vectors of all points of a flat plane are completely the same. Thus, our region segmentation depends on normal estimation from PCL [19]. Illustrated in Fig. 4(a), the normal vector of a point is  $\vec{\mathbf{n}}_i = [n_x \ n_y \ n_z]^T$ . Facing the continuous varying curvature, however, it is impossible to obtain complete flat planes. Our solution is setting an allowable angle difference between the normal of points and the representative normal vector of the region.

At the beginning, we initialize  $K$  normal vectors ( $\vec{\mathbf{r}}_1, \vec{\mathbf{r}}_2, \dots, \vec{\mathbf{r}}_K$ ) for regions ( $R_1, R_2, \dots, R_K$ ). Illustrated in Fig. 4(b), for each normal  $\vec{\mathbf{n}}_i$ , we measure the similarity between it and feature vectors of different regions respectively:

$$\alpha_{ij} = \arccos \left( \frac{|\vec{\mathbf{n}}_i \cdot \vec{\mathbf{r}}_j|}{|\vec{\mathbf{n}}_i| \cdot |\vec{\mathbf{r}}_j|} \right). \quad (4)$$

With this similarity measurement, we emphasize the region  $R_j$  whose angle difference with  $\vec{\mathbf{n}}_i$  is smallest, classifying  $\vec{\mathbf{n}}_i$  as one component of the region  $R_j$ . After the classification, each region  $R_j$  consists of several normal vectors. To improve

the representative performance of feature vectors, shown in Fig. 4(c), we use the average normal to update  $\vec{\mathbf{r}}_j$ :

$$\vec{\mathbf{r}}_j = \frac{\frac{1}{N_j} \sum_{i=1}^{N_j} \vec{\mathbf{n}}_i}{\|\frac{1}{N_j} \sum_{i=1}^{N_j} \vec{\mathbf{n}}_i\|}. \quad (5)$$

where  $\vec{\mathbf{n}}_i$  is the normal of  $i$ -th point in  $j$ -th region  $R_j$ , and  $N_j$  is the size of region  $R_j$ .

Originally,  $K$ , the number of clusters, is the only hyper-parameter of K-means clustering. Since the prior knowledge about the workpiece is unknown, it is impossible to set an optimal  $K$  in advance. To overcome this issue, our algorithm is able to tune the value of  $K$  automatically until termination, which the condition is

$$\max \alpha_{ij} < \gamma. \quad (6)$$

$i$  is the index of normal  $\vec{\mathbf{n}}_i$ ,  $j$  is the index of the point's corresponding region  $R_j$ . Here,  $\alpha_{ij}$  denotes the angle difference between each normal vector  $\vec{\mathbf{n}}_i$  and its corresponding feature vector  $\vec{\mathbf{r}}_j$  of classified region  $R_j$ .  $\gamma$  is a pre-defined angle threshold, representing the level of the segmentation.

As discussed above, the whole segmentation algorithm includes two loops, an inner loop for classification and an outer loop for increasing the value of  $K$ . For a small value  $K$ , it is general that an available solution is impossible to acquire no matter how many loops the algorithm iterates. Thus, we ought to set the maximum limit for the number of iterations. When the number of iterations  $\sigma_K$  with respect to the current  $K$  exceeds the pre-defined threshold  $\sigma^T$ , the inner loop of current  $K$  is stopped. The algorithm switches to the outer loop for increasing the value of  $K$  as  $K \leftarrow K + 1$ . Then, the algorithm enters the inter loop again to search the solution.

### C. Path Planning

After region segmentation, the shape of each region  $R_j$  is various and irregular. This path planning algorithm takes regions  $(R_1, R_2, \dots, R_{N_R})$  as input and output scanning path according to the acquisition model of line scan camera and shortest path criteria. At first, to simplify the following planning procedure, we implement an adjustment to transform each region  $R_j$  by three  $4 \times 4$  matrices, as shown in Fig. 4(d). The detailed explanation is:

- 1)  $\mathbf{T}_Z$ : Rotate  $\vec{\mathbf{r}}_j$  to  $\vec{\mathbf{Z}} = (0, 0, 1)$ .
- 2)  $\mathbf{T}_{xy}$ : Translate to x-y plane  $Z = 0$ .
- 3)  $\mathbf{T}_P$ : Rotate around Z-axis to minimize the projection length along Y-axis  $\vec{\mathbf{Y}} = (0, 1, 0)$ .

To apply our high-resolution image acquisition sub-system to inspect free-form surfaces, normal deviation is the most important factor of the sensing quality. Cooperating with the line lighting, the difference between the surface normal and the facing direction of the line scan camera should be small enough to avoid the shiny spot phenomenon. Thus, we should ensure the curvature of the free-form region with respect to a linear scanning path should not exceed the scanning range of the camera [20].

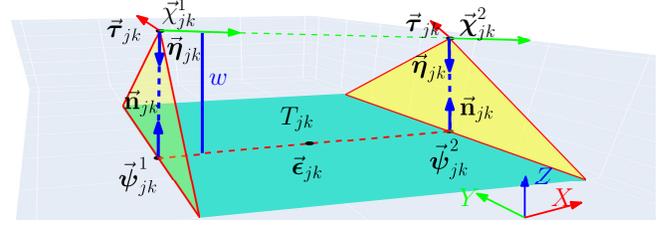


Fig. 5: Path planning with respect to a patch  $T_{jk}$ . The cyan plane represents a patch and two frames  $\vec{\mathbf{X}}_{jk}^1$  and  $\vec{\mathbf{X}}_{jk}^2$  denote two scanning poses of the camera.

To meet the facing direction requirement, we divide the region  $R_j$  into patches that can be scanned with a one-way linear motion. Similar to the above K-means region segmentation, we partition the region  $R_j$  along the motion direction of the camera into  $k_j$  patches  $(T_{j1}, T_{j1}, \dots, T_{jk_j})$  that meet the condition (Eq. 6).

After patch division, we plan the start and the end of the scanning with respect to patch  $T_{jk}$ , as shown in Fig. 5:

$$\vec{\mathbf{X}}_{jk}^1 = \vec{\mathbf{\Psi}}_{jk}^1 + w \cdot \vec{\mathbf{t}}_{jk}, \quad \vec{\mathbf{X}}_{jk}^2 = \vec{\mathbf{\Psi}}_{jk}^2 + w \cdot \vec{\mathbf{t}}_{jk} \quad (7)$$

where  $\vec{\mathbf{\Psi}}_{jk}^1 = [x_{min} \ \varepsilon_y \ \varepsilon_z]^T$  and  $\vec{\mathbf{\Psi}}_{jk}^2 = [x_{max} \ \varepsilon_y \ \varepsilon_z]^T$ ,  $x_{max}$  and  $x_{min}$  are the maximum and minimum values of patch  $T_{jk}$  along the x-axis, respectively.  $\varepsilon_y$  and  $\varepsilon_z$  are elements of  $\vec{\mathbf{e}}_{jk}$ , computed by (1) respect to patch  $T_{jk}$ .  $w$  denotes the working distance, the distance between the front end of the lens and the subject; and  $\vec{\mathbf{t}}_{jk}$  is the average normal of patch  $T_{jk}$  computed by (5).

For orientation, the normal direction of camera is  $\vec{\mathbf{n}}_{jk} = -\vec{\mathbf{t}}_{jk}$ . In addition, the moving direction of the line scan camera for patch  $T_{jk}$  is  $\vec{\mathbf{t}}_{jk} = \vec{\mathbf{x}} = (1, 0, 0)$ . After the detailed definition, the planned poses should return to their original position before the adjustment. Hence, the whole definition about the pose  $\vec{\mathbf{p}}_{jk}^i$  is:

$$\vec{\mathbf{p}}_{jk}^i = \left\{ \mathbf{T}_P^{inv} \cdot \vec{\mathbf{x}}_{jk}^i, \mathbf{T}_R^{inv} \cdot \vec{\mathbf{n}}_{jk}^i, \mathbf{T}_R^{inv} \cdot \vec{\mathbf{t}}_{jk}^i \right\} \quad (8)$$

where  $\mathbf{T}_P^{inv} = \mathbf{T}_Z^{-1} \cdot \mathbf{T}_{xy}^{-1} \cdot \mathbf{T}_P^{-1}$  and  $\mathbf{T}_R^{inv} = \mathbf{T}_Z^{-1} \cdot \mathbf{T}_P^{-1} \cdot \vec{\mathbf{x}}_{jk}^i$ ,  $\vec{\mathbf{n}}_{jk}^i$  and  $\vec{\mathbf{t}}_{jk}^i$  are  $4 \times 1$  vectors in homogeneous coordinates.

After computing the scanning path  $P_{jk}$  for patch  $T_{jk}$ , we can stack them as a complete scanning path  $P_j$  for region  $R_j$ . In order to obtain a complete path for the workpiece, we integrate the path  $P_j$  of different regions  $R_j$  into a whole scanning path based on shortest path criteria. Here, we build on the theory of Nearest Neighbor Search [21], whose principle is searching the nearest region in the rest of unscanned regions as the next one after finishing the current region.

### D. Localization

Since the above path planning is based on the object's frame, we need a transformation matrix  ${}^B\mathbf{T}_O = [ \mathbf{R}_B \vec{\mathbf{t}}_B ]$  between it and robot base's frame to control the manipulator, where  $\mathbf{R}_B$  is a  $3 \times 3$  rotation matrix and  $\vec{\mathbf{t}}_B$  is a  $3 \times 1$  translation column vector.

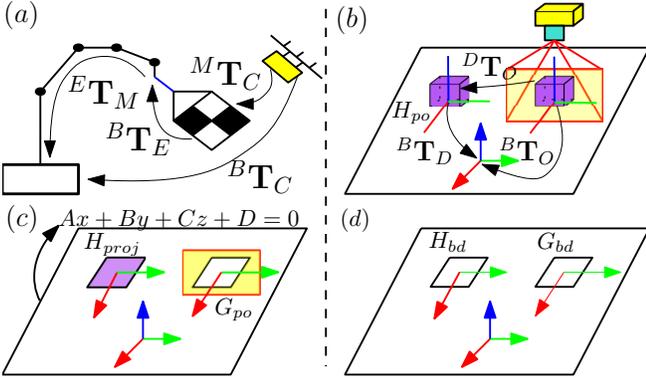


Fig. 6: Procedures of the projection registration algorithm: (a) Hand-eye calibration. (b) Real experiment setup. (c) Projection point cloud  $H_{proj}$  and capturing point cloud  $G_{po}$ . (d) Two boundary point clouds  $H_{bd}$  and  $G_{bd}$ .

To measure the transformation, we utilize a depth camera to capture the 3D points of the specular object. A hand eye calibration [22] should be conducted in advance to describe the relationship between the robot and the camera, as shown in Fig. 6(a):

$${}^B\mathbf{T}_C = {}^B\mathbf{T}_E \cdot {}^E\mathbf{T}_M \cdot {}^M\mathbf{T}_C \quad (9)$$

where C represents the depth camera, B is robot base, E represents the end-effector of the robot manipulator, and M represents AR markers on the calibration board.  ${}^B\mathbf{T}_E$  can be achieved by robot forward kinematics.  ${}^E\mathbf{T}_M$  is determined by our designed calibration board.  ${}^M\mathbf{T}_C$  is detected by the RGB-D camera via [23].

Although depth camera is able to obtain 3D points about the object, its performance is strongly affected due to the high reflection rate of the specular surface. Thus, we propose a projection registration method to deal with this issue. Fig. 6(b) illustrates the setting of our system, where an RGB-D camera is mounted on top of the workspace.

$H_{po}$  and  $G_{po}$  denote the designed pose and the actual pose of the object in robot's frame respectively. A hole appears in  $G_{po}$  because it is blocked by the specular surface of the object, as shown in Fig 6(c). Given the manipulation plane  $A \cdot x + B \cdot y + C \cdot z + D = 0$  in robot's frame, we project  $H_{po}$  to it to get  $H_{proj}$ :

$$\vec{\mathbf{h}}_i^{proj} = \vec{\mathbf{h}}_i^{po} - (\vec{\mathbf{f}} \cdot \vec{\mathbf{h}}_i^{po}) \cdot \vec{\mathbf{f}} \quad (10)$$

where  $\vec{\mathbf{f}} = [A \ B \ C \ D]^T$  is vector representation of the parametric plane;  $\vec{\mathbf{h}}_i^{po}$  and  $\vec{\mathbf{h}}_i^{proj}$  are  $i$ -th point in homogeneous coordinates of  $H_{po}$  and  $H_{proj}$  respectively. Given  $H_{proj}$  and  $G_{po}$ , we can extract their boundaries to compute their transformation matrix. [19] offered a function to extract the boundary of  $H_{proj}$ , denoted as  $H_{bd}$ . For  $G_{po}$ , we adopt a method similar to Fig. 3(c) to achieve that, denoted as  $G_{bd}$ . Finally, we can use the classical ICP [24] algorithm to derive the least square transformation matrix  ${}^D\mathbf{T}_O$  between  $H_{bd}$  and  $G_{bd}$ . Then,  ${}^B\mathbf{T}_O$  can be obtained by  ${}^B\mathbf{T}_O = {}^B\mathbf{T}_D \cdot {}^D\mathbf{T}_O$ .

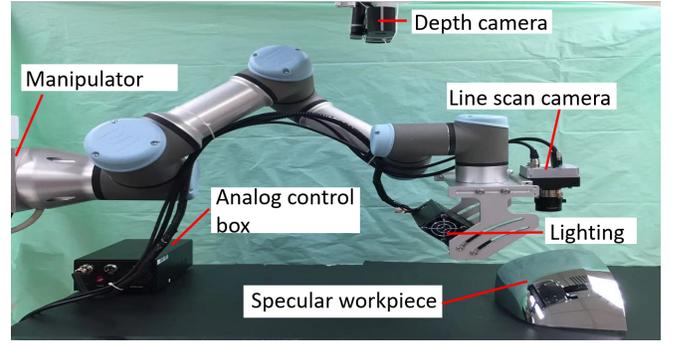


Fig. 7: Experiment setup of the inspection task. The experimental subject specular side-mirror is placed on the manipulation plane.

### E. Control

With the transformation matrix  ${}^B\mathbf{T}_O = [ \mathbf{R}_B \ \vec{\mathbf{t}}_B ]$ , we obtain the scanning path in the robot's frame by

$$\vec{\mathbf{p}}_B = \{ {}^B\mathbf{T}_O \cdot \vec{\mathbf{x}}_O, \mathbf{R}_B \cdot \vec{\boldsymbol{\eta}}_O, \mathbf{R}_B \cdot \vec{\boldsymbol{\tau}}_O \} \quad (11)$$

where  $\vec{\boldsymbol{\eta}}_O$  and  $\vec{\boldsymbol{\tau}}_O$  are  $3 \times 1$  column vectors, and  $\vec{\mathbf{x}}_O$  is a  $4 \times 1$  column vector in homogeneous coordinates.

Every pose should be a  $6 \times 1$  column vector, representing the position and the rotation vector in the robot frame to control the robot [25], [26]. Instead of using the original end-effector of the manipulator to implement the scanning path, we set the TCP (Tool Center Point) in end-effector's frame as our image acquisition sub-system to reach the scanning poses. For position,  $\vec{\mathbf{x}}_B = {}^B\mathbf{T}_O \cdot \vec{\mathbf{x}}_O$  can directly define the desired position of the end-effector. For orientation, we need to obtain the complete transformation for the desired directions ( $\vec{\boldsymbol{\eta}}_B = \mathbf{R}_B \cdot \vec{\boldsymbol{\eta}}_O$  and  $\vec{\boldsymbol{\tau}}_B = \mathbf{R}_B \cdot \vec{\boldsymbol{\tau}}_O$ ). Then, we acquire the rotation vector  $\vec{\boldsymbol{\xi}}_B$  via [27]. Finally, the  $6 \times 1$  pose vector is  $\vec{\boldsymbol{\pi}} = [\vec{\mathbf{x}}^T \ \vec{\boldsymbol{\xi}}^T]^T$ .

## IV. RESULTS

### A. Experiment Setup

The experimental setup of our system is illustrated in Fig. 7. A UR3 manipulator from Universal Robots is used to achieve a six-DOF path. The line lighting and line scan camera are equipped with the end-effector of the UR3 manipulator via our designed holder. An Orbbec astra depth camera is mounted on top of the manipulation space, sensing the visual information of the manipulation plane. The transformation matrix between the depth camera and the UR3 manipulator is calibrated by ArUco markers in advance. Velocity and acceleration of the manipulator are empirically set as  $0.05m/s$  and  $0.05m/s^2$  due to safety reasons. A specular side mirror from the automotive industry is taken as an experimental workpiece, placed on the manipulation plane. Several defects are added manually on the specular surface and the observation results from professional workers are regarded as the benchmark of this inspection task.

The lens used in our platform is Cinegon 1.8/16-0901. Working distance  $w$  of the line scan camera is set as  $15.7mm$  according to the focal length of the lens.

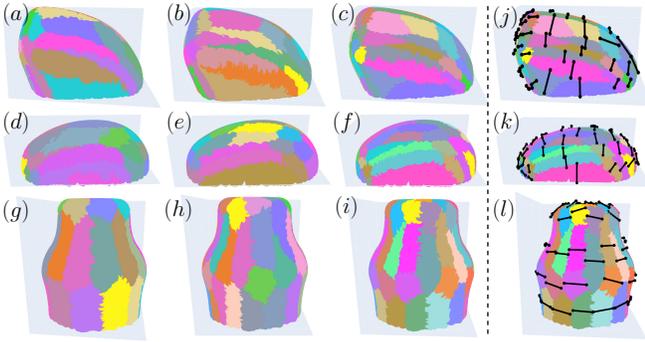


Fig. 8: (a)-(c), (d)-(f) and (g)-(i) are region segmentation results of side mirror, mouse and bottle respectively with different angle thresholds  $\gamma$ . (j)-(l) are simplified scanning paths for regions respect to side mirror, mouse and bottle respectively.

In order to comprehensively evaluate the performance of our proposed system in this challenging task, we consider three vital elements in this section:

- 1) Robustness and computation efficiency of region segmentation and path planning with respect to different free-form shapes.
- 2) Registration error between the projection boundary of the CAD model and the detected boundary after the indirect alignment.
- 3) Accuracy of automatic defects detection over the captured images with our proposed image processing.

### B. Path Planning Performance

One of the challenges of our task is the variety of object shapes. The CAD mesh models of a side mirror, a mouse, and a bottle are applied in this section to validate the robustness in terms of different free-form shapes. In addition, different input parameters are tested to compare their results.

Fig. 8 (a)-(i) (different colors represent different regions) illustrate the region segmentation results with respect to corresponding free-form objects and input parameters. It is intuitive that each region in the figure is connected and smooth. We can conclude from these intuitive results that our K-means unsupervised region segmentation algorithm is able to deal with different convex free-form objects.

The mathematical results, including the number of regions and efficiency, are shown in Table I. Given the threshold  $\gamma$ , our algorithm ensures the maximum angle difference with the representative feature vector lower than our requirement. Even if the sampling density increases and the shape of the workpiece varies, our algorithm is robust enough to find the available solution.

After region segmentation, our path planning algorithm takes the region segmentation result and the image acquisition optical parameters as input and outputs the scanning path. Fig. 8 (j)-(l) illustrates the achieved scanning path with respect to different regions. This conceptual representation shows our path planning meets the distance-optimal criteria in the face of various shapes of regions.

TABLE I: Path Planning Performance

Object	$N_E$	$\gamma$	$N_R$	T(s)
Side Mirror	4807	20	30	12
		18	36	17
		15	45	26
Mouse	3855	20	39	11
		18	40	17
		15	58	34
Bottle	7823	20	42	23
		18	55	34
		15	75	68

$N_E$ : size of exterior point cloud.  $\gamma$ : angle threshold in region segmentation.  $N_R$ : number of regions.  $T(s)$ : computation time for K-means region segmentation.

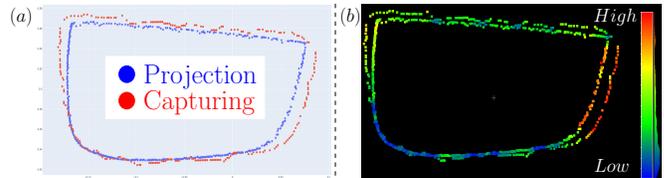


Fig. 9: Results of projection registration for specular surfaces. (a) Intuitive comparison between CAD mesh projection boundary and RGB-D camera capture boundary. (b) C2C absolute distances between them.

### C. Registration

Precise localization with respect to the workpiece in the robot's frame is another prerequisite of high-quality image acquisition. Even if the region segmentation is acceptable on the accurate CAD mesh model, the noisy registration results will make the camera unable to align with the inspection workpiece.

In this section, we take the specular side mirror as our experimental subject, as shown in Fig. 7. To test the accuracy of our indirect registration algorithm, we consider the boundary of clean CAD model projection as the ground truth. For our measurement, we apply our proposed projection registration algorithm to process the raw point cloud captured by the depth camera. Finally, we implement the ICP algorithm to registrate the ground truth and the boundary extracted from the measurement. The intuitive comparison results are shown in Fig. 9(a). Compared with the clean ground truth, the boundary of the measurement is more discrete and rough.

We take [2] as the measurement metric to compare the captured boundary with the ground truth mathematically, as shown in Fig. 9(b). The error mean and standard deviation are  $4.842 \times 10^{-3}$  and  $3.548 \times 10^{-3}$  respectively. Such low errors can guarantee the performance of our path planning with respect to the mesh model of the workpiece. In analysis, there are two components in this error. For one thing, the uncertainty of the Orbbec Astra 3D camera in depth measurements is bounded by  $\pm 15mm$  with a 2500mm range [28]. For another, we can notice that the registration error of the region with higher curvature is higher. That is because this situation is more challenging for the camera to eject the straight sensing optical lines.

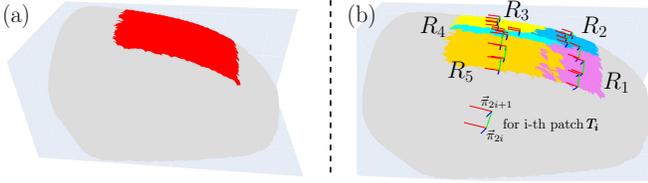


Fig. 10: Robotic experiment subject. (a) The chosen part of the specular side mirror. (b) The scanning path respect to the chosen surface.

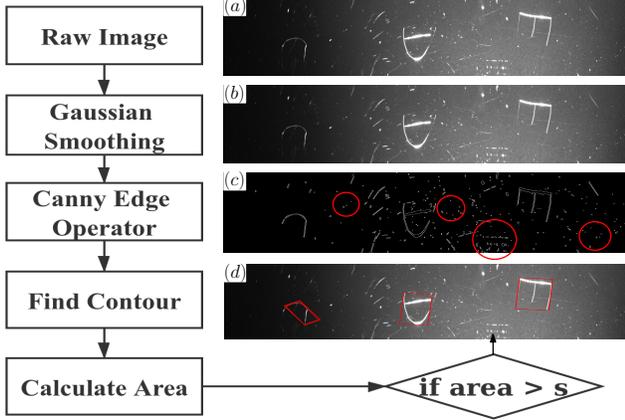


Fig. 11: Image processing pipeline and its corresponding results (a)-(d).

#### D. Detection Accuracy

After the procedures of path planning and workpiece localization, we acquire the scanning path in robot’s frame to implement the automatic inspection procedure. Constrained by the operational space of the UR3 manipulator, a small part of the specular side mirror is selected to conduct the experiments to illustrate the idea and validate our method, as shown in Fig. 10(a). Fig. 10(b) illustrates the region segmentation result and scanning path.

After the implementation of the robotic inspection procedure, we obtain several images about the specular surface, as shown in Fig. 11(a). However, there are many small white dots in the images. That’s because some dusts are captured in the view due to the high resolution of the line scan camera. For the purpose of improving the detection performance, we develop a image processing pipeline as shown in Fig. 11. Firstly, we utilize Gaussian smoothing to filter out the optical measurement noise. Next, we obtain binary images for feature extraction via Canny edge operator [29]. Due to the high contrast between the small dusts and the background, they are also considered as edges in the binary images. However, small dusts aren’t considered as unqualified defects in quality control. Thus, deeper image processing is necessary to deal with this issue. The final step is the contour sketch and area approximation [30] to connect the adjacent pixels in the image. Instead of disposing discrete pixels, it is more effective to detect defects according to the connected contour. Based on the approximated area

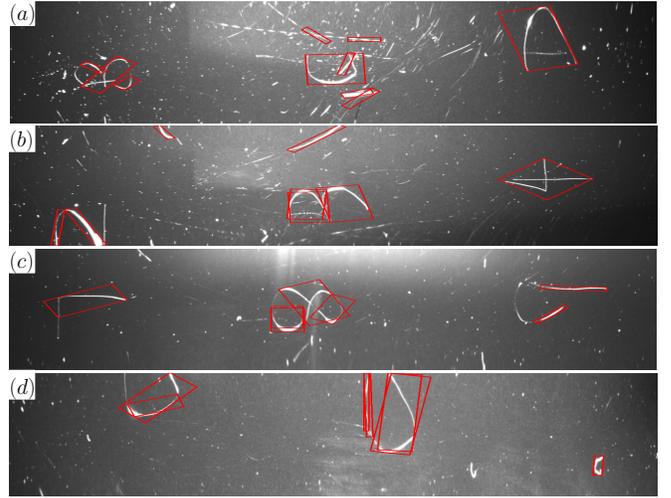


Fig. 12: (a)-(d) The captured images of the line scan camera and corresponding detection results.

TABLE II: Defect Detection Results

Part	TP	FP	FN	Precision	Recall	F-meas
$R_1$	20	3	0	0.870	1.000	0.930
$R_2$	9	3	0	0.750	1.000	0.857
$R_3$	11	1	2	0.917	0.846	0.880
$R_4$	12	4	3	0.750	0.800	0.774
$R_5$	10	5	0	0.667	1.000	0.800

TP is true positive, FP is false positive, and FN is false negative.

of contours, we tune a threshold  $s$  to judge if a contour is a defect or not, as shown in Fig. 11(d).

Fig. 12(a)-(d) illustrate other results with this image processing algorithm. The red squares represent the detected defects in the images. We use precision  $\rho$ , recall  $\nu$ , and F-measure  $\iota$  to evaluate the detection performance of this algorithm.

$$\rho = TP/(TP + FP), \quad \nu = TP/(TP + FN) \quad (12)$$

$$\iota = 2 \cdot (\rho \cdot \nu) / (\rho + \nu)$$

where  $TP$ ,  $FP$  and  $FN$  denote true positive, false positive and false negative, respectively. The accuracy of the algorithm with respect to different regions is shown in Table II. The results reveal that our system performs well in defect detection for the specular surface.

## V. CONCLUSIONS

In this paper, we demonstrate a flexible automatic defect inspection system with the help of a robotic manipulator and an image acquisition sub-system. We propose a robust region segmentation algorithm to meet the smoothness requirement of the camera. We also design a projection registration method to localize the object in robot’s frame. Experiments show the superiority of the system.

However, there are some limitations in our system. For one thing, deeper analysis about the sensing model of the image acquisition model is useful to ensure the inspection completeness. For another, the scanning path should be optimized to improve the efficiency.

## REFERENCES

- [1] C. Steger, M. Ulrich, and C. Wiedemann, *Machine vision algorithms and applications*. John Wiley & Sons, 2018.
- [2] D. Girardeau-Montaut, "Cloud compare—3d point cloud and mesh processing software," *Open Source Project*, 2015.
- [3] A. Tandiya, S. Akthar, M. Moussa, and C. Tarray, "Automotive semi-specular surface defect detection system," in *2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 285–291, IEEE, 2018.
- [4] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 50, 2014.
- [5] C.-S. Cho, B.-M. Chung, and M.-J. Park, "Development of real-time vision-based fabric inspection system," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 4, pp. 1073–1079, 2005.
- [6] C. Fernandez, C. Platero, P. Campoy, and R. Aracil, "Vision system for on-line surface inspection in aluminum casting process," in *Proceedings of IECON'93-19th Annual Conference of IEEE Industrial Electronics*, pp. 1854–1859, IEEE, 1993.
- [7] W. Fan, C. Lu, and K. Tsujino, "An automatic machine vision method for the flaw detection on car's body," in *2015 IEEE 7th International Conference on Awareness Science and Technology (iCAST)*, pp. 13–18, IEEE, 2015.
- [8] R. Hu, "Design and implementation of a defect inspection system for smooth product," Master's thesis, University of Electronic Science and Technology of China, 2018.
- [9] P. Zhang, T. J. Arre, and A. Ide-Ektessabi, "A line scan camera-based structure from motion for high-resolution 3d reconstruction," *Journal of Cultural Heritage*, vol. 16, no. 5, pp. 656–663, 2015.
- [10] H. Freeman, *Machine vision for three-dimensional scenes*. Elsevier, 2012.
- [11] Z. Les and M. Les, "Shape understanding system: understanding a convex object," *Journal of Electronic Imaging*, vol. 12, no. 2, pp. 327–342, 2003.
- [12] Y. Onodera, "High-resolution imaging system for scientific recording and digital archiving: Applications to on-the-site scanning of large-sized two-dimensional cultural heritage objects," *master degree thesis of Kyoto University*, 2013.
- [13] D. T. Pham and R. J. Alcock, *Smart inspection systems: Techniques and applications of intelligent vision*. Elsevier, 2002.
- [14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.
- [15] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, W. Lin, and P. Li, "Adaptive image-based positioning of RCM mechanisms using angle and distance features," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5403–5409, 2015.
- [16] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 339, 2017.
- [17] R. Tryon, "Cluster analysis; correlation profiles and orthometric (factor) analysis for the isolation of unities in mind and personality," *Ann Arbor: Edwards*, p. 122, 1939.
- [18] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.
- [19] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*, pp. 1–4, IEEE, 2011.
- [20] Z. Wang, X. Li, D. Navarro-Alarcon, and Y. Liu, "A unified controller for region-reaching and deforming of soft objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 472–478, 2018.
- [21] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.
- [22] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for human-robot collaboration: Fundamentals, challenges and opportunities," *Frontiers in Neurobotics*, vol. 14, p. 113, 2021.
- [23] "Ar tag tracking library for ros." [https://github.com/ros-perception/ar\\_track\\_alvar](https://github.com/ros-perception/ar_track_alvar).
- [24] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, IEEE, 2001.
- [25] D. Navarro-Alarcon, J. Qi, J. Zhu, and A. Cherubini, "A Lyapunov-stable adaptive method to approximate sensorimotor models for sensor-based control," *Frontiers in Neurobotics*, vol. 14, no. 59, pp. 1–12, 2020.
- [26] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Energy shaping methods for asymptotic force regulation of compliant mechanical systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2376–2383, 2014.
- [27] A. Baker, *Matrix groups: An introduction to Lie group theory*. Springer Science & Business Media, 2012.
- [28] S. Giancola, M. Valenti, and R. Sala, *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer, 2018.
- [29] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [30] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.