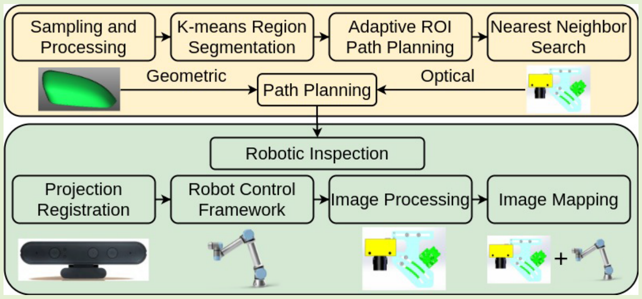
A Sensor-Based Robotic Line Scan System With Adaptive ROI for Inspection of Defects Over Convex Free-Form Specular Surfaces

Shengzeng Huo, *Graduate Student Member, IEEE*, Bin Zhang, Muhammad Muddassir,
David T. W. Chik, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—In this paper, we present a novel sensor-based system to perform defect inspection tasks automatically over free-form specular surfaces. The inspection procedure is performed by a robotic manipulator equipped with a line scanner system. Taking the geometric and optical parameters into consideration, our algorithm computes a flexible scanning path. Based on a mesh model, the system segments the convex surface into areas with similar curvatures, then adaptively adjusts the line sensor's scanning range to ensure the complete coverage of the surface. We consider the scanning efficiency through constrained optimization. To align the computed scanning path with the specular object, a real-time workpiece localization algorithm in a coarse-to-fine manner is developed. Our control framework synchronizes the motion of the manipulator and the acquisition of the line scanner for robotic inspection. We also provide an image processing pipeline for automatic defect detection. To validate the proposed formal methodology, we report a detailed experimental study with the developed robotic inspection prototype.**



***Index Terms*—Line scan sensor, depth sensor, defect inspection, path planning, sensor-guided robots, specular surfaces.**

I. Introduction

SURFACE defect inspection of products is essential for quality control (QC) in the modern manufacturing industry as defects and microcracks compromise the quality and value of the manufacturing goods. Conventionally, the modern industry relies on manpower to perform the inspection task, which is costly, time-consuming, and vulnerable to subjective results.

Although automatic defect inspection system has seen great progress in recent years (e.g. [1]–[4]), its geometric adaptability and robustness in various inspection tasks remain challenging. This is particularly critical for the inspection of free-form specular surfaces [5], as they require flexible scanning configurations (which contrasts with the *static* setups used for flat surfaces), and specialized imaging systems (to cope with the high reflection of shiny materials). Since free-form specular objects are common in industrial manufacturing, this paper aims to develop a feasible automatic solution to this problem.

Area-of-interest (AOI) based systems have been widely used in real-time defect inspection of (typically) flat surfaces due to their simplicity (in hot-rolled steel sheets [3] and fabrics [4]). Specialized setups are necessary when applying AOI-based systems to non-flat objects, including sensors, illumination, fixtures, etc. For example, to properly inspect the curved surface of a metal sphere, the development of a dual-light system is required [6]. However, these AOI-based systems usually have *fixed* scanning configurations, which limits their usage to a few types of object shapes.

In addition to geometry, the reflectivity of the surface has also been studied by researchers. [7] developed a method that visually analyzed the reflection of a known projection pattern over the surface, where both the pattern generator

Manuscript received October 8, 2021; revised November 23, 2021; accepted November 26, 2021. Date of publication December 3, 2021; date of current version January 31, 2022. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B090928001, in part by the Hong Kong Applied Science and Technology Research Institute (ASTRI) under Grant PO2019/399, in part by the Research Grants Council under Grant 14203917 and Grant F-PolyU503/18, and in part by the Jiangsu Industrial Technology Research Institute Collaborative Research Program Scheme under Grant ZG9V. The associate editor coordinating the review of this article and approving it for publication was Dr. Abhishek K. Jha. *(Corresponding author: David Navarro-Alarcon.)*

Shengzeng Huo, Bin Zhang, Muhammad Muddassir, and David Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: dna@ieee.org).

David. T. W. Chik is with the Hong Kong Applied Science and Technology Research Institute (ASTRI), Hong Kong.

This article has supplementary downloadable material available at https://doi.org/10.1109/JSEN.2021.3132428, provided by the authors.

Digital Object Identifier 10.1109/JSEN.2021.3132428

and the camera were manipulated by a robot. Based on sensor fusion, [8] addressed the automatic defect detection on car body surfaces during a painting process, where only flat regions with smooth changes in slope, concavities, edges, and corners were considered. Despite of recent developments, the automatic inspection of free-form specular surfaces still remains an open research problem.

3D surface imaging has a great potential in industrial inspection [9]. Babu *et al.* [10] and Glorieux *et al.* [11] equipped the manipulator with vision sensors to conduct shape measurement tasks on surfaces of various shapes. In this paper, we adopt similar configuration to perform defect inspection on convex [12] specular free-form surfaces.[1] Specifically, we equip a 6-DOF robot manipulator with a line scanner to achieve a flexible inspection path and utilize a depth sensor to localize the workpiece. Our sensor-based robotic system [14] consists of two subsystems, path planning and robotic inspection. The former one takes geometric (mesh model) and optical (vision sensor) parameters as inputs and computes the scanning path, while the latter one adjusts the path for the workpiece to conduct image acquisition and image processing. To summarize, the main contributions of this work are three-fold: (1) We propose a robust path planning algorithm for various specular free-form surfaces through adaptive region-of-interest (ROI) and K-means-based region segmentation. (2) We develop an autonomous defect inspection paradigm through workpiece real-time localization and sensor-based control framework. (3) We present a detailed experimental study to reveal the accuracy and robustness of our system compared with the baselines.

The rest of this paper is organized as follows: Sec. II describes the path planning algorithm, Sec. III presents the robotic inspection method, Sec. IV presents the results and Sec. V gives conclusions. The notation used in this paper is listed in Table I.

II. Path Planning

Converting the mesh model to a point cloud, this path planning algorithm adapts the ROI of the sensor to match the workpieces of various shapes, ensuring inspection completeness. The efficiency of the scanning is improved through the constrained optimization.

A. Sampling and Preprocessing

TABLE I
NOTATIONS OF THE PARAMETERS USED IN THIS STUDY

Parameter	Descriptions
$\vec{\mathbf{x}}_i^O, \vec{\mathbf{n}}_i^O, O$	i-th point and its normal in point cloud O
$\vec{\mathbf{c}}^O, \zeta_O$	the center and size of O
$\vec{\mathbf{q}}_i^O, \epsilon_i$	a specified vector of O and its exterior instance
N, n_j	the number of regions and patches
$\vec{\mathbf{s}}_j, S_j$	j-th feature vector $\vec{\mathbf{s}}_j$ of region S_j
γ_{ij}, Γ_j	the angle between $\vec{\mathbf{n}}_i^O$ and $\vec{\mathbf{s}}_j$ and the maximum
λ_j, Λ_t	termination instance of S_j and the sum at time t
β_K, β_C	angle deviation threshold of K-means and camera
θ, θ^*	rotation angle around $\vec{\mathbf{z}}$ and optimal solution
S^X, S^Y	X and Y coordinate sets of a region
$\vec{\mathbf{V}}_F$,$\vec{\mathbf{V}}_D$,$\vec{\mathbf{V}}_L$	field of view, depth of view and moving vector
$\vec{\mathbf{r}}, \delta$	projection basis and the corresponding scalar
Ω_j	the number of region division along $\vec{\mathbf{V}}_F$
$H^Y, \vec{\mathbf{h}}_{jk}, H_{jk}$	Y coordinate set and average normal of patch H_{jk}
$\vec{\boldsymbol{\psi}}$	the inspection point of the sensor on the surface
$\vec{\boldsymbol{\rho}}, \vec{\mathbf{k}}, \vec{\mathbf{i}}, \mathcal{F}_P$	position, Z-axis and X-axis of a frame
$\vec{\mathbf{p}}_j, \vec{\mathbf{p}}'_j$	the scanning poses on the edge with respect to S_j
ϕ_+, ϕ_-	rotation cost to the next X-axis $\pm\vec{\mathbf{i}}'$
$\hat{O}, O$	estimated pose and actual pose of the workpiece
φ, r	angle and distance in polar coordinates
$\vec{\mathbf{l}}_\varphi, L_\varphi$	ejection line and its corresponding buffer set
d_l, d_c	distance to ejection line and the center $\vec{\mathbf{c}}_{proj}$
$\vec{\boldsymbol{\eta}}$	manipulation plane representation
$\vec{\boldsymbol{\rho}}^B, \vec{\mathbf{k}}^B, \vec{\mathbf{i}}^B$	position and orientation of a pose in the robot's frame
$\vec{\boldsymbol{\rho}}_i, \vec{\boldsymbol{\pi}}_i, \vec{\mathbf{\Pi}}_i$	i-th position, orientation and pose of the robot
q, l	resolution and acquisition line rate of the camera
v	moving speed of the robot
$\vec{\mathbf{e}}_H, \kappa$	position vector and size of a pixel in the image frame
$\vec{\mathbf{u}}_H, \vec{\mathbf{U}}_B$	the image pixel in local and global frame
A, w	angle of view and working distance of the camera
ξ, Δ	circle of confusion and diagonal of the sensor
$F,\ f$	f-number and focal length of the len
μ, σ^2	mean value and variance

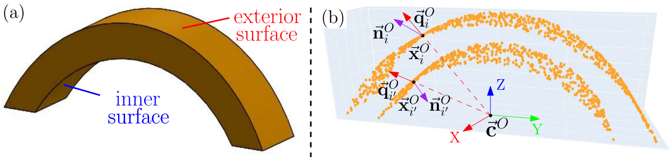


Fig. 1. 3D representation information of the workpiece. (a) Mesh model. (b) Sampled point cloud and the corresponding preprocessing procedure.

We assume that the mesh model of the workpiece is prior knowledge and the shape error between the model and the workpiece is ignorable. This assumption is reasonable due to the development of computer-aided design technology and the current manufacturing level [15]. Since mesh models consist of many vertices and surfaces (which might be difficult to process), we sample points from the mesh model to facilitate path planning. Reference [16] provides a useful sampling solution by randomly selecting a given number of points on triangles. A typical convex mesh model and its corresponding sampled point cloud are shown in Fig. 1. Due to its sampling uniformity and complete coverage, the obtained sampled point cloud O consisting of points $O = (\vec{\mathbf{x}}_1^O, \vec{\mathbf{x}}_2^O, \ldots)$ preserves the geometric information of all faces. However, only the exterior surface of the workpiece is taken into consideration in sensor-based defect inspection. Therefore, it is necessary to filter out unrelated points, as illustrated in Fig. 1(b). The center $\vec{\mathbf{c}}^O$ of the point cloud O is computed as

$$\vec{\mathbf{c}}^O = \sum_{i=0}^{\zeta_O} \vec{\mathbf{x}}_i^O / \zeta_O. \tag{1}$$

[1] A preliminary version of this work was presented in [13]. In this paper, we further extend the previous work in six aspects: (1) We utilize the characteristics of the mesh to distinguish the interior surface and the exterior surface; (2) We develop a geometric model to analyze the acquisition of the line scanner; (3) We guarantee the inspection completeness through adaptive path planning based on our proposed acquisition model; (4) We improve the efficiency of the scanning through constrained optimization; (5) We utilize the relative motion between the sensor and the object to reconstruct the defects on the CAD model; (6) We provide detailed experimental results to highlight the superiority of our system compared with the baselines on various aspects, including region segmentation, registration, image acquisition, and image processing.

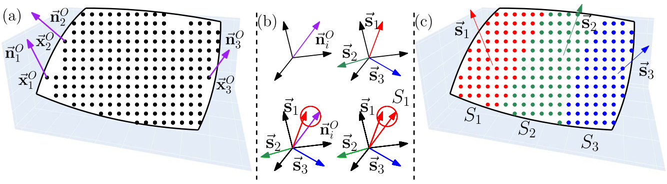


Fig. 2. Conceptual representation of K-means-based region segmentation. (a) Input information: i-th point $\vec{\mathbf{x}}_i^O$ and its corresponding normal $\vec{\mathbf{n}}_i^O$. (b) Clustering for each sample $\vec{\mathbf{n}}_i^O$. (c) Recompute the feature vector $\vec{\mathbf{s}}_j$ of each region S_j.

where ζ_O is the size of the point cloud O. Next, we utilize the normal vectors $(\vec{\mathbf{n}}_1^O, \vec{\mathbf{n}}_2^O, \ldots)$ generated from the mesh model to search for points on the exterior surface. For that, we firstly specify a vector $\vec{\mathbf{q}}_i^O = \vec{\mathbf{x}}_i^O - \vec{\mathbf{c}}^O$, then compare its direction with the corresponding surface normal $\vec{\mathbf{n}}_i^O$:

$$\epsilon_i(\vec{\mathbf{q}}_i^O \cdot \vec{\mathbf{n}}_i^O) = \begin{cases} 0, & \vec{\mathbf{q}}_i^O \cdot \vec{\mathbf{n}}_i^O < 0 \\ 1, & \text{else} \end{cases} \tag{2}$$

The data point $\vec{\mathbf{x}}_i^O$ is saved only when $\epsilon_i = 1$, which means $\vec{\mathbf{q}}_i^O$ and $\vec{\mathbf{n}}_i^O$ are in the same direction. This method effectively extracts points on the exterior surface of *convex* objects.

B. K-Means Region Segmentation

The surface of the considered free-form object has a smoothly varying curvature. To inspect defects with vision sensors, we divide the surface of the object into approximately "flat" sub-regions, which ensures that defects can be highlighted and avoids unwanted reflections caused by the light source. Then, region segmentation is formulated as a clustering analysis problem of surface geometric features (namely, position and normal). In our method, we use K-means clustering [17] to partition the point cloud O.

Our algorithm takes the points of O and their corresponding normals as input, as shown in Fig. 2(a). In standard K-means algorithms, the number of clusters N greatly affects the quality of the classification; It typically requires several trials to find an optimal N by traditional methods [18]. In this paper, we propose a new method to automatically tune N. The algorithm is based on a two-looped 1D search, with the inner loop for classification and the outer loop for updating N. The termination condition is defined as the maximum intra-class difference smaller than a threshold β_K. The entire procedure is given in Algorithm 1 and illustrated in Fig. 2.

For the outer loop, we denote the feature vectors of the N-cluster set by $[\vec{\mathbf{s}}_1, \vec{\mathbf{s}}_2, \cdots, \vec{\mathbf{s}}_N]$ and initialize them with random values. Afterward, the procedure goes into the inner loop, which is composed of two steps: 1) classification and 2) update. In classification, for each $\vec{\mathbf{n}}_i^O$, we calculate its similarities γ_{ij} with every feature vector $\vec{\mathbf{s}}_j$ as follows:

$$\gamma_{ij} = \arccos(\vec{\mathbf{n}}_i^O \cdot \vec{\mathbf{s}}_j / (\left|\vec{\mathbf{n}}_i^O\right| \cdot \left|\vec{\mathbf{s}}_j\right|)). \tag{3}$$

Then, we find the cluster S_j that produces the smallest γ_{ij} and assign $\vec{\mathbf{n}}_i^O$ to it. After the classification step, each $\vec{\mathbf{n}}_i^O$ is assigned to its corresponding region S_j.

The next step is to determine whether the classification has met the termination condition. For each cluster S_j, the termination instance λ_j is computed from the maximum intra-class difference Γ_j as:

$$\lambda_j = \begin{cases} 0, & \Gamma_j > \beta_K \\ 1, & \text{else} \end{cases} \quad \text{for} \quad \Gamma_j = \max_i \gamma_{ij} \tag{4}$$

The sum Λ_t of the instances λ_j from every region S_j at this iteration t is $\Lambda_t = \sum_{j=1}^{N} \lambda_j$. If Λ_t equals the value of the current N, the current segmentation is acceptable and the algorithm reaches termination. Otherwise, the iteration continues. In this stage, we should consider the search direction since the algorithm contains two loops, the inner one that clusters concerning N and the outer one that increases the value of N. The adjustment depends on the performance of Λ_t. If the performance decreases at the iteration step t (i.e. $\Lambda_t < \Lambda_{t-1}$), it indicates that the current N is not ideal and the inner loop must stop; a new outer loop starts with $N \leftarrow N + 1$. If the performance improves (i.e. $\Lambda_t \geq \Lambda_{t-1}$), the search within the inner loop continues.

Before switching to the next inner iteration, all feature vectors $\vec{\mathbf{s}}_j$ are updated to improve the representation level:

$$\vec{\mathbf{s}}_j = \frac{1}{\zeta_j} \sum_{i=1}^{\zeta_j} \vec{\mathbf{n}}_i^{S_j} / || \frac{1}{\zeta_j} \sum_{i=1}^{\zeta_j} \vec{\mathbf{n}}_i^{S_j} ||, \tag{5}$$

where $\vec{\mathbf{n}}_i^{S_j}$ and ζ_j are i-th normal and the size of region S_j respectively.

The proposed algorithm only takes normal features of the region S_j into consideration, which can lead to a high sparsity of the clustered points within the same region. Therefore, Euclidean cluster extraction [19] is implemented as a post-processing step to verify if it is necessary to subdivide the region S_j into two new regions according to the location of the points in it.

Algorithm 1 K-Means Region Segmentation

```
Input: $\beta_K$, $O$
Output: $S_j$, $j = 1, 2, \ldots, N$
while $\Lambda_t < N$ do
    Initialize $N$ normal vectors $\vec{\mathbf{s}}_j$ randomly, $j = 1 : N$
    while $\Lambda_t \geq \Lambda_{t-1}$ do
        for $(\vec{\mathbf{x}}_i^O, \vec{\mathbf{n}}_i^O)$, $i = 1 : \zeta_O$ do
            Compute angle $\gamma_{ij} \leftarrow (3)$
            $j^* \leftarrow \arg\min_j \gamma_{ij}$, push $\vec{\mathbf{x}}_i^O \rightarrow S_{j^*}$
        Compute $\Lambda_t \leftarrow (4)$
        Update $\vec{\mathbf{s}}_j$, $j = 1 : N \leftarrow (5)$
    $N \leftarrow N + 1$
```

C. Adaptive ROI Based Path Planning

The proposed planning algorithm takes the segmented region S_j as input and outputs the optimized scanning path. Since the scanning of the line sensor needs to synchronize with the robot's motion, the main idea of our method is to utilize the shortest path to cover the entire region S_j. However, the

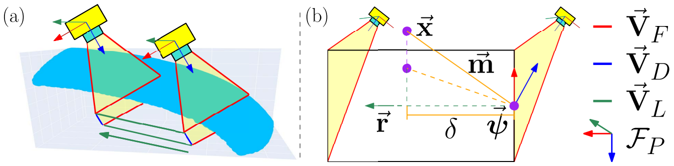


Fig. 3. Geometric acquisition model of the line scan camera with linear motion. (a) Cuboid coverage generation process. (b) Mathematical model of the coverage cuboid and completeness validation.

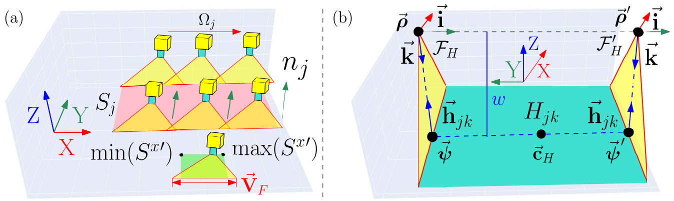


Fig. 4. Further region segmentation and linear path planning. (a) Patch segmentation along horizontal and vertical directions. (b) Linear path planning with respect to patch H_{jk}.

irregular shapes and scattered distribution make path planning very challenging. Hence, we establish a local coordinate frame $\mathcal{F}_S$ concerning each region S_j. The transformation between the local frame $\mathcal{F}_S$ and the object frame $\mathcal{F}_O$ consists of three parts $\mathbf{T}_Z, \mathbf{T}_{xy}, \mathbf{T}_Y$, the detailed explanation is:

1) $\mathbf{T}_Z$: Rotate $\vec{\mathbf{s}}_j$ to Z-axis $\vec{\mathbf{z}} = [0\ 0\ 1]^T$.
2) $\mathbf{T}_{xy}$: Translate to X-Y plane $Z = 0$.
3) $\mathbf{T}_Y$: Rotate angle θ^* around $\vec{\mathbf{z}}$ to minimize the projection length along Y-axis $\vec{\mathbf{y}} = [0\ 1\ 0]^T$. The angle θ^* is obtained by iteration search

$$S^{Y\prime} = S^X \cdot \sin\theta + S^Y \cdot \cos\theta. \tag{6}$$

$$\theta^* = \arg\min_{\theta}(\max S^{Y\prime} - \min S^{Y\prime}). \tag{7}$$

where S^X and S^Y are X and Y coordinate sets of region S_j respectively.

Our adaptive ROI method aims to adjust the scanning coverage of the vision sensor to fit the various shapes of S_j. As a 1D vision sensor, the geometry acquisition model of the line scan camera is depicted in Fig. 3(a). Capturing a single horizontal line per frame, the scanning coverage of the sensor is modeled as a rectangle at each moment, consisting of the depth of view $\vec{\mathbf{V}}_D$ and the field of view $\vec{\mathbf{V}}_F$. Combining with the linear motion of the sensor, a series of rectangles are captured and then integrated as a cuboid along the moving direction $\vec{\mathbf{V}}_L$. Hence, we model the scanning capacity of the sensor during a motion as the volume of the generated cuboid.

With the geometric scanning model, we hold a definition that the inspection completeness is achieved if all points are within the cuboid. The mathematical model is illustrated in Fig. 3(b). Let's suppose the linear scanning starts from a pose p, depicted as a frame $\mathcal{F}_P$, whose position, Z-axis and X-axis are $\vec{\rho}, \vec{\mathbf{k}}, \vec{\mathbf{i}}$ respectively. Thus, the inspection point of the sensor on the surface is $\vec{\psi} = \vec{\rho} - w \cdot \vec{\mathbf{k}}$, where w denotes the working distance between the camera and the surface. For a point $\vec{\mathbf{x}}$ of the surface, we specify a vector $\vec{\mathbf{m}} = \vec{\mathbf{x}} - \vec{\psi}$. A point $\vec{\mathbf{x}}$ being within the cuboid should meet the conditions that the projection scalar of its corresponding vector $\vec{\mathbf{m}}$ to the bases $\vec{\mathbf{V}}_D, \vec{\mathbf{V}}_F, \vec{\mathbf{V}}_L$ should be smaller than their magnitudes $|\vec{\mathbf{V}}_D|, |\vec{\mathbf{V}}_F|, |\vec{\mathbf{V}}_L|$ at the same time. The projection scalar can be computed by:

$$\delta = \vec{\mathbf{m}} \cdot \vec{\mathbf{r}} / |\vec{\mathbf{r}}| \tag{8}$$

Here, $\vec{\mathbf{r}}$ is the projection basis, which are $\vec{\mathbf{V}}_D, \vec{\mathbf{V}}_F, \vec{\mathbf{V}}_L$ in the frame $\mathcal{F}_P$ respectively.

In addition, *normal similarity* β_C, here defined as the maximum allowable angle difference between the surface normal and the camera's facing direction is also a necessary condition to guarantee the performance of the vision sensor.

To ensure the completeness of the scan based on the geometric sensing model, our algorithm further segments each region S_j into patch H_{jk}. This 1D segmentation has two aspects, horizontal and vertical directions, as shown in Fig. 4(a). For the horizontal direction, we evaluate the coverage capability over the width of the region S_j by:

$$\Omega_j = \lceil(\max(S^{x\prime}) - \min(S^{x\prime}))/\left|\vec{\mathbf{V}}_F\right|\rceil, \tag{9}$$

where $\lceil * \rceil$ is the ceiling function; $S^{x\prime} = S^X \cdot \cos\theta^* - S^Y \cdot \sin\theta^*$ is the x coordinate set of region S_j in the local frame $\mathcal{F}_S$. When Ω_j equals to one, the segmentation along $\vec{\mathbf{V}}_F$ is not required. Otherwise, we divide the region S_j into Ω_j subregions (thus the number of regions N increases). The next step is to divide each region S_j into several patches H_{jk} along the vertical direction. Similar to Algorithm 1, we segment the region S_j into n_j patches and iteratively increase n_j until meeting the termination conditions. The conditions depend on the inspection completeness, including the geometric cuboid coverage model and normal similarity, which are validated by (8) and (4), respectively.

Next, we plan a linear scanning path with the same orientation for each patch H_{jk}, see Fig. 4(b). Here, we utilize two frames $\mathcal{F}_H, \mathcal{F}_H^{\prime}$ to describe two poses of the motion. Their origin $\vec{\rho}, \vec{\rho}^{\,\prime}$ locate on:

$$\vec{\psi} = [c_x\ \ \min(H^Y)\ c_z]^T \quad \vec{\psi}^{\,\prime} = [c_x\ \ \max(H^Y)\ c_z]^T \tag{10}$$

$$\vec{\rho} = \vec{\psi} + w \cdot \vec{\mathbf{h}}_{jk}, \quad \vec{\rho}^{\,\prime} = \vec{\psi}^{\,\prime} + w \cdot \vec{\mathbf{h}}_{jk} \tag{11}$$

where $\vec{\psi}$ and $\vec{\psi}^{\,\prime}$ are two inspection points of the sensor on patch H_{jk}; c_x and c_z are elements of $\vec{\mathbf{c}}_H = [c_x\ c_y\ c_z]^T$ computed by Eq. 1 with respect to patch H_{jk}. For orientation, since $\vec{\mathbf{k}}$ is defined as $\vec{\mathbf{k}} = -\vec{\mathbf{h}}_{jk}$, the X-axis $\vec{\mathbf{i}}$ of the frame $\mathcal{F}_H$ is:

$$\vec{\mathbf{k}} = \mathbf{R}_Z \cdot \vec{\mathbf{z}}, \quad \vec{\mathbf{i}} = \mathbf{R}_Z \cdot \vec{\mathbf{x}} \tag{12}$$

where $\vec{\mathbf{x}} = [1\ 0\ 0]^T$, the rotation matrix $\mathbf{R}_Z$ is solved with an axis-angle representation.

Finally, we need a transformation between the local frame $\mathcal{F}_S$ to the original object frame $\mathcal{F}_O$ to obtain the entire path pose p:

$$p = \left\{\mathbf{T}_P \cdot \vec{\rho}, \mathbf{T}_R \cdot \vec{\mathbf{k}}, \mathbf{T}_R \cdot \vec{\mathbf{i}}\right\} = \left\{\vec{\rho}^{\,O}, \vec{\mathbf{k}}^O, \vec{\mathbf{i}}^O\right\} \tag{13}$$

where $\mathbf{T}_P = \mathbf{T}_Z^{-1}\mathbf{T}_{xy}^{-1}\mathbf{T}_Y^{-1}$ and $\mathbf{T}_R = \mathbf{T}_Z^{-1}\mathbf{T}_Y^{-1}$. Here, $\vec{\mathbf{k}}^O, \vec{\mathbf{i}}^O, \vec{\rho}^{\,O}$ represent vectors in the object's frame $\mathcal{F}_O$. Note

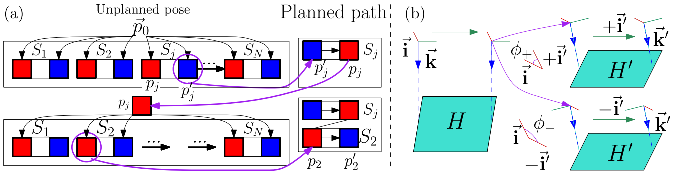


Fig. 5. Complete path optimization with respect to entire free-form surface. (a) Scanning sequence arrangement based on distance-optimal criteria. (b) Rotation angle minimization in the process of adjacent scanning pose switching.

that $\vec{\boldsymbol{\rho}}$, $\vec{\mathbf{k}}$ and $\vec{\mathbf{i}}$ are represented as 4×1 vectors in homogeneous coordinate for the transformation here.

D. Nearest Neighbor Search

Now, we need to connect all the path sections to generate the complete scanning path for the entire free-form object. To improve the inspection efficiency, the shortest path criteria are adopted, which are equivalent to minimizing the scanning time since we keep a constant velocity of the camera during the scanning task. Thus, this issue is formulated as the following constrained optimization problem:

$$\min \sum_{i=1}^{N_P-1} C(p_i, p_{i+1}) \tag{14}$$

where $C(p_i, p_{i+1})$ describe the cost switching from $\vec{\mathbf{p}}_i$ to $\vec{\mathbf{p}}_{i+1}$, including the position cost and the orientation cost. $N_P = 2 \times \sum_{j=1}^{N} n_j$ is the total number of the path poses. Considering the constraints brought by the segmentation and adaptive ROI path planning, we convert the optimization to a sorting problem and solve it by the Nearest Neighbor Search [20], which are discussed with respect to the position and orientation separately in the following.

For position, only the switch between regions can be optimized due to the fixed motion distance within each region. For each region S_j, we extract the path poses (p_j, p'_j) on the edges and obtain $2N$ poses in total. We formulate the position optimization as the coverage path planning (CPP) [21] to find the shortest path visiting every node in the graph, while the difference of our case is that the moving process is within a region rather than a node, namely the start pose and the end pose of the node are different, shown in Fig. 5(a). At first, all regions S_j locate in the unplanned region box and the planned path is empty. The initial search benchmark p_0 is the current pose of the robot manipulator. During a loop, we compute the distances between p_0 and $2N$ poses $(p_1, p'_1, \cdots, p_N, p'_N)$. Let's suppose that p'_j is nearest to the benchmark $\vec{\mathbf{p}}_0$ and we assign region S_j as the next neighbor and send it to the planned path box. After one iteration, the search benchmark switches from p_0 to p_j. At the same time, the corresponding region S_j is popped out from the unplanned pose set. Then, the search loop continues until the unplanned pose box is empty.

Another optimization section focus on the orientation, as shown in Fig. 5(b). To illustrate the idea, we define a switch between patches $H(\vec{\mathbf{k}}, \vec{\mathbf{i}}) \to H'(\vec{\mathbf{k}}', \vec{\mathbf{i}}')$. According to the characteristics of the line sensor, $\vec{\mathbf{k}}$ is fiexed while $\vec{\mathbf{i}}$ is adjustable, namely $\pm\vec{\mathbf{i}}'$ are both suitable. However, switching from $\vec{\mathbf{i}}$ to $+\vec{\mathbf{i}}'$ or $-\vec{\mathbf{i}}'$ have different costs ϕ_+, ϕ_- computed by Eq. 3. For the optimization, we should select the one with the smaller cost by

$$\begin{cases} \vec{\mathbf{i}}' \leftarrow +\vec{\mathbf{i}}' & \phi_+ \leq \phi_- \\ \vec{\mathbf{i}}' \leftarrow -\vec{\mathbf{i}}' & else \end{cases} \tag{15}$$

With this rule, we can optimize the orientation of the next patch H' according to the current patch H.

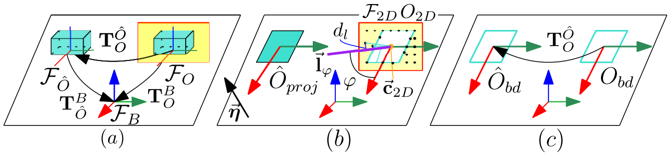


Fig. 6. Projection registration for specular surface: (a) Problem formulation. (b) Boundary extraction. (c) Final registration.

III. Robotic Inspection

After planning the scanning path on the mesh's frame, we align it to the object's frame through the workpiece's real-time localization. Then, the sensor-robot control framework [22] implements the inspection. Image processing and mapping are also provided for defect detection and visualization.

A. Sensor-Based Projection Registration

To deliver the active motion of the line sensor, a transformation $\mathbf{T}_O^B$ from the object's frame $\mathcal{F}_O$ to the robot's frame $\mathcal{F}_B$ is required, namely the workpiece localization in the robot's frame $\mathcal{F}_B$. Recently, depth sensors have been widely used for 3D vision perception [23]. However, their performance is severely affected by the high reflection of specular surfaces [24]. To deal with this issue, we design a novel projection registration algorithm in a coarse-to-fine manner as visualized in Fig. 6(a). We emphasize that the desired solution of this algorithm is the transformation $\mathbf{T}_O^B$ between the object $\mathcal{F}_O$ and the robot $\mathcal{F}_B$. Firstly, we calibrate the robot and the depth sensor via AR markers, denoted as $\mathbf{T}_C^B$. Note that the robot bridges the transformation between the line scanner and the depth sensor since the line scanner in the robot's frame $\mathcal{F}_B$ can be achieved via forwarding kinematics. $\mathcal{F}_{\hat{O}}$ denotes a coarse estimation of $\mathcal{F}_O$, whose transformation $\mathbf{T}_{\hat{O}}^B$ is a prior according to the configuration of the workspace. Mounted on the top of the workspace, the depth sensor captures the pose of the object O. As illustrated in Fig 6(b), the captured point cloud O loses the vision information of the specular surface due to its high reflection. However, the boundary of the blocked area can be used to describe the outer boundary of the object. The extraction of this boundary O_{bd} includes several steps. Firstly, we use pass-through filters to narrow down the ROI of O and transform the point cloud to the robot's frame with $\mathbf{T}_C^B$, denoted as O_{2D}. In the end, we establish a 2D local polar coordinate system $\mathcal{F}_{2D}$ in the manipulation plane whose origin is $\vec{\mathbf{c}}_{2D}$ of O_{2D}:

$$\vec{\mathbf{l}}_\varphi = [r\cos\varphi \quad -r\sin\varphi]^T \tag{16}$$

where r and φ represent the distance to $\vec{\mathbf{c}}_{2D}$ and the angle to the X-axis of $\mathcal{F}_{2D}$ respectively. The conversion to Cartesian coordinates with respect to φ is $\tan\varphi \cdot x + y = 0$. For the points of the boundary, we define them as the nearest points to the origin along different directions. Due to the discreteness of the point cloud, we use its distance d_l to the line to find out if a point is on a line

$$d_l = \left|\vec{\mathbf{x}}^{O_{2D}} \cdot \omega_\varphi\right|, \quad \omega_\varphi = [\sin\varphi \;\; \cos\varphi]^T \tag{17}$$

For each ejection line $\vec{\mathbf{l}}_\varphi$, the points whose distance d_l to $\vec{\mathbf{l}}_\varphi$ is smaller than a threshold are saved in a buffer set L_φ. Within the set, the point whose distance d_c to $\vec{\mathbf{c}}_{2D}$ is smallest is considered as a component of the boundary O_{bd}.

In addition, we also need to find out the boundary $\hat{O}_{bd}$ of the estimation pose $\hat{O}$. Firstly, we project $\hat{O}$ to the given plane $\vec{\boldsymbol{\eta}}$ to get $\hat{O}_{proj}$ by

$$\hat{O}_{proj} = \hat{O} - (\vec{\boldsymbol{\eta}} \cdot \hat{O}) \cdot \vec{\boldsymbol{\eta}} \tag{18}$$

Next, we extract the outer boundary $\hat{O}_{bd}$ of $\hat{O}_{proj}$ via [25]. Finally, we utilize ICP algorithm [26] to align $\hat{O}_{bd}$ and O_{bd}, thus obtaining the transformation matrix $\mathbf{T}_O^{\hat{O}}$ between $\mathcal{F}_{\hat{O}}$ and $\mathcal{F}_O$. Hence, combining $\mathbf{T}_{\hat{O}}^B$ and $\mathbf{T}_O^{\hat{O}}$, we can obtain the entire transformation matrix $\mathbf{T}_O^B = \mathbf{T}_{\hat{O}}^B \cdot \mathbf{T}_O^{\hat{O}}$.

B. Robot Control Framework

Given the 4×4 transformation matrix $\mathbf{T}_O^B = [\mathbf{R}_O^B | \vec{\mathbf{t}}_O^B]$, consisting of 3×3 rotation matrix $\mathbf{R}_O^B$ and 3×1 translation vector $\vec{\mathbf{t}}_O^B$, we obtain the path of the active motion

$$p^B = \left\{\mathbf{T}_O^B \cdot \vec{\boldsymbol{\rho}}^{\,O}, \mathbf{R}_O^B \cdot \vec{\mathbf{k}}^O, \mathbf{R}_O^B \cdot \vec{\mathbf{i}}^O\right\} \tag{19}$$

Note that $\vec{\boldsymbol{\rho}}^{\,O}$ is represented as 4×1 vector in homogeneous coordinates for transformation.

To define the pose of the robot end-effector for each path point, the pose vector will be

$$\vec{\boldsymbol{\Pi}}_i = \left[\, \vec{\boldsymbol{\rho}}_i^{\,T} \;\; \vec{\boldsymbol{\pi}}_i^{\,T} \,\right]^T \tag{20}$$

where $\vec{\boldsymbol{\Pi}}_i$ is a 6×1 pose vector consisting of the position vector $\vec{\boldsymbol{\rho}}_i$ and the orientation vector $\vec{\boldsymbol{\pi}}_i$. For position, $\vec{\boldsymbol{\rho}}^{\,B} = \mathbf{T}_O^B \cdot \vec{\boldsymbol{\rho}}^{\,O}$ directly defines the desired position. For orientation, we need to solve the complete rotation $\mathbf{R}_\pi$ with respect to $\vec{\mathbf{k}}^B = \mathbf{R}_O^B \cdot \vec{\mathbf{k}}^O$ and $\vec{\mathbf{i}}^B = \mathbf{R}_O^B \cdot \vec{\mathbf{i}}^O$ with axis-angle form, similar to Eq. 12. Then, we obtain the rotation vector $\vec{\boldsymbol{\pi}}$ by Rodrigues' rotation formula [27] according to $\mathbf{R}_\pi$.

The line scan camera captures one line at a time, and stack the horizontal rows of pixels to generate a 2D image, see Fig. 7. The sampling rate of the camera needs to be synchronized with the robot's motion to ensure scanning completeness and avoid repetition. The synchronization is achieved by defining:

$$|\vec{\mathbf{V}}_F|/q = (1/l) \cdot v \tag{21}$$

where q is the resolution of the vision sensor, l is the acquisition line rate with units in seconds, and v is the linear velocity of the manipulator.

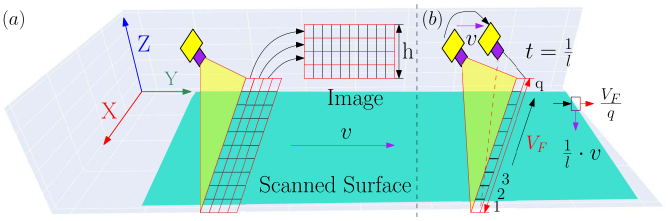


Fig. 7. Conceptual representation of automatic scanning procedure. (a) Principal of image acquisition during camera's motion. (b) Acquisition line rate adjustment for unit magnification.

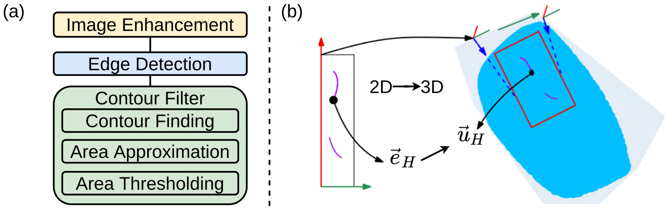


Fig. 8. Details about the automatic defect detection and visualization. (a) Complete pipeline of digital image processing for defect detection. (b) Conceptual explanation about the mathematical model of the image mapping.

C. Image Processing

Digital image processing is necessary for an automatic visual inspection system. Illustrated in Fig. 8(a), we present a pipeline of image processing for feature extraction, which is a step towards efficient defect detection.

Given the raw captured images, the enhancement is effective to improve the quality and filter out the noise. Edges displaying sudden changes in gray level play an important role in the field of image recognition. Based on the detected edges, we design a binary classifier according to the approximated area of the edges estimated by [28]. A detailed case study about various algorithms and thresholds concerning these aspects (enhancement, edge detection, and contour filter) is demonstrated in Sec. IV.

D. Image Mapping

In addition to the number of defects, the distribution of defects also provides an intuitive understanding of the quality of the products. Leveraging the model of the object, we obtain a 3D visualization result through mapping the detected defects in 2D images to the 3D model, as shown in Fig. 8(b).

As discussed above, a 2D image corresponds to a patch H_{jk} with a linear motion with fixed orientation. Thus, we assume that the distance between the image acquisition system and the inspection surface remains the same during the process. The capture of each image starts at pose $\vec{\boldsymbol{\Pi}}_H$ and stops at pose $\vec{\boldsymbol{\Pi}}'_H$. Hence, we use pose $\vec{\boldsymbol{\Pi}}_H$ as the mapping frame $\mathcal{F}_H$ to find out the 3D position $\vec{\mathbf{u}}_H$ corresponding to a pixel $\vec{\mathbf{e}}_H = (e_x, e_y)$ in the 2D image by

$$\vec{\mathbf{u}}_H = \left[\, (e_x - q/2) \cdot \kappa \;\; e_y \cdot \kappa \;\; w \,\right]^T \tag{22}$$

where $\kappa = \left|\vec{\mathbf{V}}_F\right|/q$ represents the pixel size in the image. Since $\vec{\mathbf{u}}_H$ is a 3×1 vector representing the position of

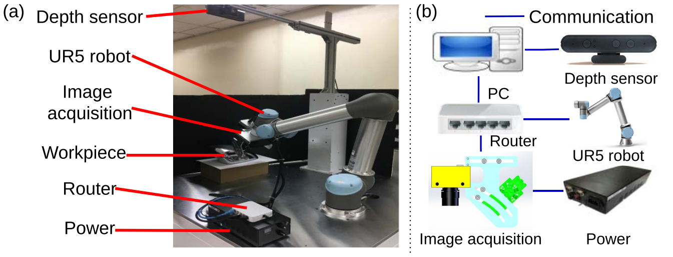


Fig. 9. The platform of our proposed autonomous robotic inspection system. (a) Experimental configuration. (b) System Architecture.

the defects in the local mapping frame $\mathcal{F}_H$, we transform it to the robot's frame $\mathcal{F}_B$ via $\vec{\mathbf{U}}_B = \mathbf{T}_H \cdot \vec{\mathbf{u}}_H$, where 4×4 transformation matrix $\mathbf{T}_H$ corresponds to pose $\vec{\mathbf{\Pi}}_H$. Note that $\vec{\mathbf{U}}_B, \vec{\mathbf{u}}_H$ are represented as 4×1 vectors in homogeneous coordinates for transformation.

IV. Results

A. Experiment Setup

Fig. 9(a) illustrates the proposed sensor-based robotic system and its components. A custom-made end-effector is developed for the robotic system, carrying the image acquisition module during the inspection. To manipulate the end-effector for the flexible scanning inspection over the workpiece, our system consists of a UR5 manipulator from Universal Robots. A depth sensor, Orbbec Astra, provides a top-down view of the workspace on the workpiece.

Fig. 9(b) conceptually depicts the architecture of the system. The image acquisition module consists of a line scan camera (Basler raL2048-48gm GigE camera) and a uniform line illumination source (LTS-2LINS300-W from LOTS company). An analog control box with a high-power strobe provides an adjustable voltage to the light source. The entire automatic inspection framework is implemented in ROS [29] on Linux PC, which is capable of real-time monitoring the sensors (line scanner, depth sensor) and controlling the actuator (manipulator) synchronously via the TCP/IP socket connection provided by the router.

With several manual trials, the performance of the image acquisition module decreases significantly when the normal similarity of β_C exceeds 20°. Thus, we set $\beta_C = 20^\circ$ during the entire inspection process. The velocity and acceleration of the manipulator are empirically set as $0.05m/s$ and $0.05m/s^2$ respectively due to safety reasons. According to [30], the optical parameters $(\vec{\mathbf{V}}_F, \vec{\mathbf{V}}_D)$ of the image acquisition module are computed by

$$\left|\vec{\mathbf{V}}_F\right| = 2w\tan(A/2), \quad \xi \approx \Delta/1500, \quad \left|\vec{\mathbf{V}}_D\right| \approx 2w^2F\xi/f^2 \tag{23}$$

where A, Δ, ξ, F, f denotes the angle of view, sensor diagonal, circle of confusion, f-number, and focal length of the sensor, respectively. We simplified that the manipulator keeps a constant speed during the trajectory execution, thus the corresponding acquisition line rate of the scanner is 670 $line/s$ according to Eq. 21.

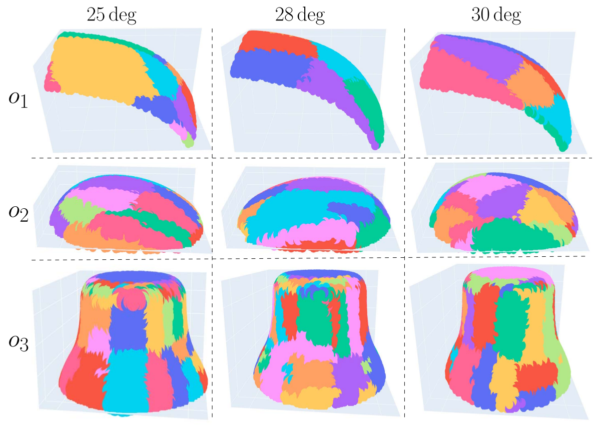


Fig. 10. Segmentation results with respect to different free-form surfaces and different termination thresholds.

B. Segmentation Performance

The diversity of object shapes is a great challenge for surface inspection path planning. Region segmentation is the key guidance for path planning in automatic surface defect inspection, where the desired goal is to assign a label to every element in the point cloud such that points sharing certain characteristics have the same label, which is similar to the image segmentation [33]. To intuitively show the robustness and accuracy of our region segmentation algorithm, we implement it on various objects (o_1, o_2, o_3) with different thresholds. The results are presented in Fig. 10, from which we conclude that our K-means-based unsupervised region segmentation is suitable for various convex free-form surfaces under different conditions. As the coarse segmentation for the following path planning, we select $\beta_C = 28^\circ$ since it minimizes the total number of patches N_P with a relatively uniform size.

We also evaluate our region segmentation method by implementing the existing baselines to the same objects shown in Fig. 10 and comparing the results with those of our method. We define the metrics for evaluation based on Otsu's algorithm [34]:

$$\begin{aligned}
&\mu_S = \frac{1}{N} \cdot \sum_{j=1}^{N} \zeta_j, \quad \sigma_S^2 = \frac{1}{N} \cdot \sum_{j=1}^{N} (\zeta_j - \mu_S)^2 \\
&\sigma_I^2 = \frac{1}{N} \cdot \sum_{j=1}^{N} \frac{1}{\zeta_j} \sum_{i=1}^{\zeta_j} \gamma_{ij} \\
&\vec{\boldsymbol{\mu}}_B = \frac{1}{N} \cdot \sum_{j=1}^{N} \vec{\mathbf{s}}_j, \quad \sigma_B^2 = \frac{1}{N} \cdot \sum_{j=1}^{N} \gamma\left(\vec{\mathbf{s}}_j, \vec{\boldsymbol{\mu}}_B\right)
\end{aligned} \tag{24}$$

where $\gamma\left(\vec{\mathbf{s}}_j, \vec{\boldsymbol{\mu}}_B\right)$ computes the angle between $\vec{\mathbf{s}}_j$ and $\vec{\boldsymbol{\mu}}_B$ according to Eq. 3. Specifically, the ideal segmentation solution should satisfy three conditions: (1) $\max_j \Gamma_j$ is adjustable according to the desired segmentation accuracy; (2) As the indicator of the regional distribution, N_S should be small with a uniform partition σ_S^2, which benefits the inspection efficiency. (3) Minimize the intra-class variance σ_I^2 and maximize the between-class variance σ_B^2. We compare our proposed method against three baselines in point cloud processing: (1) Difference of Normals (DON) [31], (2) Conditional Euclid-

TABLE II
COMPARISON OF THE SEGMENTATION PERFORMANCE OF DIFFERENT ALGORITHMS ON THREE FREE-FORM OBJECTS

	$\max_j \Gamma_j(°)$			N_S			σ_S^2			σ_I^2			$\sigma_B^2(°)$		
Algorithm	o_1	o_2	o_3	o_1	o_2	o_3	o_1	o_2	o_3	o_1	o_2	o_3	o_1	o_2	o_3
Difference of Normals [31]	33.8	90.3	87.1	33	8	22	135.2	401.8	1030.7	0.6	4.7	2.7	25.0	32.9	50.0
Conditional Euclidean [19]	32.3	85.0	87.6	39	76	126	91.3	32.3	120.7	0.5	0.5	0.9	23.8	55.3	41.6
Region Growing [32]	34.7	20.6	17.8	53	561	1060	90.2	0.2	0.3	0.4	0.04	0.01	21.2	36.0	64.0
Our Method Alg. 1	26.8	24.8	27.99	6	26	66	22.8	4.0	9.9	12.8	9.3	8.3	31.7	50.0	57.7

N_S : Number of segmented regions. σ_S^2 : Variance of region size. $\max_j \Gamma_j$: Maximum angle difference among different regions S_j. σ_I^2 : Intra-class variance of region S_j. σ_B^2 : Between-class variance of region S_j.

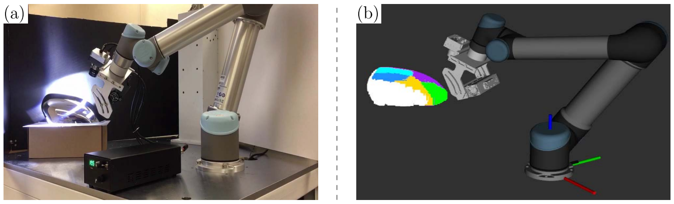


Fig. 11. Designed robotic validation experiment. (a) Real inspection experiment. (b) Simulation environment with the segmented regions results.

TABLE III
RESULTS OF REGISTRATION ACCURACY

Methods	$E_{min}(cm)$	$E_{max}(cm)$	$\mu_E(cm)$	$\sigma_E(mm)$
Raw	4.782	8.051	6.326	0.917
NoCoarse	15.893	20.352	19.303	3.153
Our	0.414	1.745	0.974	0.143

E_{min} and E_{max} is minimum and maximum value of the error. μ_E and σ_E is the mean and standard variance of the error respectively.

ean Clustering (CEC) [19] and (3) Region Growing (RG) [32]. Note that all the parameters of the baselines have been finetuned to obtain their best performances in several trials. The comparison results are shown in Table. II. Setting the allowable angle threshold as 28°, our algorithm guarantees a suitable solution for the condition $\max_j \Gamma_j < 28°$ with a uniform segmentation in terms of size and deviations, while other algorithms fail to ensure the availability of the segmentation solution. In essence, the failure and low-quality results of other algorithms are caused by the inconsistency between their parameters and the metrics. The common idea of them is to look for points meeting the predefined conditions (angle differences and Euclidean distances) among the search range of initialized points and absorb them into the corresponding clusters. The drawback of the reference algorithms is that they only consider the local information around the points on the edges of the clusters instead of the global performance of the clusters. Conversely, our method maintains stable performance for different free-form surfaces.

C. Registration

A specular side-mirror workpiece is taken as the experimental subject in this section, as shown in Fig. 11. The experiments in this section investigate the accuracy of our proposed real-time projection registration method, which is the basis to align the scanning path.

We run series of ablations to evaluate the importance of the components of our coarse-to-fine registration method; the results averaged over all eight trials can be found in Table III. We implement evaluations on the estimated pose $\hat{y}$ and its corresponding ground-truth y (both are transformed from the mesh model) by the Root Mean Square Errors (RMSE) E_R:

$$E_R(\hat{y}_i, y_i) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{25}$$

In real applications, industrial factories localize the workpiece with some designed fixtures. In this section, we leverage AR markers attached to the designed holder to observe the ground truth of the object posture. Note that the markers and the holder are only for collecting truth data in the experiments, but not components of our proposed system since they require manual assistance. Our ablations are designed to analyze two aspects of our registration algorithms.

1) The Benefit of Using Projection Boundary Compared With the Raw Observation: We modify our registration algorithm to leverage the raw observation from the depth sensor as input and match it to the complete workpiece with the ICP algorithm (denoted as *Raw* in Table. III). In this ablation, the raw observation is broken due to the specular reflection, causing the ICP algorithm tending to be trapped in a local minimum due to the huge difference between the source and the target. In contrast, in our method (denoted as *Our* in Table. III) the projection boundary acquires the complete contour from a top-down view to align the exterior boundary; thus this equivalent registration leads to a global optimum solution. Note that our algorithm is robust to the background of the workspace since it relies on the geometric feature instead of the color space. Our approach leads to a great improvement.

2) The Benefit of Using a Coarse-to-Fine Manner Compared With the Direct Registration: In this ablation, we utilize the exterior boundary extracted from the mesh model to align the captured boundary directly (denoted as *NoCoarse* in Table. III). Without the initialization, the failure probability increases due to the enlarging search space. Conversely, our algorithm (denoted as *Our* in Table. III) is robust to the object's pose with a mean error less than $1cm$. This ablation study indicates that coarse initialization is crucial for a satisfactory result.

D. Image Acquisition

To validate the detection performance of our proposed system, we use a specular side-mirror from the automotive

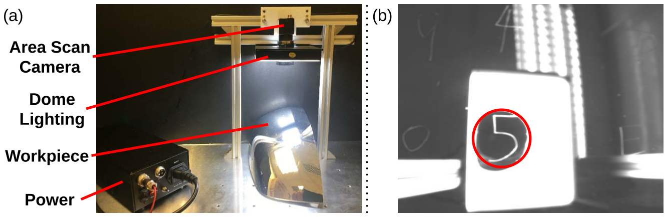


Fig. 12. Image acquisition with area scan vision sensor. (a) Experimental setup. (b) Example of high-quality image acquisition with dome light as illumination.

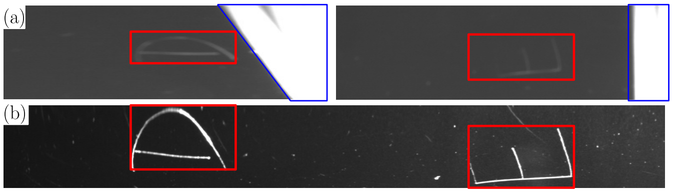


Fig. 13. Comparison of the image acquisition quality (red regions represent the captured defects, blue regions display the spot phenomenon). (a) Captured by area scan vision sensor. (b) Captured by our line scan vision sensor.

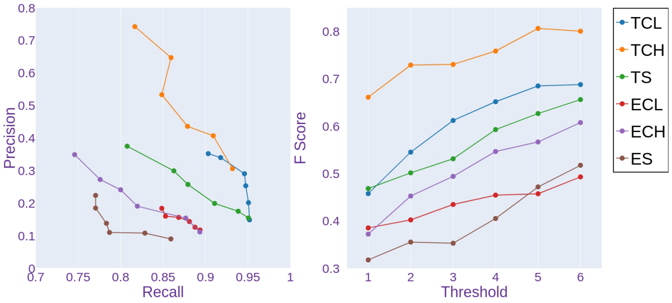


Fig. 14. Statistical results of defect detection results with various baselines in the case study (T: Thresholding, E: Histogram equalization, CL: Canny detector with low value (100,200), CH: Canny detector with high value (200,300), S: Sobel edge detector). (a) PR curve. (b) F score.

industry as an experimental subject. Several defects are added manually on the specular surface and the observation results from professional workers are regarded as the benchmarks of this inspection task. Fig. 11 illustrates the designed experiment. The upper free-form specular surface is selected as our experiment subject. The colored point cloud visualizes the segmentation result of our K-means-based region segmentation.

To highlight the advantages of our system, a traditional inspection configuration with an area scan sensor [3] is used for comparison, see Fig. 12(a). Dome light is employed to avoid an illumination spot caused by the specular surface. With the illumination condition, we can obtain high-quality images when the defects are spotted, as shown in Fig. 12(b).

As illustrated in Fig. 13(a), the area scan sensor is unable to focus on the ROI over the free-form surface, which compromises the quality of the image. Compared with the fixed acquisition configuration, our flexible image acquisition subsystem with a line scan sensor enables the adjustment of the scanning path according to the shape of the surface. Thus, the alignment between the vision sensor and the surface is ensured to obtain high-quality images, as shown in Fig. 13(b).

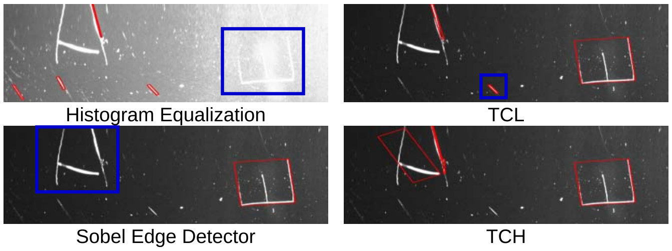


Fig. 15. Visualization of the detection results in the case study.

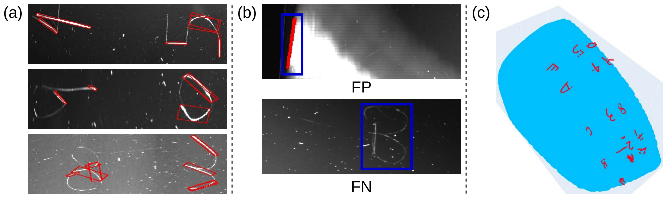


Fig. 16. Defect detection results conducted by the optimal pipeline (TCH). (a) Satisfactory result. (b) Some failure cases. (c) The distribution results via image mapping based on the detected defects.

E. Defect Detection

The automatic visual inspection process [35] is divided into two aspects, image acquisition and image processing. Although there are plenty of state-of-the-art defect detection algorithms for flat plates [36], free-form surfaces receive less attention. In this section, we provide a case study about defect detection with various common processing algorithms based on our captured images. Our objectives here are validating the acquisition quality of our robotic system and the feature extraction function of our proposed image processing pipeline, which are the basis of future research.

We evaluate the performance of different image processing methods based on the following indexes [37]:

$$P = TP/(TP+FP), \quad R = TP/(TP+FN),$$
$$F = (1+\beta_P^2) \times P \times R/(\beta_P^2 \times P + R) \tag{26}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively, and P, R and F are the precision, recall and scores respectively. F score provides an indicator of taking the trade-off between precision and recall [38], where $\beta_P = 2$ is set since we emphasize recall more in defect detection tasks.

As discussed in Sec. III, our proposed image processing pipeline consists of image enhancement, edge detection, and contour filter. In this case study, we include some common algorithms for comparison.

Specifically, we include the histogram equalization [39] as an alternative option relative to the pixel thresholding in image enhancement; For edge detection, we compare the Sobel operator [40] with Canny detectors [41] under various values. Combining them into different detection algorithms, the detailed comparison results are shown in Fig. 14. Note that the statistical results consider the observation of professional workers as the ground truth and false negatives are defined as none of any segment of a manually predefined defect is detected. For each implementation, we evaluate the performance with respect to various area thresholds (unit in pixels), with a range of (50, 100, 150, 200, 250, 300). As the area threshold increases, fewer samples are considered as the detected defects. Although *TCL* achieves high recall, its low precision results in a poorer F score. Fig. 14(b) illustrates that *TCH* is more robust and accurate under various thresholds, which is considered as the optimal solution in our case.

Fig. 15 graphically demonstrates the cause of the quantitative results in the case study, where the red boxes describe the detected results via the corresponding baseline and the blue boxes illustrate the failure cases. Although histogram equalization improves the contrast of the image, the regions with defects are easily overexposed. Sobel operator is susceptible to noise, resulting in the omission of certain defects. Compared with *TCH*, *TCL* achieves lower precision because of mistaking additional environmental dust as defects. Other detection results obtained by the *TCH* algorithm are shown in Fig. 16 (sharing the same specification with Fig. 15). Fig. 16(a) displays the successful results. Here, we consider the detection that captures only a part of the defects as true positives, which are caused by the disconnection of defects. Besides, we also present some failure cases for further research in Fig. 16(b). The edges of the workpiece are considered as defects due to the sudden gray level change, playing as false-positive cases. In addition, some true defects are missed due to the low contrast between the defects and the background.

Fig. 16(c) illustrates the defect visualization result via image mapping. Note that the displayed defects are selected among the true positives by operators. Although we cannot obtain a precise 3D reconstruction model with the proposed method, this visualization result can provide an intuitive distribution of the defects on the specular surface.

V. Conclusion

In this paper, we developed a novel sensor-guided automatic defect inspection system for convex free-form specular surfaces. Mounted on a robotic manipulator, a high-resolution vision sensor can flexibly execute surface scanning. Based on the prior mesh model of the workpiece, K-means-based region segmentation shows robustness to segment free-form surfaces to relatively flat regions without explicit tuning. Our path planning algorithm ensures inspection completeness through patch segmentation and adaptive ROI further refinement. With the real-time workpiece localization through the perception of the depth sensor, our system operates this visual inspection task automatically. The experimental results demonstrate the accuracy and robustness of our sensor-based system.

There are a few limitations that exist in the proposed robotic system which need further investigation. Firstly, the performance of the system is validated in a laboratory environment. Thus, it requires industrial compliance tests with various workpieces under real manufacturing environments. Secondly, some improvement about the design of the custom-built end-effector is required to increase the scanning space for the image acquisition subsystem. Moreover, the proposed image processing algorithm cannot classify defects at this stage. Further, the computationally expensive tasks like segmentation and classification of defects may utilize some data-driven methods, such as convolutional neural network (CNN) [42].

References

[1] O. Duran, K. Althoefer, and L. D. Seneviratne, “State of the art in sensor technologies for sewer inspection,” *IEEE Sensors J.*, vol. 2, no. 2, pp. 73–81, Apr. 2002.

[2] J. Gan, Q. Li, J. Wang, and H. Yu, “A hierarchical extractor-based visual rail surface inspection system,” *IEEE Sensors J.*, vol. 17, no. 23, pp. 7935–7944, Dec. 2017.

[3] C.-S. Cho, B.-M. Chung, and M.-J. Park, “Development of real-time vision-based fabric inspection system,” *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1073–1079, Aug. 2005.

[4] Q. Luo and Y. He, “A cost-effective and automatic surface defect inspection system for hot-rolled flat steel,” *Robot. Comput.-Integr. Manuf.*, vol. 38, pp. 16–30, Apr. 2016.

[5] H. Freeman, *Machine Vision for Three-Dimensional Scenes*. Amsterdam, The Netherlands: Elsevier, 2012.

[6] Y.-J. Chen, J.-C. Tsai, and Y.-C. Hsu, “A real-time surface inspection system for precision steel balls based on machine vision,” *Meas. Sci. Technol.*, vol. 27, no. 7, Jul. 2016, Art. no. 074010.

[7] S. Akhtar, A. Tandiya, M. Moussa, and C. Tarry, “A robotics inspection system for detecting defects on semi-specular painted automotive surfaces,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8928–8934.

[8] J. Molina, J. E. Solanes, L. Arnal, and J. Tornero, “On the detection of defects on specular car body surfaces,” *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 263–278, Dec. 2017.

[9] J. Geng and J. Xie, “Review of 3-D endoscopic surface imaging techniques,” *IEEE Sensors J.*, vol. 14, no. 4, pp. 945–960, Apr. 2014.

[10] M. Babu, P. Franciosa, and D. Ceglarek, “Spatio-temporal adaptive sampling for effective coverage measurement planning during quality inspection of free form surfaces using robotic 3D optical scanner,” *J. Manuf. Syst.*, vol. 53, pp. 93–108, Oct. 2019.

[11] E. Glorieux, P. Franciosa, and D. Ceglarek, “Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection,” *Robot. Computer-Integrated Manuf.*, vol. 61, Feb. 2020, Art. no. 101843.

[12] M. Les, “Shape understanding system: Understanding a convex object,” *J. Electron. Imag.*, vol. 12, no. 2, p. 327, 2003, doi: 10.1117/1.1557157.

[13] S. Huo, D. Navarro-Alarcon, and D. T. Chik, “A robotic defect inspection system for free-form specular surfaces,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11364–11370.

[14] A. Cherubini and D. Navarro-Alarcon, “Sensor-based control for collaborative robots: Fundamentals, challenges, and opportunities,” *Frontiers Neurorobotics*, vol. 14, p. 113, Jan. 2021.

[15] Z. Bi and X. Wang, *Computer Aided Design and Manufacturing*. Hoboken, NJ, USA: Wiley, 2020.

[16] D. Girardeau-Montaut, “Cloud compare-3D point cloud and mesh processing software,” *Open Source Project*, to be published.

[17] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. Berkeley Symp. Math. Statist. Probab.*, vol. 1, no. 14, 1965, pp. 281–297.

[18] A. M. Mehar, K. Matawie, and A. Maeder, “Determining an optimal value of k in K-means clustering,” in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, Dec. 2013, pp. 51–55.

[19] R. B. Rusu, “Semantic 3D object maps for everyday manipulation in human living environments,” *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[20] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *J. ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[21] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, 2013.

[22] D. Navarro-Alarcon, J. Qi, J. Zhu, and A. Cherubini, “A Lyapunov-stable adaptive method to approximate sensorimotor models for sensor-based control,” *Frontiers Neurorobot.*, vol. 14, p. 59, Sep. 2020.

[23] K. Litomisky, *Consumer RGB-D Cameras and Their Applications*, vol. 20. Francisco, CA, USA: Rapport Technique, Univ. California, 2012, p. 28.

[24] M. G. Diaz, F. Tombari, P. Rodriguez-Gonzalvez, and D. Gonzalez-Aguilera, “Analysis and evaluation between the first and the second generation of RGB-D sensors,” *IEEE Sensors J.*, vol. 15, no. 11, pp. 6507–6516, Nov. 2015.

[25] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[26] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Modeling*, 2001, pp. 145–152.

[27] A. Baker, *Matrix Groups: An Introduction to Lie Group Theory*. London, U.K.: Springer, 2007.

[28] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Comput. Graph. Image Process.*, vol. 1, no. 3, pp. 244–256, Nov. 1972.

[29] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3. Kobe, Japan, 2009, p. 5.

[30] M. W. Burke, *Image Acquisition: Handbook of Machine Vision Engineering*, vol. 1. Springer, 2012.

[31] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Visualizat. Transmiss.*, Oct. 2012, pp. 501–508.

[32] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 88–100, Jun. 2015.

[33] Y. Xiao, L. Daniel, and M. Gashinova, "Image segmentation and region classification in automotive high-resolution radar imagery," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6698–6711, Mar. 2021.

[34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[35] R. G. Rosandich, *Intelligent Visual Inspection: Using Artificial Neural Networks*. London, U.K.: Chapman & Hall, 1997.

[36] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 348–363, Jan. 2008.

[37] K. Qiu, L. Tian, and P. Wang, "An effective framework of automated visual surface defect detection for metal parts," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20412–20420, Sep. 2021.

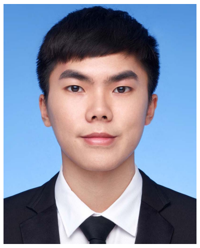
[38] N. Chinchor and B. Sundheim, "MUC-5 evaluation metrics," in *Proc. 5th Conf. Message Understand. (MUC)*, Aug. 1993, pp. 69–78.

[39] S. M. Pizer *et al.*, "Adaptive histogram equalization and its variations," *Comput. Vis., Graph., Image Process.*, vol. 39, no. 3, pp. 355–368, 1987.

[40] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[41] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI–8, no. 6, pp. 679–698, Nov. 1986.

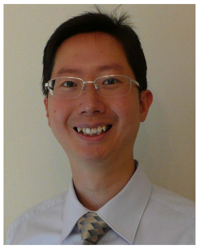
[42] X. Miao, J. Wang, Z. Wang, Q. Sui, Y. Gao, and P. Jiang, "Automatic recognition of highway tunnel defects based on an improved U-Net model," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11413–11423, Dec. 2019.



Shengzeng Huo (Graduate Student Member, IEEE) received the B.Sc. degree in vehicle engineering from the South China University of Technology, China, in 2019. He is currently pursuing the Ph.D. degree in mechanical engineering with The Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include automation, data-driven visual servoing, and robot learning.

Bin Zhang received the bachelor's degree from Beihang University in 2017 and the master's degree from the China Academy of Space Technology in 2020. He is pursuing the Ph.D. degree with the Department of Mechanical Engineering, The Hong Kong Polytechnic University. His research interests include multi-agent systems and control theory.

Muhammad Muddassir received the master's degree in control science and engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2017. He is pursuing the Ph.D. degree with the Mechanical Engineering Department, The Hong Kong Polytechnic University. He joined as a Research Assistant Robotics and Machine Intelligence Laboratory (ROMI Labs), The Hong Kong Polytechnic University, in 2018. His current research interests include the development and control of sensor-guided robotic systems.



David T. W. Chik received the Ph.D. degree from The University of Hong Kong in 2005. He is a Lead Engineer at the Hong Kong Applied Science and Technology Research Institute. His current research interests are on deep reinforcement learning and its applications in manufacturing and healthcare areas. He also collaborates with local companies to commercialize innovative products.

David Navarro-Alarcon (Senior Member, IEEE) received the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Shatin, Hong Kong, in 2014.

Since 2017, he has been with The Hong Kong Polytechnic University (PolyU), Hung Hom, Hong Kong, where he is an Assistant Professor at the Department of Mechanical Engineering, the Principal Investigator of the Robotics and Machine Intelligence Laboratory, and an Investigator at the Research Institute for Smart Ageing. Before joining PolyU, he was a Researcher at the CUHK T Stone Robotics Institute from 2014 to 2017. He has had visiting appointments at the University of Toulon, France, and the Technical University of Munich, Germany. His current research interests include perceptual robotics and control theory.