A Sensor-based Robotic Line Scan System with Adaptive ROI for Inspection of Defects over Convex Free-Form Specular Surfaces

Shengzeng Huo, Bin Zhang, Muhammad Muddassir, David TW 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 system is composed of a robotic manipulator equipped with a line scanner system. Our scanning approach uses a mesh model to segment the convex surface of the object into areas with similar curvatures, then, adaptively adjusts the camera's scanning range to ensure the complete coverage of the surface. We propose a new projection registration algorithm that uses measurements from a depth sensor to robustly localize and register the specular workpiece in real-time. We develop a complete image processing pipeline for automatic defect detection from the captured images. 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 is important in modern manufacturing, as defects compromise the quality and value of products. The common practice of many factories is to employ numerous workers to manually perform the inspection task, which is costly, time-consuming, and vulnerable to subjective results. With the development of effective vision sensors and image processing algorithms [1], automatic vision-based defect inspection system presents a feasible and sustainable solution to the existing problems.

Although automatic defect inspection system has seen great progress in recent years (e.g. [2–5]), its geometric adaptability and robustness in various inspection tasks remain challenging. That is particularly critical for the inspection of free-form specular surfaces [6], 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). Free-form specular objects are common in many industries, particularly the automotive industry, therefore, it is important to develop effective methods to automate their defect inspection. The goal of this paper is to develop a feasible robotic solution to this problem.

S. Huo, B. Zhang, M. Muddassir and D. Navarro-Alarcon are with Dept. of Mechanical Engineering, The Hong Kong Polytechnic University (PolyU), Kowloon, Hong Kong. D. Navarro-Alarcon is also with the Research Institute for Smart Ageing of PolyU. Corresponding author: dna@ieee.org.

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

This work is supported in part by: ASTRI (grant PO2019/399), the Research Grants Council (grants 14203917, F-PolyU503/18), the Jiangsu Industrial Technology Research Institute Collaborative Research Program Scheme (grant ZG9V), the Key-Area Research and Development Program of Guangdong Province 2020 (project 76), and PolyU (grants G-YBYT, 4-ZZHJ, UAKU).

In the past decade, several automatic inspection systems leveraging on various sensors have been applied in different areas, such as sewers [2], rail surfaces [3], textured materials [4], LED chips [5], etc. Due to their simple structure, area-of-interest (AOI) based systems are widely used in real-time defect inspections of (typically) flat surfaces, such as hot-rolled steel sheets [7] and fabrics [8]. Specialised setups are required when applying AOI-based systems into non-flat objects, including sensors, illumination, fixtures, etc. For example, [9] developed a dual-light system to properly inspect the curved surface of a metal sphere. However, these AOI-based systems usually have *fixed* scanning configurations, which limits their usage to few types of object shapes.

In addition to the geometry, the reflectivity of the surface has also been studied by researchers. [10] developed a method that visually analyzed the reflection of a known projection pattern over the surface, where both the pattern generator and the camera were manipulated by a robot. Based on sensor fusion, [11] 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 these recent promising results, the automatic inspection of free-form specular surfaces still remains an open research problem.

Robotic dimensional inspection presents a more flexible alternative for the inspection task. This approach uses a computer-controllable manipulator equipped with 2D/3D vision sensors [12, 13] to scan various surface shapes with active robot motions. In this paper, we build on this kind of configuration to perform defect inspection on *convex* specular surfaces [14]. The conceptual overview of our new sensor-guided solution is shown in Fig. 1. The original contributions of this work are as follows[1]:

- A path planning algorithm with an adaptive region-of-interest (ROI) and K-means segmentation for inspecting free-form specular surfaces based on the acquisition model of the adopted sensors.
- A new registration algorithm based on iterative closest point (ICP) for real-time localization of observed specular

[1]An earlier version of this paper will be presented in June 2021 at the IEEE Int. Conf. on Robotics and Automation (ICRA) [15]. In contrast with the preliminary paper [15], this version has the following new methods/content: (i) An adaptive ROI algorithm that guarantees the completeness of the inspection task; (ii) A new geometric model that enables the analysis of the line scan camera; (iii) An optimization algorithm that improves the path planning efficiency; (iv) A detailed comparison study of the proposed method with various baselines.

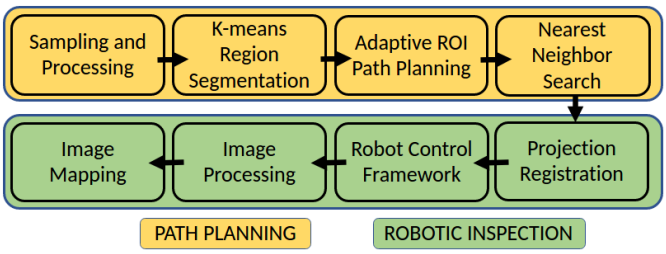

Fig. 1: Overview of the automatic robotic defect inspection pipeline. Given a mesh model of the workpiece, we generate a feasible scanning path. Subsequently, our robotic system executes the designed procedures to finish the inspection task.

objects.

- A detailed experimental study to validate the performance of our automatic inspection system and compare it with existing baselines.

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

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 p^B in the robot's frame
$\vec{\boldsymbol{\rho}}_i, \vec{\boldsymbol{\pi}}_i, \vec{\boldsymbol{\Pi}}_i$	i-th position and orientation pose for manipulator
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 frame and in 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
Ψ, χ, τ	precision, recall and accuracy in defect detection

II. PATH PLANNING

In this section, we present a *complete* and *formal* methodology to plan the active motions of the line scan sensor.

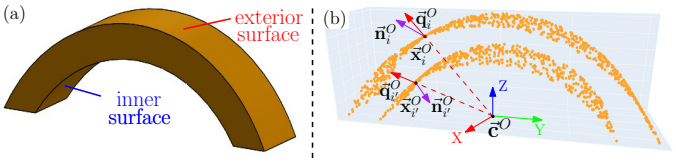

Fig. 2: 3D representation information of the workpiece. (a) Mesh model. (b) Sampled point cloud and the corresponding preprocessing procedure.

A. Sampling and Preprocessing

Since the mesh models consist of many vertices and surfaces (which might be difficult to process), we sample points from the mesh model to facilitate the path planning. [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. 2. Due to its sampling uniformity and complete coverage, the obtained sampled point cloud O consisting of points $O = (\vec{\mathbf{x}}_1^O, \vec{\mathbf{x}}_2^O, ...)$ 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. 2(b). The center $\vec{\mathbf{c}}^O$ of the point cloud O is computed as

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

where ζ_O is the size of the point cloud O. Next, we utilize the normal vectors $(\vec{\mathbf{n}}_1^O, \vec{\mathbf{n}}_2^O, ...)$ 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 [17] of surface geometric features (viz. positions and normals). In our method, we use K-means clustering [18] to partition the point cloud O.

Our algorithm takes the points of O and their corresponding normals as input, as shown in Fig. 3(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 [19]. 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 when the maximum intra-class

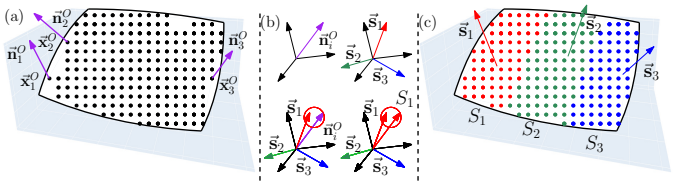

Fig. 3: Conceptual representation of K-means-based region segmentation. (a) Input information: i-th point $\vec{\mathbf{x}}_i^O$ and its corresponding normal $\vec{\mathbf{n}}_i^O$. (b) Clustering for each sample $\vec{\mathbf{n}}_i^O$. (c) Recompute the feature vector $\vec{\mathbf{s}}_j$ of each region S_j.

difference is smaller than a threshold β_K. The entire procedure is given in Algorithm 1 and illustrated in Fig. 3.

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. Afterwards, 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:

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

Then, we find the cluster S_j that produces the samllest 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 to 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 that clusters with respect to N and the outer that increases the value of N. The adjustment depends on the performance of Λ_t. If at the iteration step t, the performance decreases (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 [20] 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 β_K, O.
Output $S_j, j = 1, 2, ..., N$
Routine

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 $\vec{\mathbf{x}}_i^O \rightarrow S_{j^*}$
    end for
    Compute $\Lambda_t \leftarrow (4)$
    Update $\vec{\mathbf{s}}_j, j = 1 : N \leftarrow (5)$,
  end while
  $N \leftarrow N + 1$
end while

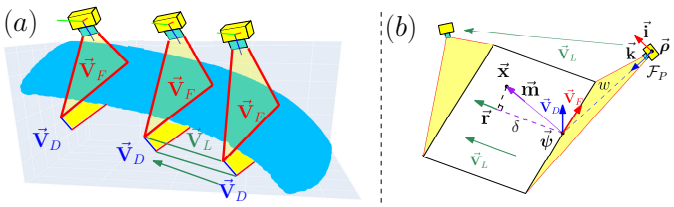

Fig. 4: Geometric acquisition model of the line scan camera with linear motion. (a) Cuboid coverage generation process. (b) Mathematical model of the coverage cuboid and completeness validation.

C. Adaptive ROI Based Path Planning

The proposed planning algorithm takes the segmented regions S_j as inputs, 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 a shortest path to cover the entire region S_j. However, the irregular shapes and scattered distribution make the path planning very challengiing. Hence, we establish a local coordinate frame $\mathcal{F}_S$ with respect to 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.

The aim of our adaptive ROI method is 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. 4(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

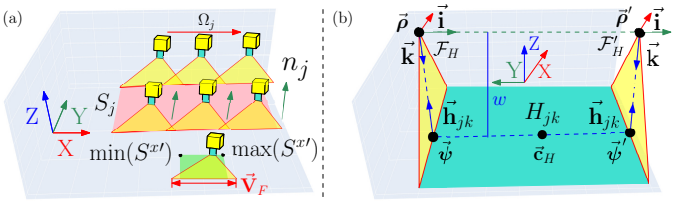

Fig. 5: Further region segmentation and linear path planning. (a) Patch segmentation along horizontal and vertical directions. (b) Linear path planning with respect to patch H_{jk}.

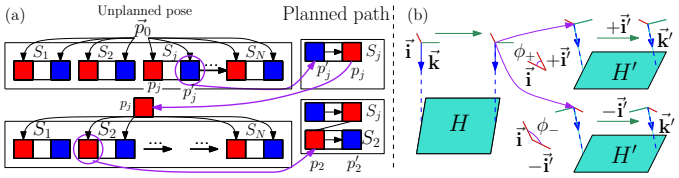

Fig. 6: Complete path optimization with respect to entire free-form surface. (a) Scanning sequence arrangement based on distance-optimal criteria. (b) Rotation angle minimization in the process of adjacent scanning pose switching.

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. 4(b). Let's suppose the linear scanning starting from a pose p, depicted as a frame $\mathcal{F}_P$, whose position, Z-axis and X-axis are $\vec{\boldsymbol{\rho}}, \vec{\mathbf{k}}, \vec{\mathbf{i}}$ respectively. Thus, the inspection point of the sensor on the surface is $\vec{\boldsymbol{\psi}} = \vec{\boldsymbol{\rho}} - w \cdot \vec{\mathbf{k}}$, 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 $\vec{\mathbf{m}} = \vec{\mathbf{x}} - \vec{\boldsymbol{\psi}}$. 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 $\vec{\mathbf{V}}_D, \vec{\mathbf{V}}_F, \vec{\mathbf{V}}_L$ respectively.

In addition, *normal similarity*, 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 steps, each along the horizontal and the vertical directions, as shown in Fig. 5(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 region S_j into Ω_j sub-regions (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 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. 5(b). Here, we utilize two frames $\mathcal{F}_H, \mathcal{F}'_H$ to describe two poses of the motion. Their origin $\vec{\boldsymbol{\rho}}, \vec{\boldsymbol{\rho}}'$ locate on:

$$\vec{\boldsymbol{\psi}} = [c_x \ \min(H^Y) \ c_z]^T \ \ \vec{\boldsymbol{\psi}}' = [c_x \ \max(H^Y) \ c_z]^T \tag{10}$$

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

where $\vec{\boldsymbol{\psi}}$ and $\vec{\boldsymbol{\psi}}'$ 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:

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

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 six degrees of freedom (DoF) path pose p:

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

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, $\vec{\mathbf{k}}^O, \vec{\mathbf{i}}^O, \vec{\boldsymbol{\rho}}^O$ represent vectors in the object's frame $\mathcal{F}_O$. Note that $\vec{\boldsymbol{\rho}}$, $\vec{\mathbf{k}}$ and $\vec{\mathbf{i}}$ are represented as 4×1 vectors in homogeneous coordinates for 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 is adopted, which equals to minimizing the scanning time since we keep the velocity of the camera constant during the scanning task. Thus, this issue is formulated as the following constrained optimization problem:

$$\text{minimize} \sum_{i=1}^{N_P-1} C(p_i, p_{i+1})$$

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 the adaptive ROI path planning, we convert the optimization to a sorting problem and solve it by the Nearest Neighbor Search [21], which are discussed with respect to the position and the 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

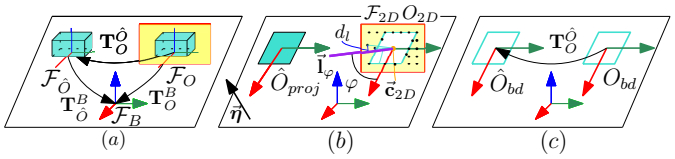

Fig. 7: Projection registration for specular surface: (a) Problem formulation. (b) Boundary extraction. (c) Final registration.

edges and obtain $2N$ poses in total. We formulate the position optimization as the coverage path planning (CPP) [22] 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. 6(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. 6(b). To illustrate the idea, we define a switch between patches $H(\vec{\mathbf{k}}, \vec{\mathbf{i}}) \rightarrow H'(\vec{\mathbf{k}}', \vec{\mathbf{i}}')$. 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

$$\begin{cases} \vec{\mathbf{i}}' \leftarrow +\vec{\mathbf{i}}' & \phi_+ \leq \phi_- \\ \vec{\mathbf{i}}' \leftarrow -\vec{\mathbf{i}}' & else \end{cases} \tag{14}$$

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

III. ROBOTIC INSPECTION

In this section, we provide the detailed *robotic* procedures about the *automatic* sensor-based defect inspectioin.

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]. Hence, we design a novel projection registration algorithm to deal with this issue, formulating it as a coarse-to-fine registration issue, as shown in Fig. 7(a).

The hand-eye calibration $\mathbf{T}_C^B$ is implemented via AR markers. $\mathcal{F}_O$ is the object's frame whose transformation to the robot's frame $\mathcal{F}_B$ is desired. $\mathcal{F}_{\hat{O}}$ 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 7(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}:

$$\vec{\mathbf{l}}_\varphi = [r\cos\varphi \;\; -r\sin\varphi]^T \tag{15}$$

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

$$d_l = \left|\vec{\mathbf{x}}^{O_{2D}} \cdot \omega_\varphi\right|, \;\; \omega_\varphi = [\sin\varphi \;\; \cos\varphi]^T \tag{16}$$

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

$$\hat{O}_{proj} = \hat{O} - (\vec{\boldsymbol{\eta}} \cdot \hat{O}) \cdot \vec{\boldsymbol{\eta}} \tag{17}$$

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 $\mathbf{T}_O^B = \left[\mathbf{R}_O^B | \vec{\mathbf{t}}_O^B\right]$, 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

$$p^B = \left\{\mathbf{T}_O^B \cdot \vec{\boldsymbol{\rho}}^O, \mathbf{R}_O^B \cdot \vec{\mathbf{k}}^O, \mathbf{R}_O^B \cdot \vec{\mathbf{i}}^O\right\} \tag{18}$$

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

$$\vec{\boldsymbol{\Pi}}_i = \left[\begin{array}{cc} \vec{\boldsymbol{\rho}}_i^T & \vec{\boldsymbol{\pi}}_i^T \end{array}\right]^T \tag{19}$$

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 $\mathbf{R}_\pi$.

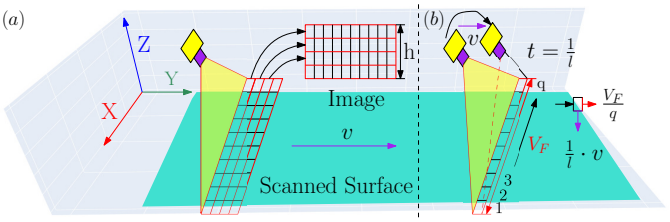

Fig. 8: Conceptual representation of automatic scanning procedure. (a) Principal of image acquisition during camera's motion. (b) Acquisition line rate adjustment for unit magnification.

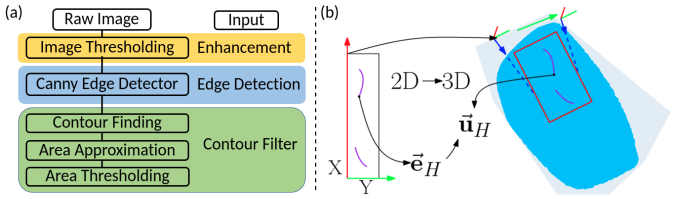

Fig. 9: Details about the automatic defect detection and visualization. (a) Complete pipeline of digital image processing for defect detection. (b) Conceptual explanation about the mathematical model of the image mapping.

The line scan camera captures one line at a time, and stack the horizontal rows of pixels to generate a 2D image, see Fig. 8. The sampling rate of the camera needs to be synchronized with the robot's motion to ensure the scanning completeness and avoid repetition. The synchronization is achieved by defining:

$$|\vec{\mathbf{V}}_F|/q = (1/l) \cdot v \tag{20}$$

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.

C. Image Processing

Digital image processing is also necessary for the proposed sensor-based defect inspection solution. In this section, we design an image processing pipeline for our system based on OpenCV [28], as shown in Fig. 9(a).

Taking raw images as input, we firstly adopt image thresholding for enhancement. Edge features are suitable for our task by displaying sudden changes in gray level. Therefore, we next extract edge features from the enhanced images by adopting a robust Canny edge detector [29]. Some dust may also be detected by the high-resolution sensor and disturb the defect inspection. Therefore, we include the deeper contour sketching and area approximation [30] in our algorithm to enrich the feature descriptors. Finally, we tune the allowable defect area threshold for judgement. This criteria also benefits the operators to exclude the "fake" defects.

D. Image Mapping

In addition to the number of defects, the distribution of defects also provides an intuitive understanding about 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. 9(b).

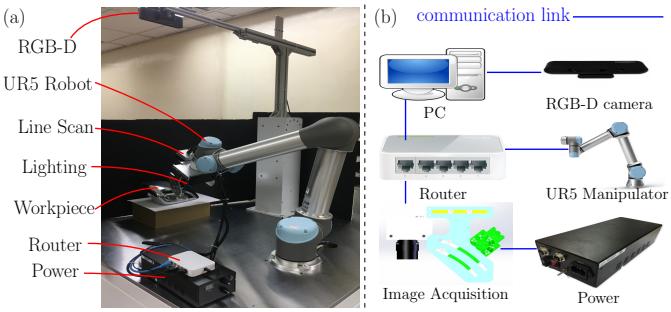

Fig. 10: The platform of our proposed autonomous robotic inspection system. (a) Experimental configuration. (b) Syetem Architecture.

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 $\vec{\mathbf{\Pi}}_H$ and stops at pose $\vec{\mathbf{\Pi}}'_H$. Hence, we use pose $\vec{\mathbf{\Pi}}_H$ 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

$$\vec{\mathbf{u}}_H = \begin{bmatrix} (e_x - q/2) \cdot \kappa & e_y \cdot \kappa & w \end{bmatrix}^T \tag{21}$$

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 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. 10(a) illustrates the configuration of our proposed robotic platform. An Orbbec Astra depth sensor is mounted on the top of the manipulation space for real time workpiece localization. To control the image acquisition system to achieve a 6-DOF scanning path over the workpiece, our system includes a UR5 manipulator from Universal Robots. We also design an instrumented customised holder to equip the image acquisition system with the end-effector of the manipulator under a flexible and precise illumination configuration.

Fig. 10(b) conceptually depicts the architecture of the system. The image acquisition sub-system 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 high power strobe provides adjustable voltage to the light source. To communicate different components in the system, we establish a network with the help of a router via the TCP/IP socket and ROS [31]. Hence, the Linux PC is able to simultaneously control all components in real time.

Velocity and acceleration of the manipulator are empirically set as $0.05m/s$ and $0.05m/s^2$ respectively due to safety

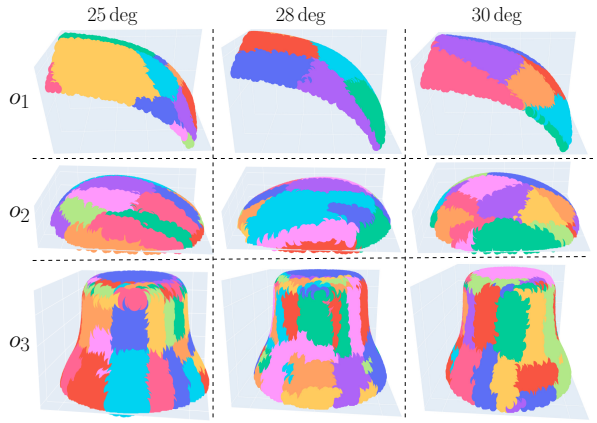

Fig. 11: Segmentation results with respect to different free-form surfaces and different termination thresholds.

reasons. The magnitude of $\vec{\mathbf{V}}_F$ and $\vec{\mathbf{V}}_D$ of the vision sensor can be calculated according to [32]

$$\left|\vec{\mathbf{V}}_F\right| = 2w\tan(A/2),\ \xi \approx \Delta/1500,\ \left|\vec{\mathbf{V}}_D\right| \approx 2w^2F\xi/f^2 \tag{22}$$

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.

B. Segmentation Performance

Flat region segmentation is one of the prerequisites of high-quality image acquisition, whose biggest challenge is the diversity of object shapes. To validate the robustness of our proposed K-means based region segmentation algorithm, we present the implementation results in terms of different mesh models, denoted as o_1, o_2 and o_3 respectively. Inspired by Otsu's algorithm [35], we design the segmentation deviations of intra-class and between-class to measure the segmentation performance statistically,

$$\begin{aligned} \mu_S &= \frac{1}{N}\cdot\sum_{j=1}^{N}\zeta_j, \quad \sigma_S^2 = \frac{1}{N}\cdot\sum_{j=1}^{N}(\zeta_j-\mu_S)^2 \\ \sigma_I^2 &= \frac{1}{N}\cdot\sum_{j=1}^{N}\frac{1}{\zeta_j}\sum_{i=1}^{\zeta_j}\gamma_{ij} \\ \vec{\boldsymbol{\mu}}_B &= \frac{1}{N}\cdot\sum_{j=1}^{N}\vec{\mathbf{s}}_j, \quad \sigma_B^2 = \frac{1}{N}\cdot\sum_{j=1}^{N}\gamma(\vec{\mathbf{s}}_j,\vec{\boldsymbol{\mu}}_B) \end{aligned} \tag{23}$$

where $\gamma(\vec{\mathbf{s}}_j,\vec{\boldsymbol{\mu}}_B)$ computes the angle between $\vec{\mathbf{s}}_j$ and $\vec{\boldsymbol{\mu}}_B$ according to Eq. 3. For comparison, three commonly used segmentation algorithms in point cloud processing, namely Difference of Normal (DON) [33], Conditional Euclidean Clustering (CEC) [20] and Region Growing (RG) [34], are considered as references. The comparison results are shown in Table. II.

Note that all the parameters of the reference algorithms have been fine-tuned to obtain best performances. Among the results, $\max_j \Gamma_j$, measuring the maximum intra-class difference, is the most important evaluation criteria. Setting the termination threshold as 28^o, our algorithm is able to find the optimal solution regarding to the corresponding conditions,

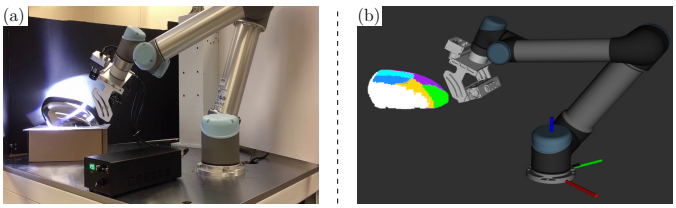

Fig. 12: Designed robotic validation experiment. (a) Real inspection experiment. (b) Simulation environment with the segmented regions results.

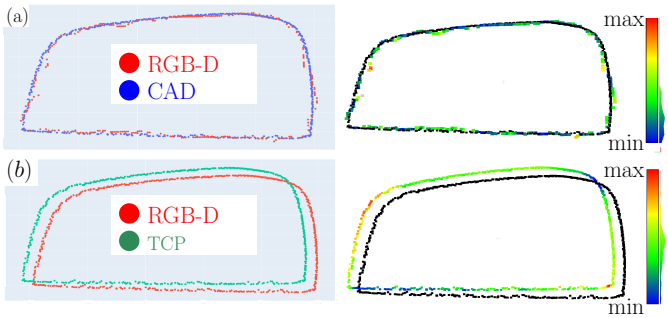

Fig. 13: Conceptual results of workpiece localization. (a) Comparison between depth camera capturing and Computer-aided Design (CAD) mesh model. (b) Comparison between depth camera capturing and measurement by the robot manipulator.

while other algorithms fail to ensure the availability of the final segmentation. The other evaluation indexes, $N_S, \sigma_S^2, \sigma_I^2, \sigma_B^2$ are auxiliary indicators for the clusters distribution. The optimal segmentation is to divide the free-form surface evenly in terms of size and position, while minimizing the intra-class difference and maximizing the between-class difference. The results show that our method can maintain stable performance with respect to different free-form surfaces. Basically, the failure and bad results of other algorithms are caused by the inconsistency between their parameters and the criteria. The common idea of them is to look for points meeting the predefined conditions, such as the 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.

To validate the robustness of our proposed algorithm in terms of shapes and required smoothness, we conduct multiple segmentation trials on several free-form objects with different thresholds, as shown in Fig. 11. We can conclude from these intuitive results that our K-means unsupervised region segmentation is suitable for various convex free-form objects under different conditions.

C. Registration

Another prerequisite of image acquisition is the precise localization for the workpiece in the robot's frame. A specular side-mirror workpiece is taken as the experimental subject in this section, as shown in Fig. 12.

To evaluate the performance of our indirect registration algorithm, we take the shape errors and the position errors into account. The ground truth of the shape and position are provided by the mesh model and the measurements from the end-

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

	$\max_j \Gamma_j(^\circ)$			N_S			σ_S^2			σ_I^2			$\sigma_B^2(^\circ)$		
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 [33]	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 [20]	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 [34]	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.

TABLE III: RESULTS OF REGISTRATION ACCURACY

Types	$d_{min}(cm)$	$d_{max}(cm)$	$\mu_D(cm)$	$\sigma_D(mm)$
Shape	0.004	0.680	0.167	1.134
Position	0	1.288	0.765	2.182

d_{min} and d_{max} is minimum and maximum value of the Euclidean distance. μ_D and σ_D is the mean and standard variance of the distance respectively.

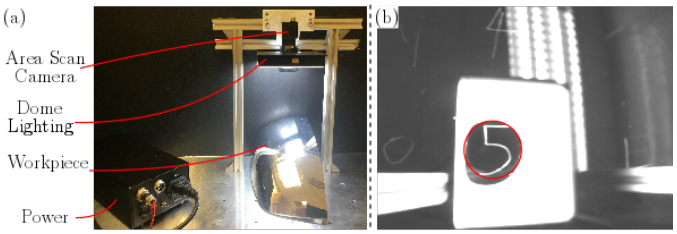

Fig. 14: Image acquisition with area scan vision sensor. (a) Experimental setup. (b) Example of high-quality image acquisition with dome light as illumination.

effector of the manipulator. To quantify the error, Hausdorff distance [36] is used, and the mathematical comparison results are shown in Table. III. Fig. 13(a) shows the comparison between the boundary extracted from the point cloud captured by depth camera and the boundary extracted from the mesh model. Since we focus on the shape retrieval in this part, the comparison is executed after the ICP alignment. This comparison visualizes that our indirect registration is suitable to describe the original shape of the workpiece. Fig. 13(b) illustrates the position registration error. The low localization error can guarantee the scanning performance of the registered path.

D. Image Acquisition

To validate the detection performance of our proposed system, we take a specular side-mirror from the automotive 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. 12 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 area scan sensor [7] is used for comparison, see Fig. 14(a). Dome light is employed to avoid an illumination spot caused by the specular surface. With the illumination condition, we are able to obtain high quality images when the defects are spotted, as shown in Fig. 14(b).

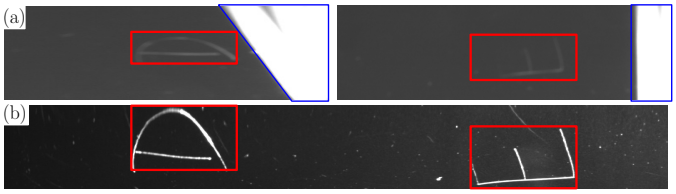

Fig. 15: Comparison of the image acquisition quality (red regions represent the captured defects, blue regions display the spot phenomenon). (a) Captured by area scan vision sensor. (b) Captured by our line scan vision sensor.

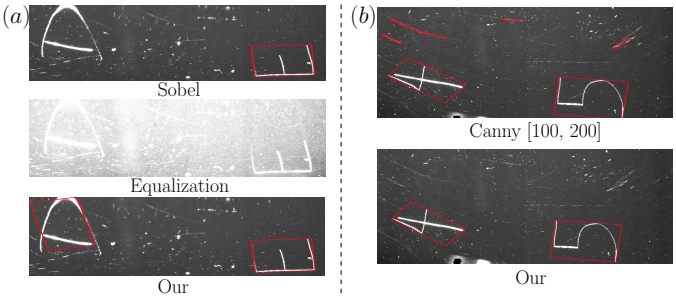

Fig. 16: Conceptual comparison about image processing for defects detection. (a) Comparison with histogram equalization and Sobel operator. (b) Case study about the parameters of the Canny Edge Detector.

TABLE IV: COMPARISON OF THE DEFECT DETECTION PERFORMANCE OF DIFFERENT PIPELINES ON THE SELECTED WORKPIECE

Pipeline	**TP**	**FP**	**FN**	Ψ	χ	τ
Sobel Operator [37]	40	5	10	0.889	0.8	0.818
Equalization [38]	59	120	9	0.330	0.868	0.760
Canny [29] [100, 200]	44	40	2	0.524	0.957	0.870
Our method	51	8	2	0.864	0.962	0.943

TP, FP and FN are true positive, false positive and false negative respectively. Ψ, χ and τ are precision, recall and accuracy respectively.

As illustrated in Fig. 15(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 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. 15(b).

E. Defect Detection

The final step of the automatic defect inspection is the image processing and the unqualified defects statistics. In this section, we also include some common image processing techniques as references to highlight the advantages of our pipeline. Specifically, the histogram equalization [38] is adopted as a

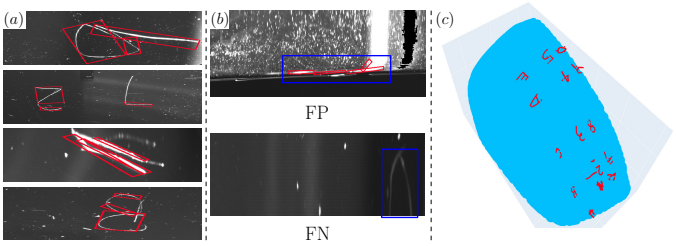

Fig. 17: Defects detection results conducted by our pipeline. (a) Satisfactory result. (b) Some failure cases. (c) The distribution visualization results via image mapping based on the detected defects

comparison in the image enhancement step; the Sobel operator [37] is chosen as a comparison in the edge detection; we also tune the parameters of the Canny detectors to see their influence on the detection accuracy.

Based on the ground truth provided by the professional workers, we implement the comparison with the baselines, shown in Table. IV, where the precision Ψ and recall χ are used to evaluate the performance.

$$\Psi = TP/(TP+FP), \quad \chi = TP/(TP+FN) \tag{24}$$

In Eq. 24, TP, FP and FN denotes the true positive, false positive and false negative respectively. Since we weigh the missing defects more than the wrong predictions, recall plays a more important role than precision in our experiment. Hence, the final accuracy is defined as a weighted sum of the precision and recall: $\tau = 0.2 \times \Psi + 0.8 \times \chi$. Although it involves some FP cases, our algorithm can minimize the probability of missing defects. Hence, our algorithm has the best performance compared with the references. Illustrated in Fig. 16(a), Sobel operator is so susceptible to the noise that it misses some defects occasionally. Although histogram equalization is good at improving the contrast of the image, the regions with defects are easily overexposed. As a double-threshold algorithm, the edge identification by Canny detectors depends on the minimum and maximum thresholds. When the thresholds are set as $(100, 200)$, the algorithm is able to detect the majority of defects, but it involves more FP cases compared with that of our final selection $(200, 300)$.

Other detection results obtained by our proposed algorithm is shown in Fig. 17. Fig. 17(a) displays the successful results. Note that the detection that captures only a part of the defects is also considered as the true positive samples. Besides, there are some failure cases shown in Fig. 17(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. 17(c) illustrates the defect visualization result via image mapping. Note that the displayed defects are selected 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. CONCLUSIONS

In this paper, we developed a new sensor-guided automatic defect inspection system for convex free-form specular surfaces. Mounted on a robotic manipulator, a high-resolution vision sensor is able to execute surface scanning in a flexible manner. Based on the prior mesh model of the workpiece, our K-means based region segmentation is robust to segment free-form surfaces to relative flat regions without troublesome tuning. Our path planning algorithm ensures the inspection completeness through the further patch segmentation and the analysis of the vision sensor model. The defect inspection is fully automatic through real-time projection registration feedback and image processing procedure. The experiment results illustrate that our sensor-based system is effective in automatic defect inspection and has superiority over the reference methods.

However, our proposed prototype exists some limitations. Firstly, the performance of the system is merely validated in the ideal laboratory environment. More tests with more workpieces under real manufacturing environments are required to improve the robustness of our system. Secondly, the design of the custom-built end-effector needs improvements to enlarge the scanning space of the image acquisition subsystem. Finally, the proposed image processing algorithm cannot classify unqualified defects at this stage. Further research can be conducted on the image processing regarding to the recognition and classification using deep neural network (DNN) [39].

REFERENCES

[1] C. Steger, M. Ulrich, and C. Wiedemann, *Machine vision algorithms and applications*. John Wiley & Sons, 2018.
[2] O. Duran, K. Althoefer, and L. D. Seneviratne, "State of the art in sensor technologies for sewer inspection," *IEEE Sensors journal*, vol. 2, no. 2, pp. 73–81, 2002.
[3] J. Gan, Q. Li, J. Wang, and H. Yu, "A hierarchical extractor-based visual rail surface inspection system," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7935–7944, 2017.
[4] A. Kumar and G. K. Pang, "Defect detection in textured materials using gabor filters," *IEEE Transactions on industry applications*, vol. 38, no. 2, pp. 425–440, 2002.
[5] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, "Automated defect inspection of led chip using deep convolutional neural network," *Journal of Intelligent Manufacturing*, vol. 30, no. 6, pp. 2525–2534, 2019.
[6] H. Freeman, *Machine vision for three-dimensional scenes*. Elsevier, 2012.
[7] 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.
[8] Q. Luo and Y. He, "A cost-effective and automatic surface defect inspection system for hot-rolled flat steel," *Robotics and Computer-Integrated Manufacturing*, vol. 38, pp. 16–30, 2016.
[9] Y.-J. Chen, J.-C. Tsai, and Y.-C. Hsu, "A real-time surface inspection system for precision steel balls based on machine vision," *Measurement Science and Technology*, vol. 27, no. 7, p. 074010, 2016.
[10] S. Akhtar, A. Tandiya, M. Moussa, and C. Tarry, "A robotics inspection system for detecting defects on semi-specular painted automotive surfaces," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8928–8934, IEEE, 2020.
[11] J. Molina, J. E. Solanes, L. Arnal, and J. Tornero, "On the detection of defects on specular car body surfaces," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 263–278, 2017.
[12] 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," *Journal of Manufacturing Systems*, vol. 53, pp. 93–108, 2019.
[13] E. Glorieux, P. Franciosa, and D. Ceglarek, "Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101843, 2020.
[14] Z. Les and M. Les, "Shape understanding system: understanding a convex object," *Journal of Electronic Imaging*, vol. 12, no. 2, pp. 327–342, 2003.
[15] S. Huo, D. Navarro-Alarcon, and D. Chik, "A robotic defect inspection system for free-form specular surfaces," in *IEEE International Conference on Robotics and Automation (ICRA) (in press)*, IEEE, 2021.
[16] D. Girardeau-Montaut, "Cloud compare—3d point cloud and mesh processing software," *Open Source Project*, 2015.
[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] A. M. Mehar, K. Matawie, and A. Maeder, "Determining an optimal value of k in k-means clustering," in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 51–55, IEEE, 2013.
[20] 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.
[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] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
[23] K. Litomisky, "Consumer rgb-d cameras and their applications," *Rapport technique, University of California*, vol. 20, p. 28, 2012.
[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 journal*, vol. 15, no. 11, pp. 6507–6516, 2015.
[25] 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.
[26] 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.
[27] A. Baker, *Matrix groups: An introduction to Lie group theory*. Springer Science & Business Media, 2012.
[28] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
[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.
[31] 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.
[32] M. W. Burke, *Image Acquisition: Handbook of machine vision engineering*, vol. 1. Springer Science & Business Media, 2012.
[33] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 501–508, IEEE, 2012.
[34] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015.
[35] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[36] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.
[37] I. Sobel, *Camera Models and Machine Perception*. Memo (Stanford artificial intelligence Laboratory), Department of Electrical Engineering, Stanford University.
[38] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.
[39] 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 Journal*, vol. 19, no. 23, pp. 11413–11423, 2019.