Planar Image-Space Trajectory Planning Algorithm for Contour Following in Robotic Machining

Wade R. MacMillan, Rishad A. Irani and Mojtaba Ahmadi

Department of Mechanical and Aerospace engineering, Carleton University, Ottawa, ON K1S 5B6, Canada.

Abstract

This paper proposes the *Planar Image-Space Trajectory* (PIST) planning algorithm for contour following using computer vision and a force control scheme. The PIST algorithm generates machining trajectories offset to a planar workpiece with an applied force direction so that a force controller can be utilized. The PIST algorithm starts with an image of the workpiece, corrects for the perspective of the camera, defines a contour following trajectory in image space, and then transfers the trajectory into the robot's workspace for playback. A case study is presented where the PIST algorithm is applied to deburring sheet metal parts for validation through experimentation. A comparison is performed between the PIST algorithm and CAD/CAM software using CAD data for path design and a laser scanner for localization. The PIST planner achieved a positional accuracy greater than 1.7 mm compared to the CAM/CAM accuracy of 1.2 mm. An admittance controller successfully accounted for the positional error to achieve satisfactory and equivalent surface finish results for both trajectory planning approaches. The PIST planner is a beneficial alternative for trajectory planning as it significantly reduced the setup and process time while adapting to deviations from the design dimensions and manufacturing imperfections

Keywords: Computer Vision, Trajectory Planning, Contour Following, Robotic Deburring, Image Processing, Planar Homography

1. Introduction

A ubiquitous task for an industrial robot in machining is tracking the periphery of a planar contour, present in operations such as robotic deburring, polishing and grinding. Currently, it is standard practice to program industrial robots through teach and playback methods where the robot is manually taught a finite series of waypoints which are replayed in operation. A skilled operator performs the teaching process manually, which can require a significant amount of setup time. For each unique contour, teaching must be repeated, making teach and playback impractical for high-mix, low-volume applications. The following paper develops an automated offline trajectory planner for designing planar contour following paths using computer vision and a force control scheme.

Path planning is the process of defining an ordered geometric path for a robot to follow. Trajectory planning is an extension of path planning, through adding time parameterization to the path. There is a significant amount of research on developing automated path and trajectory planners for programming industrial robots. The most prevalent automated planning method for robotic machining uses Computer-Aided Manufacturing (CAM) software to design a path relative to a CAD model of a workpiece [1]. In CAD/CAM software, the path is defined relative to the CAD model and is transformed into the robot's workspace based on the workpiece's location. A skilled operator can localize the workpiece manually, referencing a datum, or the robot can automatically localize the workpiece using a 3D sensor [2, 3, 4, 5] or a touch-off probe [6, 7, 8]. CAD/CAM software is inflexible to manufacturing imperfections or variations from the reference CAD model [9]. The additional localization step increases the total process time, complexity, and cost.

Computer vision can provide the necessary spatial information for path planning, removing the need for CAD data and a localization step. A field of research for contour tracking with computer vision is online visual servoing where a camera is used to provide online feedback on the location of the tool tip relative to the workpiece. The camera may be mounted on the end effector of the robot in an eye-in-hand configuration [10] or fixed in the workspace in an eye-to-hand configuration [11] . Eye-in-hand visual servoing has been implemented in deburring in Zhang et al. [12], and in robotic welding in Rout et al. [13]. Since an eye-in-hand configuration only obtains a local view of the trajectory, at complex curves or corners, local predictions may be incorrect and cause the robot to diverge from the surface [14]. Nakhaeinia et al. [14] combined eye-in-hand visual servoing with a global offline path planner to give a global reference of the contour.

Applications in robotic welding have shown that offline

path planners using computer vision have achieved sufficient accuracy for welding without any online feedback. Dinham et al. [15] used an eye-in-hand stereo vision system for offline path planning for robotic welding and achieved an accuracy within ±1 mm. In Micallef et al. [16] robotic welding path planning was performed using a single image, from an eye-in-hand monocular camera to weld planar workpieces. A single image only provides 2D spatial information. To identify the 3D path in the robot's workspace, Micallef et al. assumed the camera was parallel to the planar surface at a known distance. The perpendicular assumption has been made in Jinno et al. [17] and Shah et al. [18] as well, which impedes the accuracy of the generated path as misalignments in the placement of the camera is neglected. In Chang et al. [11], planar contour following was performed using a monocular camera with the planar surface localized relative to the camera. Chang et al. used planar homography to determine the depth of each point along the 3D path.

For high accuracy robotic machining applications, a force control scheme is often employed to improve edge tracking. Jinno et al. [17] used an eye-to-hand monocular camera to design deburring and chamfering paths on complex planar workpieces. The machining paths were offset to the detected workpiece to account for the radius of the deburring tool and a force control scheme was used to apply a force normal to the contour. Jinno et al. did not discuss the offset procedure or the selection of the force control direction. LeoPrincely et al. [19] and Lai et al. [20, 21] used a monocular eye-to-hand camera for deburring planar workpieces. In Lai et al. , the offset procedure was performed by exporting the detected path into CAD/CAM software as a series of linear segments and using the software for offsetting. The presented deburring papers did not address a solution for designing an offset path or applied force direction automatically.

The contribution of this paper is the development of the *Planar Image-Space Trajectory* (PIST) algorithm for contour following using computer vision and a force control scheme. The PIST algorithm is an offline trajectory planner aiming to eliminate the significant setup time required for manual teaching or CAD/CAM software. Computer vision is used to provide the necessary spatial information for path planning, eliminating the need for any pre-existing knowledge of the workpiece. The computer vision system used is a monocular camera in a fixed eye-to-hand configuration which provides global information of the workpiece. The novelty of the PIST algorithm is the simple and automated procedure used to define an offset trajectory relative to a captured image of a contour and enhance the concept of trajectory planning to include force trajectories as required by machining process. Planar homography is used to correct the perspective of the captured image to obtain a to-scale representation of the contour. Image processing techniques are utilized to detect the workpiece in the image and automatically define an offset path with an applied force direction. The paper is organized as follows. Section 2 introduces the planar trajectory planning problem and defines the notation for the paper. Section 3 presents the PIST algorithm for contouring planar surfaces. Section 4 describes a case study where the PIST algorithm is applied to deburring sheet metal parts. An experiment is performed to evaluate the PIST algorithm and compare it to a CAD/CAM software approach.

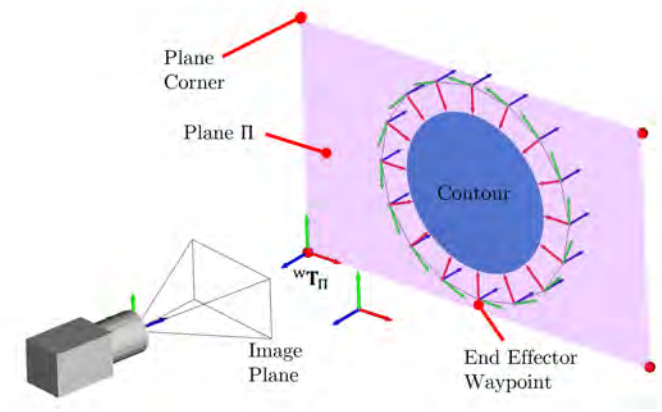

Fig. 1: The geometry of the PIST algorithm workspace and sample end effector waypoints.

2. Problem Definition

The PIST algorithm designs planar contour following trajectories for robotic machining using a monocular camera with the following assumptions:

- The monocular camera is calibrated and fixed at a known location.
- The contour is static and resides in a known plane.
- The camera has an unobstructed view of the contour, and provides an image with adequate resolution and quality to detect the contour.

The machining path designed by the PIST planner is around the periphery of the workpiece, using a cylindrical tool. The robot's end effector is assumed to be located at the center of the tool, therefore the path is offset to the contour to account for the tool radius. A force control scheme is used to improve the edge tracking accuracy of the robot, which requires the direction normal to the surface in which the applied force is controlled. Figure 1 displays the PIST algorithm workspace where a contour following path around a circle is generated. The output of the trajectory planner is a series of ordered waypoints $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, ...\mathbf{w}_n\}$ with spatial and timing information, which are interpolated between to generate the trajectory on the robot. The workspace waypoints are defined with the end effector position $^{w}\mathbf{p}_{ee}$, the end effector pose $^{w}\mathbf{R}_{ee}$ relative to the world frame, the force pose $^{w}\mathbf{R}_{f}$ relative to the world frame, fee-

drate v, and applied force F with,

$$\begin{aligned}\mathbf{w}_i = \{ &{}^{\mathrm{w}}\mathbf{p}_{\mathrm{ee,i}},\ {}^{\mathrm{w}}\mathbf{R}_{\mathrm{ee,i}},\ {}^{\mathrm{w}}\mathbf{R}_{f,i},\ F_i,\ v_i | \\ &{}^{\mathrm{w}}\mathbf{p}_{\mathrm{ee,i}} \in \mathcal{R}^3, \\ &{}^{\mathrm{w}}\mathbf{R}_{\mathrm{ee,i}} \in SO(3), \\ &{}^{\mathrm{w}}\mathbf{R}_{f,i} \in SO(3), \\ &F_i, v_i \in \mathcal{R}^1 \}.\end{aligned} \tag{1}$$

A point $\mathbf{p}$ or pixel $\mathbf{q}$ in reference to the coordinate frame a is denoted as ${}^{a}\mathbf{p}_{b}$ where b is a description of the point. In Fig. 1, the waypoints are represented as axes located at the end effector position ${}^{\mathrm{w}}\mathbf{p}_{\mathrm{ee}}$ with the force pose ${}^{\mathrm{w}}\mathbf{R}_{f}$ where the x axis (red) is directed normal to the contour.

The PIST algorithm uses a calibrated monocular camera for detection and planning. The pinhole camera model is used to describe the projection of point features ${}^{\mathrm{c}}\mathbf{p} \in \mathcal{R}^3$ relative to the camera onto pixels in an image $\mathbf{q} \in \mathcal{R}^2$ in homogeneous coordinates with,

$$\begin{bmatrix} {}^{\mathrm{I}}\mathbf{q} \\ 1 \end{bmatrix} = \lambda\ {}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}}\ {}^{\mathrm{c}}\mathbf{p}, \tag{2}$$

where ${}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}}$ is the camera intrinsic matrix of the form,

$${}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

which transforms points in the camera coordinate frame into pixels in image-space. In the equations (2) and (3), λ is a non-zero scaling factor to transfer into homogeneous pixel coordinates, f_x and f_y are the focal lengths in pixels, and (c_x, c_y) is the optical center of the image. The camera intrinsic matrix is determined when calibrating the camera using the procedure presented in Zhang et al. [22]. Point features ${}^{\mathrm{c}}\mathbf{p}$ are in reference to the camera coordinate frame, which is often not aligned with the world coordinate frame. To use equation (2) referencing the world coordinate frame, the points are first transformed into the camera coordinate frame using a rigid body transformation $\mathbf{T} \in SE(3)$ of the form,

$${}^{b}\mathbf{T}_{a} = \begin{bmatrix} {}^{b}\mathbf{R}_{a} & {}^{b}\mathbf{t}_{a} \\ 0 & 1 \end{bmatrix}. \tag{4}$$

with the notation that a mapping $\mathbf{M}$ from space a to space b is denoted as ${}^{b}\mathbf{M}_{a}$. The rigid body transformation is applied to the points in equation (2) as,

$$\begin{bmatrix} {}^{\mathrm{I}}\mathbf{q} \\ 1 \end{bmatrix} = \lambda\ {}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}} \begin{bmatrix} {}^{\mathrm{c}}\mathbf{R}_{\mathrm{w}} & {}^{\mathrm{c}}\mathbf{t}_{\mathrm{w}} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{w}}\mathbf{p} \\ 1 \end{bmatrix}, \tag{5}$$

where the transformation ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{w}}$ has its final row omitted to map from $\mathcal{R}^3$ to $\mathcal{R}^2$. The transformation ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{w}}$ is known as the camera extrinsic matrix.

The PIST algorithm creates trajectories around planar surfaces; therefore, the end effector position is constrained within a plane $\Pi \subset \mathcal{R}^3$. The projection of a planar surface onto an image results in an elegant simplification of equation (5). First, not that a rotation matrix is composed of three orthonormal unit vectors as,

$$\mathbf{R} = \begin{bmatrix} \hat{\mathbf{n}} & \hat{\mathbf{s}} & \hat{\mathbf{a}} \end{bmatrix}, \tag{6}$$

where $\hat{\mathbf{n}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{a}} \in \mathcal{R}^3$ are unit vectors directed toward the x, y and z axis of the rotated frame. By assigning a coordinate frame to the planar surface, restricting the projection to points on the plane ${}^{\Pi}\mathbf{p}$ and expanding the rotation matrix, equation (5) reduces to,

$$\begin{aligned} \begin{bmatrix} {}^{\mathrm{I}}\mathbf{q} \\ 1 \end{bmatrix} &= \lambda\ {}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}} \begin{bmatrix} {}^{\mathrm{c}}\hat{\mathbf{n}}_{\Pi} & {}^{\mathrm{c}}\hat{\mathbf{s}}_{\Pi} & {}^{\mathrm{c}}\hat{\mathbf{a}}_{\Pi} & {}^{\mathrm{c}}\mathbf{t}_{\Pi} \end{bmatrix} \begin{bmatrix} {}^{\Pi}\begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \\ 1 \end{bmatrix} \\ &= \lambda\ {}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}} \begin{bmatrix} {}^{\mathrm{c}}\hat{\mathbf{n}}_{\Pi} & {}^{\mathrm{c}}\hat{\mathbf{s}}_{\Pi} & {}^{\mathrm{c}}\mathbf{t}_{\Pi} \end{bmatrix} \begin{bmatrix} {}^{\Pi}\begin{bmatrix} x \\ y \end{bmatrix} \\ 1 \end{bmatrix} \\ &= {}^{\mathrm{I}}\mathbf{H}_{\Pi} \begin{bmatrix} {}^{\Pi}\mathbf{p} \\ 1 \end{bmatrix}. \end{aligned} \tag{7}$$

The projection from planar features onto the image is now performed with a 3×3 matrix ${}^{\mathrm{I}}\mathbf{H}_{\Pi}$ defined as,

$${}^{\mathrm{I}}\mathbf{H}_{\Pi} = \lambda\ {}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}} \begin{bmatrix} {}^{\mathrm{c}}\hat{\mathbf{n}}_{\Pi} & {}^{\mathrm{c}}\hat{\mathbf{s}}_{\Pi} & {}^{\mathrm{c}}\mathbf{t}_{\Pi} \end{bmatrix}. \tag{8}$$

Planar homography can be expanded to mappings between two planes or two images [23]. An image-to-image mapping is referred to as a perspective transformation. A general mapping between points from plane a onto plane b using homography is described as,

$$\begin{bmatrix} {}^{b}\mathbf{p} \\ 1 \end{bmatrix} = {}^{b}\mathbf{H}_{a} \begin{bmatrix} {}^{a}\mathbf{p} \\ 1 \end{bmatrix}. \tag{9}$$

A minimum of four known correspondences between points ${}^{a}\mathbf{p}$ and ${}^{b}\mathbf{p}$ are required to identify a homography matrix ${}^{b}\mathbf{H}_{a}$ using the *Direct Linear Transformation* (DLT) algorithm as described in Hartley et al. [23]. Planar homography is essential to the PIST algorithm, as it provides a mapping from features in an image onto a surface in the robot's workspace.

3. Planar Image-Space Trajectory Algorithm

The *Planar Image-Space Trajectory* (PIST) algorithm creates contour following trajectories for robotic machining that is offset to the workpiece with a known normal direction for a force control scheme. Figure 2 describes the PIST algorithm, which begins with a single image of a planar contour for tracking and outputs a series of waypoints. First, in Section 3.1, planar homography is used to transform the perspective of the image into a perpendicular, to-scale, view of the planar contour. Next, Section 3.2

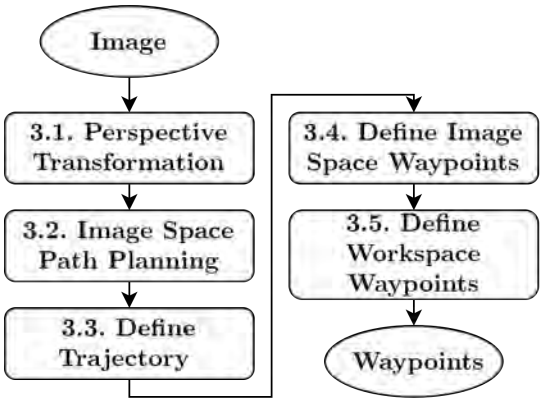

Fig. 2: Process of the PIST algorithm.

performs path planning in the to-scale image by detecting the contour of interest and planning an offset path and normal direction. Section 3.3 designs the feedrate along the planned path. Finally, Section 3.4 describes the trajectory in image-space with a minimal set of waypoints, which is transformed into the robot's workspace in Section 3.5.

3.1. Perspective Transformation

The geometry of the projection of a planar surface onto an image is shown in Fig. 3 in the left camera plane. The planar surface appears distorted due to the angled perspective of the original camera. With planar homography, the perspective of the camera is corrected to obtain a perpendicular perspective of the plane. The corrected image is a to-scale representation of the planar surface, in which measurements, positions, and paths in the image, can be scaled onto the surface. The homography matrix ${}^{\Pi}\mathbf{H}_{\mathrm{I}}$ is defined from the perspective of the camera, onto a corrected view perpendicular to the contour using the DLT method. The correspondences are from points on the plane projected into the first image ${}^{\mathrm{I}}\mathbf{q}$, onto their desired location in the second image ${}^{\Pi}\mathbf{q}$. To maintain uniform scaling, the correspondences are chosen so the points on the planar surface have the same shape as the corrected image. In Fig. 3 the correspondences are between the projection of four rectangular points on the plane onto image one, and the four corners of the corrected image of a similar shape to the rectangular pattern. In the transformed perspective, a path can be planned in image-space and then linearly scaled onto the planar surface in the workspace.

3.2. Image-Space Path Planning

The corrected perspective image provides a to-scale representation of the planar surface in which path planning is performed. The task of image space path planning is to detect the contour on the workpiece, and define an ordered list of pixels offset to the contour ${}^{\Pi}\mathbf{q}_{\mathrm{ee}}$ and an angle normal to the contour θ_f. The pixel path and angle provide spatial and directional information for the robot end effector to traverse. Figure 4 images A to D describes the

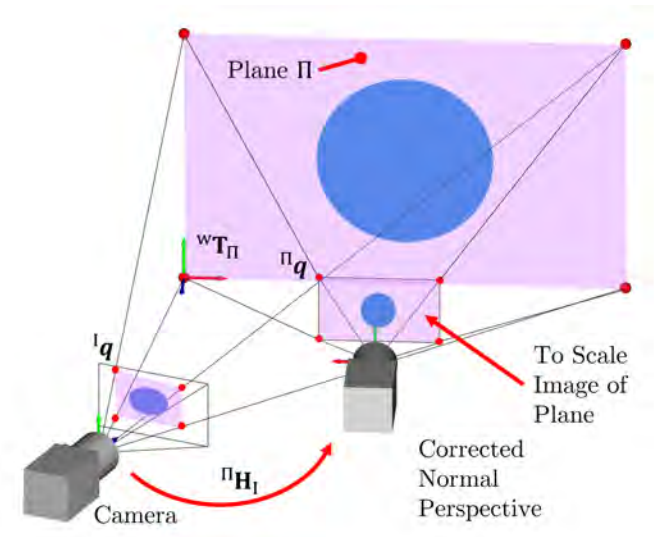

Fig. 3: Perspective correction using planar homography. The corrected perspective on the right is perpendicular to the plane and provides a to-scale representation of the surface.

four-step process for identifying a contour following path using image processing.

3.2.1. Identify the Mask Image

There are an assortment of established techniques for detecting an object in an image, such as thresholding by the pixel intensity, colour detection, blob detection, or background subtraction. The workpiece contour is detected in the image, and a binary mask image is produced which identifys the contour. Image A in Fig. 4 displays a sample binary mask image of an identified circle.

3.2.2. Distance Transform

The distance transform is a transformation taken on a binary image, which scales the intensity of each high pixel based on the distance to the nearest low pixel. Using the distance transform, Image B in Fig. 4 is obtained in which the pixel intensity describes the distance to the identified circle. Notice that the pixel intensity increases uniformly in the radial direction away from the center of the circle, while it is constant at a fixed radius away from the center.

3.2.3. Distance Transform Gradient

The spatial gradient of an image yields a vector at each pixel describing the directional rate of change of pixel intensity. The gradient is taken of the distance transform image to create a new image where the pixel intensity represents the angular orientation away from the contour, for designing the applied force direction. The spatial gradient is calculated through convolution with a discrete differentiation operator in the x and y direction. Derivatives in the x and y direction are converted into polar form to obtain the gradient magnitude and orientation. Image C in Fig. 4 displays the distance transform gradient orientation, where the pixel intensity describes the angle θ_{g} pointing

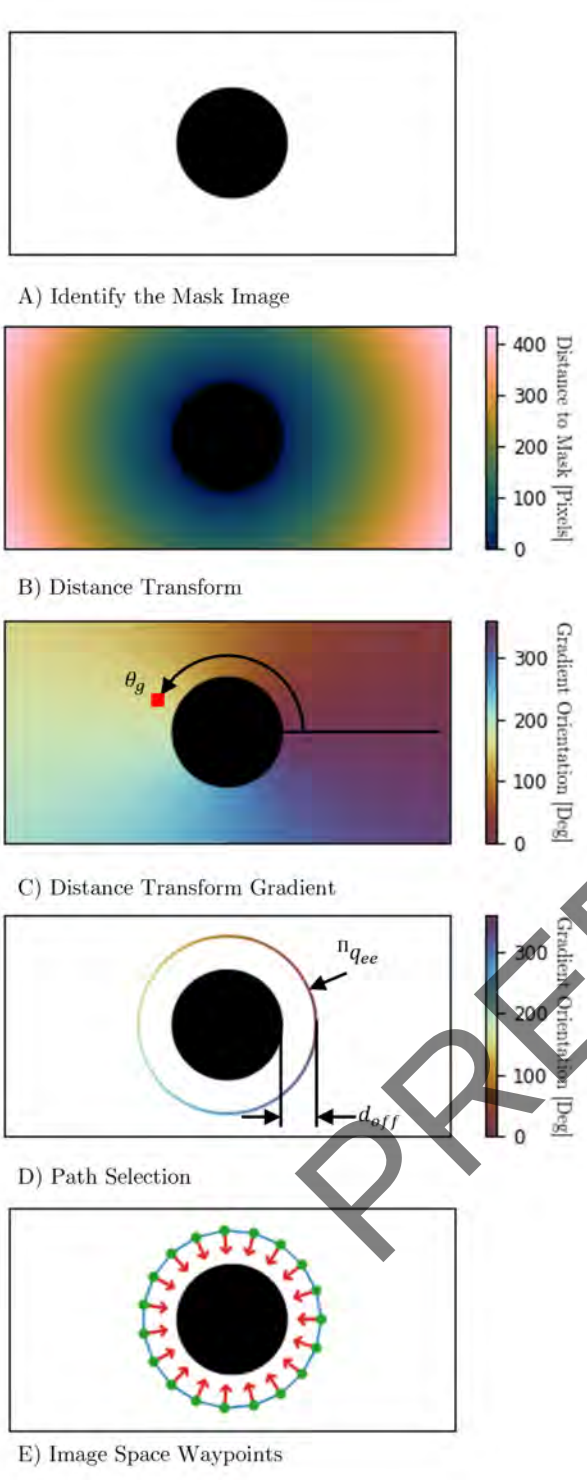

Fig. 4: Identifying an offset path in image-space using the distance transform and gradient. A) a binary image of a circle as an example of a detected object, B) the distance transform of the binary image, C) the gradient orientation of the distance transform and D) the selected path, offset to the mask with the gradient orientation, E) the selected image-space waypoints with position (green) and pose (red arrow) information, which are interpolated (blue) to define the path. The gradient orientation is an angle, in degrees from the x axis of the image.

normal to and away from the circle. Notice that the pixel intensity is constant in the radial direction from the center of the circle, while it increases in the counter clockwise direction at a fixed radius away from the center. The applied force orientation θ_{f} is the angle pointing towards the contour which is $\theta_{\mathrm{f}} = \theta_{\mathrm{g}} - 180$.

3.2.4. Path Selection

The PIST algorithm uses the distance transform and its gradient to select an offset path around the detected planar contour with a known normal direction. An offset mask is obtained by thresholding the distance transform at a specified offset distance d_{off} in pixel units. The border of the thresholded image is the desired offset path in image-space. The offset procedure is similar to the process presented in Xu et al. [24].

When offsetting at a constant distance d_{off} away from a mask, sharp external corners are projected into arcs as they maintain a constant distance away from the corner, while sharp internal corners are maintained. Sharp corners in the path should be avoided; to enforce a minimum corner radius r_{min} at all corners, offset outwards from the workpiece by $d_{\mathrm{off}} + r_{\mathrm{min}}$, then offset inwards by r_{min}.

Pixels along the border of the offset mask describe the end effector path and can be extracted with Canny Edge detection [25] or the boarder following algorithm presented in Suzuki et al. [26]. The direction normal to the workpiece is retrieved at each pixel in the path by referencing the gradient orientation. Results of the procedure are shown in image D of Fig. 4, where the offset path pixels ${}^{\Pi}\mathbf{q}_{\mathrm{ee}}$ are drawn next to the mask image. The pixel intensity describes the angle normal to the detected contour. The image-space path for the end effector is described by the ordered offset pixels ${}^{\Pi}\mathbf{q}_{\mathrm{ee}}$ and the applied force orientation θ_{f}. The path provides the geometric information which now requires timing information for the trajectory.

3.3. Define Trajectory

In machining applications, the resulting finish depends on the interaction between the tool and the workpiece. The end effector path is defined relative to the tool center, while material removal occurs at the outer radius. If the end effector trajectory has a constant feedrate, the observed feedrate between the tool and the workpiece will vary at complex curvatures in the path. The end effector feedrate is modulated to ensure the observed feedrate, and machining results, are constant. The expected contact point between the tool and the workpiece ${}^{\Pi}\mathbf{q}_{\mathrm{c}}$ is calculated by projecting each pixel ${}^{\Pi}\mathbf{q}_{\mathrm{ee}}$ in the force orientation θ_{f} direction by the offset distance d_{off} in pixel units with,

$$\begin{bmatrix} {}^{\Pi}\mathbf{q}_{\mathrm{c}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{2\times 2}(\theta_f) & \begin{bmatrix} d_{\mathrm{off}} \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{\Pi}\mathbf{q}_{\mathrm{ee}} \\ 1 \end{bmatrix}, \tag{10}$$

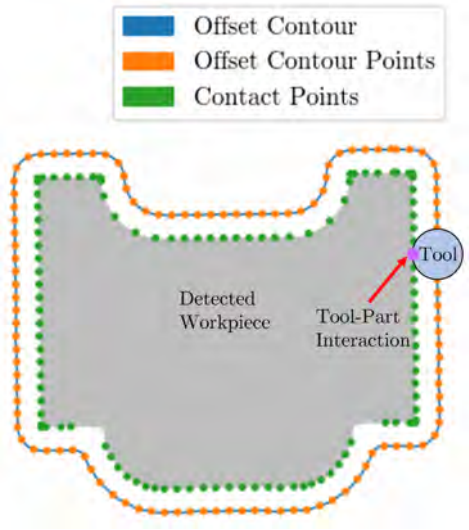

Fig. 5: Projected tool contact points. The points (orange) on the offset contour (blue) are projected in the force orientation direction to estimate the contact point (green). The upper internal arcs have a larger spacing between contact points than the lower external arcs.

where $\mathbf{R}_{2\times2}(\theta_f) \in SO(2)$ is a rotation matrix about θ_f. The image-space path and contact points for a complex part are displayed in Fig. 5. Pixel positions on the offset path are initially uniformly distributed and projected in the force orientation direction. The curvature in the contour, and consequently the force orientation, cause a non-uniform spacing of the contact points. The difference between the spacing between the contact points and the path positions, reveals the expected change in the observed feedrate relative to a nominal end effector feedrate v_{nom}. For compensation, The feedrate at the end effector v is scaled by the inverse of the expected change in the observed feedrate as,

$$v_i = \frac{v_{\text{nom}}}{|| \ {}^{\Pi}\mathbf{q}_{c,i} - {}^{\Pi}\mathbf{q}_{c,i-1}||}. \tag{11}$$

The results of the feedrate modulation is displayed in Fig. 6, the feedrate is reduced on internal corners and increased on external corners. Sharp external corners create very large feedrates, therefore the resulting v_i is saturated at double the nominal feedrate to prevent excessive velocities. The image-space trajectory data is now completely described by the position, orientation and feedrate at a series of pixel locations. The following section will export the trajectory data into a minimal set of waypoints.

3.4. Define Image-Space Waypoints

The task now is to generate a set of image-space waypoints from the derived trajectory data. Image-space waypoints are defined as,

$$\begin{aligned} {}^{\Pi}\mathbf{w}_i = \{ & {}^{\Pi}\mathbf{q}_{\text{ee,i}}, \ \theta_{f,i}, \ F_i, \ v_i | \\ & {}^{\Pi}\mathbf{q}_{\text{ee,i}} \in \mathcal{R}^2, \ \theta_{f,i} \in \mathcal{S}^1, \ F_i, v_i \in \mathcal{R}^1 \}, \end{aligned} \tag{12}$$

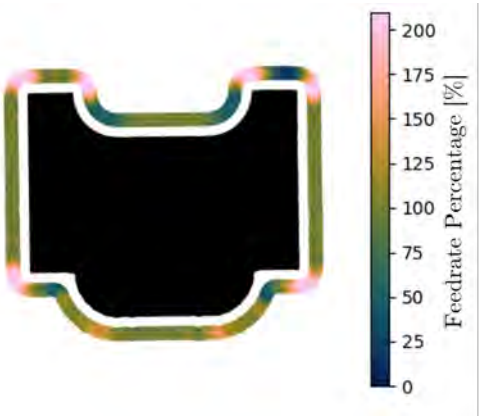

Fig. 6: Feedrate modulation along th eselected contour. The feedrate is shown as a percentage of the nominal feedrate.

where the applied force F along the path is specific to the contouring application. The position ${}^{\Pi}\mathbf{q}_{\text{ee}}$ and force orientation θ_f data is specified at each pixel along the image-space trajectory. Waypoints are initially chosen at each pixel location, evenly distributed based on the image resolution. To obtain a minimal set of waypoints, the waypoints are first re-sampled and then redundant waypoints are removed.

3.4.1. Re-Sampling Waypoints

The re-sampling factor is the ratio between the current pixel spacing and the desired waypoint spacing. The re-sampling ratio is converted to a fraction $\frac{\text{num}}{\text{den}}$, where the numerator is the up-sampling factor and the denominator is the down sampling factor. All waypoint data is up sampled, lowpass filtered at the cut-off frequency f_c, then down sampled. The cut-off frequency f_c is calculated from the Nyquist frequency f_n as,

$$f_c = \frac{f_n}{\max(\text{num}, \text{den})}. \tag{13}$$

3.4.2. Removal of Redundant Waypoints

Waypoints are interpolated between to obtain a trajectory. Waypoints along linear sections of a path with a constant force orientation are redundant. To identify redundant points, the derivative of the orientation θ_f is taken using a backwards difference approximation and waypoints are removed in areas with low rates of change. The remaining waypoints describe the image-space trajectory. Image E of Fig. 4 presents the image-space waypoints, which are next mapped into the robot's workspace.

3.5. Define Workspace Waypoints

The output of the PIST algorithm is a series of waypoints in the robot's workspace for the end effector to follow. The image-space waypoints from equation (12) are now mapped into the workspace waypoints described in equation (1). First, the end effector position ${}^{\text{w}}\mathbf{p}_{\text{ee}}$ is determined

by scaling the pixel path onto the plane in the workspace. Next, the force pose ${}^{\mathrm{w}}\mathbf{R}_{\mathrm{f}}$ is established using the force orientation. The feedrate v and force magnitude F are the same as previously specified in image-space. The end effector pose ${}^{\mathrm{w}}\mathbf{R}_{\mathrm{ee}}$ is specific to the robot configuration and its selection is purposefully excluded from the description.

3.5.1. End Effector Position

Features or contours detected in the corrected image are linearly scaled onto points on the workspace plane ${}^{\Pi}\mathbf{p}$ and then rigidly transformed into the workspace coordinate frame. The combination of a rigid transformation and scaling is performed using a similarity transformation $\mathbf{S} \in Sim(3)$ of the form,

$$ {}^{b}\mathbf{S}_{a} = \begin{bmatrix} {}^{b}\mathbf{R}_{a} & {}^{b}\mathbf{t}_{a} \\ 0 & s^{-1} \end{bmatrix}, \tag{14} $$

where $s \in \mathcal{R}^1$ is the scaling factor. The end effector position in image-space ${}^{\Pi}\mathbf{q}_{\mathrm{ee}} \in \mathcal{R}^2$ is mapped into the workspace R^3 with,

$$ \begin{bmatrix} {}^{\mathrm{w}}\mathbf{p}_{ee} \\ 1 \end{bmatrix} = {}^{\mathrm{w}}\mathbf{S}_{\Pi} \begin{bmatrix} \begin{bmatrix} {}^{\Pi}\mathbf{q}_{\mathrm{ee}} \\ 0 \end{bmatrix} \\ 1 \end{bmatrix}, \tag{15} $$

where the scaling factor is the ratio between the image size in pixels and the measured size of the detected plane.

3.5.2. Force Pose

The applied force orientation in image-space $\theta_{\mathrm{f}} \in \mathcal{S}^1$ can be transformed into a rotation matrix in the workspace with,

$$ {}^{\mathrm{w}}\mathbf{R}_{\mathrm{f}} = {}^{\mathrm{w}}\mathbf{R}_{\Pi}\mathbf{R}_{z}(\theta_{\mathrm{f}}). \tag{16} $$

where $\mathbf{R}_{\hat{\mathbf{n}}}(\theta) \in SO(3)$ is the rotation around the $\hat{\mathbf{n}}$ axis by θ. The rotation $\mathbf{R}_{z}(\theta_{\mathrm{f}})$ is the force orientation as a rotation matrix in the plane Π, and the rotation ${}^{\mathrm{w}}\mathbf{R}_{\Pi}$ transforms the force orientation matrix into the world coordinate frame. The force pose x axis ${}^{\mathrm{w}}\hat{\mathbf{n}}_{\mathrm{f}}$ now describes the direction for the robot to apply force onto the contour. The workspace waypoints are now completely described. Figure 1 displays the resulting waypoints around the circular contour with the end effector position and force pose displayed.

4. Case Study on Robotic Deburring

Within the following validation case study, the PIST algorithm is applied to deburring sheet metal parts. Burrs are sharp imperfections present on parts after manufacturing, which can compromise its functionality, onset fatigue, and present a safety risk when handling and in service. An industrial robot can remove minor burrs and imperfections by machining the periphery of the part with a cylindrical abrasive tool using a force control scheme to maintain consistent contact force and surface finish. In the deburring process, no large geometrical modifications are expected to be done on the workpieces, thus the combination of precision control with an appropriate applied force is adequate to achieve the final machining quality. Section 4.1 describes the testing setup, including the robot platform and computer vision system. Section 4.2 details the implementation of the PIST algorithm tailored for robotic deburring. The PIST algorithm is validated through a deburring experiment. The experiment procedure is detailed in Section 4.3 and the results are presented in Section 4.4.

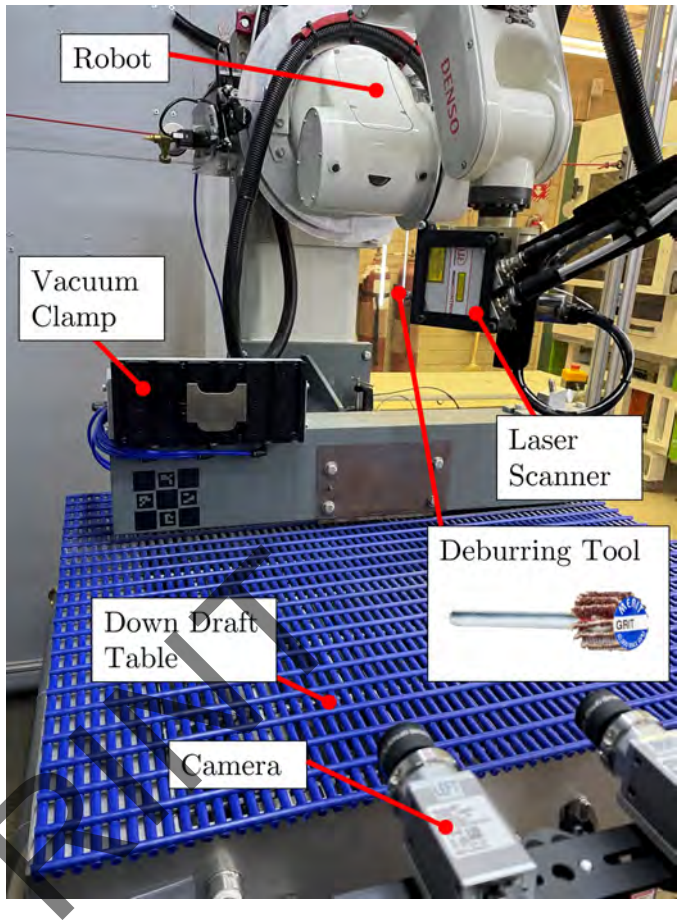

Fig. 7: Experimental setup used for all tests indicating the robot, the vacuum clamp, camera, and laser scanner locations.

4.1. Experimental Setup

The specialized robotic deburring platform is displayed in Fig. 7, which uses a 6 degree of freedom Denso VS SERIES 650 serial manipulator robot. Equipped to the robot's end effector is a Nakanishi NR-3060S spindle, ATI Gamma force-torque sensor and Micro-Epsilon scanCONTROL 2950 laser scanner as displayed in Fig. 8. The computer vision system comprises a Basler a2A3840-13gcPRO camera with an Edmund Optics Cr series 8.5 mm fixed focal length lens. A custom vacuum clamp is used to hold sheet metal parts in place for deburring. The vacuum clamp is localized relative to the robot with the laser scanner. The robot's tool frame and laser scanner frame have been locally calibrated around the clamp to a positional accuracy of $\pm.15$ mm.

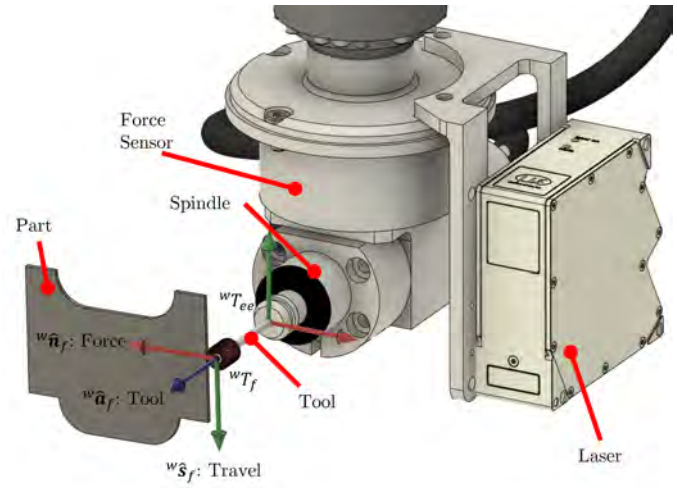

Fig. 8: Robot end effector and the interaction between the tool and part.

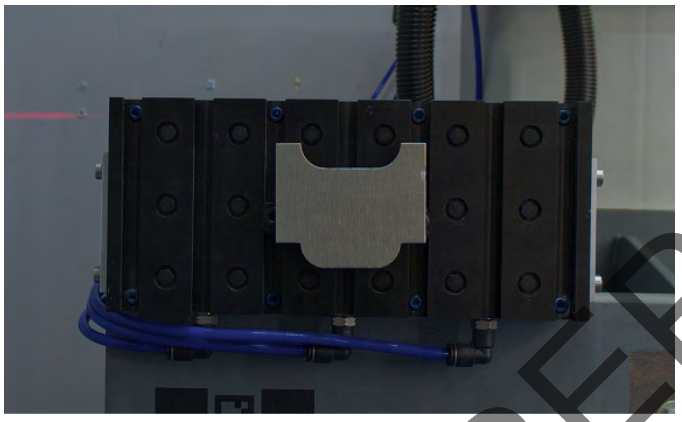

Fig. 9: Example image of a test part taken by the calibrated monocular camera. The test part is 3" × 2.5" in size.

Figure 9 contains an image taken from the camera of a test part for deburring. The camera is calibrated to correct for lens distortion and calculate the cameras intrinsic ${}^{\mathrm{I}}\mathbf{K}_{\mathrm{c}}$ and extrinsic ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{w}}$ parameters. The mean re-projection error after the camera calibration is 0.21 pixels which projects to 0.03 mm error in the clamping plane. Figure 10 outlines items of interest from the camera's perspective, such as the test part and vacuum clamp. The vacuum clamp is at a known location in the workspace with transformation ${}^{\mathrm{w}}\mathbf{T}_{\mathrm{cb}}$ attached to the bottom left corner of the clamp base. Planar sheet metal parts are placed on the vacuum clamp and therefore constrained to the clamping plane Π_{cp}, which is normal to, and offset in the clamp base z axis. An additional transformation ${}^{\mathrm{w}}\mathbf{T}_{\mathrm{cp}}$ describes the location of the clamping plane, with the same pose as ${}^{\mathrm{w}}\mathbf{T}_{\mathrm{cb}}$. The PIST algorithm was implemented in Python using OpenCV library for image processing and Scipy library for data processing.

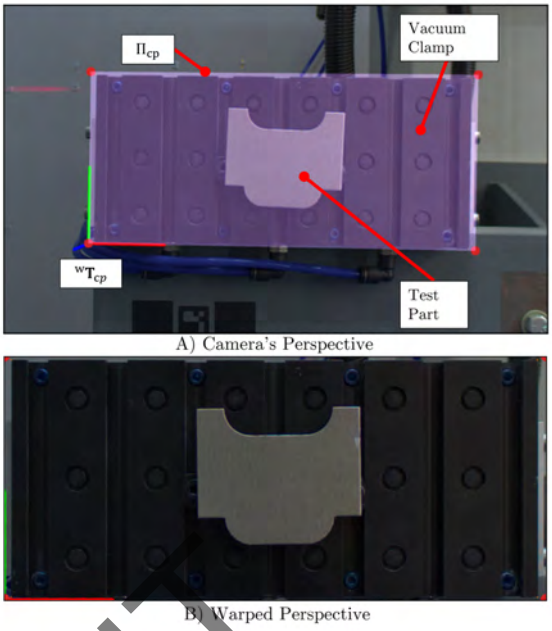

Fig. 10: Image A is the camera's perspective with the objects of interest identified. Image B is the transformed perspective perpendicular to and aligned with the clamping plane.

4.2. PIST Algorithm Implementation

The PIST algorithm is applied to robotic deburring to remove burrs present on sheet metal parts. The camera system is used to capture an image of the part for path detection. An example image is presented in Fig. 9 which is used to design the trajectory. Results of the image-space path planning procedure are displayed in Fig. 11.

The first step of the PIST algorithm is to use planar homography to transform the perspective of the camera into a perpendicular view of the clamping plane. The homography matrix between the image plane and clamping plane ${}^{\mathrm{cp}}\mathbf{H}_{\mathrm{I}}$ is calculated using the DLT method using the correspondences between the four corners of the clamping plane projected onto the original image and the four desired corners of the corrected image. The four corners of the clamping plane projected onto the original image is illustrated in Fig. 10 as red dots. The results of the perspective transformation, and then the following image-space path planning is presented in Fig. 11.

The applied force and end effector pose selection are excluded from the PIST algorithm description as they are process specific, therefore they are now introduced, tailored to robotic deburring.

4.2.1. Applied Force Selection

Force control ensures the tool remains in contact with the part by allowing deviations from the identified path. The minimum corner radius imposed in the path causes the tool to lose contact with the part at sharp internal corners. At sharp external corners, the observed feedrate v_c is zero,

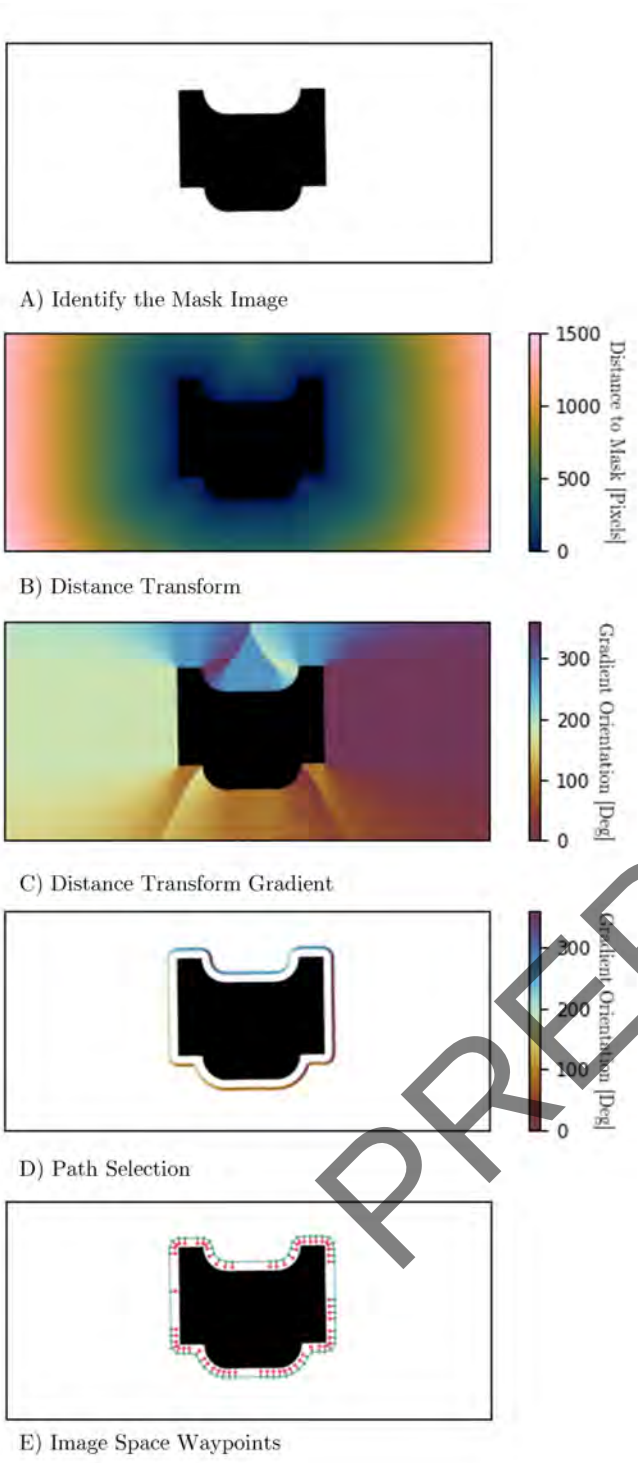

Fig. 11: See description for Fig. 4

causing the tool to dwell at a single point and round the corner; therefore, the applied force F_i is removed when contact is not expected or when at sharp external corners. To estimate if the tool should be in contact with the part, the Euclidean distance between each tool contact point ${}^{\Pi}\mathbf{q}_t$ and the detected foreground mask is referenced. The tool is estimated to be at a sharp external corner when

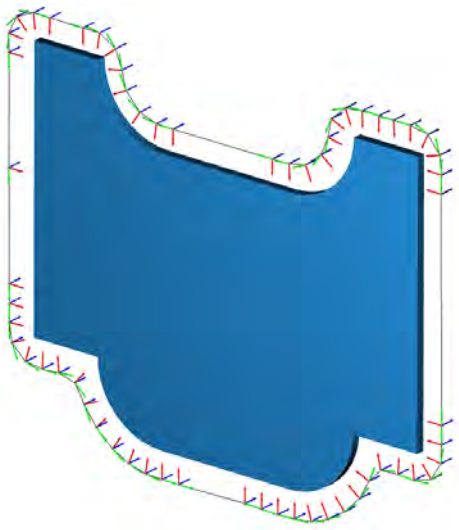

Fig. 12: PIST algorithm output waypoints with the end effector position and force pose in the workspace. The waypoints are coarsely spaced for visual purposes

the observed feedrate v_t is near zero. The selection of the applied force completes the description of the workspace waypoints from Section 3.4, which are mapped into the workspace to describe the trajectory of the end effector.

4.2.2. Workspace Waypoints

The workspace waypoints for deburring is defined from the PIST algorithm in Section 3.5. The end effector pose was excluded as it is process specific. Figure 12 displays the force pose and end effector position from the generated waypoints. The interaction between the robot and part is displayed in Fig. 8 for deburring with a cylindrical tool. The force pose x axis is the direction to apply force onto the part, the y axis is the direction of motion and the z axis is colinear to the tool. The end effector pose ${}^{\mathrm{w}}\mathbf{R}_{\mathrm{ee}}$ for deburring is simply selected to ensure the tool is perpendicular to the planar part. Material removal may occur anywhere around the circumference of the tool, so the end effector y axis is selected at a convenient direction for the robot [27]. With the end effector pose defined, the PIST algorithm is now completely implemented for experimentation with robotic deburring.

4.3. Experiment Procedure

The PIST algorithm is validated and compared to CAD/CAM software, developed with equivalent force and feedrate modulation. All tests were performed with an applied force of 2 N, an observed feedrate of 5 mm/s and a 3/8 in diameter flap wheel tool shown in Fig. 7. The part used for the tests are 3×2.5 in in size with minor defects.

The procedure for the PIST algorithm:

1. Place the test part in the workspace.
2. Obtain an image of the part on the vacuum clamp.

3. Generate the trajectory using the PIST algorithm. Note that the background model is pre-defined for every part on the vacuum clamp.
4. Perform the deburring trajectory.

The procedure for the CAM trajectory planner:

1. Place the test part in the workspace.
2. Plan a local deburring trajectory offline using CAM software.
3. Obtain a 3D point cloud of the test part with the laser on the end effector of the robot.
4. Fit the CAD file to the point cloud with an ICP algorithm [28].
5. Transform the local trajectory into the workspace based on the part localization.
6. Perform the deburring trajectory.

The CAM trajectory planning procedure introduces a new localization step performed with a laser scanner. In the current setup, the laser scanner required for localization is significantly more expensive than the camera for the PIST algorithm. There are potentially large cost and time savings when switching to the PIST algorithm, provided it can yield satisfactory results.

4.4. Results

The goal of the experimentation is to validate the PIST algorithm and see if it can provide comparable deburring results to the CAM approach, proving it is a viable alternative. Three test parts are deburred for each trajectory planning approach, each time at a new location on the vacuum clamp. The same tool was used for all tests, but the tool was indexed in the end effector z axis (${}^{w}\hat{\mathbf{a}}_{f}$ from Fig. 8) after each test to expose a fresh section of the tool. The results are displayed in Fig. 13 which displays the position error for each test against the distance travelled along the path and the surface finish is displayed in Fig. 14. Figure 15 provides a reference for the tool position and image location on the part based on the distance travelled. The metrics that the trajectory planners are evaluated on are, the position error in the detected path, the error in the applied force, the process time, and the part finish.

4.4.1. Detected Path Position Error

Admittance control is used to improve the edge tracking accuracy of the robot by admitting a positional error in the pre-defined path to maintain a set contact force. The positional accuracy of each trajectory planner is evaluated by measuring the tool's positional error, as it deviates from the defined path to contact the part. Figure Fig. 13 contains the positional error along the path for each trajectory planner. The maximum position error of the three CAM tests are 0.985, 0.967, and 1.148 mm. The max position error for the three PIST tests are 1.214, 1.205, and 1.688 mm. The path positional accuracy for both trajectory planning approaches were sufficient for satisfactory deburring with admittance control.

The accuracy reported is affected by the compliance and wear of the flap wheel tool, thus the actual accuracy of each trajectory planner is understated. The tool compliance is observed in the positional error of all tests between 0 mm and 25.4 mm as the tool initially contacts the part. The tool wear reduces the tool radius and increases the positional error as each test progresses. It is estimated that the tool wears 0.5 mm on each test based on a first order fit of the positional error. The positional accuracy was deemed to be the least important metric, as it does not directly affect the part finish when deburring with admittance control.

4.4.2. Applied Force Error

The applied force onto the part had a significant impact on the part finish. Figure 13 also displays the magnitude of the applied force along the path. Both trajectory planners maintained the set contact force of 2 N for deburring with similar trends in the applied force results. There are two minor differences between the trajectory planners' performance in controlling the applied force. The first difference is that the spikes in the applied force were slightly larger for the PIST tests than the CAM tests. With admittance control, a large change in the positional error causes a spike in the applied force. The PIST tests had lower accuracy and therefore larger changes in the position errors at corners, causing larger force spikes. The second difference occurs at the arc features. The PIST algorithm discretizes arcs into multiple linear segments, which reduces the positional accuracy and causes fluctuation in the positional error and consequently the applied force. The performance reduction is prevalent at $55-65$ mm and $90-100$ mm at the internal arcs. At the external arcs at $200-220$ mm and $245-265$ mm there are larger oscillations, displayed on both trajectory planners. Figure 14 displays close-up images of the test part before and after deburring, and Image 2 is taken at the external arc between $200-220$ mm. Before deburring there are prominent burr features at the external arcs which causes fluctuations the applied force while deburring. The differences in the applied force are minor and not reflected in the surface finish.

4.4.3. Process Time

All deburring tests required the robot to be in operation for 100 s. The CAM approach required the part to be localized in the workspace using the laser scanner which required an additional 20 s, thus increasing the process time by 20%. The PIST algorithm achieved a faster process time and essentially eliminated the setup time for testing as it does not require any pre-defined CAD information of the part or the design of a scanning path. The faster process time and setup time makes the PIST algorithm beneficial for automating deburring on a production line.

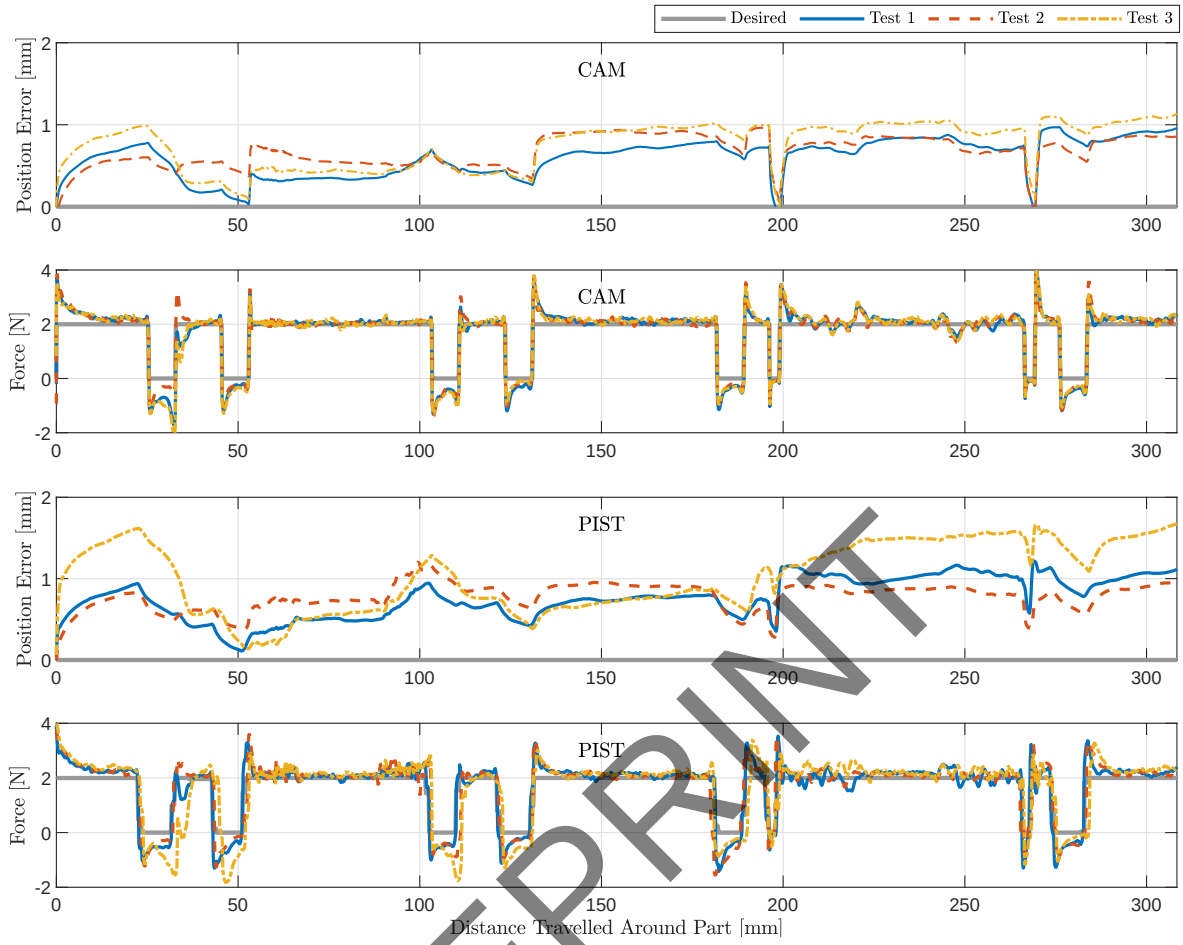

Fig. 13: The positional error and applied force along the deburring path for the CAM tests (above) and the PIST tests (below). All data displayed is low pass filtered at 5 Hz.

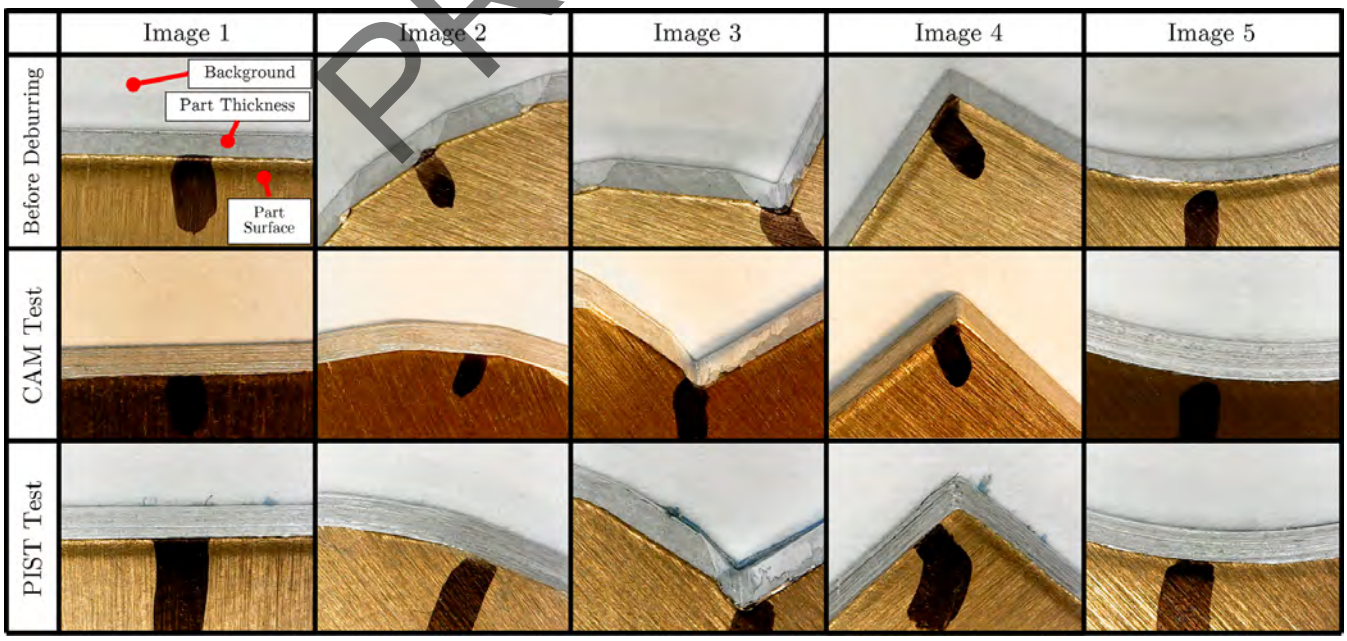

Fig. 14: Images of the surface finish before and after deburring for both trajectory planners. See Fig. 15 for image locations on the part.

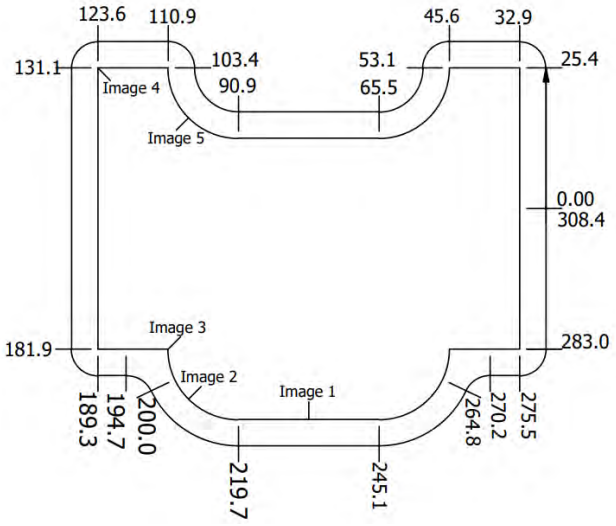

Fig. 15: Part position as the tool travels around the part in mm and the locations images are taken at in Fig. 14 .

4.4.4. Part Finish

Figure 14 displays close-up images of the parts edges before and after deburring to display the surface finish. Figure 15 provides a reference for the image location on the part. Using the PIST algorithm, the path is developed referencing the observed part, while the CAD/CAM approach uses an ideal model of the workpiece. If a large defect/burr is present, the PIST system will not identify the defect. The current experimentation shows that minor burrs and imperfections are adequately removed with the contour tracking trajectory provided by the PIST algorithm. Both trajectory planners were able to successfully remove burrs along the part and achieve a smooth and consistent surface finish at all accessible locations. At sharp internal corners, the tool cannot contact the part and the surface finish is untouched. At sharp external corners, the force is removed and the corners are not rounded.

With admittance control, the discrepancy in the positional accuracy and minor fluctuations in the applied force did not alter the surface finish. The surface finish between both trajectory planners are indistinguishable. The PIST algorithm is a viable alternative for robotic machining as it provided equivalent surface finish results, with the benefits of not requiring pre-emptive knowledge of the workpiece and adapting to manufacturing imperfections. Additionally, is has been shown that the PIST algorithm reduces the setup and processing time in absence of expensive infrastructure like the laser scanner.

5. Conclusion

This paper proposed the PIST algorithm for designing planar contour following paths for robotic machining using computer vision and a force control scheme. Path planning was performed entirely in image-space using the distance transform and its gradient to select an offset path with a known normal direction. The image space path was exported into a set of waypoints and mapped into the robot's workspace for playback. The PIST algorithm was validated with a robotic deburring experiment and obtained a positional accuracy within 1.7 mm and an acceptable surface finish. A comparison was performed with a CAM trajectory planner which provided a positional accuracy within 1.2 mm. The positional error reported was due to inaccuracies in the designed path, tool compliance and tool wear. With admittance control, the discrepancy in the positional accuracy was not reflected in the part finish. The PIST algorithm improved on the CAM planning method through a 20% faster processing time and virtually removing the setup time, in the absence of expensive infrastructure like the laser scanner. The PIST algorithm did not require a CAD model of the workpiece and can adapt to manufacturing imperfections.

Acknowledgment

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [grants RGPIN-2017-06967, RGPIN-2015-04169, and CRDPJ-514258-17]. The authors also acknowledge the financial and technical support from Arnprior Aerospace Inc. (AAI). The authors acknowledge the technical support and feedback of Steffan Lloyd and Reza Dehghanitafti, as well for their work in the development of the experimental platform and aid in testing.

References

[1] I. F. Onstein, O. Semeniuta, M. Bjerkeng, Deburring Using Robot Manipulators: A Review, 2020 3rd International Symposium on Small-Scale Intelligent Manufacturing Systems, SIMS 2020 (2020). doi:10.1109/SIMS49386.2020.9121490.

[2] R. Béarée, J. Y. Dieulot, P. Rabaté, An innovative subdivision-ICP registration method for tool-path correction applied to deformed aircraft parts machining, International Journal of Advanced Manufacturing Technology 53 (5-8) (2011) 463–471. doi:10.1007/s00170-010-2875-0.

[3] H. Kosler, U. Pavlovčič, M. Jezeršek, J. Možina, Adaptive Robotic Deburring of Die-Cast Parts with Position and Orientation Measurements Using a 3D Laser-Triangulation Sensor, Strojniski Vestnik/Journal of Mechanical Engineering 62 (4) (2016) 207–212. doi:10.5545/sv-jme.2015.3227.

[4] E. Villagrossi, C. Cenati, N. Pedrocchi, M. Beschi, L. Molinari Tosatti, Flexible robot-based cast iron deburring cell for small batch production using single-point laser sensor, International Journal of Advanced Manufacturing Technology 92 (1-4) (2017) 1425–1438. doi:10.1007/s00170-017-0232-2.

[5] M. Dehghani, R. A. Mckenzie, R. A. Irani, M. Ahmadi, Robotics and Computer-Integrated Manufacturing Robot-mounted sensing and local calibration for high-accuracy manufacturing, Robotics and Computer-Integrated Manufacturing 79 (April 2022) (2023) 102429. doi:10.1016/j.rcim.2022.102429. URL https://doi.org/10.1016/j.rcim.2022.102429

[6] K. M. Murphy, R. J. Norcross, F. M. Proctor, CAD directed robotic deburring, Proceedings of the second international symposium on robotics and manufacturing research, education, and applications, Albuquerque, NM (1988).

[7] H. C. Song, B. S. Kim, J. B. Song, Tool path generation based on matching between teaching points and CAD model for robotic deburring, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM (2012) 890–895doi:10.1109/AIM.2012.6265921.

[8] J. Hu, A. M. Kabir, S. M. Hartford, S. K. Gupta, P. R. Pagilla, Robotic deburring and chamfering of complex geometries in high-mix/low-volume production applications, IEEE International Conference on Automation Science and Engineering 2020-Augus (2020) 1155–1160. doi:10.1109/CASE48305.2020.9217042.

[9] A. Kuss, M. Drust, A. Verl, Detection of Workpiece Shape Deviations for Tool Path Adaptation in Robotic Deburring Systems, Procedia CIRP 57 (2016) 545–550. doi:10.1016/j.procir.2016.11.094.
URL http://dx.doi.org/10.1016/j.procir.2016.11.094

[10] J. Baeten, J. De Schutter, Hybrid vision/force control at corners in planar robotic-contour following, IEEE/ASME Transactions on Mechatronics 7 (2) (2002) 143–151. doi:10.1109/TMECH.2002.1011251.

[11] W. C. Chang, M. Y. Cheng, H. J. Tsai, Image feature command generation of contour following tasks for SCARA robots employing Image-Based Visual Servoing—A PH-spline approach, Robotics and Computer-Integrated Manufacturing 44 (2017) 57–66. doi:10.1016/j.rcim.2016.08.002.
URL http://dx.doi.org/10.1016/j.rcim.2016.08.002

[12] H. Zhang, H. Chen, N. Xi, G. Zhang, J. He, On-line path generation for robotic deburring of cast aluminum wheels, IEEE International Conference on Intelligent Robots and Systems (2006) 2400–2405doi:10.1109/IROS.2006.281679.

[13] A. Rout, B. B. Deepak, B. B. Biswal, Advances in weld seam tracking techniques for robotic welding: A review, Robotics and Computer-Integrated Manufacturing 56 (June 2017) (2019) 12–37. doi:10.1016/j.rcim.2018.08.003.
URL https://doi.org/10.1016/j.rcim.2018.08.003

[14] D. Nakhaeinia, P. Payeur, R. Laganiere, Adaptive robotic contour following from low accuracy RGB-D surface profiling and visual servoing, Proceedings - Conference on Computer and Robot Vision, CRV 2014 (2014) 48–55doi:10.1109/CRV.2014.15.

[15] M. Dinham, G. Fang, Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding, Robotics and Computer-Integrated Manufacturing 29 (5) (2013) 288–301. doi:10.1016/j.rcim.2013.01.004.
URL http://dx.doi.org/10.1016/j.rcim.2013.01.004

[16] K. Micallef, , G. Fang, , M. Dinham, Automatic Seam Detection and Path Planning in Robotic Welding, Robotic Welding, Intelligence and Automation (2011) 23–32doi:https://doi.org/10.1007/978-3-642-19959-2_3.

[17] M. Jinno, M. Uenohara, J. Oaki, K. Tatsuno, Teaching-less robot system for finishing workpieces of various shapes using force control and computer vision, IEEE International Conference on Intelligent Robots and Systems 1 (1999) 573–578. doi:10.1109/iros.1999.813065.

[18] H. N. M. Shah, M. Sulaiman, A. Z. Shukor, Z. Kamis, A. A. Rahman, Butt welding joints recognition and location identification by using local thresholding, Robotics and Computer-Integrated Manufacturing 51 (December 2017) (2018) 181–188. doi:10.1016/j.rcim.2017.12.007.
URL https://doi.org/10.1016/j.rcim.2017.12.007

[19] F. Leo Princely, T. Selvaraj, Vision assisted robotic deburring of edge burrs in cast parts, Procedia Engineering 97 (2014) 1906–1914. doi:10.1016/j.proeng.2014.12.344.
URL http://dx.doi.org/10.1016/j.proeng.2014.12.344

[20] Z. Lai, R. Xiong, H. Wu, Y. Guan, Integration of Visual Information and Robot Offline Programming System for Improving Automatic Deburring Process, 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018 (2018) 1132–1137doi:10.1109/ROBIO.2018.8665148.

[21] R. Xiong, Z. Lai, Y. Guan, Y. Yang, C. Cai, Local Deformable Template Matching in Robotic Deburring, 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018 (2018) 401–407doi:10.1109/ROBIO.2018.8665325.

[22] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11) (2000) 1330–1334. doi:10.1109/34.888718.

[23] A. Z. Richard Hartley, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, New York, 2000.

[24] K. Xu, Y. Li, B. Xiang, Image processing-based contour parallel tool path optimization for arbitrary pocket shape, International Journal of Advanced Manufacturing Technology 102 (5-8) (2018) 1091–1105. doi:10.1007/s00170-018-3016-4.

[25] J. Canny, A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (6) (1986) 679 – 698. doi:10.1109/TPAMI.1986.4767851.

[26] S. Suzuki, K. A. Be, Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics and Image Processing 30 (1) (1985) 32–46. doi:10.1016/0734-189X(85)90016-7.

[27] N. Asakawa, K. Toda, Y. Takeuchi, Automation of chamfering by an industrial robot; For the case of hole on free-curved surface, Robotics and Computer-Integrated Manufacturing 18 (5-6) (2002) 379–385. doi:10.1016/S0736-5845(02)00006-6.

[28] P. J. B. McKay, N. D., A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256. doi:10.1109/34.121791.