Introducing data analytics to the robotic drilling process

Toufik Al Khawli , Hamza Bendemra and Muddasar Anwar
Aerospace Research and Innovation Center, Khalifa University of Science Technology and Research, Abu Dhabi, United Arab Emirates
Dewald Swart
Research and Development, Strata Manufacturing PJSC, Al Ain, United Arab Emirates, and
Jorge Dias
Electrical and Computer Engineering, Khalifa University of Science Technology and Research, Abu Dhabi, United Arab Emirates

Abstract

Purpose – This paper presents a method for extracting the geometric primitives of a circle in a three-dimensional space from a discrete point cloud data set obtained by a laser stripe sensor. This paper aims to first establish a reference frame for the robotic drilling process by detecting the position and orientation of a reference hole on structural parts in a pre-drilling step, and second, to perform quality inspection of the hole in a post-drilling step.

Design/methodology/approach – The method is divided into the following steps: a plane is initially fitted on the data by evaluating the principle component analysis using singular value decomposition; the data points or measurements are then rotated around an arbitrary axis using the Rodrigues' rotation formula such that the normal direction of the estimated plane and the *z*-axis direction is parallel; the Delaunay triangulation is constructed on the point cloud and the confidence interval is estimated for segmenting the data set located at the circular boundary; and finally, a circular profile is fitted on the extracted set and transformed back to the original position.

Findings – The geometric estimation of the circle in three-dimensional space constitutes of the position of the center, the diameter and the orientation, which is represented by the normal vector of the plane that the circle lives in. The method is applied on both simulated data set with the addition of several noise levels and experimental data sets. The main purpose of both the tests is to quantify the accuracy of the estimated diameter. The results show good accuracy (mean relative error < 1 per cent) and high robustness to noise.

Research limitations/implications – The proposed method is applied here to estimate the geometric primitives of only one circle (the reference hole). If multiple circles are needed, an addition clustering procedure is required to cluster the segmented data into multiple data sets. Each data set represents a circle. Also, the method does not operate efficiently on a sparse data sets. Dense data are required to cover the hole (at least ten scans to cover the hole diameter).

Practical implications – Researchers and practitioners can integrate this method with several robotic manufacturing applications where high accuracy is required. The extracted position and orientation of the hole are used to minimize the positioning and alignment errors between the mounted tool tip and the workpiece.

Originality/value – The method introduces data analytics for estimating the geometric primitives in the robotic drilling application. The main advantage of the proposed method is to register the top surface of the workpiece with respect to robot base frame with a high accuracy. An accurate workpiece registration is extremely necessary in the lateral direction (identifying where to drill), as well as in the vertical direction (identifying how far to drill).

Keywords Data analytics, 3D point cloud

Paper type Research paper

1. Introduction

Aero structural parts make extensive use of drilled fastener holes for the ease of assembly and disassembly of panels and covers. A major criterion in drilling is the quality of the hole, as a low-quality one can negatively affect the production cycle and cause critical failures of the aircraft because of corrosion propagation, inappropriate sealing, local stresses, etc. It is investigated that 70 per cent of the airframe fatigue failures occurs at the joint parts between different structures, and 80 per cent of these failures take place around the fastener holes (Wang and Feng 2008). Drilling these holes is a challenging task because of the increasing number of fastener holes, as well as the usage of materials which are difficult to machine such as carbon fiber and titanium alloy (Yan and Chen 2015). These

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
45/3 (2018) 371–378
© Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-01-2018-0018]

This research was performed as part of the Aerospace Research and Innovation Center (ARIC) program which is jointly funded by STRATA Manufacturing PJSC (a Mubadala company) and Khalifa University of Science and Technology.

Received 29 January 2018
Revised 2 April 2018
Accepted 3 April 2018

challenges are causing the manual drilling to become time-consuming and labor-intensive, while simultaneously a good drilling quality is not guaranteed. To increase the efficiency in massive production, aircraft companies like Airbus and Boeing have been recently investigating, developing and deploying robotic drilling systems in their shop floor. High viability of these systems is because of intrinsic high flexibility, relatively lower investment and achievable good drilling quality (Olsson et al., 2010; Xu, 2008).

During the robotic drilling process, a set of reference holes are drilled in advance on the workpiece. Metrology systems like laser stripe sensors, photogrammetry and vision systems are then applied to perform measurements on these reference holes. The outcome is a set of images or point cloud data that are transformed into a computer-aided design (CAD) model. This model is used as a basis for providing the actual positions and orientations of these holes in the world coordinate system (Zhu et al., 2014). As a circular hole exists in the image or point cloud data, a circle detection algorithm is essential for the metrology system to recognize the circle itself and subsequently render tool positioning and alignment. One of the major parameters that is estimated from the CAD model is the surface-normal direction. It is a key parameter in robotic drilling process, as it greatly influences the verticality and the cylindricity of the fastener hole (Biao et al., 2015). There inevitably exists a certain dimensional tolerance in the measurements between the workpiece and its CAD model. Therefore, external global metrology systems such as laser trackers help providing some corrections to minimize the deviation between the actual position and the nominal position of the reference holes and the alignment mismatch of the drilling tip direction and the surface normal directions.

The topic of circle extraction has attracted a lot of attention in computer vision and visualization community. It has been applied to several industrial applications such as process and task automation, precision manufacturing and quality inspection. The task of circle detection usually consists of two main steps, namely, edge extraction and circle fit (Abramov et al., 2016). In literature, the solution to edge extraction problem can be classified into geometric models, statistical models or estimating the normal on sharp edges. In computational geometry, Pinchasi et al. (2006) derived a mathematical model that seeks empty convex holes in a planar point set. Hervias et al. (2014) proposed a method to find voids that builds the largest possible empty or almost empty polygons around them starting from local longest-edges in a Delaunay triangulation. Wei (2008) and Kalogerakis et al. (2009) have used approaches that are based on the calculation of an angle, as well as the distance between points and their neighbors in point cloud. Alternatively, the implementation of edge extraction techniques using robust statistics was investigated thoroughly. Fleischman et al. (2005) introduced a robust moving least-squares technique for reconstructing a piecewise smooth surface from a point cloud data set. This work was extended later by Daniels et al. (2008) who identified sharp features in a point-based model that return a set of smooth spline curves aligned along the edge. Bazazian et al. (2016) proposed a fast and precise method to detect sharp edge features by analyzing the eigenvalues of the covariance matrix that are defined by each point's k-nearest neighbors. Furthermore, previous related work has been done by estimating the normal on sharp edges. Gumhold et al. (2001) published a method to extract feature lines directly from a surface point cloud by assigning a penalty weight to each point that indicates the unlikelihood of a neighbor graph. In this method, a set of feature patterns is produced by extracting a sub-graph of the neighbor graph that minimizes the edge penalty weights. Weber et al. (2010) presented a technique for detecting sharp features on point-sampled geometry that is based on computing a Gauss map clustering on local neighborhoods to discard all points which are unlikely to belong to a sharp feature. Feng et al. (2014) proposed a method to efficiently extract planar surfaces from organized data by segmenting the point cloud into groups and using them as nodes to create a graph. On the graph, they used an agglomerative hierarchical clustering to merge nodes on the same plane.

The proposed paper presents a methodology to register the top surface of the workpiece with a high accuracy by using a laser profile sensor. This is vital for the pre-drilling steps such as positioning and alignment between the drill tool and the workpiece; and the post-drilling steps such as countersinking, inspection and riveting. The focus is to detect the reference hole, as well as the surface normal, from a three-dimensional point cloud data. The point cloud is acquired from a two-dimensional laser stripe sensor which is mounted on a six degrees of freedom robot. The movement of the robot allows generating point cloud measurements that are confined to a plane in three-dimensional space. The plane is estimated via principle component analysis (PCA). Then the Rodrigues' rotation formula is used to rotate the data such that the normal plane and the z-axis direction are parallel. After rotating the data, the three-dimensional data are projected onto a 2D plane to generate the Delaunay triangulation. After generating the triangulation, the area of the triangles is calculated and a confidence interval is estimated to segment the data points that lie exactly at the circular edge which will be preceded by a circle fit step.

The paper is organized as follows: in Section II, the steps of detecting a circle in a three-dimensional point cloud are discussed in detail. In Section III, the method is applied to simulated data where its z-coordinates are corrupted by three levels of noise. In Section IV, the method is applied to establish a reference frame for the experimental robotic drilling process and the results are presented. Finally, in Section V, the conclusions and future work are discussed.

2. Methodology

The circle detection method takes a point cloud data P of size $n_p \times 3$ and returns a circle with a center $\vec{\Theta}$, a diameter ϕ and an orientation denoted by a normal vector $\vec{n}$. The proposed methodology involves the following seven main steps:

2.1 Data acquisition

To demonstrate a better overview on every step and to simulate a workpiece plane, a generated three-dimensional point cloud data set of size 6,900 points containing a circular hole with a 3-mm diameter and a center positioned at (1, 1 and 0 mm) is first generated on a rectilinear grid on x-y plane at z = 0 mm.

Then, the set is transformed homogeneously by a translation vector T = [3, 4 and 5 mm] and a rotation of 32° around x-axis, 0° around y-axis and 20° around z-axis counter-clockwise, respectively. The transformed three-dimensional point cloud set P, in Figure 1(a), is defined by:

$$P = \left\{\left(P_i^x, P_i^y, P_i^z\right)\right\}_{i=1}^{n_p} \tag{1}$$

where P^x, P^y, P^z represent the relative position of the data with respect to a predefined world reference frame with Cartesian coordinates O and n_p is the total number of the point cloud data set.

2.2 Plane estimation

To detect the orientation of the plane, the normal is estimated by applying PCA. PCA aims to find the structure in the data that holds the most variance by calculating the covariance matrix. This covariance matrix is calculated according to the singular value decomposition (SVD) method (Lindsay, 2002). Basically, the plane normal is evaluated by performing the following three steps:

1 compute the centroid vector $\vec{\mu}$ of the 3D point cloud which is defined by:

$$\vec{\mu} = (\mu^x, \mu^y, \mu^z), \tag{2}$$

where,

$$\mu^\alpha = \frac{1}{n_p}\sum_{i=1}^{n_p} P_i^\alpha \quad \text{with} \quad \alpha \in [x, y, z]\cdot \tag{3}$$

2 compute matrix Y which is the translation of all the measurements data P from the centroid $\vec{\mu}$ to the origin according to:

$$Y = P - \vec{\mu}\ \cdot \tag{4}$$

3 apply SVD to the matrix Y which is determined according to:

$$Y = U\Sigma V^T, \tag{5}$$

where, U is a matrix of size $n_p \times n_p$, Σ is a diagonal matrix of size $n_p \times 3$ and V is a matrix of size $n_p \times 3$. The last column vector of V^T denotes the normalized plane normal $\vec{n}$, which is defined as:

$$\vec{n} = (n^x, n^y, n^z). \tag{6}$$

As a result, a plane with equation $\vec{n}P = d$ is estimated, as shown by the green plane in Figure 1(b).

2.3 Data transformation

To transform the data in such a way that the plane normal $\vec{n}$ and z-axis, represented by vector $\vec{k} = (0, 0, 1)$ are parallel, the angle of rotation θ and the arbitrary rotation vector $\vec{r}$ are calculated according to:

$$\vec{r} = \frac{\vec{n} \times \vec{k}}{|\vec{n} \times \vec{k}|} \quad \text{with} \quad \theta = \arccos\left(\vec{n}.\vec{k}\right)\cdot \tag{7}$$

Consequently, a quaternion $\vec{q}$ is expressed as follows:

$$\vec{q} = (q^w, q^x, q^y, q^z), \tag{8}$$

where, $q^w = \cos\left(\frac{\theta}{2}\right)$ and $q^j = \sin\left(\frac{\theta}{2}\right) r^j$ with $j \in [x, y, z]$.

After defining the quaternion, the rotation set P_{rot} of the point cloud P is obtained by applying the Rodrigues' formula:

$$P_{rot} = \left(I\cos(\theta) + \sin(\theta)K + (1 - \cos(\theta))K^2\right).Y \tag{9}$$

where,

$$K = \begin{bmatrix} 0 & -\hat{q}^z & \hat{q}^y \\ \hat{q}^z & 0 & -\hat{q}^x \\ -\hat{q}^y & \hat{q}^x & 0 \end{bmatrix}, \ \hat{q}^j = \frac{q^j}{\sqrt{q^x + q^y + q^z}}, \tag{10}$$

and I is the Identity matrix. P_{rot} is represented graphically by the blue points in Figure 2.

2.4 Delaunay triangulation

Because of the fact that the first principle component of the rotated data lies on the x-y plane, one can simply eliminate the z-coordinate. Therefore, the data dimension space is reduced from three-dimensional to two-dimensional space by projecting the data set $\hat{P} = \left(P_{rot}^x, P_{rot}^y\right)$ on the x-y plane as demonstrated in Figure 3(a).

At this stage, the Delaunay triangulation is applied to the reduced two-dimensional data set $\hat{P}$. The Delaunay method is a triangulation that is similar to the nerve of the cells in a Voronoi diagram. When applied to two-dimensional data set, the triangulation ensures that the circumcircle associated

Figure 1 (a) 3D plot of the point cloud that contains a circular hole shape and (b) the green plane represents the estimated plane of the data using the PCA method

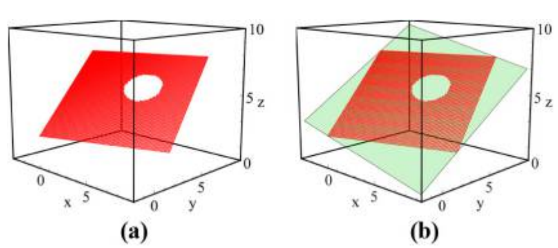


Figure 2 Rotation of the original data (red points) according to Rodrigues transformation. The rotated points are represented in blue

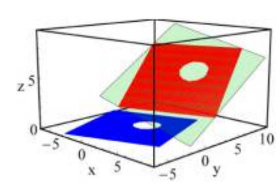
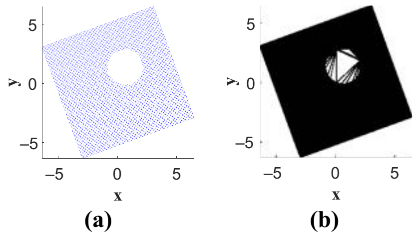


Figure 3 (a) Dimension reduction from 3D to 2D of the blue points shown in Figure 2 and (b) the corresponding Delaunay triangulation



with each triangle does not contain any other points in its interior (Okabe *et al.*, 1992). Therefore, it allows to distinguish how far or close the data points are positioned with respect to each other. The Delaunay triangulation for $\hat{P}$ is illustrated in Figure 3(b). As shown in the figure, the triangulation structure or the mesh contains multiple triangle edges and areas. Basically, the areas and edges of the triangles inside the circle are larger than the other triangles around the circle. This initiates the concept of detecting the circle by selecting the vertices of the triangles that have larger areas than the remaining ones. The concept is demonstrated in the following section.

2.5 Data segmentation

To detect the position of the center of the hole, a segmentation procedure is required to extract the data positioned at the circular edge. This is done by first calculating the area of all the triangles, second by estimating the confidence interval to filter out the vertices of the triangles with larger areas and third by segmenting the data points which are positioned at the circular edge by constructing the convex hull and filtering out the points that lie inside it. Basically, for every triangle containing the vertices $\hat{P}_A$, $\hat{P}_B$, and $\hat{P}_C$, the corresponding area is calculated according to the following equation:

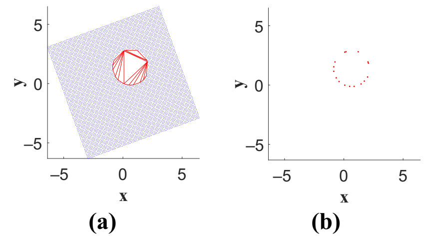
$$Area(\hat{P}_A, \hat{P}_B, \hat{P}_C) = 0.5 \times \left| \det \begin{bmatrix} 1 & 1 & 1 \\ \hat{P}_A^x & \hat{P}_B^x & \hat{P}_C^x \\ \hat{P}_A^y & \hat{P}_B^y & \hat{P}_C^y \end{bmatrix} \right|. \tag{11}$$

When the entire areas of the triangles of size n_t are considered as one data set $Areas = \{Area_j\}_{j=1}^{n_t}$ that is normally distributed around a mean value, μ_t and a standard deviation σ_t, a confidence interval (CI) can be calculated. Assuming that the data set is large, the 95 per cent CI is estimated using the following equation:

$$\mu_t - \frac{1.96\ \sigma_t}{\sqrt{n_t}} < CI(Areas) < \mu_t + \frac{1.96\ \sigma_t}{\sqrt{n_t}} \tag{12}$$

The 95 per cent CI is used as a criterion to classify the triangles with larger areas compared to the mean of the area. As shown in equation (12), the upper bound of the CI is equal to $\mu_t + \frac{1.96\ \sigma_t}{\sqrt{n_t}}$. As illustrated by the red triangles in Figure 4(a), any triangle area greater than the upper bound will be classified as triangle inside the hole.

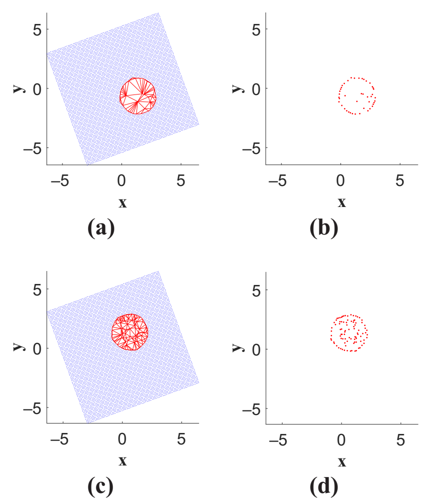
Figure 4 (a) Classification of the triangles using the upper bound (95 per cent) confidence interval and (b) the vertices of the classified triangles



The example demonstrated in Figure 4 is based on a data generated from a simulation model. However, it is important to mention that in experimental environments, the generated point cloud data set is affected by accuracy of the laser sensor, as well as noise. The noise is mainly caused by reflections of the laser profile especially when it scans shiny surfaces like aluminum. As a result, noisy data measurements are produced everywhere including inside the hole itself.

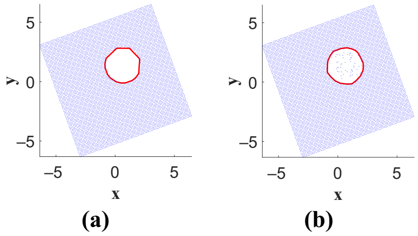
To make the proposed method robust with a noisy data that are located inside the hole, two randomly generated data sets with two different sizes (10 and 100 points) are appended inside the hole. After constructing the Delaunay triangulation and classifying the triangles that have larger areas than the upper bound, the results are illustrated in Figure 5. Figure 4(b) and Figure 5(b) and (c) show that the proposed circle detection method is able to filter out the vertices that lie on the circular edges, as well as the included noise. As the vertices that lie on the circular edge are the ones of interest, one can simply construct the convex hull and segment the vertices that lie on it as illustrated by Figure 6.

Figure 5 Adding artificial noise to the hole



Notes: The left column shows the classified hole triangles based on the confidence level. The right column shows the corresponding vertices of the classified triangles

Figure 6 Constructing the convex hull and segmenting the data points that lie on it:



Notes: (a) when no additional points are appended inside the hole; (b) when 100 randomly generated points are appended inside the hole

2.6 Circle estimation

Once the data points $\hat{S}$ that lie on the circular edge are segmented, a circle Θ_{rot} is estimated as shown in Figure 7. The circle estimation is based on applying the least square fit which aims to minimize the mean squared distance between the data points and the estimated circle (Gander *et al.*, 1996). The cost function C is defined by:

$$C = \min \sum_{i=1}^{n_{\hat{S}}} d_i^2 = \min \sum_{i=1}^{n_{\hat{S}}} \left(\sqrt{(x_i - a)^2 + (y_i - b)^2} - \phi \right)^2, \tag{13}$$

where, d_i is the Euclidean distance from points $x_i, y_i \in \hat{S}$ and the estimated circle which is parameterized by the center (a, b) and the radius ϕ. The resulting estimated circle center is defined by:

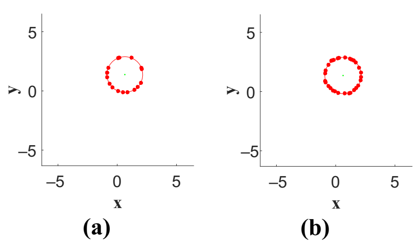
$$\overrightarrow{\Theta}_{rot} = (a, b, 0)\cdot \tag{14}$$

2.7 Results transformation

The final step is to transform the circle center back to the 3D space which maintains an orientation with the normal vector $\overrightarrow{n}$. This is achieved by:

$$\begin{aligned} Rot &= \left(I\cos(\theta) + \sin(\theta)K + (1 - \cos(\theta))K^2\right)^T\cdot \\ \overrightarrow{\Theta} &= Rot\overrightarrow{\Theta}_{rot} + \overrightarrow{\mu}\cdot \end{aligned} \tag{15}$$

Figure 7 Estimating a circle on the segmented data set $\hat{S}$ with a center (a,b) denoted by the green point for the data sets



Notes: (a) When no additional points are appended inside the hole; (b) when 100 randomly generated points are appended inside the hole. The red dots in the two figures denote the data set

The flow diagram of the proposed circle estimation method is summarized in Figure 8. The method takes a three-dimensional point cloud data set P as an input and generates the following outcomes:

- a circle with a center $\overrightarrow{\Theta}$;
- the diameter of the circle ϕ; and,
- the orientation with a normal vector $\overrightarrow{n}$.

3. Simulation results

In this section, the proposed method is applied to a simulated point cloud data, where the ground truth is easily labeled. The presence of ground truth allows to quantify the error when compared with the estimated center location, the diameter and the orientation. Additionally, in pursuance of quantifying the accuracy of the method in the presence of noisy data, the z values of the simulated point cloud are corrupted by four levels of additive Gaussian noise with a standard deviation σ_{noise} of 0, 10, 20 and 30 per cent. The preliminary results of the proposed method on the four generated data sets are shown in Figure 9.

The generated three-dimensional point cloud is shown on the left column, whereas the detected circle with its position in and the corresponding orientation are shown on the right. The absolute errors e_{Θ}^x, e_{Θ}^y, e_{Θ}^z, e_{ϕ}, e_{α}, e_{β}, e_{γ} are shown in Table I, representing the position errors in the x, y and z coordinates, the diameter errors and the orientation errors around x-, y- and z-axis, respectively.

The results show that the proposed method is not affected by noise and produce accurate results even under heavy noise conditions. The estimated position of the center has an accuracy less than 5 μm, diameter accuracy less than 18 μm and an orientation accuracy less than 0.2°. Additionally, the results show that increasing the noise in the z-direction deteriorates the accuracy of the estimated diameter. This occurs when some of the data points that lie on the circular edge become absorbed by the confidence interval according to equation (12).

To evaluate the proposed method performance in terms of time, the total computation time for the different steps is given in Table II which is a summation of the following:

- the time required to estimate the plane;
- time required to construct the Delaunay Triangulation; and
- time required to segment the points that lie on the circular edge.

The algorithm is implemented in MATLAB where the Delaunay triangulation is generated using the *Delaunay Triangulation* function. The runs are conducted on a computer with an Intel Core i7-6700 CPU of 2.60GHz and 16GB of RAM without any parallel processing step.

As shown in Table II, when the total number of the point cloud data set increases, the computational time increases. Additionally, more than 90 per cent of the total time of the proposed method is majorly consumed by the segmentation phase where the following steps take place:

- calculating the areas of the triangles;
- filtering the hole vertices according to the confidence interval upper bound; and
- segmenting the data points that lie on the circular contour.

Figure 8 The Flow diagram of the proposed method

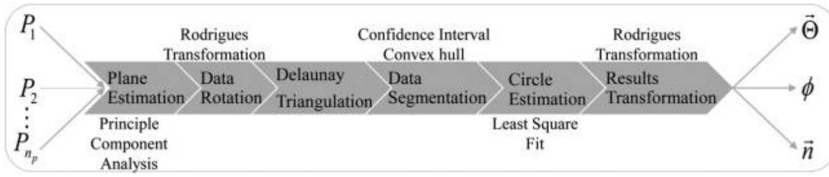
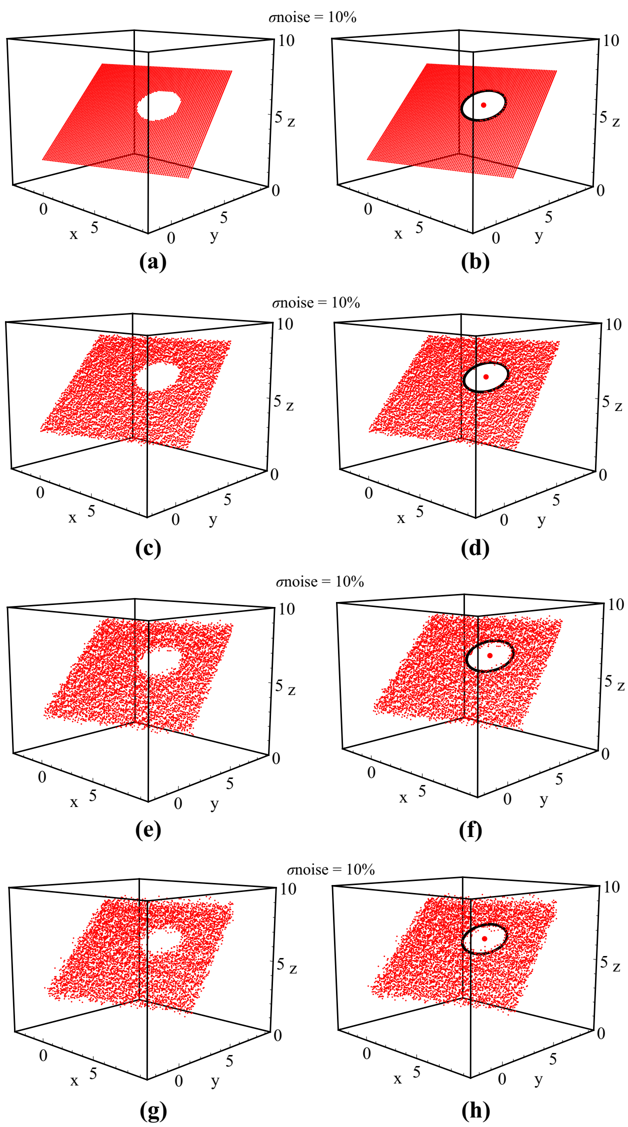


Figure 9 Four data sets where the *z*-values are corrupted by Gaussian noise



Notes: The left column shows the generated 3D point cloud. The right column shows the detected circle in its corresponding orientation

4. Experimental results

To achieve a high-quality control in the automated drilling process, the drilled holes need to be inspected with a high precision. This is done by first detecting the drilled holes and then measuring several quality criteria such as positioning accuracy, diameter, circularity and cylindricity. In this section, the integration of the circle detection method to a non-contact 3D robotic inspection system for measuring the diameter accuracy is presented. As shown in Figure 10(a), an AccuProfile 820-60 laser stripe sensor is mounted on a Mitsubishi RV-6SDL arm robot. The movement of the robot allows generating 3D point cloud measurements that are confined to a plane. As a circular hole exists in the point cloud data, the circle detection method is applied to recognize the circle itself and estimate its diameter. Therefore, the performance of the circle detection method is evaluated with an inclined composite workpiece that has test holes drilled by a 8.3 mm diameter tool as shown in Figure 10(a). The entire inspection process including robot manipulation, data acquisition and the circle detection method is controlled by an operating computer.

The inspection process including the laser sensor calibration and the camera calibration is currently still under investigation. The basic steps are summarized as following:

- acquire an image from a mounted camera on the robot;
- use computer vision techniques such as canny edge detection to detect a circle and estimate its center position in the world Cartesian coordinates;

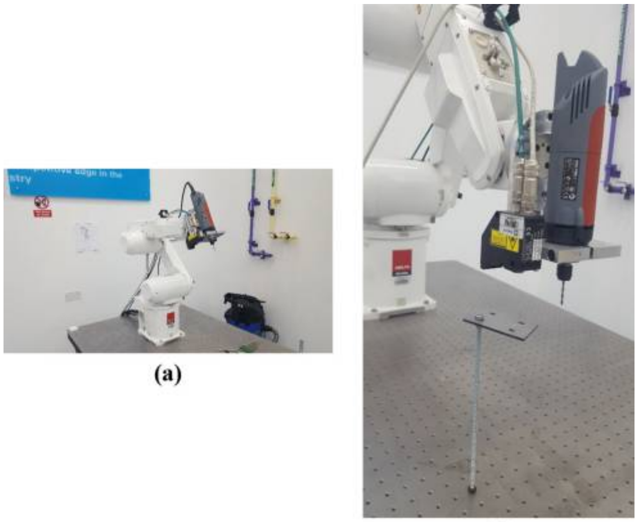
Table I Validation of the accuracy of the center e^x_Θ, e^y_Θ, e^z_Θ, diameter e_ϕ and orientation e_α, e_β, e_γ of the circle detection method using simulated data corrupted by Gaussian noise. The units of the absolute error are measured in μm for the center and diameter, and in degrees for the orientation

σ_{noise}	e^x_Θ	e^y_Θ	e^z_Θ	e_ϕ	e_α	e_β	e_γ
0%	3.70	2.98	0.01	7.7	2e-4	2e-4	6e-5
10%	4.32	4.51	0.53	15.3	0.04	0.04	2e-4
20%	3.91	4.32	0.75	15.6	0.05	0.03	0.06
30%	3.74	3.04	0.34	17.5	0.07	0.10	0.07

Table II The computation time of multiple data sets with different sizes n_p. The units are measured in seconds

n_p	416	1,572	6,733	9,498
Plane estimation	0.01	0.02	0.30	0.59
Triangulation	0.01	0.01	0.02	0.03
Segmentation	0.18	0.60	2.40	3.41
Total time	0.20	0.62	2.72	4.03

Figure 10 (a) The General view of the 3D robotic inspection system and (b) experiment configuration of scanning a hole on an inclined composite material



(a)

(b)

- manipulate the robot such that the tool center point of the laser stripe sensor is positioned on the center of the hole while taking into consideration the acceptable measuring range (between 60 and 120 mm for the AccuProfile 820-60);
- perform a scan while moving the robot in *x*-direction and generate a point cloud data as shown in Figure 11(a); and
- estimate the circle using the circle detection method as shown in Figure 11(b).

To evaluate the repeatability of the measured diameter accuracy, five measurements acquired from the 3D robotic inspection are compared with five measurements acquired from a commercial 3D coordinate measurement machine (CMM). The CMM is operated by a user who acquires data by navigating a probe manually to touch the inner circular contour of the drilled hole (at least with four different points). Once the data are generated, the Geomet software operated by the machine has a circle detection method which estimates the center position in a predefined home position, as well as the circle diameter. The estimated diameters for the 3D robotic inspection system and the CMM for every trial are listed in Table III.

Table III The Estimated diameters (in mm) of five trials of the robotic and CMM inspection system

Inspection	*Trial*1	*Trial*2	*Trial*3	*Trial*4	*Trial*5
Robotic	8.4728	8.4237	8.4213	8.4980	8.4884
CMM	8.3080	8.3103	8.3061	8.3094	8.30434

From these results, the corresponding means μ_R and μ_C and the standard deviations σ_R and σ_C for the measurements acquired from the robotic and CMM inspection systems, respectively, are calculated to estimate the repeatability of the two systems. For the robotic system, the estimated mean μ_R is equal to 8.461 mm and the standard deviation σ_R is equal to 0.032 mm. For the CMM system, the estimated mean μ_C mm is equal to 8.3076 mm, and the standard deviation σ_C is equal to 0.0024 mm. The measurement accuracy of the CMM is estimated to be 7 ± 2.4 μm, whereas the accuracy of the 3D robotic system is around 160 ± 32 μm.

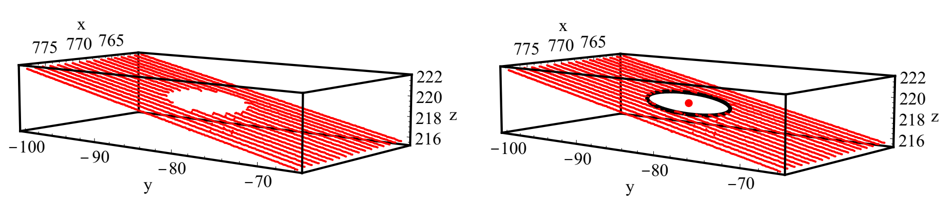
The main advantage of the robotic inspection system is achieving a higher flexibility of the inspection process. This is because of the gain of additional three degrees of freedom in the robotic system compared to the CMM inspection process. Another advantage is the speed up in the overall time required for the inspection using the robotic inspection process. The main reason for this speed up is that the process of inspection using the robotic method is automated, whereas the inspection using the CMM process requires manual operation. From the results, one disadvantage is the deterioration of the measured diameter accuracy. The main reason for this deterioration is caused by the tolerance in the accuracy of the robot, calibration of the laser sensor and the measuring procedure.

5. Conclusions

In this paper, a circular contour extraction method from an unorganized 3D point cloud is presented. Starting with an acquired point cloud data set, the proposed method is summarized by incorporating the following steps:

- estimating a plane by SVD to determine the arbitrary rotation vector and angle of rotation required for Rodrigues' transformation;
- rotating the data;
- generating the Delaunay triangulation to extract the hole vertices by selecting the areas of the triangle that are larger than the upper bound of the confidence interval;
- segmenting the points that lie on the convex hull of the extracted vertices; and

Figure 11 (a) 3D Plot of the point cloud of size of 4531 and (b) the detected circle is centered at $\vec{\Theta}$ =(769.033 mm, −83.632 mm and 218.781 mm) with a diameter ϕ = 8.579 mm and an orientation $\vec{n}$ =(0.01, 0.18 and 0.98). The time required to detect the circle was around 1.5 s



(a)

(b)

- estimating the circle then rotating the results back to the original orientation.

The results show significant accuracy of the detected hole in virtual environment (simulation) with high robustness against large noise. Additionally, the results show good repeatability in real (experimental) environments. The method can be applied to several industrial applications such as automation, precision manufacturing and quality inspection where hole detection is required. Further progress will focus on first analyzing and comparing the circle detection step by using Delaunay technique with several edge segmentation algorithms and second extending the algorithm to perform multiple circle detection. Additionally, the authors are currently focusing on implementing different solutions to improve the robotic inspection system accuracy (<50 μm) such as improving the laser calibration accuracy, evaluating different data acquisition procedure and using global metrology such as laser trackers.

References

Abramov, N.A., Bolsunovskaya, M.V., Leksashow, A.V. and Barinov, D.S. (2016), "Algorithms for detecting and tracking of objects with optical markers in 3d space", *IEEE International Conference on Soft Computing and Measurements (SCM), St. Petersburg*, pp. 362-364.

Bazazian, D., Casas, J.R. and Ruiz-Hidalgo, J. (2016), "Fast and robust edge extraction in unorganized point clouds", *Digital Image Computing: Techniques and Applications (DICTA), Int. Conference IEEE*, pp. 1-8.

Biao, M., Weidong, Z., Guorui, Y. and Yinglin, K. (2015), "A new elliptic contour extraction method for reference hole detection in robotic drilling", *Pattern Analysis and Applications*, Vol. 18 No. 3, pp. 695-712.

Daniels, J., Ochotta, T. and Ha, L.K. (2008), "Spline-based feature curves from point sampled geometry", *The Visual Computer*, Vol. 24 No. 6, pp. 449-462.

Feng, C., Taguchi, Y. and Kamat, V. (2014), "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6218-6225.

Fleischman, S., Cohenor, D. and Silva, T. (2005), "Robust moving least-squares fitting with sharp features", *ACM Transactions on Graphics*, Vol. 24 No. 3, pp. 37-49.

Gander, W., Golub, G.H. and Strebel, R. (1996), "Least squares fitting of circles and ellipses", *Bulletin of the Belgian Mathematical Society*, Vol. 34 No. 4, pp. 558-578.

Gumhold, S., Wang, X. and Mcleod, R. (2001), "Feature extraction from point clouds", *Proceedings of 10th International Meshing Roundtable*.

Hervias, C., Hitschfeld-Kahler, N., Campusano, L.E. and Font, G. (2014), "On finding large polygonal voids using delaunay triangulation, the case of planar point sets", in Sarrate, J. and Staten, M. (Eds), *Proceedings of the 22nd International Meshing Roundtable, Springer, Cham*, pp. 275-292.

Kalogerakis, E., Nowrouzezahrai, D., Simari, P. and Singh, K. (2009), "Extracting lines of curvature from noisy point clouds", *Journal Computer-Aided Design archive, USA*, Vol. 41 No. 4, pp. 282-292.

Lindsay, S. (2002), "A tutorial on principal component analysis", available at: http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

Okabe, A., Boots, B. and Sugihara, K. (1992), *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, Wiley, New York, NY*.

Olsson, T., Haage, M., Kihlman, H., Johansson, R. and Nilsson, K. (2010), "Cost-efficient drilling using industrial robots with high bandwidth force feedback", *Robot Cim-Int Manuf*, Vol. 26 No. 1, pp. 24-38.

Pinchasi, R., Radoicic, R. and Sharir, M. (2006), "On empty convex polygons in a planar point set", *J. Comb. Theory, Ser. A*, Vol. 113 No. 3, pp. 385-419.

Wang, L. and Feng, T. (2008), "Application of digital automatic drillriveting technology in aircraft manufacture", *Aeronautical Manufacturing Technology*, Vol. 11, pp. 42-45.

Weber, C., Hahmann, S. and Hagen, H. (2010), "Sharp feature detection in point clouds", *Shape modelling international conference, IEEE*, pp. 175-186.

Wei, S. (2008), "Building boundary extraction based on lidar point clouds data", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII*, pp. 157-161.

Xu, G. (2008), "Automatic assembly technology for large aircraft", *Acta Aeronautica et Astronautica Sinica*, Vol. 29, pp. 734-740.

Yan, Q. and Chen, W. (2015), "Automatic modification of local drilling holes via double pre-assembly holes", *World Journal of Engineering and Technology*, Vol. 3 No. 3, pp. 191-196.

Zhu, W., Mei, B. and Yan, G. (2014), "Measurement error analysis and accuracy enhancement of 2d vision system for robotic drilling", *Robotics and Computer Integrated Manufacturing*, Vol. 30 No. 2, pp. 160 -171.

Corresponding author

Toufik Al Khawli can be contacted at: toufik.alkhawli@kustar.ac.ae