



HAL
open science

Three-dimensional point cloud analysis for automatic inspection of complex aeronautical mechanical assemblies

Hamdi Ben Abdallah, Jean-José Orteu, Igor Jovančević, Benoît Dolives

► To cite this version:

Hamdi Ben Abdallah, Jean-José Orteu, Igor Jovančević, Benoît Dolives. Three-dimensional point cloud analysis for automatic inspection of complex aeronautical mechanical assemblies. *Journal of Electronic Imaging*, 2020, 29 (04), pp.041012-1-041012-22. 10.1117/1.JEI.29.4.041012. hal-02557867

HAL Id: hal-02557867

<https://imt-mines-albi.hal.science/hal-02557867>

Submitted on 29 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Three-dimensional point cloud analysis for automatic inspection of complex aeronautical mechanical assemblies

Hamdi Ben Abdallah,^{a,b} Jean-José Orteu,^a Igor Jovančević,^{b,*} and Benoit Dolives^b

^aUniversité de Toulouse, Institut Clément Ader, CNRS, IMT Mines Albi, INSA, UPS, ISAE, Albi, France

^bDIOTASOFT, Labège, France

Abstract. We present a robust approach for detecting defects on an aircraft electrical wiring interconnection system in order to comply with the safety regulations such as the forbidden interference and allowed bend radius of cables and/or harness in mechanical assemblies. For this purpose, we exploit 3-D point clouds acquired with a 3-D scanner and the 3-D computer-aided design (CAD) model of the assembly being inspected. Our method mainly consists of two processes: an offline automatic selection of informative viewpoints and an online automatic treatment of the acquired 3-D point cloud from said viewpoints. The viewpoint selection is based on the 3-D CAD model of the assembly and the calculation of a scoring function, which evaluates a set of candidate viewpoints. After the offline viewpoint selection is completed, the robotic inspection system is ready for operation. During the online inspection phase, a 3-D point cloud is analyzed for measuring the bend radius of each cable and its minimum distance to the other elements in the assembly. For this, we developed a 3-D segmentation algorithm to find the cables in the point cloud, by modeling a cable as a collection of cylinders. Using the segmented cable, we carried out a quantitative analysis of the interference and bend radius of each cable. The performance of the inspection system is validated on synthetic and real data, the latter being acquired by our precalibrated robotic system. Our dataset is acquired by scanning different zones of an aircraft engine. The experimental results show that our proposed approach is accurate and promising for industrial applications.

Keywords: robotic inspection system; CAD model; viewpoint selection; cable segmentation; 3-D point cloud analysis; nonrigid/flexible part.

1 Introduction

In the late 1980s, aircraft electrical wiring interconnection system (EWIS) safety concerns became an issue due to aircraft accidents. According to the Federal Aviation Administration, investigations found common degrading factors in aircraft EWIS to be the cause of some accidents. These findings led to extensive investigations of wiring, done by different industries and agencies.

There are many different factors that play a role in wiring degradation, such as: (i) interference between a cable and another element: e.g., if a cable is close to some hot elements, high temperatures can cause melting and burning. (ii) Bend radius: a cable can get damaged if it is bent too much.

Today, these defects are detected by visual and tactile inspection using instruments such as boroscopes. On the other hand, wiring inspection is a difficult task for a human operator. Indeed, the human operators have implicit limits like difficulties in following a cable trajectory and low repeatability rate.

*Address all correspondence to Igor Jovančević, E-mail: igorjovan@gmail.com

Therefore, in order to automate the inspection process and to reduce the human error, many aerospace companies aim to automate numerous and complicated operations of quality control of aircraft EWIS, staying complied with the various security requirements.

In this paper, we present a robust approach for detecting defects on aircraft EWIS in order to comply with the growing amount of safety regulations such as forbidden interference between elements and allowed bend radius of cables and/or harness in mechanical assemblies. Our method mainly consists of two processes: automatic selection of informative viewpoints (offline preparation of the inspection), and second, automatic treatment of the acquired 3-D point cloud from said viewpoints (online inspection process).

First, offline automatic selection of informative viewpoints is performed. The proposed method exploits the 3-D model of the assembly and it consists of two steps: generating candidate viewpoints sampled on a visibility sphere and constructing scoring functions, which evaluate each candidate viewpoint. The viewpoint selection method will not be described in this paper, as it has been detailed in our recent paper.¹

Second, after the viewpoints have been selected, the robotic inspection system is ready for operation. The 3-D scanner mounted on the robot acquires a one-shot 3-D point cloud. Then our algorithm analyzes the 3-D point cloud for measuring the bend radius of a cable and its minimum distance to the other elements in the assembly. For this purpose, we focus on developing a segmentation algorithm to find the cables in the point cloud received from the 3-D scanner and to reconstruct them as collections of cylinders of varying radius, length, and orientation. First, we propose a fully automated registration process that allows for alignment of the 3-D point cloud data with the 3-D computer-aided design (CAD) model. The resulting process encompasses two preprocessing steps: noise filtering and data sampling. Our point cloud filtering algorithm examines local neighborhoods to identify and remove outliers. After the preprocessing stage, a two-step global-to-local registration is applied. A keyframe-based SLAM algorithm is employed to get an initial estimate for our camera pose. The result of the global registration is further refined by a local registration process that uses the iterative nonlinear ICP method and Levenberg–Marquardt algorithm. Next, after the registration process, we propose an original method for cable segmentation and reconstruction based on cylinder fitting. Finally, using the segmented cable, we can carry out quantitative analysis of the interference between a cable and its surrounding environment, and we can compute the bend radius of the cable being analyzed.

The contributions of the proposed segmentation method are summarized as follows: (1) fast, almost immediate segmentation of a cable point cloud out of the 3-D measurement data; (2) segmentation robust to noise and outliers; (3) functioning on 3-D point clouds captured from a single position (the one-shot acquisition process leads to sparse and incomplete 3-D point clouds); (4) fully automatic; and (5) can be used to segment objects other than cables and pipes (e.g., arteries in the medical related fields and tree branches in forestry applications).

2 Related Work

2.1 Robot-Based Inspection

The type of inspection robot that we will be investigating is a standard industrial robot that uses a vision-based sensor system mounted on its end effector and may handle the acquisition system, including the camera and the illumination.²

Typical examples of similar works are the visual weld seam inspection for car manufacturing presented by Worn et al.³ and the robot arm equipped with an optical 3-D scanner for inspection of mechanical components by Raffaelli et al.⁴

Recently, systems for aircraft inspection have been developed as well, such as the mobile robot named Air-Cobot presented in Ref. 5. Air-Cobot uses a Pan-Tilt-Zoom 2-D camera and a 3-D scanner mounted on its end effector. The camera provides 2-D images and the 3-D scanner provides 3-D point clouds that are processed to perform the inspection of the aircraft. Seher et al.⁶ presented an overview of the Aging Aircraft Program for developing automation tools for deployment of a robotic mobile nondestructive inspection system for the military sector. Wu et al.⁷ proposed a comprehensive internal defect inspection system of pipeline based on

an in-pipe inspection robot. This robot is equipped with an active stereo omnidirectional vision sensor. Oh et al.⁸ developed a bridge inspection robot with the aim of checking the safety status of a real bridge, gathering accurate data, and performing maintenance. This work emphasizes the system integration method to design and control the entire robot system. Recently, Qin et al.⁹ presented a new methodology to inspect the power line from point clouds acquired by a cable inspection robot.

2.2 Segmentation Using a 3-D Cylindrical Shape Model

The literature can be divided into two categories, according to the type of the target application: segmentation of rigid cylindrical shapes and segmentation of flexible (nonrigid) cylindrical shapes.

2.2.1 Segmentation of rigid cylindrical shapes

These methods aim to fit a cylinder model to a point cloud using RANSAC^{10,11} or Hough-based methods,¹² which are popular techniques for parametric model calculation.

Recently, Lalonde et al.¹³ presented an algorithm based on principal component analysis (PCA) to extract cylinder and estimate its parameters. First, PCA allows one to estimate the position and direction of cylinder. Then 3-D cylinder points are projected onto the plane perpendicular to the principal direction (principal component). To estimate the radius of cylinder, a circle is fitted to these projected 2-D points. In the same way, Nurunnabi et al.¹⁴ combined robust principal component analysis (RPCA) with robust regression. The RPCA allows one to estimate cylinder direction and a circle fitting algorithm following robust regression allows one to adjust the cylinder radius. Liu et al.¹⁵ and Qiu et al.¹⁶ presented an algorithm based on Gaussian sphere to extract primitives in large-scale point clouds of a pipeline plant. First, normals of input points are estimated and projected onto a Gaussian sphere. Next, the principal directions are estimated by mapping all the point normals onto the Gaussian sphere. Further, they detect great circle patterns. In the same context, Chaperon and Goulette¹⁷ combined the Gaussian sphere with RANSAC method to extract cylinders and estimate their parameters.

2.2.2 Segmentation of nonrigid cylindrical shapes

The methods belonging to the second category are usually based on iterative process to fit sub-cylinders (collections of cylinders) to a point cloud. Moreover, the problem of fitting multiple cylinders is more challenging since it is easily affected by noise/outliers and depends on prior segmentation results. However, this kind of methods has been barely investigated.

The EWIS cable segmentation problem can be related to tree branches segmentation. Tree branches are locally approximated as a sequence of cylinders with varying radius, length, and orientation. In this context, Raunonen et al.¹⁸ presented a comprehensive tree reconstruction based on cylinder fitting. Each cylinder is fitted in the least-squares sense to the point cloud data corresponding to a subcylinder. Geometrical analysis is used to find good initial values for the iterative least-squares cylinder fitting. Finally, the branches are modeled as collections of cylinders (set of subcylinders).

2.3 3-D Point Cloud Analysis for Automatic Inspection Using CAD

An advantage of industrial inspection is that, very often, the 3-D model of the part to be inspected is available from the design process.¹⁹ In the work presented in this paper, we assume that such a model is available.

Various studies have been conducted on the data processing of point clouds obtained from cylindrical object. For instance, Taguchi and Morimoto²⁰ have presented a system focusing on wire harness inspection. This system uses an RGB-D sensor to get both color image and depth image, simultaneously. Phong and Choi²¹ compared the point cloud data with the 3-D CAD model for on-site dimensional inspection of industrial plant piping systems.

3 General Concept and System Overview

3.1 Robot-Based Inspection

Our robot-based inspection platform consists of a mobile robot equipped with three sensors (two cameras and a 3-D scanner) mounted on the robot arm end effector (see Fig. 1).

This setup allows one to perform a certain range of actions: (1) localization: a camera with a wide field of view (FOV) allows one to precisely localize the effector with respect to the part to be controlled; (2) inspection: a high-resolution camera with a reduced FOV allows one to capture the details and to observe the elements very finely with 7 degrees of freedom (robot arm); (3) 3-D scan: a structured light stereo sensor complements the sensor's capabilities by digitizing the observed areas; (4) controlling: the whole system is controlled by an industrial computer that manages data acquisition and localization of the effector; and (5) processing: processing of the acquired data and making decision.

3.2 Different Types of Inspection

Several types of verification can be carried out on a mechanical assembly. For instance, check that the elements of the assembly are presented and that they have been mounted correctly (the CAD model is the reference), check if two flexible elements (e.g., cables and harnesses) or a flexible and a rigid element (e.g., pipe and support) are at a safe distance from each other and check that the bend radius of cable complies with safety standards. According to the type of verification to be performed, one of the two following general strategies has been employed:

1. Model-based 2-D image analysis (also called 2-D inspection). This method is easy to deploy since it only uses two RGB cameras (one for localization and one for inspection) mounted on the robot (see Fig. 1) and the CAD model of the object to be inspected. This method is presented in our recent paper.¹
2. Model-based 3-D point cloud analysis (also called 3-D inspection). This method uses one RGB camera (for localization) and the 3-D scanner mounted on the robot and the CAD model. In the present paper, we will focus on the second strategy only.

First strategy is preferred whenever possible, for time reasons (2-D image acquisition and analysis are both less time-consuming than 3-D point cloud acquisition and analysis).

Second strategy is used when 2-D image processing is not sufficient. Indeed, sometimes, using 3-D data is necessary, for example, for checking if the distance between two cables

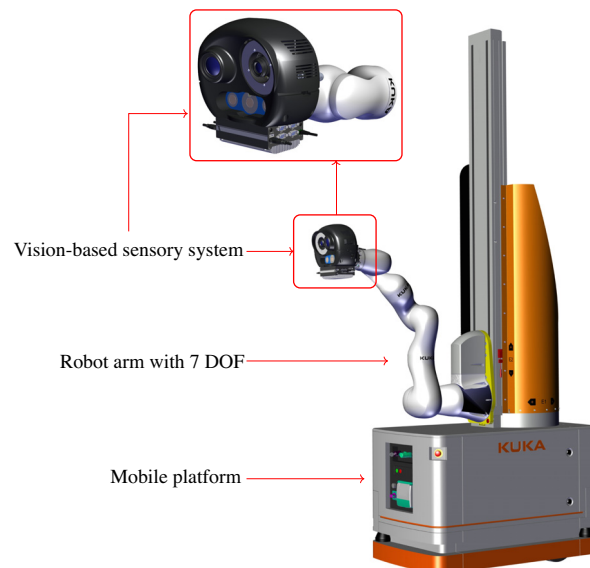


Fig. 1 Our inspection system: robot with its vision-based sensory system.

conforms to the security standards. These types of defects are undetectable with an RGB camera because the cables have the same color. Moreover, obtaining distance measurements is challenging in the absence of depth information. More precisely, it is required to check if two flexible elements (e.g., cables and harnesses) or a flexible and a rigid element (e.g., pipe and support) are at a safe distance from each other. Figure 2 illustrates that 2-D image analysis is not sufficient to solve the inspection problem.

3.3 Data: Point Clouds

The dataset is captured from a single position (one-shot 3-D point cloud) using a precalibrated Ensensio N35 3-D scanner (see Fig. 3) mounted on our mobile robot (see Fig. 1). The specifications of the scanner are shown in Table 1.

An example of our dataset is shown in Fig. 4. As Fig. 4(c) shows, the one-shot acquisition process leads to sparse and incomplete 3-D point clouds. The cables surface is incomplete (approximately only one-third of the cables surface has been captured by the 3-D sensor). As we will see in Sec. 4, this will be challenging for the 3-D segmentation of the cables.

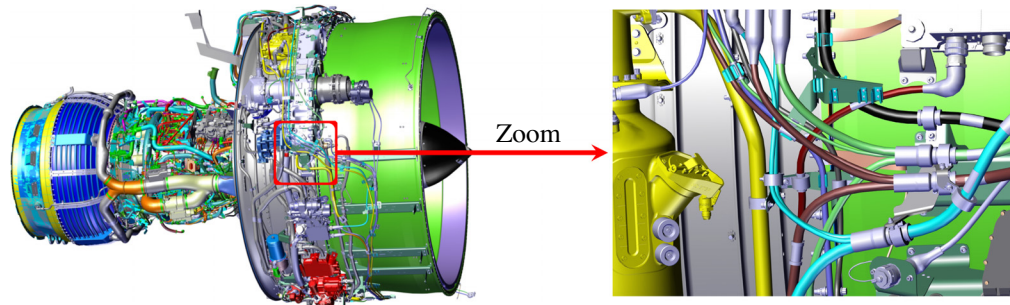


Fig. 2 Example of 3-D inspection: (a) airplane engine, (b) the interference between some cables cannot be determined by 2-D image analysis.

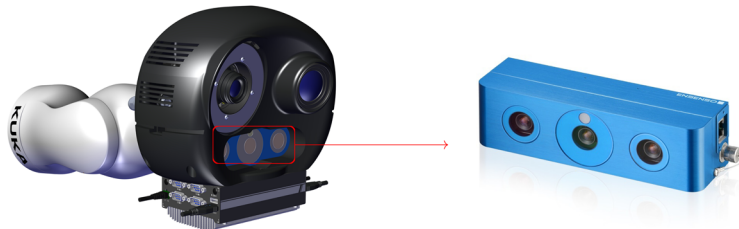


Fig. 3 Precalibrated Ensensio N35 3-D scanner.

Table 1 3-D scanner specifications.

Name	Scanner 3-D Ensensio N35
Sensors	Two global shutter CMOS sensors + one pattern projector
Resolution	1280 × 1024 (1.3 MP)
Focal length	6 to 16 mm
Working distance max	3000 mm
Fps max	10 (2× binning : 30) and 64 disparity levels

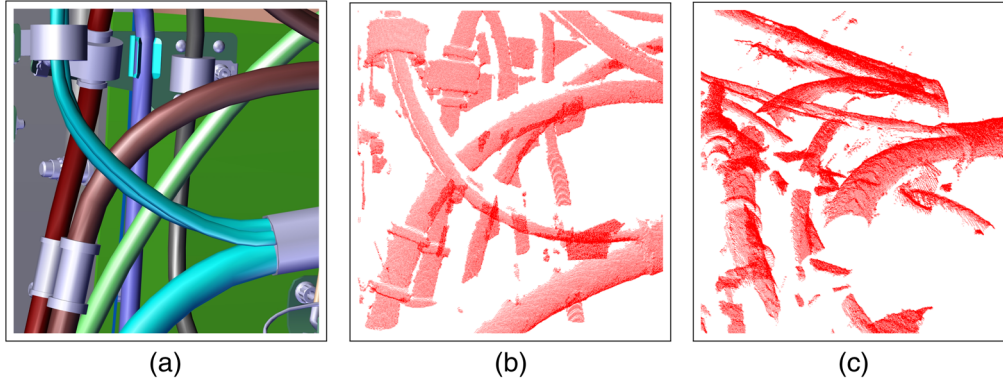


Fig. 4 Examples of our dataset: (a) CAD model and (b), (c) corresponding 3-D point clouds provided by the 3-D scanner from two different points of view.

4 3-D Segmentation of a Prespecified Cable

Usually, the inspection of manufactured parts in industry is done by performing a comparison between a reference model, such as a CAD model, and 3-D measurements taken on the surface.²² However, the matching with 3-D CAD model becomes more problematic when the target objects are nonrigid (flexible), or in any way prone to change in shape or appearance (e.g., cables and harnesses). To solve this problem, we present an original method for cable segmentation and reconstruction based on cylinder fitting. Once the cable is segmented, we carry out a quantitative analysis of the interference and bend radius of each cable.

The automatic segmentation process of a prespecified cable is illustrated in Fig. 5. In this figure, we can see four main blocks as follows.

1. Block ① is dedicated to pose estimation.
2. Block ② is dedicated to the preprocessing of the CAD model (creation of a synthetic 3-D point cloud by mesh sampling) and the input real point cloud (noise filtering).
3. The estimated pose, the synthetic point cloud, and the real filtered point cloud are used in block ③ dedicated to the initialization of the 3-D segmentation (see Sec. 4.1).

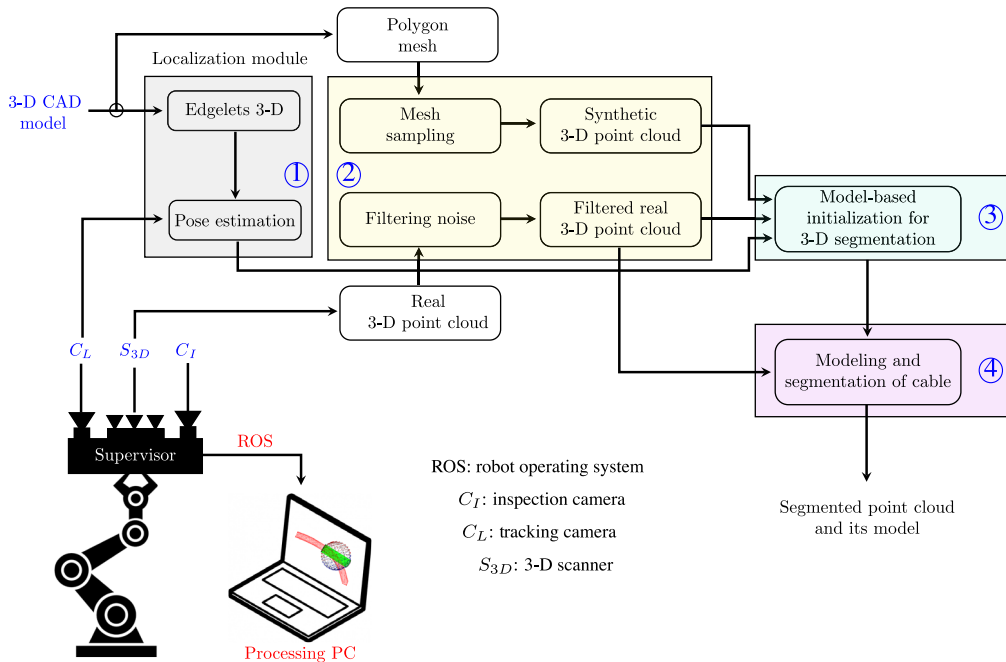


Fig. 5 Overview of the automatic segmentation process.

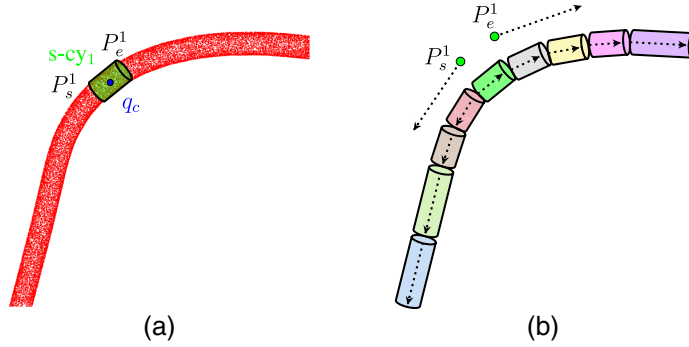


Fig. 6 The steps of cable segmentation: (a) first subcylinder and (b) propagation of subcylinders search.

4. Block ④ is dedicated to the 3-D modeling and segmentation of the cable.

The 3-D segmentation of a cable is carried out in several steps.

1. Model-based initialization of the segmentation (in order to find a 3-D point q_c belonging to the cable to be segmented) [see Fig. 6(a)].
2. Search for the first subcylinder (noted $s\text{-cy}_1$) [see Fig. 6(a)].
3. Once the initial subcylinder is built, we adjust the parameters of the subcylinder on the real point cloud.
4. To search for the next subcylinder, the segmentation algorithm automatically propagates the search structure in both directions from both ends of the initial subcylinder (P_s^1 , P_e^1) [see Fig. 6(b)].

4.1 Model-Based Initialization for the 3-D Segmentation of a Prespecified Cable

In order to initialize the segmentation, we need to find a 3-D point that belongs to the cable. For this, we first perform a registration of the real 3-D point cloud with the synthetic point cloud generated from the CAD model. After this registration, we are able to locate the area where the 3-D initialization point should be searched for.

4.1.1 Registration of the two clouds

The two 3-D point clouds (synthetic point cloud and real point cloud) are automatically registered using a two-step global-to-local registration method.

- Step 1 (global registration): A keyframe-based SLAM algorithm is used to get an initial estimate for the RGB camera (with a wide FOV) pose in the model frame. This algorithm has been implemented by DIOTA and it is used in our work as a black-box that provides us with the required camera pose. Since our 2-D–3-D system is previously calibrated, it is straightforward to obtain the pose of our 3-D scanner in the model frame to align the 3-D data with the 3-D CAD model by applying the obtained transformation parameters R_c (rotation matrix) and T_c (translation matrix).
- Step 2 (local registration): The result of the global registration is refined by a local registration process that uses an iterative nonlinear ICP proposed by Besl and McKay²³ and Levenberg–Marquardt algorithm.^{24,25}

Figures 7(a) and 7(b) show the results of applying the two-step global-to-local registration: first a keyframe-based SLAM and then an iterative nonlinear ICP. As we can see on these figures, after this process, the CAD model is getting closer to the point cloud.

As shown in Fig. 7(c) (blue area), the differences between the CAD model and the real 3-D point cloud are very small in the rigid areas (near the connectors that guarantee a rigidity of a portion of cable).

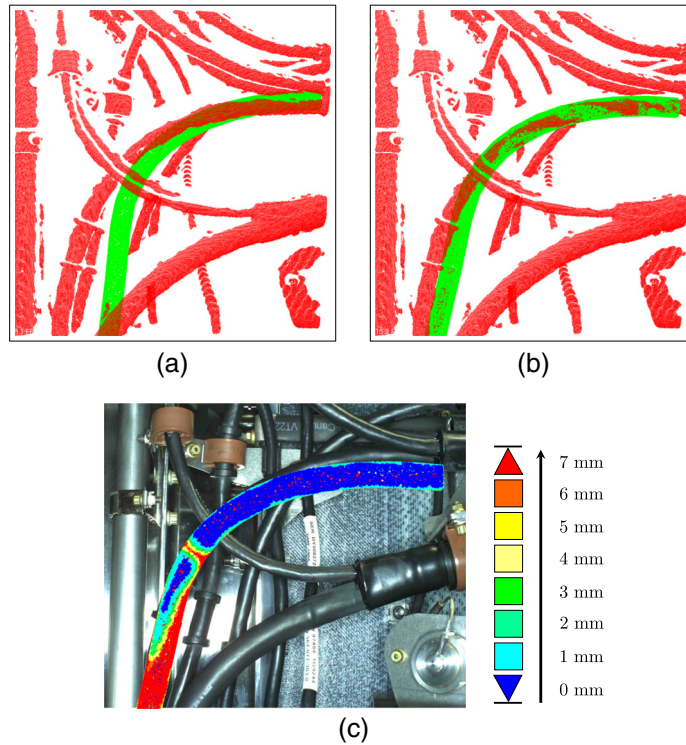


Fig. 7 (a) Step 1: SLAM-based global registration, (b) step 2: ICP-based local registration, and (c) projection of the differences between the CAD model of the cable and the real point cloud onto the 2-D image (blue: small differences and red: large differences).

The blue area in Fig. 7(c) is the one in which we are more likely to find a cable point within the real point cloud. It is, therefore, in this area that we will search for a 3-D point to initialize the segmentation.

4.1.2 Search for a 3-D point to initialize the segmentation

We consider each point p_i of the synthetic cable cloud as the center of a sphere with a radius r equal to the cable radius in the CAD model. We automatically calculate this theoretical radius. Then we count the number of 3-D points of the real point cloud that are in this sphere. Finally, we sort the p_i points in descending order according to the number of neighboring points within each sphere.

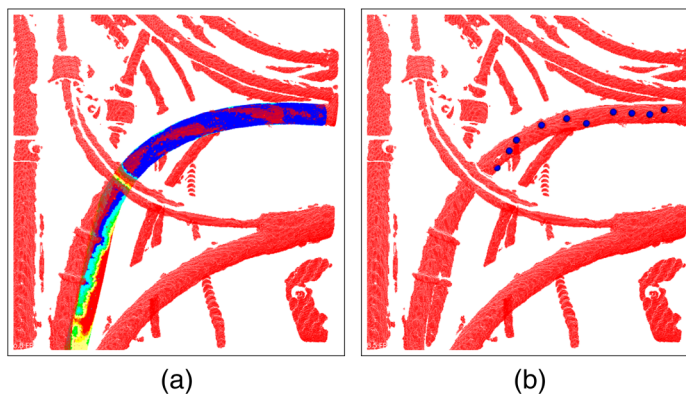


Fig. 8 (a) The differences between the CAD model of the cable and the real point cloud (blue: small differences and red: large differences). (b) 10 best p_i points found are mainly concentrated in the rigid area (blue area).

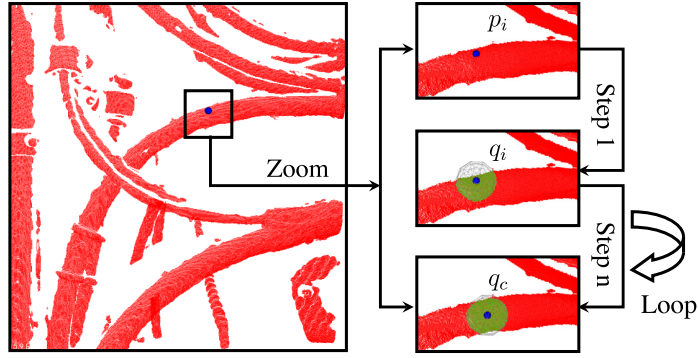


Fig. 9 Illustration of our method to estimate the initialization point.

For example, Fig. 8(b) shows the 10 best p_i points that have been found. As we can see on this figure, these points are mainly located in the rigid area (blue area), where the differences between the cable CAD model and the point cloud are very small.

Once the p_i points of the synthetic cloud are sorted in descending order, we start with the first point p_i that has the largest number of real neighbors in its sphere. To this point p_i of the synthetic cloud, we associate the nearest point in the real cloud (q_i).

We look for all the points of the real cable contained in the sphere centered on the point q_i (of the same radius as the cable model) (see the green points in step 1 of Fig. 9).

Among these points, we seek, through an iterative process, the point that has the greatest number of neighbors within a sphere centered on this point.

At the end of the iterative process, we find a point q_c (see step n of Fig. 9). This point q_c is chosen as the initialization point for segmentation.

4.2 Search for the First Subcylinder

The 3-D point q_c found that previously will allow one to initialize the 3-D segmentation of the cable. We search for the first subcylinder (noted $s\text{-cy}_1$) thanks to an innovative search structure that we have proposed and that will be described in the next section.

4.2.1 Search structure

The proposed search structure consists of a virtual sphere and a rotating cylinder [see Fig. 10(a)]. The sphere is sampled yielding a set of \mathcal{N} 3-D points uniformly distributed on the sphere's surface according to the range of longitude θ and colatitude δ that are located at a distance

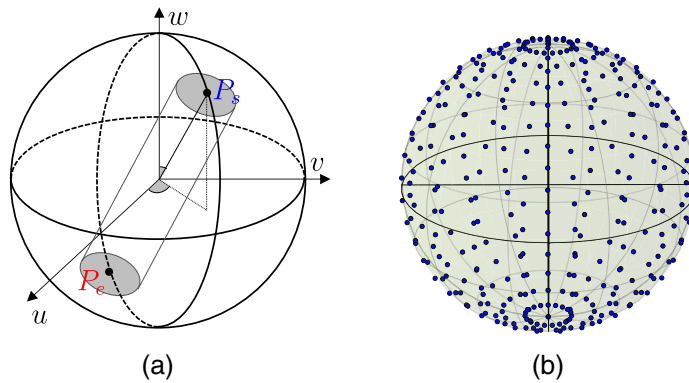


Fig. 10 Search structure: (a) virtual rotating cylinder with three degrees of freedom and (b) set of antipodal points evenly distributed on the sphere's surface according to the range of longitude θ and colatitude δ .

r from a given center [see Fig. 10(b)]. Next, the rotating cylinder is defined by two antipodal points on the sphere.

The number of samples \mathcal{N} is defined using an angular step ω (angular spacing) between 3-D points on the sphere's surface as follows:

$$\mathcal{N} = \frac{(\theta_{\max} - \theta_{\min}) \times (\delta_{\max} - \delta_{\min})}{2 \times \omega^2}, \quad (1)$$

where the range of longitude is $\theta_{\min} \leq \theta \leq \theta_{\max}$ and the range of colatitude is $\delta_{\min} \leq \delta \leq \delta_{\max}$. The angular step ω allows one to control the movement of the rotating cylinder.

Equation (1) states that ω and \mathcal{N} are inversely proportional. Therefore, when the angular spacing ω is small, the number of points \mathcal{N} is high, the algorithm is more robust, and the execution time is higher.

4.2.2 Search algorithm

The search structure is positioned on the initialization point q_c described in Sec. 4.1. The rotating virtual cylinder of the search structure rotates within the virtual sphere with three degrees of freedom (three rotations) for each pair of antipodes. For each rotation, we calculate the number of points inside the virtual rotating cylinder. The position of the cylinder with the highest number of points is taken as the initial subcylinder s-cy₁.

Once the initial subcylinder s-cy₁ has been found, we will adjust its parameters so that it fits the cable as well as possible.

4.3 Fitting the Subcylinder Parameters

4.3.1 Fitting the main axis direction

The start-point P_s and endpoint P_e of the subcylinder are defined by the two antipodal points, which form a cylinder with the greatest number of cloud points inside it. The main axis of subcylinder is initially defined by these two antipodal points on the sphere.

The main axis of subcylinder model is further refined using the point-to-point ICP to the point cloud data corresponding to the cable (see Fig. 11).

Some practical experiments were conducted in order to choose the number of iterations used in the point-to-point ICP. The distance d_i between the point data corresponding to the cable and the surface of subcylinder model is computed after each iteration. The computations are based purely on local neighborhood. Given a raw 3-D point cloud corresponding to the cable

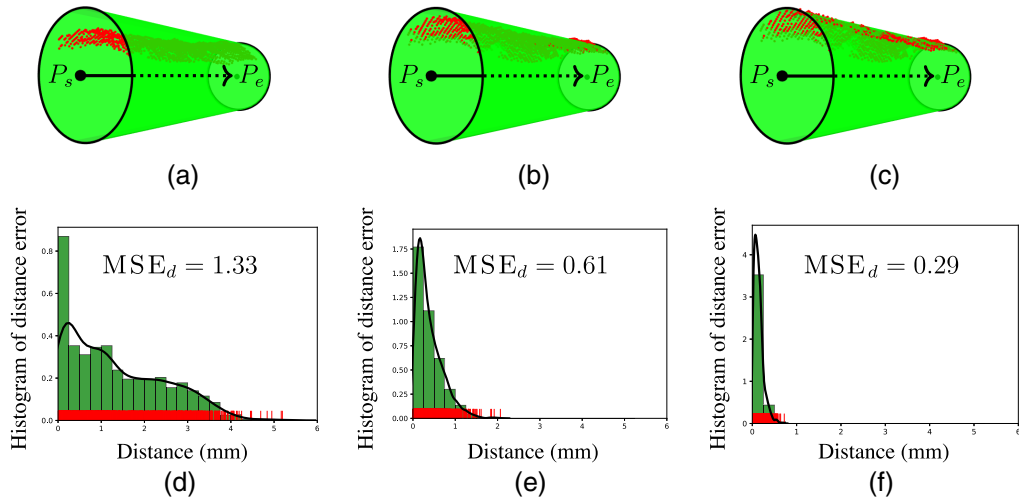


Fig. 11 Iterative fitting for subcylinder model s-cy₁ (in green) to the point cloud data corresponding to the cable (in red): (a) after first iteration, (b) after fifth iteration, and (c) after tenth iteration. Histogram of distance error: (d) after first iteration, (e) after fifth iteration, and (f) after tenth iteration.

$$\mathcal{P} = p_1, p_2, \dots, p_n,$$

a k - d tree structure is constructed. Further, for each point p_i , the distance d_i to its nearest neighbor point from the sampled subcylinder model is calculated using the k -nearest neighbor (KNN) method.

In order to choose the number of iterations used in the point-to-point ICP, the mean squared error MSE_d of the distance was computed after each iteration as follows:

$$MSE_d = \frac{1}{n} \sum_{i=1}^n (d_i)^2. \quad (2)$$

The MSE_d of different iterations is plotted in Fig. 11.

The error gradually drops close to zero after a few iterations (10th iteration) as shown on the histograms of distance error. The MSE_d reduces and converges to a small value after 10th iteration. Therefore, the number of iterations was set to 10 in all the experiments.

The advantage of using the proposed search structure is that the point cloud data corresponding to the cable is always within the subcylinder model. Therefore, only inlier points (points inside cylinder) are forwarded to the subcylinder fitting module. This allows faster convergence and more accurate alignment.

Other methods could have been used to estimate the cylinder axis. For instance, in order to estimate the axis of a pipe, Kawashima et al.²⁶ have projected normal vectors of the grouped points to a unit sphere surface. Then a plane is fitted to the projected normal vectors. Finally, the normal vector of the plane is determined as an initial axis vector of cylinder.

Another possible method could be to compute the average of normal vectors. If a point lies on a cylinder, its normal is perpendicular to the cylinder axis. Qiu et al.¹⁶ proposed a calculation of average normal and searching a cylinder axis as a vector perpendicular to this average normal. However, normal estimation is sensitive to outliers, to noise, and to variations of point density.

4.3.2 Fitting the length (distance between start-point P_s and endpoint P_e):

Depending on the level of the cable curvature, each subcylinder has a different length (see Fig. 12).

The fitting process is illustrated in Fig. 13. For each iteration, the length is computed by projecting the real 3-D points inside the cylinder (between P_s and P_e) onto its main axis. Once the points are projected, a weighting of these projected points makes it possible to estimate the length of the cylinder.

4.3.3 Fitting the radius:

For each iteration, the mean distance of all 3-D points inside cylinder to its main axis gives the estimated cable radius \tilde{r} . The estimated cable radius was computed as follows:

$$\tilde{r} = \frac{1}{N} \sum_{i=1}^n d_i, \quad (3)$$

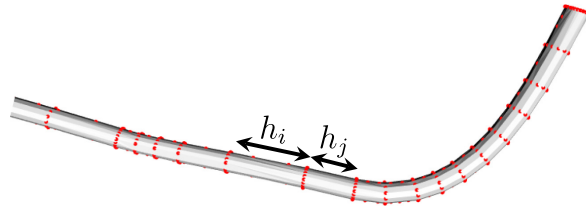


Fig. 12 Depending on the size of the cable curvature each subcylinder has a different length (e.g., $h_i \neq h_j$). The red points represent the set of vertices.

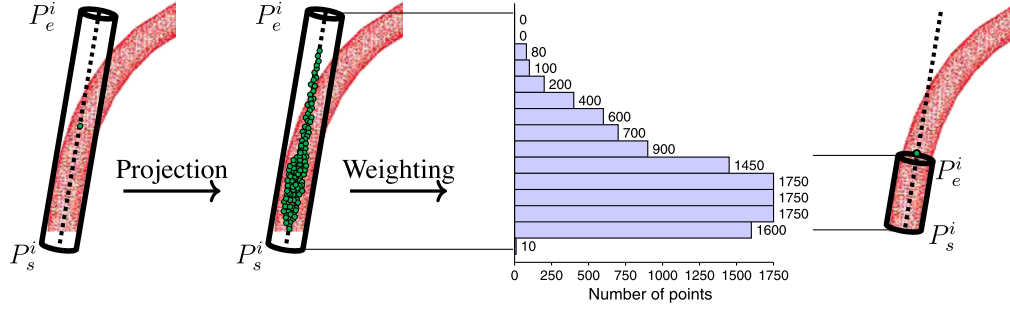


Fig. 13 Fitting the subcylinders parameters: (first column) point cloud data corresponding to the cable and rotating cylinder, (second column) projection of the point cloud inside rotating cylinder onto main axis, (third column) bar graph showing the amount of points per bin, and (fourth column) the fitted subcylinder.

where d_i represents the distance between a real point p_i , and its orthogonal projection p_i^* onto the cylinder axis.

Another possible way to estimate the cable radius could be to project the points onto a plane orthogonal to the axis, and then fit a circle to the projected points in the least square sense.¹⁸ We have tested this method but it proved to be less robust than the one we have implemented. In addition, let us keep in mind that we are using a one-shot scanning approach. Therefore, the acquired data are mostly incomplete and sometimes locally missing, which makes the fitting circle approach not efficient.

4.4 Propagation of Search Structure

Once the first subcylinder s-cy₁ has been adjusted on the point cloud data corresponding to the cable, the subcylinder search/adjustment process is iteratively extended by propagating along the cable in both directions (see Figs. 6 and 14).

Figure 15(a) illustrates the final cylinders model made of 13 consecutive subcylinders labeled s-cy₁ through s-cy₁₃, obtained on one prespecified cable. Figure 15(b) provides the details of the 13 subcylinders: their radii, lengths, angles between two consecutive main axes, and directions of the propagation, where “+” denotes first direction and “-” denotes second direction.

4.5 Experimental Results of 3-D Segmentation

Our 3-D segmentation method has been extensively tested on real and synthetic 3-D point clouds.

4.5.1 Results obtained by processing real data

Our 3-D segmentation method has been extensively tested on 74 different real point clouds acquired during a period of several months in factory conditions. All the visible cables within each cloud have been successfully segmented, as can be seen in Fig. 16, which provides several examples of segmented cables with different radii.

In order to quantify the quality of segmentation we have used the Meshlab[®] tool for semi-manual segmenting/annotating of the cables. The quality of segmentation is evaluated by calculating the error distance (MSE_{distance}) between the set of semimanual segmented points and the set of points segmented by our algorithm, defined by the following equation:

$$MSE_{\text{distance}} = \frac{1}{n} \sum_{i=1}^n MSE_{d_i}, \quad (4)$$

where MSE_d is defined in Eq. (2), and n represents the number of samples (dataset).

By applying this quantitative comparison on our dataset made of 74 different point clouds, we have obtained a mean value of the MSE_{distance} equal to 0.3 mm.

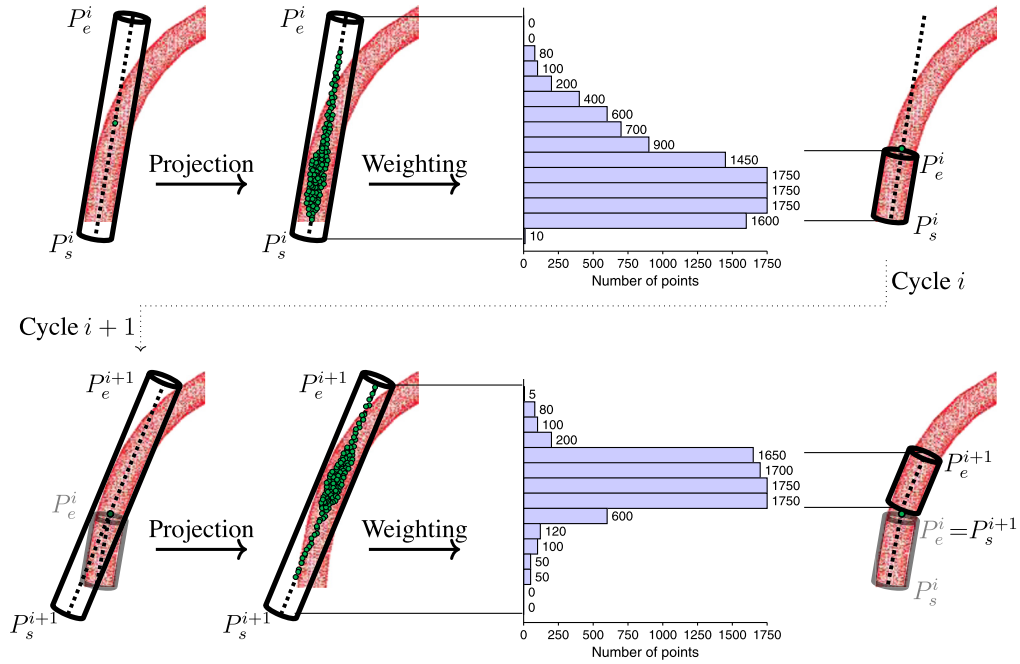
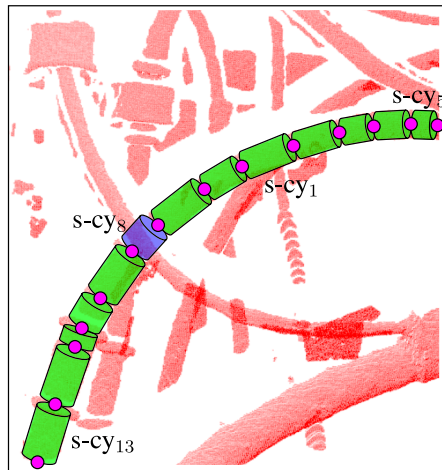


Fig. 14 Propagation of the fitting process: (first column) point cloud data corresponding to the cable and the rotating cylinder, (second column) projection of the point cloud inside cylinder onto the main axis, (third column) bar graph showing the amount of points per bin, and (fourth column) the fitted subcylinder.



Sub-cylinder	Radius	Length	Angle	Dir
s-cy ₁	5.81	33.55	–	+
s-cy ₂	5.93	22.09	8.51	+
s-cy ₃	5.89	17.83	2.12	+
s-cy ₄	5.97	17.97	11.4	+
s-cy ₅	5.91	19.26	3.7	+
s-cy ₆	5.90	16.43	4.12	–
s-cy ₇	5.91	10.01	2.02	–
s-cy ₈	–	13.20	3.22	–
s-cy ₉	5.93	22.01	1.7	–
s-cy ₁₀	5.98	10.66	1.1	–
s-cy ₁₁	5.92	19.44	4.3	–
s-cy ₁₂	5.79	11.81	12.3	–
s-cy ₁₃	5.80	18.02	11.2	–

(a)

(b)

Fig. 15 (a) The final cylinders model made of 12 subcylinders labeled s-cy₁ through s-cy₁₃ (excluding s-cy₈) in green and, in blue, a detected hole (s-cy₈) in the cable due to two overlapping cables, and (b) the details of the 13 subcylinders: their radii, lengths, and angles between consecutive main axis and directions of propagation, where “+” denotes first direction and “–” denotes second direction. It should be noted that s-cy₈ is a detected hole due to the overlapping cables and its radius is equal to the radius of previous subcylinder s-cy₇.

4.5.2 Results obtained by processing synthetic data

We have created our synthetic data using parts extracted from the engine CAD model (see Fig. 2). In order to represent the 3-D CAD model in a point cloud format, we have used weighted uniform sampling to generate different point clouds with different resolutions (\mathcal{R}_p).

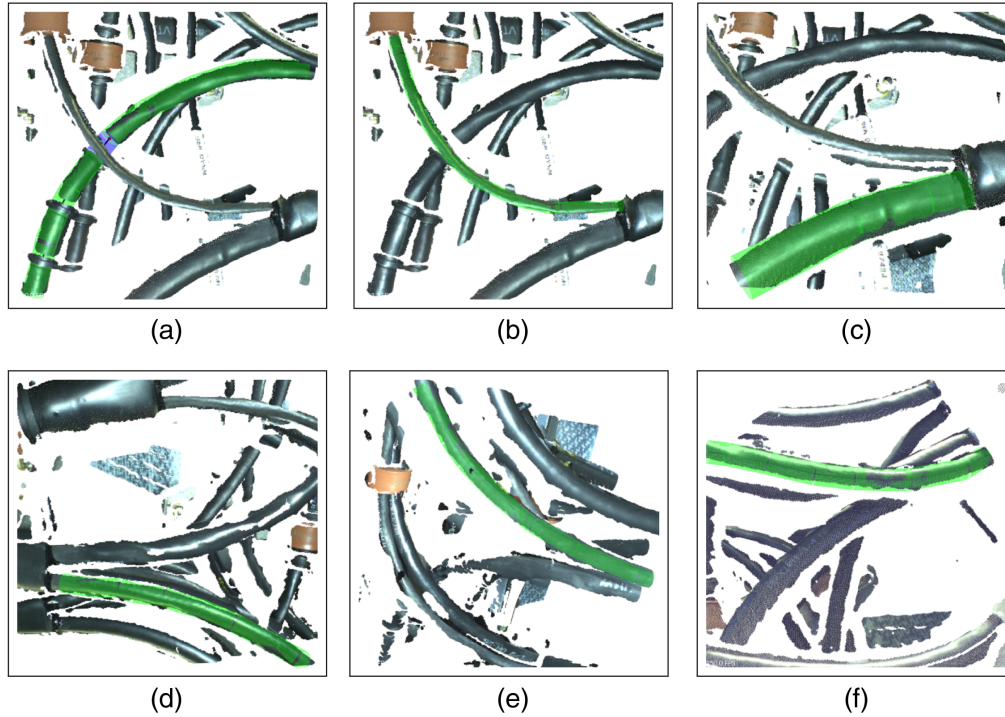


Fig. 16 (a)–(f) Segmentation results on different real data acquired by the 3-D scanner. Segmented cables are in green.

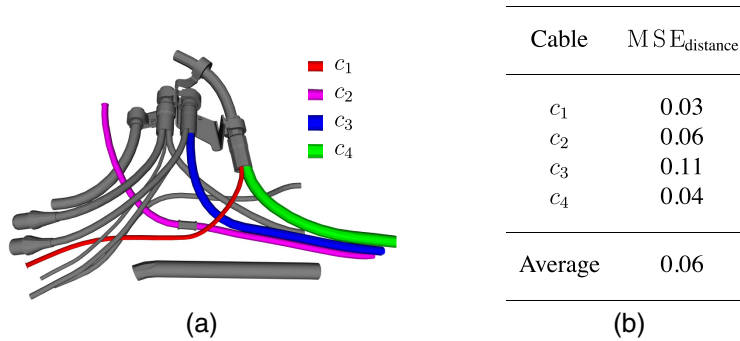


Fig. 17 Results on synthetic data generated from CAD model with three different resolutions (\mathcal{R}_p) and 10 different levels of noise (σ). (a) The four synthetic cables that have been processed and (b) the error distances that have been computed.

In addition, a Gaussian noise with mean ($\mu = 0$) and different values of standard deviation (σ) for the normal distribution $\mathcal{N}(0, \sigma)$, which defines the noise level in all (x , y , and z) directions was added to the point cloud data.

Figure 17(b) shows the results for four different synthetic cables shown in Fig. 17(a). Each MSE value in this figure is obtained as the average on 30 different clouds generated from one cable with 3 different sampling resolutions and 10 different levels of noise.

The robustness level of our method is demonstrated by the curve in Fig. 18. For a standard deviation up to $\sigma = 1.3$, we have a curve with low slope.

5 Inspection of a 3-D Segmented Cable

The global objective of the inspection process is to measure the bend radius of each segmented cable (in order to check that it complies with safety standards) and the minimum distance between each segmented cable and the other elements in the mechanical assembly (in order to check a possible interference problem).

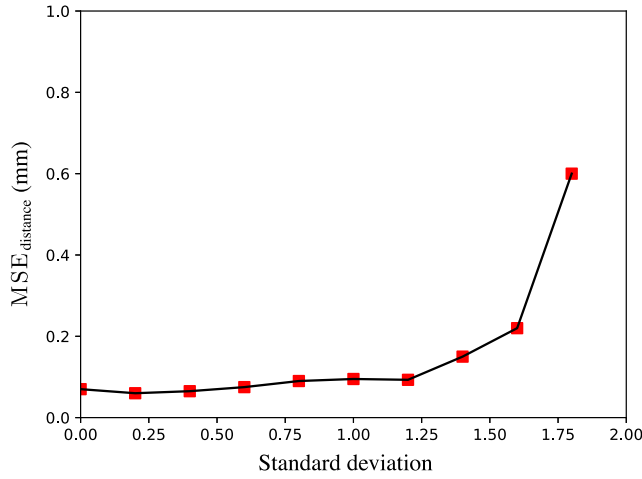


Fig. 18 Robustness in the presence of different levels of noise added to the synthetic data obtained from different cables.

5.1 Computation of the Minimum Bend Radius

5.1.1 Description of the method

The minimum bend radius is the smallest allowed radius the cable is allowed to be bent around. The 3-D segmentation method presented in Sec. 4 produces almost immediately a cable model segmented from the measurement data. Using this model, we can carry out quantitative analysis of the bend radius of the cable. By projecting the set of start-points P_s and endpoints P_e of all the subcylinders [Fig. 19(b)] onto a plane that fits the set of 3-D points, a circle is fitted to the projected points using a least squares minimization method.

The fitting algorithm had to be split into three steps.

- Step 1: Using the least-squares method, find the best fitting plane to the set of start-points P_s and endpoints P_e of all the subcylinders.
- Step 2: Project the set of start-points P_s and endpoints P_e of all the subcylinders onto the calculated plane.
- Step 3: Using the least-squares method, fit a circle to the projected 2-D coordinates and get the circle center and radius.

Finally, the bend radius is estimated from the radius of the fitted circle [see Fig. 19(b)]. This method gives satisfying results.

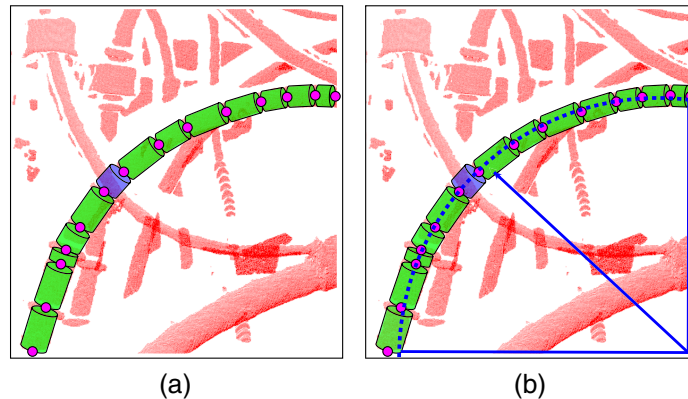


Fig. 19 Computation of the bend radius: (a) the set of start-points P_s and endpoints P_e of all the subcylinders and (b) the corresponding bend radius.

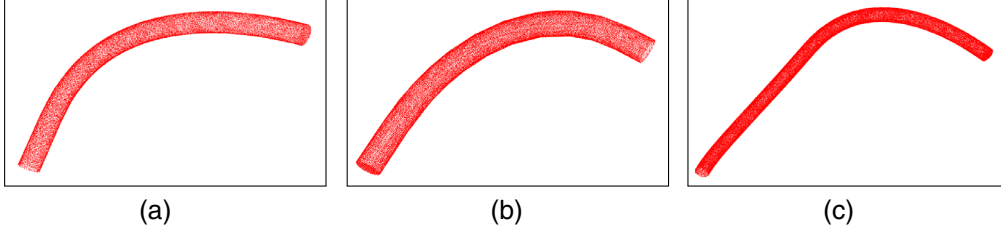


Fig. 20 A few examples of our dataset: three different synthetic cables with different bend radii (R_c). (a) $R_c = 13.65$ cm, (b) $R_c = 14$ cm, and (c) $R_c = 46.55$ cm.

5.1.2 Experimental results

Since we are not in the possession of needed instrumentation to measure precisely the bend radius of the cable, we have decided to test our approach on synthetic data generated from the CAD model. A few examples of our dataset are shown in Fig. 20.

The results are evaluated on the basis of the difference between the ground truth bend radius R_c and the estimated bend radius \tilde{R}_c :

$$\text{MSE}_{\text{b.radius}} = \frac{1}{n} \sum_{i=1}^n (R_{c_i} - \tilde{R}_{c_i})^2. \quad (5)$$

As expected, the accuracy of the measure is directly related to the robustness of the segmentation algorithm, fitting plane algorithm, and fitting circle algorithm.

On a set of three different cables with different resolutions and different bend radii, we found $\text{MSE}_{\text{b.radius}} = 1.23$ mm.

In our experiments, we have found that the result is better as the number of subcylinders is higher. Indeed, the fitting of a plane and the fitting of a circle are both more accurate when the set of start-points P_s and endpoints P_e is bigger.

5.2 Interference Detection

5.2.1 Description of the method

In this section, we will focus on the detection of a possible interference between a cable and the other surrounding elements (another cable, a support, etc.) present in the mechanical assembly.

After the segmentation process, we have two main point clouds: the segmented point cloud, which we will denote by \mathcal{P}_s , and all the other points, which we will denote by $\overline{\mathcal{P}}_s$ (see Fig. 21). If we name \mathcal{P} the whole filtered input 3-D point cloud provided by the 3-D scanner, we have $\mathcal{P} = \mathcal{P}_s + \overline{\mathcal{P}}_s$. \mathcal{P}_s and $\overline{\mathcal{P}}_s$ are, therefore, the inputs for our interference detection module.

The objective of the interference detection is to determine if the segmented cable is at a safe distance d_T from the other surrounding elements present in the mechanical assembly.

In order to measure the distance between the segmented point cloud \mathcal{P}_s (that represents a cable) and the surrounding elements belonging to $\overline{\mathcal{P}}_s$, we divide the set of points $\overline{\mathcal{P}}_s$ into smaller parts (clusters) having salient characteristics. If the minimum distance between the segmented point cloud (a cable) and a cluster belonging to $\overline{\mathcal{P}}_s$ is lower than the given tolerance value d_T , then the cable is in interference with the cluster.

Namely, we say that a cable (3-D points $p_i \in \mathcal{P}_s$) is in interference with a cluster $\mathcal{C} = \{p_j \in \mathcal{C} \subset \overline{\mathcal{P}}_s\}$ if:

$$\min \|p_i - p_j\|_2 < d_T. \quad (6)$$

In order to divide the set of points $\overline{\mathcal{P}}_s$ into meaningful clusters, we use the Euclidean clustering algorithm proposed in Ref. 27. This algorithm explores the local neighborhood of each data point present in the input and assembles points into small clusters having salient characteristics.

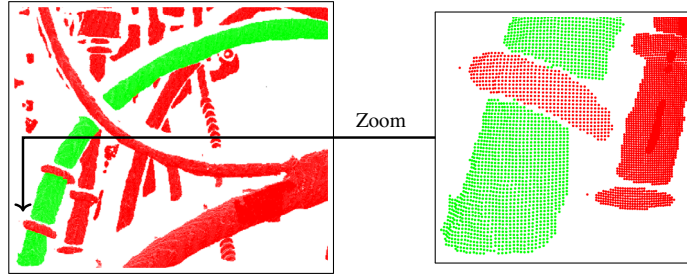


Fig. 21 The segmented point cloud, which we will call \mathcal{P}_s in green and all the other points, excluding the segmented point cloud, which we will call $\overline{\mathcal{P}}_s$ in red.

This is achieved by comparing the Euclidean distance between neighboring points using the KNN method.

The algorithm is controlled by a tolerance `ClusterTolerance`, which indicates how close two points are required to be to belong to the same cluster. The parameters of the Euclidean clustering method are presented in Table 2.

This approach provides a rough segmentation of many objects of interest in the scene. Figure 22(b) shows the results of such clustering to the set of points $\overline{\mathcal{P}}_s$ [shown in red in Fig. 22(a)]. The Euclidean clustering algorithm was applied with a `ClusterTolerance` of 0.1 m, a `MinCluster` of 100 cluster points, and a `MaxCluster` of 100.000 cluster points. Each cluster in the scene is assigned a random color. Figure 22(b) illustrates various clusters in the scene, corresponding to various objects including cables, pipes, connectors, and clamps.

After the clustering phase, we have two main point clouds, the segmented cable point cloud \mathcal{P}_s and all the other points $\overline{\mathcal{P}}_s$ divided into clusters \mathcal{C}_j , where $j = 1 \dots n$.

Table 2 Euclidean clustering algorithm parameters.

Parameter	Role
<code>ClusterTolerance</code>	Minimum imposed distance between two clusters
<code>MinCluster</code>	Minimum number of points that a cluster needs to contain in order to be considered valid
<code>MaxCluster</code>	Maximum number of points that a cluster needs to contain in order to be considered valid

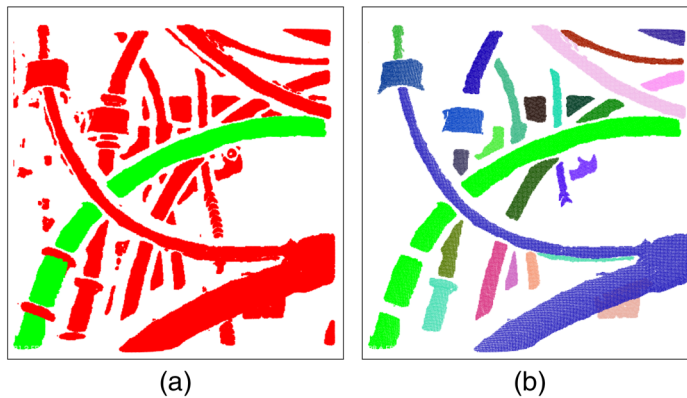


Fig. 22 Clustering result using Euclidean distance clustering approach for the remaining points $\overline{\mathcal{P}}_s$: (a) the set of points $\overline{\mathcal{P}}_s$ in red and (b) list of 26 clusters \mathcal{C}_1 through \mathcal{C}_{26} . Each separate cluster is represented with a different color.

For each point $p_i \in \mathcal{P}_s$, we will compute the distance d_{ij} between this point and its nearest neighbor point \tilde{p}_j from each cluster \mathcal{C}_j as follows:

$$d_{ij} = \|p_i - \tilde{p}_j\|_2. \quad (7)$$

Then the distance $d_j = \min(d_{ij})$, where $p_i \in \mathcal{P}_s$ is considered a distance between a segmented point cloud (cable) and the cluster \mathcal{C}_j . If d_j is lower than a given tolerance value d_T , then the cluster \mathcal{C}_j is considered a cluster that causes interference problem with the segmented point cloud. For example, Table 3 shows a list of 26 clusters \mathcal{C}_1 through \mathcal{C}_{26} of Fig. 22(b), and their details: label, size, and distance d_j from the segmented point cloud \mathcal{P}_s .

Table 3 List of 26 clusters \mathcal{C}_1 through \mathcal{C}_{26} , and their details: label, size, and distance d_j from the segmented point cloud \mathcal{P}_s .

Cluster	Size	Distance
\mathcal{C}_1	17655	15.1459
\mathcal{C}_2	10115	35.956
\mathcal{C}_3	4091	9.58378
\mathcal{C}_4	1931	26.5427
\mathcal{C}_5	1871	37.096
\mathcal{C}_6	1825	52.319
\mathcal{C}_7	1787	13.270
\mathcal{C}_8	1740	52.854
\mathcal{C}_9	1685	31.828
\mathcal{C}_{10}	1642	38.69
\mathcal{C}_{11}	1520	39.727
\mathcal{C}_{12}	1251	32.317
\mathcal{C}_{13}	1091	27.355
\mathcal{C}_{14}	943	18.194
\mathcal{C}_{15}	913	10.181
\mathcal{C}_{16}	889	36.137
\mathcal{C}_{17}	842	44.962
\mathcal{C}_{18}	786	52.264
\mathcal{C}_{19}	746	40.558
\mathcal{C}_{20}	719	57.968
\mathcal{C}_{21}	674	47.760
\mathcal{C}_{22}	627	47.490
\mathcal{C}_{23}	600	32.479
\mathcal{C}_{24}	399	45.798
\mathcal{C}_{25}	372	42.684
\mathcal{C}_{26}	331	29.022

Figure 23 shows that there are two clusters (\mathcal{C}_3 and \mathcal{C}_{15}) with the distances d_3 and d_{15} lower than $d_T = 12$ mm.

The result of the interference detection process is shown in Fig. 24.

5.2.2 Experimental results

In this part, we will present some results regarding the final output of our approach, which is the distance between a cable and neighboring elements. Obtaining ground truth for the physical distance between elements on the engine is not straightforward, especially with millimetric precision. Therefore, for the moment, we have decided to evaluate our measure on synthetic data.

For this purpose, we have used the available CAD model. We have identified four cables and measured the distance between each cable and its surrounding elements in the assembly. We have done this with a specialized modeling software CATIA[®]. Finally, we have run our algorithm on all generated clouds and obtained the minimal distance between each segmented cable and the other elements. We have compared this result with the measure obtained with CATIA[®]. Figure 25 and Table 4 show that the results obtained with our algorithm are close to the results obtained with CATIA[®] considered as the ground truth.

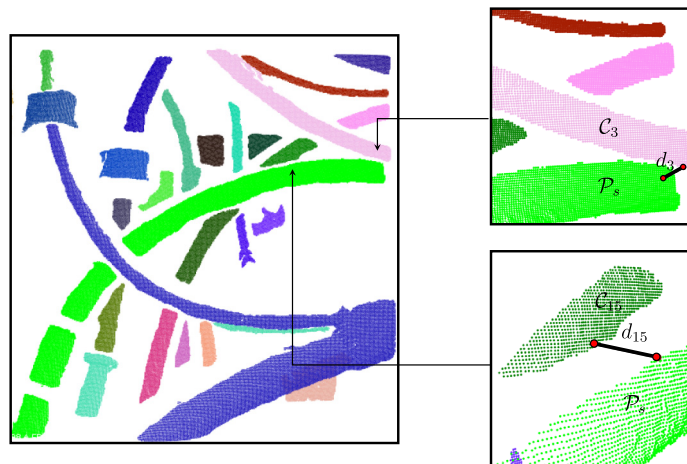


Fig. 23 Interference detection results for a tolerance $d_T = 12$ mm. There are two clusters (\mathcal{C}_3 and \mathcal{C}_{15}) shown in Table 3 with distances d_3 and d_{15} lower than $d_T = 12$ mm.

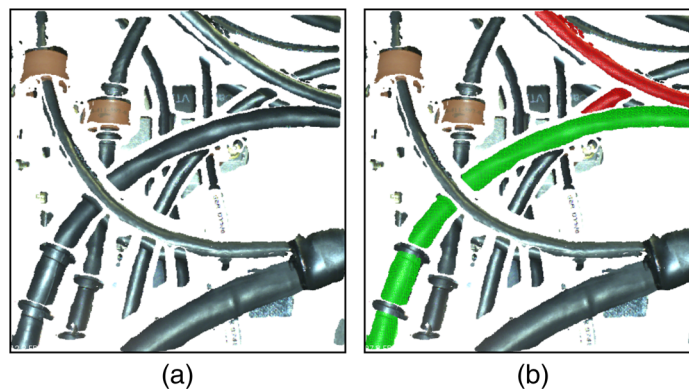


Fig. 24 Results of interference detection process: (a) the input point cloud and (b) segmented point cloud in green and two representative clusters that cause an interference problem in red.

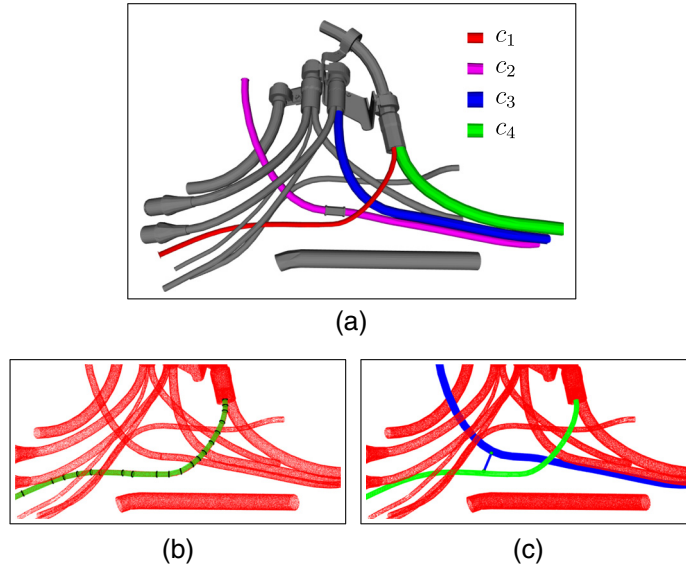


Fig. 25 (a) CAD model with four different cables, (b) an example of a segmentation result on a cable c_1 , and (c) result of interference detection: the segmented cable (green) interferes with the cable c_2 (blue).

Table 4 Interference measurement results compared with modeling software CATIA®.

Pair of cables	Measure obtained with CATIA	Measure obtained by our algorithm
$c_1 - c_2$	40.7	41.2
$c_1 - c_3$	17.2	17.0
$c_1 - c_4$	0.1	0.2
$c_2 - c_3$	11.3	10.7
$c_2 - c_4$	37.0	37.2
$c_3 - c_4$	11.3	11.1

6 Conclusion

In this paper, we propose an original framework for detecting defects on aircraft EWIS in order to comply with the growing amount of safety regulations, such as the forbidden interference between elements and minimal allowed bend radius of cables. We address this problem of automatic robotic inspection in two parts.

First, a CAD-based automatic selection of informative viewpoints is performed. Next, we start our online inspection process by several preprocessing steps: noise filtering and data resampling. A point cloud filtering algorithm is developed to identify and remove outliers. After the preprocessing stage, a two-step global-to-local registration is applied. A model-based keyframe SLAM algorithm and a camera with a wide FOV are used to get an initial estimate for our 3-D sensor localization. The results of the global registration are then refined by a local registration process based on the well-known iterative nonlinear ICP and Levenberg–Marquardt algorithm.

Further, a method for cable segmentation based on cylinder collection fitting and aided by the available CAD model is proposed and evaluated. Using the segmentation result, we can detect the possible interference of each cable with other objects as well as analyze its bend radius. The experimental results show that our method is highly accurate in detecting defects on aircraft EWIS.

For now, we are detecting meaningful point cloud clusters interfering with our cable of interest. Our further research will focus on classifying these clusters by exploiting the CAD model of the mechanical assembly.

We are also working on a more general framework for the segmentation of cylindrical shapes other than cables and pipes (e.g., arteries in the medical field, tree branches in forestry applications, etc.). In parallel, we are looking into making 3-D information richer by fusing multiple one-shot scans acquired from multiple views.

Acknowledgments

This research work has been carried out within a PhD thesis co-funded by the French “Région Occitanie.” We would like to thank the “Région Occitanie” for its financial support.

References

1. H. B. Abdallah et al., “Automatic inspection of aeronautical mechanical assemblies by matching the 3D CAD model and real 2D images,” *J. Imaging* **5**, 81–108 (2019).
2. C. Eitzinger, S. Zambal, and P. Thanner, “Robotic inspection systems,” in *Integrated Imaging and Vision Techniques for Industrial Inspection*, Z. Liu et al., Eds., pp. 321–350, Advances in Computer Vision and Pattern Recognition, Springer, London (2015).
3. H. Worn, T. Langle, and M. Gauss, “ARIKT: adaptive robot based visual inspection,” *Kunstliche Intell.* **17**, 33–35 (2003).
4. R. Raffaeli, M. Mengoni, and M. Germani, “Context dependent automatic view planning: the inspection of mechanical components,” *Comput.-Aided Des. Appl.* **10**, 111–127 (2013).
5. I. Jovancevic et al., “3D point cloud analysis for detection and characterization of defects on airplane exterior surface,” *J. Nondestr. Eval.* **36**, 74 (2017).
6. C. Seher, M. Siegel, and W. M. Kaufman, “Automation tools for nondestructive inspection of aircraft: promise of technology transfer from the civilian to the military sector,” in *Fourth Annu. IEEE Dual-Use Technol. and Appl. Conf.*, SUNY Institute of Technology at Utica/Rome, IEEE, Griffis AFB, New York (1994).
7. Z. Wu et al., “3D shapenets: a deep representation for volumetric shapes,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1912–1920 (2015).
8. J.-K. Oh et al., “Design and control of bridge inspection robot system,” in *Int. Conf. Mechatron. and Autom.*, pp. 3634–3639 (2007).
9. X. Qin et al., “Detecting inspection objects of power line from cable inspection robot LiDAR data,” *Sensors* **18**, 1284 (2018).
10. R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for point-cloud shape detection,” *Comput. Graphics Forum* **26**, 214–226 (2007).
11. Z. Su et al., “Extracting wood point cloud of individual trees based on geometric features,” *IEEE Geosci. Remote Sens. Lett.* **16**, 1294–1298 (2019).
12. T. Rabbani and F. Van Den Heuvel, “Efficient Hough transform for automatic detection of cylinders in point clouds,” in *Proc. ISPRS Workshop Laser Scan, ISPRS Arch*, p. 36 (2005).
13. J.-F. Lalonde, N. Vandapel, and M. Hebert, “Automatic three-dimensional point cloud processing for forest inventory,” Tech. Rep. CMU-RI-TR-06-21, Carnegie Mellon University, Pittsburgh, Pennsylvania (2006).
14. A. Nurunnabi, Y. Sadahiro, and R. Lindenbergh, “Robust cylinder fitting in three-dimensional point cloud data,” *ISPRS Int. Arch. Photogramm Remote Sens. Spat. Inf. Sci.* **XLII-1/W1**, 63–70 (2017).
15. Y.-J. Liu et al., “Cylinder detection in large-scale point cloud of pipeline plant,” *IEEE Trans. Visual Comput. Graphics* **19**, 1700–1707 (2013).
16. R. Qiu, Q.-Y. Zhou, and U. Neumann, “Pipe-run extraction and reconstruction from point clouds,” *Lect. Notes Comput. Sci.* **8691**, 17–30 (2014).
17. T. Chaperon and F. Goulette, “Extracting cylinders in full 3D data using a random sampling method and the Gaussian image,” in *Proc. Vision Model. and Visual. Conf.*, Stuttgart, Germany, pp. 35–42 (2001).

18. P. Raumonon et al., “Fast automatic precision tree models from terrestrial laser scanner data,” *Remote Sens.* **5**, 491–520 (2013).
19. J. L. Alarcon-Herrera, X. Chen, and X. Zhang, “Viewpoint selection for vision systems in industrial inspection,” in *IEEE Int. Conf. Rob. and Autom.*, pp. 4934–4939 (2014).
20. K. Taguchi and M. Morimoto, “Appearance inspection of wire harness using RGB-D sensor,” in *Joint 7th Int. Conf. Inf., Electron. and Vision and 2nd Int. Conf. Imaging, Vision and Pattern Recognit.*, pp. 416–419 (2018).
21. H. Phong and Y. Choi, “Comparison of point cloud data and 3D CAD data for on-site dimensional inspection of industrial plant piping systems,” *Autom. Constr.* **91**, 44–52 (2018).
22. A. E. Jaramillo, P. Boulanger, and F. Prieto, “On-line 3-D inspection of deformable parts using FEM trained radial basis functions,” in *IEEE 12th Int. Conf. Comput. Vision Workshops*, pp. 1733–1739 (2009).
23. P. Besl and N. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992).
24. A. Fitzgibbon, “Robust registration of 2D and 3D point sets,” *Image Vision Comput.* **21**, 1145–1153 (2003).
25. C. Kim, H. Son, and C. Kim, “Fully automated registration of 3D data to a 3D CAD model for project progress monitoring,” *Autom. Constr.* **35**, 587–594 (2013).
26. K. Kawashima, S. Kanai, and H. Date, “As-built modeling of piping system from terrestrial laser-scanned point clouds using normal-based region growing,” *J. Comput. Des. Eng.* **1**, 13–26 (2014).
27. R. B. Rusu, “Semantic 3D object maps for everyday manipulation in human living environments,” PhD Thesis, Computer Science Department, Technische Universitaet Muenchen, Germany (2009).

Hamdi Ben Abdallah received his engineering degree in industrial control from the National Engineering School of Sousse, Tunisia, in 2016. He received his PhD in computer vision in 2020 from IMT Mines Albi (Albi, France), a French “Grande Ecole” specialized in process engineering, under the supervision of Professor Jean-José Orteu and Dr. Igor Jovancevic. Currently, he is working at Institut Clément Ader, Albi, as a research engineer focusing on computer vision applications on the problems of inspection and manufacturing process monitoring.

Jean-José Orteu graduated with an engineering degree in electrical and software engineering and a master’s thesis in automatic control from a French “Grande Ecole” (ENSEIRB, Bordeaux, France) in 1987. He received his PhD in computer vision from Université Paul Sabatier (Toulouse, France) in 1991. Currently, he is a full professor at IMT Mines Albi (Albi, France), a French “Grande Ecole” specialized in process engineering. He carries out his research work in the Institut Clément Ader Laboratory (200 people). For more than 15 years, he has developed computer vision-based solutions for 3-D measurements in experimental mechanics (photomechanics) and for a few years he is more specifically involved in the application of computer vision on NDE, inspection, and manufacturing process monitoring.

Igor Jovancevic graduated with a mathematics degree (specialty in computer science) from the Faculty of Natural Science and Mathematics at the University of Montenegro in 2008. He graduated from joint Erasmus Mundus Master Program in Computer Vision and Robotics (VIBOT) conducted by University of Burgundy (Le Creusot, France), University of Girona (Girona, Spain), and Heriot Watt University (Edinburgh, UK) in 2011. He received his PhD in computer vision in 2016 from IMT Mines Albi (Albi, France), a French “Grande Ecole” specialized in process engineering, under the supervision of Professor Jean-José Orteu and Dr. Thierry Sentenac. Currently, he is working at Diotasoft, Toulouse, as a research engineer focusing on computer vision applications on the problems of inspection and manufacturing process monitoring.

Benoit Dolives graduated with a master’s degree (specialty in robotics and computer science) from University Paul Sabatier of Toulouse in 2010. He has spent 7 years working on camera localization and 3-D reconstruction for autonomous rover exploration and UAV. He is currently working at Diotasoft, Toulouse, as a research engineer focusing on robotics and computer vision applications on the problems of inspection and manufacturing process monitoring.