



HAL
open science

Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model

Ilisio Viana, Jean-José Orteu, Nicolas Cornille, Florian Bugarin

► To cite this version:

Ilisio Viana, Jean-José Orteu, Nicolas Cornille, Florian Bugarin. Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model. *Journal of Electronic Imaging*, 2015, 24 (6), pp.061118. 10.1117/1.JEI.24.6.061118 . hal-01351022

HAL Id: hal-01351022

<https://hal.science/hal-01351022>

Submitted on 2 Aug 2016

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.

Journal of Electronic Imaging

JElectronicImaging.org

Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model

Ilisio Viana
Jean-José Orteu
Nicolas Cornille
Florian Bugarin

Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model

Ilisio Viana,^{a,*} Jean-José Orteu,^a Nicolas Cornille,^b and Florian Bugarin^a

^aUniversité de Toulouse, INSA, UPS, Mines Albi, ISAE, Institut Clément Ader, Campus Jarlard, 81013 Albi, France

^bG²Metric, 40 Chemin Cazalbarbier, 31140 Launaguet, France

Abstract. We focus on quality control of mechanical parts in aeronautical context using a single pan-tilt-zoom (PTZ) camera and a computer-aided design (CAD) model of the mechanical part. We use the CAD model to create a theoretical image of the element to be checked, which is further matched with the sensed image of the element to be inspected, using a graph theory-based approach. The matching is carried out in two stages. First, the two images are used to create two attributed graphs representing the primitives (ellipses and line segments) in the images. In the second stage, the graphs are matched using a similarity function built from the primitive parameters. The similarity scores of the matching are injected in the edges of a bipartite graph. A best-match-search procedure in the bipartite graph guarantees the uniqueness of the match solution. The method achieves promising performance in tests with synthetic data including missing elements, displaced elements, size changes, and combinations of these cases. The results open good prospects for using the method with realistic data. © 2015 SPIE and IS&T [DOI: 10.1117/1.JEI.24.6.061118]

Keywords: computer-aided inspection; image analysis; contour registration; feature matching; graph matching; pan-tilt-zoom camera.

Paper 15521SSP received Jun. 28, 2015; accepted for publication Nov. 25, 2015; published online Dec. 23, 2015.

1 Introduction

For a company, checking product conformity is a guarantee to ensure good-quality delivery for its consumers, to potentially save costs by avoiding product return, and, in crucial applications, to prevent accidents. As a result, many efforts are continuously made by security organizations, industries, and research institutions in order to improve quality control methodologies. Defect inspection using computer vision is a widely studied topic.¹⁻⁴ It is suitable for contactless inspection and it ensures uniform, consistent, and quick defect control. In the primary days of inspection using camera information, a black-and-white image of a nondefective element was used as a reference that was compared with the tested and potentially defective element. Over the years, different approaches have been used, such as correlation of defect-free image with test image, contour comparison, and more sophisticated approaches such as comparison of a model representing the nondefective element with the potentially defective element, as in Refs. 5 and 6.

In this paper, we address the computer-aided inspection problem, trying to make sure the tested element is congruent with its computer-aided design (CAD) model. In our approach, the reference is an image derived from a CAD model. This model contains primitives (including line segments and ellipses) representing the element to be controlled. Our goal consists of comparing this theoretical image with an image acquired with a pan-tilt-zoom (PTZ) camera. This type of approach was used by Far⁷ to help in three-dimensional (3-D) reconstruction and by Bourgeois⁸ for mechanical object alignment. Karabagli⁹ used theoretical

versus real image comparison to check machining setup element placement in a high-speed-machining application. This paper addresses the problem of mechanical assembly in an aeronautical context under the constraint of robustness and flexibility. In other words, the method has to be able to detect defects in an evaluated element of the inspected part while being especially careful of the false alarm rate. In fact, it was reported that if the false alarm rate is very important, users of the application are reluctant to use it. In all the mentioned works, contours were the information used. In some cases, like in Ref. 9, contours were associated with other types of information (skeletons, geometric metrics, and so on). In our work, we extract geometric features from contours and characterize them mathematically. Our work belongs then to the group of geometry-based matching methods.

In Sec. 2, we describe how data used for inspection are obtained. In Sec. 3, we describe the proposed matching method based on the matching of line segments and ellipses present in both reference and test images. In Sec. 4, we explain how scores of line segment and ellipse matching are combined to make a global score. Then, we present and discuss the results in Sec. 5. The paper ends with the conclusion and future work in Sec. 6.

2 Setup: Pan-Tilt-Zoom Camera Associated with Directed Light Device and Computer-Aided-Design Model

This section describes our CAD-guided inspection method setup and how both sensed images and data from the CAD model are exploited in order to make acceptance/rejection

*Address all correspondence to: Ilisio Viana, E-mail: ilisio.viana@mines-albi.fr

decisions, presented later in this paper. Our goal is to check a mechanical assembly made of several elements and whose CAD model is known by processing only two-dimensional (2-D) data generated from the CAD model on one hand and the acquired image on the other. Our acquisition system is composed of a high-definition PTZ dome and one-directional light device moving synchronously with the PTZ camera (cf. Fig. 1).

In our method, we use a PTZ camera because the final goal of the application is to be used in an hangar. For that reason, a PTZ camera appears as a good choice in order to cover a variable distance range. Using a robot arm with 6 deg of freedom equipped with a camera in its end effector could be another solution, but it is potentially more expensive. This acquisition system is provided by the company G²Metric, which has developed inspection software named Lynx that carries out the inspection task by comparing an acquired image with a defect-free reference image stored in a database. Our goal using the CAD model is to work on a new version of Lynx in order to replace the current reference image with an image derived from the CAD model. Data used during the inspection process are obtained through two different stages. First, the CAD-derived image is generated as a screenshot of our CAD model at one given position previously decided by the user as a representative view of the element to be controlled. From the CAD model, we coarsely estimate a lookup table for the pan and tilt parameters of our PTZ camera. Also, we estimate a congruent zoom factor for the PTZ camera in order to have the same scale for both theoretical and real elements. Second, we use all these parameters to direct our acquisition system toward the element to be inspected. The inspection is guided by the CAD model. After the selection of an element of the CAD model in a given position, the PTZ camera is positioned and an image of the element is taken in this position. At the end of this procedure, we have a set of images made of theoretical and real images (cf. Fig. 2).

The hypothesis here is to consider that the parameters used to generate the theoretical image are sufficient to help us position the real camera in order to have two comparable images. In fact, our main goal during this work is to be able to compare one theoretical image with one real image, assuming that they are aligned. This means that the image generated from the CAD model and the sensed image share the same coordinate frame. If the test image is nondefective, then the object of interest is located at nearly the same position in both images. It is important to highlight that



Fig. 1 Acquisition system: PTZ camera and the directional light device.

the goal is to compare two images. The reference image, which is a theoretical image, has to be compared with the test image in which the object of interest can be incorrectly mounted by the operator who can mount an object having a wrong size, or an object that can be displaced/rotated relatively to what is expected in the CAD-derived image. Therefore, we will be considering that perspective distortions can be neglected. However, a 2-D image registration in terms of rotation and translation may be suited in order to be able to compare the test image with its reference. Comparing a theoretical image with a real image is a challenging task because the first is created using mathematical object descriptions and the second captures the object description via optics. In that sense, the first question to ask is: What kind of information can be used to achieve a robust comparison of these images? In the next section, we describe our CAD-based matching method to compare an image generated from a CAD model with an image acquired with the PTZ camera, using primitives extracted from the contours.

3 Computer-Aided-Design-Based Matching of Primitives from a Reference and a Test Image

In this section, we describe our matching method, starting with a short literature review in Sec. 3.1. In Sec. 3.2, we describe which primitives are used in the matching process and how they can be extracted from the images. Also in this section, we explain how the primitive extraction problem is related to the problem of image registration in the peculiar context of image matching. The matching process is accomplished in two stages. First, in Sec. 3.3, two attributed graphs representing the reference and the test objects are constructed. Then, we compute a similarity function that provides a matching score for each pair of graph nodes. Finally, the similarity scores are used in Sec. 3.4 as edge weights of a bipartite graph that is optimized in order to keep only the best weighted edges. After bipartite graph optimization, we guarantee one-to-one matching at the primitive level, which means that we find the correct line segments and ellipses in the test image that best match their counterparts in the reference image.

3.1 Related Works

Matching is the task of finding correspondences between elements of data schema or data instances.¹⁰ Matching is applied in numerous fields of science such as computer science (namely in data warehousing and e-business), bioinformatics, and image analysis. In image analysis and computer vision, the matching task has been widely studied.¹¹⁻¹³ It has applications in 3-D reconstruction, object retrieval or object recognition, automated inspection, disease diagnosis, and so on. The matching task varies a lot and so do matching techniques. In image analysis, matching techniques include descriptor-based methods [scale-invariant feature transform (SIFT)¹² and speeded up robust features (SURF)¹⁴], cross-correlation (CC)-based methods (CC, normalized cross-correlation, zero mean normalized cross-correlation bounded partial correlation¹⁵), template matching,¹⁶ shape-based matching, model-based matching, geometry-based matching, and hybrid approaches. In our method, the goal is to match a CAD-derived image with an image acquired by a camera. In that sense, approaches relying on image intensity such as descriptor-based matching

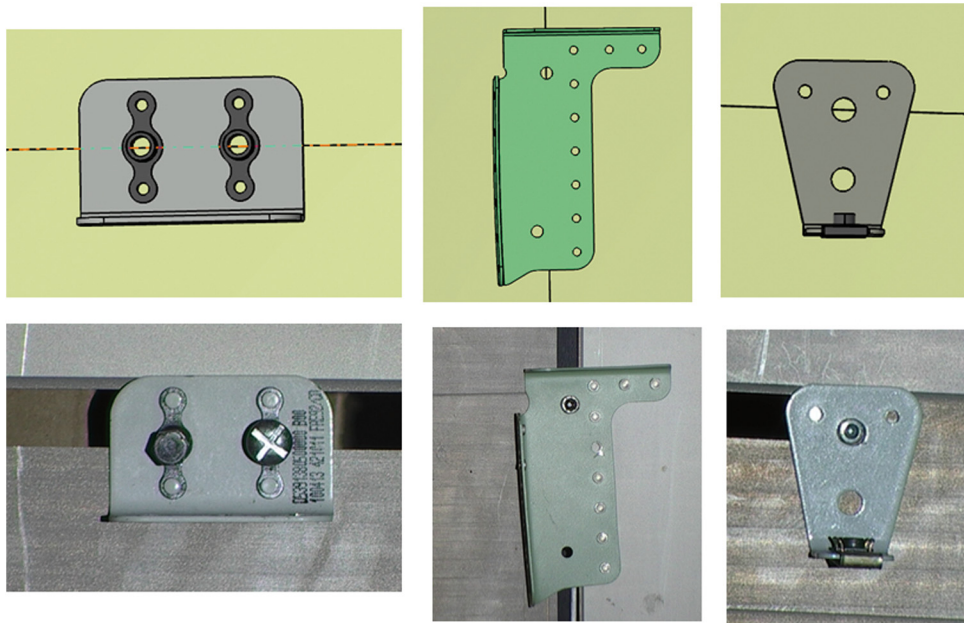


Fig. 2 Set of images generated from CAD model (top) and real images acquired with PTZ camera (bottom).

techniques (like SIFT and SURF), classic template matching, or classic correlation-based matching techniques are not applicable to our data. In our approach, we try to match geometric information, as it is the only really relevant information in our data. Thus, our method belongs to the group of geometry-based matching methods. This group of methods has received a lot of attention from the computer vision and image analysis communities, and it includes works on topology matching^{17–19} as well as works relying on similarity measurements of models.^{13,20,21} Our approach is inspired by the method suggested by Fishkel,²⁰ which is a 3-D application of graph matching using a bipartite graph and tree search. In our 2-D method, first, we compute similarity measurements between two attributed graphs. The similarity scores are later used as edge weights in a bipartite graph in which a mutual best match search is carried out. The final step of our work concerns the calculation of a global score, which combines the result of ellipse matching with the result of line matching in order to validate or refuse the inspected element.

3.2 Feature Extraction and Contour Registration

Most of our mechanical elements (Fig. 2) contain several elliptical elements. Segments are also one very common feature in our images. Therefore, these are the features used in our approach. Extracting these features is a known problem in computer vision. Before diving into the core of the line and ellipse extraction problem, another point that needs to be taken into consideration is contour registration. Given two or more images representing the same scene (taken at different times, points of view, or with different sensors), image registration^{22,23} is the process of bringing them into the same coordinate frame so that they can be overlaid.²³ In our case, we have an image generated from the CAD model and a sensed image acquired with a PTZ camera, as shown in Fig. 2. The raw information on these images cannot be used directly for inspection purposes. Indeed, they represent the same object, but the only common information

they share is the object geometry. Thus, we use contours of objects in both the CAD-derived image and the sensed image. The application is designed to work with real sensed images in the end. However, in this study, the sensed image will be simulated by an image also generated from the CAD model, exhibiting some comparative differences to the reference image. Several examples of test images and their respective contours are shown in Fig. 3. The ultimate goal is to extend the study and compare the results obtained with synthetic data with those of future tests that will be done using experimental data. This will be the object of our next work.

As explained in Sec. 2, objects of interest in a test image can be mounted incorrectly by the operator compared to what is defined in the CAD model. As a result, a 2-D image registration may be suited before image comparison. An example of such a situation is shown in Fig. 4. The element in the test image is translated compared to the reference image. We will explain how this registration is done using contours. When we look Fig. 4, it is obvious that it is necessary to register the reference image in green with the test image in magenta before comparing them. The registration stage will yield the transformation matrix, from which we can compute the parameters (rotation angle and translation vector) that allow the overlaying of green and magenta contours from Fig. 4. Image registration has received a lot of attention from researchers from both the image analysis and computer vision communities. In fact, several applications in different fields of these domains, such as 3-D reconstruction, medical image analysis, inspection, and others require images to be aligned. Depending on the type of images to be aligned, methods such as SIFT, phase correlation, optimization using regular step gradient descent, and so on have been successfully applied. In our application, most of the images can be aligned using regular step gradient descent optimization in a process that tries to minimize mean-square error or to maximize mutual information.²³ For other images,

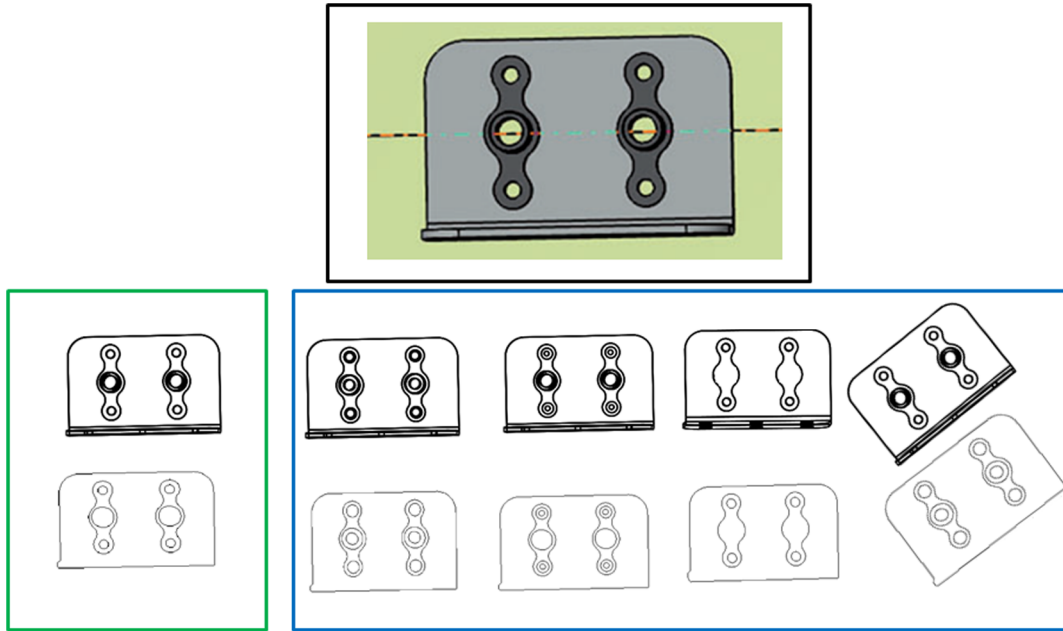


Fig. 3 First row: screenshot of CAD model of an element to be inspected. Second row: left, scalable vector graphics (SVG) image generated from the CAD model (reference image) and its contour extracted after bitmap conversion; right, SVG image generated from the altered CAD model (test images) and their respective contours extracted after bitmap conversion.

registration is accomplished using a phase correlation algorithm.^{24,25} For the example shown in Fig. 4, we obtained the result presented in Fig. 5 after a registration using regular step gradient descent minimizing the mean-square error between the two input contours. For a 2-D rigid registration of two images, the transformation matrix (TM) composed of a rotation matrix $R(\theta)$ and a translation vector $T = (t_x, t_y)^t$ is written as follows:

$$TM = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

The visual result shown in Fig. 5 is obtained by applying the transformation matrix [Eq. (2)] to the reference contour, causing it to be aligned with the test contour. From Eq. (2), we can obtain the registration parameters (rotation angle $\theta_1 = 0.2$ deg and translation vector $T_1 = [-127.804, -65.295]^t$):

$$TM_1 = \begin{pmatrix} 0.999 & 0.0004 & -127.804 \\ 0.0004 & 0.999 & -65.295 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

Now, with the two contour images registered, we can move a step ahead in our goal to compare them. As said previously, we are going to extract primitives (line segments and ellipses) from the two images. Later, they will be described and matched.

Mainly, two large families of methods are used to extract primitives like ellipses or line segments from an image. The first one may derive from the Hough transform²⁶ and uses a vote system or is based on curve fitting using an optimization least square-like process. It has been proven that these categories of methods can be used for both line and ellipse detections. The standard Hough transform²⁶ is a reference for line detection, and it was generalized to deal with other parametric curves such as ellipses. Improved versions of the Hough transform, e.g., the randomized Hough transform, have also been successfully used in line and ellipse

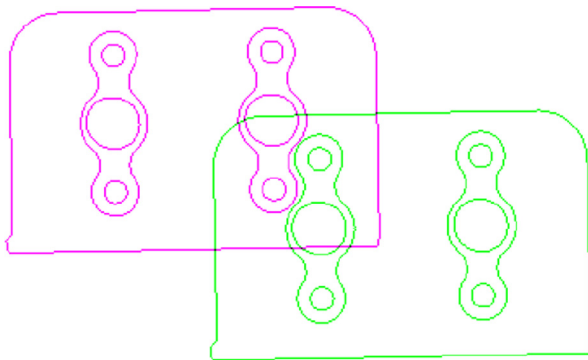


Fig. 4 Two contours not registered: contour of test (left) and contour of reference image (right).

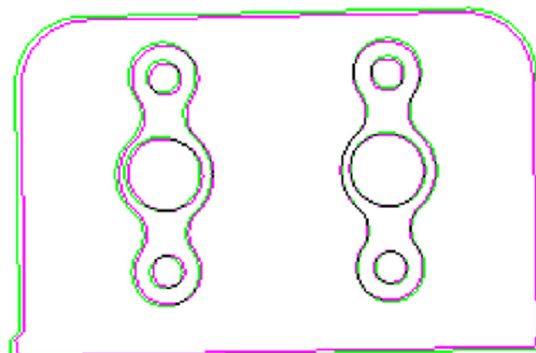


Fig. 5 Test and reference images after the registration process.

extractions.²⁷ A new reference line segment algorithm is presented in Ref. 28. Prasad et al.²⁹ presented an ellipse-fitting algorithm allowing good ellipse extraction. Patraucean et al.³⁰ presented a parameterless ellipse and line segment extraction algorithm. The second category of methods for parametrized curves is based on edge grouping.^{31,32} Methods in this group have been successfully used for ellipse and elliptical arc detection. It is worthy of mention that other classifications may be different from this one. For instance, Wang et al.³¹ divide ellipse extraction algorithms into three groups, including in their classification genetic algorithm-based methods.

In this work, we are not concerned about developing a new feature extractor but rather with using one of the known algorithms to extract our primitives that will be further matched. The Prasad algorithm²⁹ performs well in our images for ellipses detection as soon as the images are correctly segmented. Currently, our images are segmented using some preprocessing strategies based on mathematical morphology combined with connected component labeling techniques. Another algorithm that has been proven to work well for ellipse and line detection is the ellipse and line segment detector (ELSD) by Patraucean et al.³⁰ The advantage of using ELSD is that we obtain both ellipses and line segments using one single algorithm. However, even using well-segmented images, the result of the ELSD algorithm may need to be filtered, as it can yield short segments or concentric ellipses. For that reason, we chose to use the Prasad algorithm²⁹ to find ellipses in our images and the classic Hough transform algorithm for finding line segments. Figure 6 shows one example of primitives extracted from the reference and the test images after contour registration. These two sets of primitives are the input of our matching method described in the next sections. Ellipses in the reference image are labeled from 1 to P ; in this example $P = 6$. Therefore, the labels for ellipses in the test image start from $P + 1$. Similarly, line segments in the reference image are labeled from 1 to L ($L = 4$ in this example). Subsequently, line segments in the test image contain labels starting from $L + 1$. The same labeling technique is adopted in the next section when we present attributed graphs (cf. Fig. 7). As stated previously, we consider two types of primitives: ellipses and line segments. Each type of primitive can be described by a set of attributes. For instance, one ellipse is represented by five parameters: its center coordinates, the

orientation angle, and the half-lengths of the two axis. A line can be described by its endpoints.

3.3 Feature Matching Using a Match Function

Line segment matching has been a widely studied topic over the last 40 years, as it can be used in several applications such as 3-D reconstruction,³³ object retrieval, and so on. This is also the case for primitives like ellipses.³⁴ In this section, we explain our primitive matching method. Figure 6 shows an illustration of such primitives.

3.3.1 Preliminaries

Given two sets of primitives X and Y , with X representing primitives in the theoretical image and Y representing primitives in the real image, we will show how a match score $s(X, Y)$ denoting the similarity between these two sets of primitives can be computed. For ellipses, from the initial five parameters, we create an attribute vector with four variables: the center coordinates, the orientation angle, and the ellipse area (computed using the half-lengths of the two axis). For line segments, from the endpoints, we construct the attribute vector also composed of four variables, the midpoint coordinate of the line, its length, and its orientation angle. For each primitive set, we construct a graph (basic concepts of graph theory are presented in the next section). The graph representing theoretical primitives is composed of N nodes (for N theoretical primitives) and the one for the real dataset contains K nodes (representing the K primitives in the real image). Each graph node is associated with the attribute vector describing the represented primitive. Each graph also contains a set of edges linking two nodes. Two primitives are linked by relative neighborhood relationship.³⁵ However, any other proximity relationship could be convenient, as we are not exploiting the information from graph edges; all the relevant information (attributed vector) is assigned to the nodes. The choice of using an attribute graph was made because this graph is an effective way of representing objects³⁶ and is standardized and easy to handle (we can assign as many attributes as we wish to the nodes and to the edges). Figure 7 shows four graphs corresponding to the two sets of ellipses and line segments shown in Fig. 6. The attribute graphs have their nodes associated with an attribute vector. In Fig. 7, we illustrate this by adding an arrow to one node of each graph.

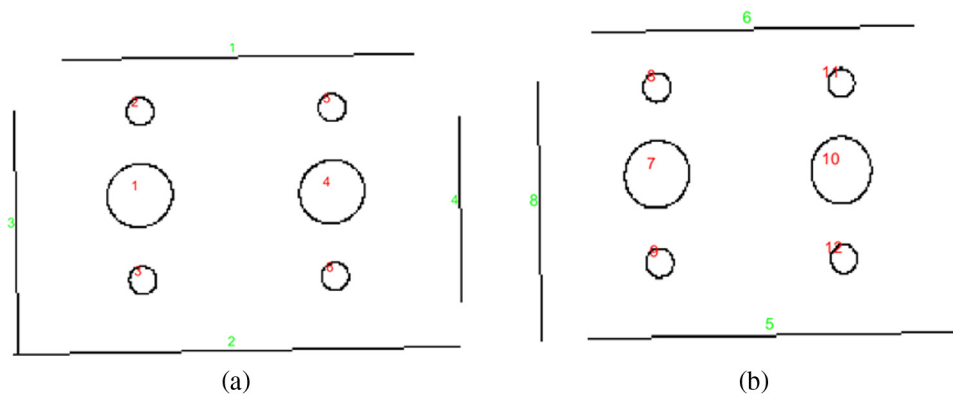


Fig. 6 Two sets of primitives (line segments and ellipses) extracted from the reference image (a) and test image (b) after contour registration.

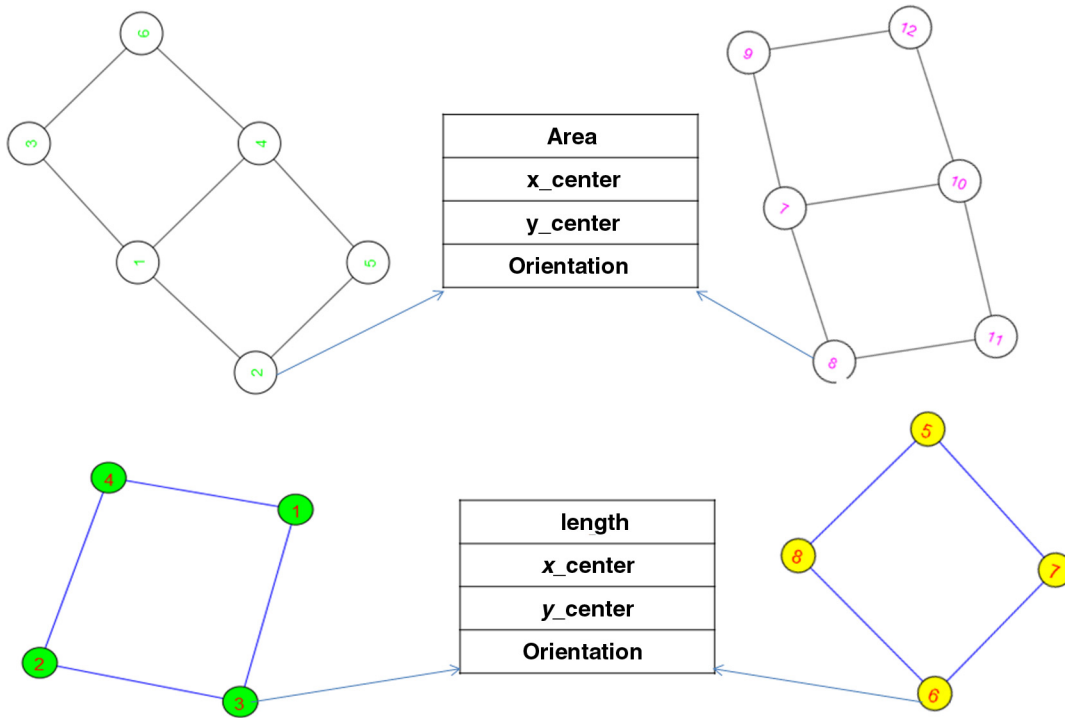


Fig. 7 Attributed graphs for the two sets of primitives shown in Fig. 6. First row: graph corresponding to ellipses in reference image (left), and graph corresponding to ellipses in test image (right). Second row: graph corresponding to line segments in reference image (left), and graph corresponding to line segments in test image (right).

3.3.2 Basic graph theory concepts

Given a graph $G = (V, E)$, the set $N_G(v) = \{u \in V(G) | uv \in E\}$ denotes the neighborhood of a vertex v . The degree of a vertex v is $|N_G(v)|$ and is denoted by $d_G(v)$. If $d_G(v) = 0$, then v is called an isolated vertex. The in-degree of a vertex v is the number of edges having the arrow pointing toward v . The out-degree of a vertex v is the number of edges having the arrow going out from v . A graph $G = (V, E)$ is said to be bipartite if $V(G)$ can be partitioned into two disjoint sets X and Y such that every edge $e \in E$ joins a vertex in X to another vertex in Y . A partition (X, Y) of V is called a bipartition. A bipartite graph with bipartition (X, Y) of V is denoted by $G = (X, Y, E)$. Most of these definitions can be found in Ref. 37 and in the Python igraph library documentation. Other standard concepts in graph matching include graph or subgraph isomorphism and maximal common subgraph.³⁸ In graph matching, as often as possible, exact graph matching is carried out through isomorphism and maximal common subgraph. Other times, inexact graph matching is needed.¹³ As noted in Refs. 21 and 38, in real-world applications, it is not always possible to use concepts such as graph isomorphism or maximal common subgraph because of the imperfection of the real-world data. In such cases, an error-tolerant graph matching approach or a method that computes a measure of similarity between two given graphs is suitable, which is the case in the work described in this paper.

3.3.3 Match function

The match score is computed in the following manner. We divide the attribute variables into two classes ($c = [c_1, c_2]$).

Attribute variables such as area or length belong to the class c_1 , as we can compute their ratio [cf. Eq. (3)]. Computing the ratio for attribute variables such as center coordinate or orientation is, however, not meaningful. Therefore, these attributes will be in class c_2 . For such attributes, the similarity score is computed using the absolute difference between the theoretical and the test primitives [cf. Eq. (4)]. In this calculation, a maximal accepted disparity δ_i between the theoretical and the test primitives is included in order to normalize the score. For the i 'th attribute, the maximal accepted disparity δ_i is a threshold that specifies the maximal expected difference between the test attribute and its theoretical counterpart. For instance, if we are considering a line orientation, then the maximal accepted disparity stands for the maximal difference between the angle value expected for the theoretical line and the actual value of the angle measured for the test line. This maximal accepted disparity δ_i is chosen based on the user requirements, so it can vary from one application to another one.

To summarize, for ellipses, the class c_1 contains ellipse areas as attribute variables; for both reference and test primitives, the ratio of the two areas is computed and normalized to be in the range $[0, 1]$. The class c_2 contains ellipse orientations and ellipse centers as attribute variables for the two types of primitives (reference and test). The center coordinates are used to compute the euclidean distance between the two primitives. A score produced by attributes from the class c_2 is in the range $[-inf, 1]$. In our method, only candidates having scores in the range $[0, 1]$ for both c_1 and c_2 are considered for matching purposes. For line segments, the class c_1 contains the segment lengths as attribute variables for the reference and the evaluated line segments. The class

c_2 contains the midpoints and the orientations of the segments as attribute values. The similarity between a theoretical and a test primitive is then the summation of the scores of each individual attribute (length or area, orientation, and euclidean distance). This match function was proposed by McIntosh and Mutch³⁹ for line matching. We extended the concept in order to use it with ellipses as well:

$$s_i = \frac{\min(X_{k_i}, Y_{m_i})}{\max(X_{k_i}, Y_{m_i})}, \quad \text{if } i \in c_1, \quad (3)$$

$$s_i = \frac{\delta_i - \text{abs}(X_{k_i} - Y_{m_i})}{\delta_i}, \quad \text{if } i \in c_2, \quad (4)$$

$$s(X_k, Y_m) = \text{MF}(X_k, Y_m) = \sum s_i w_i, \quad (5)$$

where X_k stands for a primitive in the theoretical image, whereas Y_m is a primitive in the test image and i denotes the i 'th attribute for a given primitive. δ_i is the maximal accepted disparity for the i 'th attribute in the class c_2 . The scalar s_i is the match score between one theoretical primitive with one test primitive for the i 'th attribute. As some attributes may be more relevant than others,¹³ a priority weighting function may be suitable. In our case, we follow the weighting technique proposed by McIntosh and Mutch.³⁹ It is represented in Eq. (5) by w_i . For each attribute variable for which a score is computed, a priority P_i is defined between 1 and 10. A value $Q_i = 11 - P_i$ is then created. Subsequently, $T = \sum Q_i$ is created; $w_i = Q_i/T$. The important point in this weighting function is how to define P_i . Assigning a low priority to an attribute variable corresponds to giving it a high weight value. If it is not possible to distinguish which attribute variables deserve which priority, then it is better to give them the same priority. This technique leads to a value w_i , which is always in the range $[0, 1]$. Finally, $s(X_k, Y_m)$ denotes the match score between X_k with Y_m .

The matching algorithm is accomplished by exploring all the nodes in the theoretical graph and by searching for the most similar nodes in the graph representing the test image. The similarity criterion used during the search process is the match function described previously. During this process, a matching matrix is computed. The matching matrix links a primitive in the set X with a primitive in Y (cf. Table 1). This matching matrix is very important, as it is used later to construct the bipartite graph.

For the matching matrices shown in Tables 1 and 2, the maximal disparity for attributes corresponding to the center coordinates is 40 pixels and 10 deg for the orientation. In this example, both reference and test images have the same number of ellipses and line segments, and the two contour images were correctly registered. Therefore, there is no need to do an optimization process in order to find best matches, as we have one-to-one correspondences already. Nonetheless, in some situations, a one-to-several match is possible (see Fig. 8). In this example, the reference image has 6 ellipses, whereas the test image has 10. One green ellipse from the reference image may match two magenta ellipses from the test image (cf. Fig. 8, second row). This is highlighted in

Table 1 Matching matrix between test and reference ellipses shown in Fig. 6.

		Ellipses from test image					
		1	2	3	4	5	6
Ellipses from reference image	1	0.980	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.971	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.980	0.0	0.0	0.0
	4	0.0	0.0	0.0	0.983	0.0	0.0
	5	0.0	0.0	0.0	0.0	0.973	0.0
	6	0.0	0.0	0.0	0.0	0.0	0.975

Table 2 Matching matrix between test and reference line segments shown in Fig. 6.

		Line segments from test image			
		1	2	3	4
Line segments from reference image	1	0.0	0.969	0.0	0.0
	2	0.863	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.976
	4	0.0	0.0	0.859	0.0

different colors in the matching matrix of ellipses shown in Fig. 9.

In such a situation, we have to find the best matches in order to satisfy the uniqueness constraint of the matching. In the next section, we explain how the best match search is performed, using a bipartite graph.

3.4 Search of Mutual Best Match in a Bipartite Graph

Bipartite graph matching has been widely studied and has applications in various fields of science, such as data mining,⁴⁰ mathematics,³⁷ computer vision, and image analysis.^{20,21,41} By its definition, it is obvious that it is particularly suitable for a two-class matching problem. In our method, it is used as follows. First, we count the number M of occurrences of a score in the matching matrix meeting a predefined threshold. A bipartite directional graph containing M edges is then created. This graph holds in it two types of nodes (X, Y), representing theoretical and real primitives, respectively. An X -type node in the bipartite graph is connected to a Y -type node when their similarity score satisfies the threshold condition. The edge linking these two nodes is weighted with their similarity score. The threshold is in the range $[0, 1]$, and it has to be set to a low value, depending on the level of similarity desired in the application. The use of a threshold is optional and aims at reducing the size of the bipartite graph without discarding potential good candidates

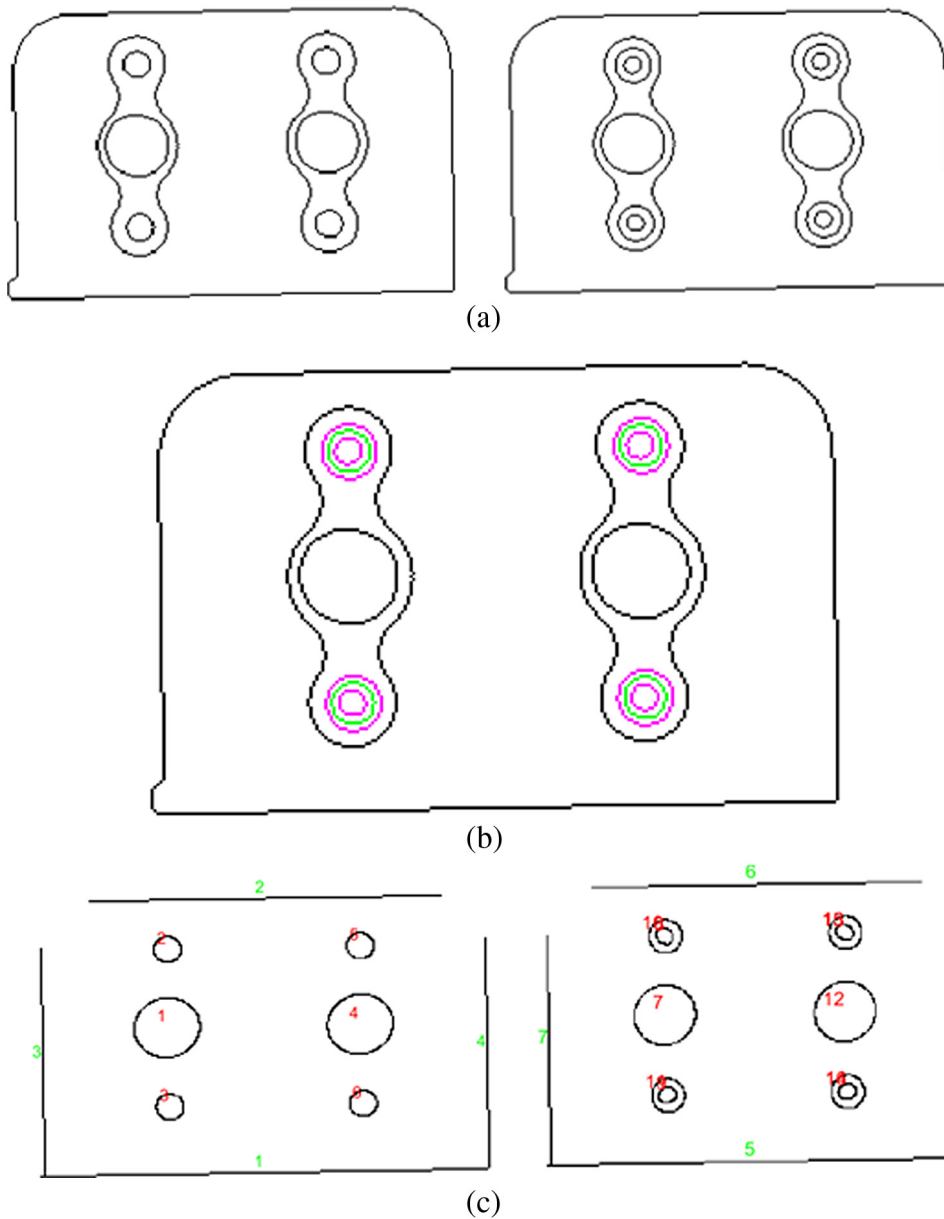


Fig. 8 (a) Contours of reference (left) and test (right) images. (b) Contours of reference and test images aligned. (c) Extracted primitives from the contours.

for matching. If we choose not to use the threshold, the bipartite graph will be larger, as each reference node will be connected to all test nodes. In the bipartite graph, one *X*-type node may be connected to more than one *Y*-type node and vice versa. In that sense, an optimization process may be required in order to filter out edges having low scores and uniquely keep the mutual best matches. The mutual best

match search algorithm in the bipartite graph is divided into two stages. In the first stage, best matches are found for edges linking a *Y*-type node with an *X*-type node as follows.

Let *k* be a *Y*-type node; if the in-degree of *k* is 1, add the edge linking *k* and one *X*-type node to the best match list. Otherwise, if the in-degree of *k* is greater than 1, then search the connected edge to *k* holding the maximal weight. Add this edge to the best match list. Add the other edges to the pruning list. As the edges in the bipartite graph are oriented, this step can be considered a target-to-source best search. In the second stage, the best matches in a source-to-target sense are found using the out-degree of the nodes. The mutual best match is kept and all nonmutual best match edges are eliminated from the graph. The mutual best match search guarantees respect of the uniqueness constraint of the matching. Figure 10 shows the bipartite graph

		Ellipses from test image									
		1	2	3	4	5	6	7	8	9	10
Ellipses from reference image	1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.864	0.0	0.830	0.0	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.876	0.0	0.837	0.0	0.0	0.0	0.0	0.0
	4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	5	0.0	0.0	0.0	0.0	0.0	0.0	0.870	0.0	0.821	0.0
	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.889	0.0	0.837

Fig. 9 Matching matrix illustrating one-to-multiple correspondences for the ellipses in Fig. 8, third row.

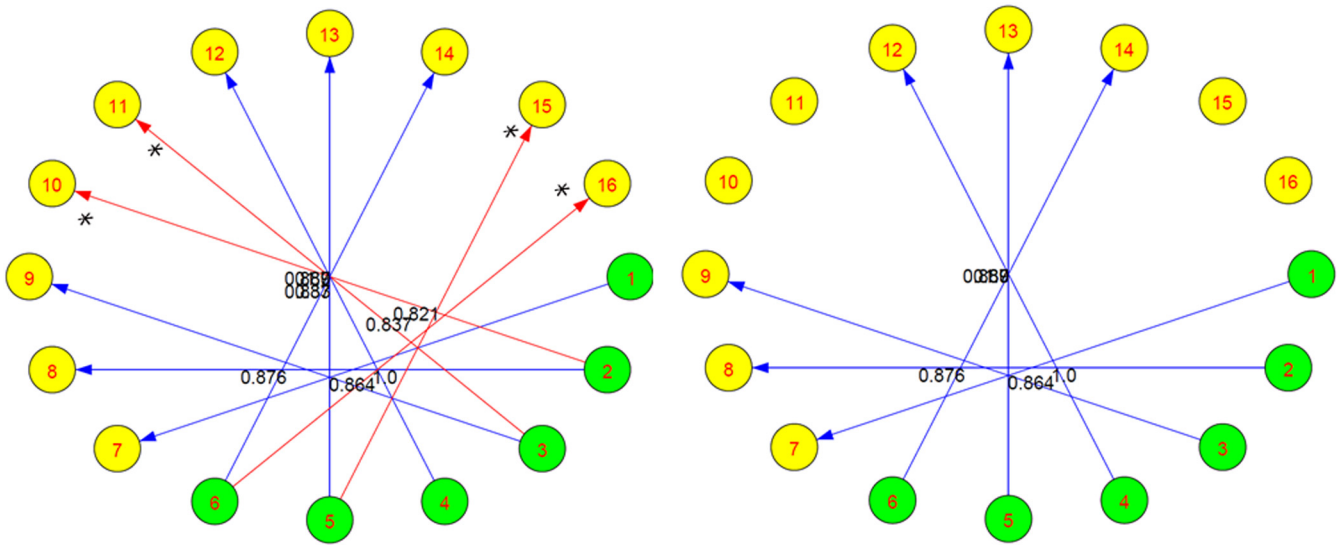


Fig. 10 Best match search illustration. Bipartite graph before mutual best match search, with nonbest matches (annotated with a star) (left). Bipartite graph after best match search (right).

corresponding to the ellipses match presented in Fig. 8 before and after the mutual best match search.

Once the uniqueness constraint of matching is verified, we can move to the final step of the method and combine the results of ellipse matching with those of line segment matching in order to achieve a global score that indicates how close the test image is to the reference image.

4 Making the Conformity Decision

Before combining the results of ellipse and line segment matching, it is worthwhile to remember that the input images (contours) undergo a rigid registration stage before matching. Therefore, registration parameters (rotation angle and translation vector) have to be taken into account in the decision process. Thus, they will be provided in addition to the final score. In fact, we cannot set limit values for these parameters, as they are highly related to customers' application. For instance, a user "A" may decide that elements of his inspected part are not defective for a certain translation value, whereas another user "B" will say the opposite for the same translation value. In order to keep the method as flexible as possible, we imagine registration parameters to be provided as gauge indicators that have to be set out by the user.

As said previously, if we have one-to-one matches for both ellipse and line segment matching, we can combine them to make a global score. This is done as follows:

$$s_{e,l} = \frac{1}{2} \left(\frac{1}{P} \sum_{i=1}^{M_1} e_i + \frac{1}{L} \sum_{j=1}^{M_2} l_j \right). \tag{6}$$

In Eq. (6), $s_{e,l}$ represents the global score taking into consideration ellipses e and lines l . P represents the number of expected ellipses in the reference image and L stands for the number of line segments expected in the reference image. M_1 and M_2 are the number of ellipses and line segments from the reference image that matches ellipses and line segments from the test image; e_i and l_j are the match scores for i 'th ellipse and j 'th line segment in the reference image. Having a global score, it is easy to make the conformity decision. For instance, taking into consideration the matching matrix for

ellipses (Table 1) and for line segments (Table 2), the test image from Fig. 4 yields the following global score:

$$\text{score}_1 = \frac{1}{2} \left(\frac{0.980 + 0.971 + 0.980 + 0.983 + 0.973 + 0.975}{6} + \frac{0.969 + 0.863 + 0.859 + 0.976}{4} \right) = 0.9469.$$

As said previously, the decision has to be made in a flexible way that allows the application to be used by different customers. Thus, the global score and rigid registration parameters [cf. transformation matrix (Table 2)] should be fixed according to the customer's requirement. In that sense, the decision parameters for this element will be

- score: 0.9469;
- translation: $t_{x_1} = 127.804 \text{ px} \rightarrow 25.4 \text{ mm}$, $t_{y_1} = 65.295 \text{ px} \rightarrow 13.0 \text{ mm}$ [object size: (46 mm \times 30 mm \rightarrow 231 px \times 150 px)]; and
- rotation angle: $= \theta_1 = 0.2 \text{ deg}$.

The decision elements are compared with the limit values preset by the user. If they are in the range fixed by the user, then the element is said to be nondefective.

5 Results and Discussion

The present method was tested with synthetic data and has proved to work well. It is capable of finding correct matches for ellipses and for line segments. Test cases include missing elements, displaced elements, size changes, and combinations of these cases. In all these cases, the method performs well. In addition to the test image from Fig. 4, taken to illustrate the approach, the next paragraphs show the results for some other cases.

Let us consider the test image from Fig. 8. This test image has 10 ellipses, whereas the reference image has only 6 ellipses. The method was capable of matching the correct ellipses, as shown in the bipartite graphs in Fig. 10. For this image, the final result at the end of the procedure is

- score: 0.958. For this element, the score is high in spite of the fact that the test element has more ellipses than the reference. As we know it as a prior information, we can use an alert to let the user know that in addition to what was expected to be in the CAD model, we found four supplementary ellipses in the test image;
- translation: $t_{x_2} \sim 0$ mm, $t_{y_2} \sim 0$ mm; and
- rotation angle: $\theta_2 \sim 0$ deg.

This result takes into consideration the matching matrix (Fig. 9) for ellipses, which was combined with the matching matrix for line segments (we have a perfect match for line segments). It also includes the registration parameters extracted from following transformation matrix [Eq. (7)], which is obtained with a regular step gradient descent that minimizes the mean-square error between the test and reference images:

$$TM_2 = \begin{pmatrix} 1.0 & 0.0004 & -0.0012 \\ 0.0004 & 1.0 & -0.0059 \\ 0 & 0 & 1 \end{pmatrix}. \quad (7)$$

Now, consider the input contours shown in Fig. 11. In this example, the test image is translated and it has fewer ellipses than what is expected in the reference image. Therefore, the score is also lower than in the previous case.

For this element, the registration parameters come from the following transformation matrix:

$$TM_3 = \begin{pmatrix} 1.0 & 0.001 & 1.7 \\ 0.001 & 1.0 & 172.1 \\ 0 & 0 & 1 \end{pmatrix}. \quad (8)$$

After the registration and primitives matching, we obtain the matches shown in Tables 3 and 4.

At the end of the procedure, the final result is

- score: 0.7945. This is the global score (gathering the contribution of ellipses and line segments) given by the application. Additionally, we report that there are missing elements;
- translation: $t_{x_2} \sim 1.7$ px \rightarrow 0.3 mm, $t_{y_2} \sim 172$ px \rightarrow 34.4 mm; and
- rotation angle: $\theta_2 \sim 1$ deg.

One important point that deserves to be highlighted concerns rotated elements. Finding the transformation matrix to register a rotated test image with the reference image is not a problem. Indeed, we can find the transformation matrix that allows the registration to be done, as long as the rotation is not too important (up to 45 deg). This is illustrated by Fig. 12, in which a case of a 45-deg rotation between the test and the theoretical primitives is presented. If the CAD model is correctly exploited to generate the theoretical image, this case of an important rotation between the test image and the reference image would never occur. However, as the application deals with the inspection of part assembly, one element may be awkwardly assembled; this is the reason why we would like to be capable of assessing such cases as well. A transformation matrix for the rotated test image in Fig. 12 is

$$TM_4 = \begin{pmatrix} 0.707 & -0.707 & -113.4 \\ 0.707 & 0.707 & 708.6 \\ 0 & 0 & 1 \end{pmatrix}. \quad (9)$$

In this section, we have shown some test cases, including translation, rotation, and missing elements. Even if tests exhibiting size changes were carried out, we did not show examples with varying zoom factors. In fact, all the tests were carried out using synthetic data. We plan to show tests with experimental data in the future. Then the impact of varying the zoom factor will be quantified. Currently, we can assess cases in which the test element and its reference

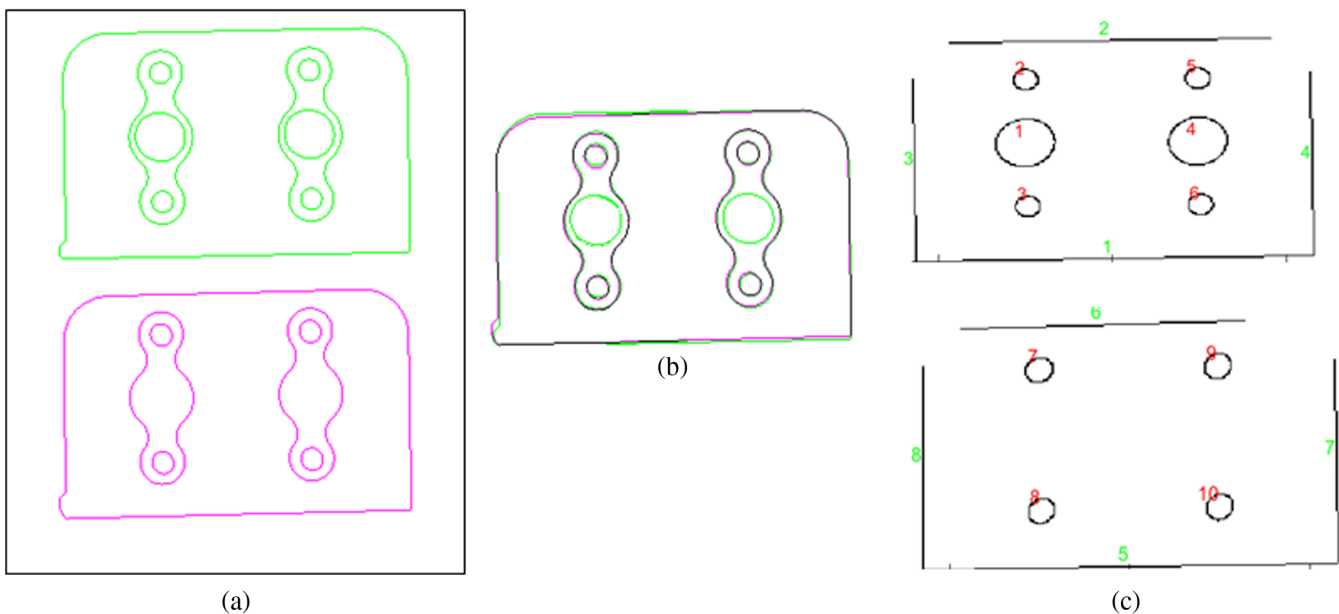


Fig. 11 Illustration of a test case with missing elements. (a) Input contours of reference (top) and test (bottom) images before registration. (b) Input contours after registration. (c) Primitives extracted from the input contours after registration.

Table 3 Matching matrix between test and reference ellipses segments shown in Fig. 11.

	Ellipses segments from test image				
	1	2	3	4	
Ellipses segments from reference image	1	0.0	0.0	0.0	0.0
	2	0.964	0.0	0.0	0.0
	3	0.0	0.962	0.0	0.0
	4	0.0	0.0	0.0	0.0
	5	0.0	0.0	0.969	0.0
	6	0.0	0.0	0.0	0.969

Table 4 Matching matrix between test and reference line segments shown in Fig. 11.

	Line segments from test image				
	1	2	3	4	
Line segments from reference image	1	0.99	0.0	0.0	0.0
	2	0.0	0.877	0.0	0.0
	3	0.0	0.0	0.0	0.966
	4	0.0	0.0	0.985	0.0

have different sizes, which is equivalent to changing the zoom factor. In such cases, correspondences can still be established and a global score can be given as easily as it is done in cases of rotation or translation. Nevertheless, an exhaustive benchmark to quantify the impact of the zoom factor is to be realized later with realistic data. Also, even if tests using synthetic data have shown good

performance, namely in terms of establishing correspondences between the test element and its theoretical counterpart, it is fundamental to test the application with realistic data, as it is intended to work in a real-world environment. As said in Sec. 6, using realistic data is a work still in progress, and it will be shown in a future paper.

6 Conclusion and Future Work

We present a CAD-based inspection of mechanical parts by means of comparison between a reference and a test image. First, we extract contours from both images and then we register them before extracting primitives (line segments and ellipses) that are described and used to construct two attributed graphs. The graphs are matched using a similarity measurement function. In order to guarantee one-to-one matching, an optimization process is carried out using a bipartite graph. Finally, we compute a global score combining the matching result for line segments and for ellipses. Tests with synthetic data proved that the method works in cases such as missing elements, displaced elements, size changes, and combinations of these cases.

We have shown that as soon as the test image and the reference image are registered, it is easy to make the comparison using our graph matching method based on a similarity function. However, problems may appear after the registration. In fact, during the registration stage, pixels undergo an interpolation. As a result, the contour image obtained from the registration process may be downsampled. In such a condition, it is more difficult to extract primitives from the contour image, as one ellipse can break into several connected components. For this reason, a regular ellipse fitting algorithm may not work. We are currently investigating the best way to tackle this problem concerning element rotation, either by using a more adapted primitive extractor or by changing the registration position in the method flowchart.

Our method relies on the matching of two types of primitives (ellipses and line segments); in the future, we want to enlarge the approach to include other shapes, as ellipses and line segments may not be sufficient to describe all the images. In this paper, we present results of 2-D/2-D comparison using synthetic data. Tests with real experimental data are in progress. Another approach that consists of not using

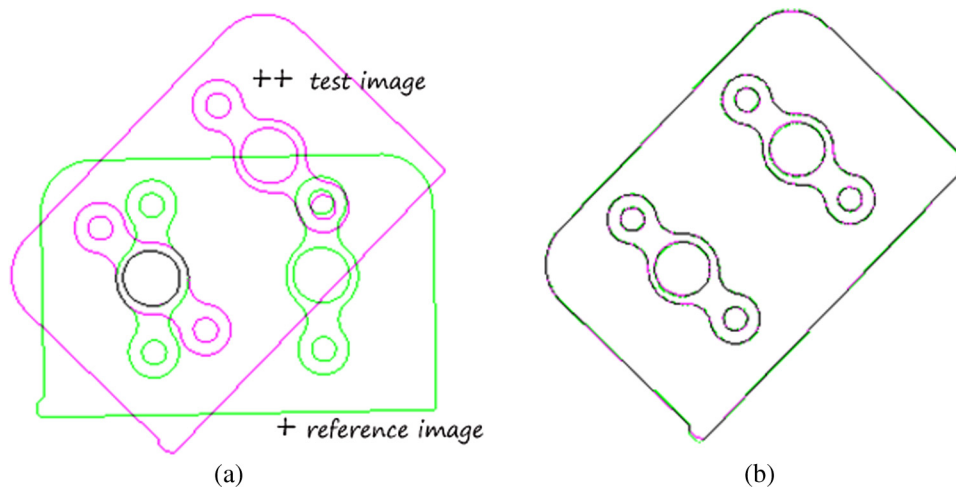


Fig. 12 Example of a rotated image before and after registration with the reference image. (a) Reference image (+) and test image (++) before registration. (b) Reference and test images after registration.

any synthetic images computed from the CAD model is being investigated. In that approach, we will directly match 3-D primitives extracted from the CAD model with 2-D primitives extracted from the real image to be checked. After calibration of the real camera, its pose can be estimated and used to project 3-D primitives from the CAD model onto the image plane (augmented reality) then it will be possible to perform a direct 3-D/2-D matching.

References

1. B. Modayur, L. Shapiro, and R. Haralick, "Visual inspection of machined parts," in *Computer Vision and Pattern Recognition*, pp. 393–398 (1992).
2. T. Newman and A. K. Jain, "A survey of automated visual inspection," *Comput. Vision Image Understanding* **61**, 231–262 (1995).
3. E. Malamas et al., "A survey on industrial vision systems, applications and tools," *Image Vision Comput.* **21**, 171–188 (2003).
4. H. Golnabi and A. Asadpour, "Design and application of industrial machine vision systems," *Rob. Comput. Integr. Manuf.* **23**, 630–637 (2007).
5. M. Bukovec, Z. Spiclin, and F. Pernus, "Automated visual inspection of imprinted pharmaceutical tablets," *Meas. Sci. Technol.* **18**, 2921–2930 (2007).
6. M. Mozina et al., "Automated visual inspection of imprint quality of pharmaceutical tablets," *Mach. Vision Appl.* **24**, 63–73 (2013).
7. A. B. Far, "Analyse multi-images application à l'extraction contrôlée d'indices images et à la détermination de descriptions scéniques," PhD Thesis, Louis Pasteur University, France (2005).
8. S. Bourgeois, "Alignement d'objets mécaniques complexes par vision monoculaire," PhD Thesis, Clermont Ferrand University, France (2006).
9. B. Karabagli, T. Simon, and J.-J. Orteu, "A new chain-processing-based computer vision system for automatic checking of machining set-up application for machine tools safety," *Int. J. Adv. Manuf. Technol.*, 1–22 (2015).
10. S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its applications to schema matching," in *Proc. of 18th Int. Conf. on Data Engineering*, pp. 117–128 (2002).
11. M. Bator and M. Nieniewski, "Detection of cancerous masses in mammograms by template matching: optimization of template brightness distribution by means of evolutionary algorithm," *J. Digital Imaging* **25**, 162–172 (2012).
12. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**, 91–110 (2004).
13. R. M. Cesar, Jr. et al., "Inexact graph matching for model-based recognition: evaluation and comparison of optimization algorithms," *Pattern Recognit.* **38**, 2099–2113 (2005).
14. H. Bay et al., "SURF: speeded up robust features," *Comput. Vision Image Understanding* **110**, 346–359 (2008).
15. L. D. Stefano, S. Mattoccia, and F. Tomabari, "ZNCC-based template matching using bounded partial correlation," *Pattern Recognit. Lett.* **26**, 2129–2134 (2005).
16. K. Briechele and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *Proc. SPIE* **4387**, 95–102 (2001).
17. X. Bai and L. J. Latecki, "Path similarity skeleton graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 1282–1292 (2008).
18. Y. Xu et al., "Skeleton graph matching based on critical points using path similarity," *Lect. Notes Comput. Sci.* **5996**, 456–465 (2010).
19. M. Hilaga et al., "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, pp. 203–212 (2001).
20. F. Fishkel, A. Fischer, and S. Ar, "Verification of engineering models based on bipartite graph matching for inspection applications," *Lect. Notes Comput. Sci.* **4077**, 485–499 (2006).
21. A. Shokoufandeh and S. Dickinson, "Applications of bipartite matching to problems in object recognition," in *Proc. of Int. Conf. on Computer Vision on Graph Algorithms and Computer Vision* (1999).
22. D. Dubey, A. Jain, and U. P. Singh, "An overview on: image alignment and open issues," *Int. J. Adv. Res. Comput. Sci. Software Eng.* **2**(4), 137–142 (2012).
23. B. Zitová and J. Flusser, "Image registration methods: a survey," *Image Vision Comput.* **21**, 977–1000 (2003).
24. B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Process.* **5**, 1266–1271 (1996).
25. R. Kokila and P. Thangavel, "FFT based image registration using corner response," *Int. J. Comput. Sci. Electron. Eng.* **1**, 607–613 (2013).
26. R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM* **15**, 11–15 (1972).
27. R. A. McLaughlin, "Randomized Hough transform: improved ellipse detection with comparison," *Pattern Recognit. Lett.* **19**, 299–305 (1998).
28. R. G. von Gioi et al., "LSD: a line segment detector," *Image Process. Online* **2**, 35–55 (2012).
29. D. K. Prasad, M. K. Leung, and C. Quek, "Ellifit: an unconstrained, non-iterative, least squares based geometric ellipse fitting method," *Pattern Recognit.* **46**, 1449–1465 (2013).
30. V. Patraucean, P. Gurdjos, and R. G. von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *Proc. of the 12th European Conf. on Computer Vision Part II*, pp. 572–585 (2012).
31. G. Wang et al., "A fast and robust ellipse-detection method based on sorted merging," *Sci. World J.* **2014**, 1–15 (2014).
32. T. M. Nguyen, S. Ahuja, and Q. Wu, "A real-time ellipse detection based on edge grouping," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 3280–3286 (2009).
33. H. Bay, V. Ferrari, and L. V. Gool, "Wide-baseline stereo matching with line segments," in *Proc. of Computer Vision and Pattern Recognition*, Vol. 1, pp. 329–336 (2005).
34. M. Hutter and N. Brewer, "Matching 2-D ellipses to 3-D circles with application to vehicle pose identification," in *Proc. of Int. Conf. Image and Vision Computing*, pp. 153–158 (2009).
35. C. D. Correa and P. Lindstrom, "Locally-scaled spectral clustering using empty region graphs," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp. 1330–1338, ACM (2012).
36. E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," PhD Thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France (2002).
37. B. Panda and D. Pradhan, "Minimum paired-dominating set in chordal bipartite graphs and perfect elimination bipartite graphs," *J. Comb. Optim.* **26**, 770–785 (2013).
38. H. Bunke, "Graph matching: theoretical foundations, algorithms, and applications," in *Int. Conf. on Vision Interface* (2000).
39. J. H. McIntosh and K. M. Mutch, "Matching straight lines," *Comput. Vision Graphics Image Process.* **43**, 386–408 (1988).
40. H. Zha et al., "Bipartite graph partitioning and data clustering," in *Proc. of the Tenth Int. Conf. on Information and Knowledge Management*, pp. 25–32 (2001).
41. C. Wang and K. K. Ma, "Bipartite graph-based mismatch removal for wide-baseline image matching," *J. Visual Commun. Image Represent.* **25**, 1416–1424 (2014).

Ilisio Viana graduated from Paul Sabatier University of Toulouse, France, in electrical and computer engineering in 2010. He received his master's degree in artificial intelligence, pattern recognition, and robotics from the same university. Currently, he is a PhD student at École des Mines, Albi, France, and works at Institut Clément Ader (ICA) laboratory, where he studies how CAD models can be employed in vision-based inspection of mechanical parts.

Jean-José Orteu received his PhD in computer vision in 1991. Currently, he is the deputy director of the ICA-Albi Research Center (70 people) located within the École des Mines d'Albi in France. He is also the head of the ICA's Metrology, Identification, Control and Monitoring Group (30 people). His main interest topics are computer vision and automatic control, and he is now more specifically involved in the application of computer vision to 3-D metrology, photomechanics, process monitoring, and NDT.

Nicolas Cornille completed his dual PhD in mechanical engineering and computer science in 2005. He is now R&D manager at G²Metric, a French company specializing in metrology solutions for industrial applications. G²Metric has been developing Lynx software for 2 years and intends to take a big step forward with the use of a direct CAD model.

Florian Bugarin graduated from École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, d'Hydraulique et des Télécommunications in Toulouse, France. He received his PhD in 2012 from Institut National Polytechnique, Toulouse. He was a research engineer at ICA in the Metrology, Identification, Control and Monitoring Group. Since September 2014, he has been an assistant professor at Paul Sabatier University, Toulouse, and a permanent researcher at ICA in the Modeling of Mechanical Systems and Microsystems Group.