A new chain-processing-based computer vision system for automatic checking of machining set-up - Application for machine tools safety

Bilal Karabagli, Thierry Simon, Jean-José Orteu

Journal of advanced manufacturing Technology manuscript No.
(will be inserted by the editor)

A new chain-processing-based computer vision system for automatic checking of machining set-up
Application for machine tools safety

Bilal Karabagli · Thierry Simon ·
Jean-José Orteu

Received: date / Accepted: date

Abstract In high-speed machining it is of key importance to avoid any collision between the machine tool and the machining set-up. If the machining set-up has not been assembled correctly by the operator and does not conform to the 3D CAD model sent to the machining unit, such collisions may occur. This paper presents a new chain-processing-based computer vision system to automatically avoid collision between tool and machining set-up components by checking that the actual machining set-up is in conformity with the desired 3D CAD model used to generate the tool trajectory. This computer vision system utilizes a single camera to automatically check conformity before the start of the machining operation. The proposed solution was tested in different kinds of machining set-ups, and each step of the proposed chain was evaluated. The results show the robustness of the solution for different kinds of machining set-ups.

Keywords Machine tool safety · Machining set-up · Computer-vision-based inspection · Automatic checking

Bilal Karabagli
Université de Toulouse, IUT de Figeac, LRPmip, Avenue de Nayrac, 46100 Figeac, France
Université de Toulouse ; Mines Albi ; ICA (Institut Clément Ader) ; Campus Jarlard, F-81013 Albi, France
E-mail: bilal.karabagli@mines-albi.fr

Thierry Simon
Université de Toulouse, IUT de Figeac, Avenue de Nayrac, 46100 Figeac, France
Université de Toulouse ; Mines Albi ; ICA (Institut Clément Ader) ; Campus Jarlard, F-81013 Albi, France
E-mail: thierry.simon@univ-tlse2.fr

Jean-José Orteu
Université de Toulouse ; Mines Albi ; ICA (Institut Clément Ader) ; Campus Jarlard, F-81013 Albi, France
E-mail: jean-jose.orteu@mines-albi.fr

1 Introduction

"Closed-door machining" is a term used with increasing frequency by aerospace and automotive enterprises to describe a highly automated machining process that aims to shorten production cycles and reduce repair costs. Repairs may be required due to collisions of machine tools with the machine set-up components. Very often, the operator of the machine will be making multiple types of parts every day, which increases the probability that the operator may make a mistake when assembling the machining set-up, leading to a different set-up from the one specified in the 3D Computer-Aided Design model (CAD model) used to generate the tool trajectory. Any proposed solution to this problem involving the installation of a computer vision system within the machine would need to respect industrial constraints, such as the **time response**, **operational simplicity** and **system configuration**. In order to detect a possible collision between the machine tool and the assembly, a computer-vision-based method was developed to automatically verify the machining set-up before launching the machining process.

The proposed solution was tested for different machining set-ups as shown in Figure 1. Column (a) shows the machining set-up composed of a rectangular part fixed by 3 pins and rods. Column (b) shows the machining set-up to provide a horizontal support for a milling operation. It is composed of 5 pins, 2 rods and a moulded part. The last column (c) represents the machining set-up to provide the support of a joystick used for aircraft control. It is composed of a rectangular part and 8 rods.

The set-up shown in Figure 1a presents a sideways positioning and clamping of the part that is to be machined, a parallelepiped, non-complex blank. The support contains numerous fixing holes. Machining is carried out from above, with the tool descending on either side of the clamp to obtain the desired shape. The path taken by the tool could encounter the clamp or the fixing pins if they are not in the correct place. Figure 1b presents a set-up with a pre-shaped blank that has an additional thickness on the right-hand side. This additional thickness must not be interpreted as an additional part. This more complex blank must be perfectly positioned to avoid collisions. Finally, Figure 1c presents a set-up with few holes and with clamping from above. This configuration gives a higher risk of collision if the clamps are badly aligned. The set-up geometry is structured by the boundary lines.

The variety of examples chosen here covers several industrial cases, although not all, since certain companies do not wish to divulge their machining techniques.

In reality, the manufacturing process is sequential and can be divided into three main steps, as shown in Figure 2.

1. The product design step, in which the designer generates a 3-D CAD model(STEP, IGES files format) of the machining set-up.
2. CADCAM programme computation, e.g ISO 3592, using machine type information (3, 4 or 5 axes) and a 3D CAD model.

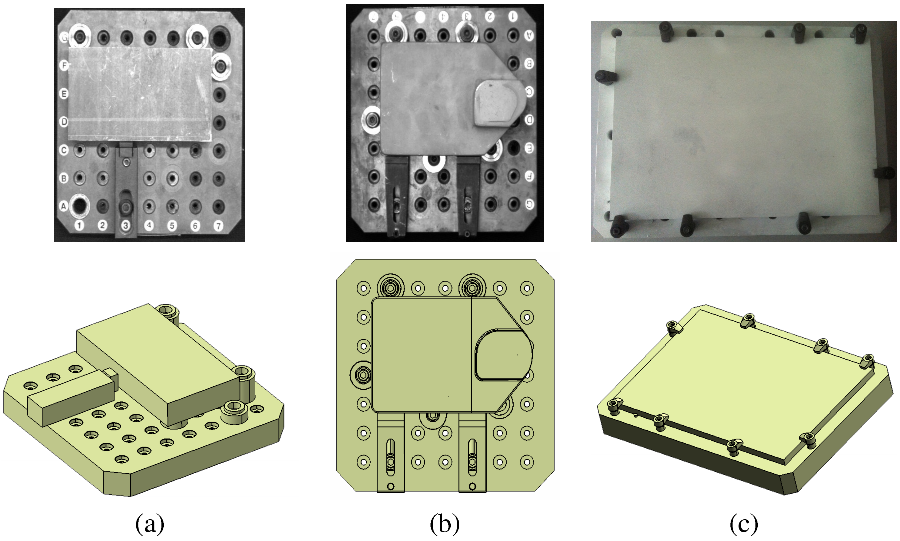


Fig. 1 Different machining set-up configurations.

3. Finally, carrying out the machine set-up and integrating the CADCAM programme into the machine to generate numerical command (NC-code) e.g ISO 6983.

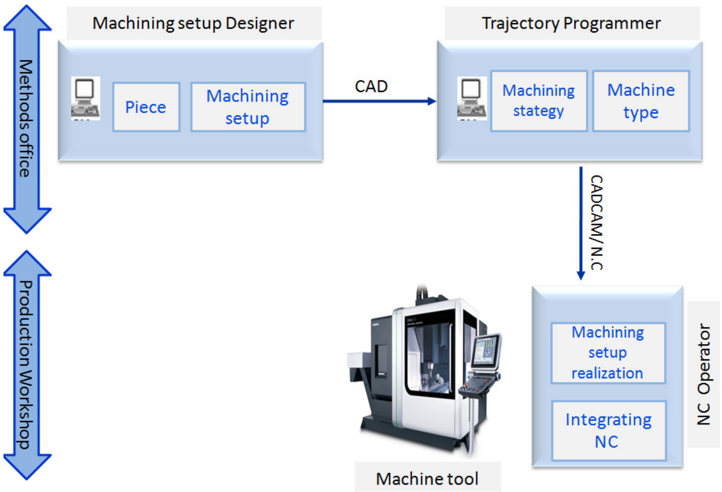


Fig. 2 Actual numerical chain treatment of machining.

In both steps (1) and (2), the designer and the programmer use a simulation software (CATIA©, Mastercam©,. . .) to check the tool trajectory within the

3D CAD model. In the workshop (see step 3), the operator has to assemble the machining set-up defined by the 3D CAD model and integrate the CADCAM programme into the machine. The NC-code is generated by the CAM software and uploaded to the machine controller after being tested on the machining simulation software. This operation is usually unsupervised in the type of case under consideration here (High Speed Machining), where, in general, the operator has to carry out many assembling operations during a day. Figure 3 shows an example of different machining set-ups and the corresponding 3D CAD model (Figure 3e).

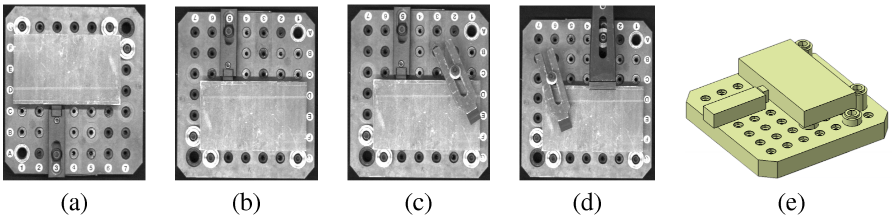


Fig. 3 Examples of errors in the assembly of machining set-ups: (a) machining set-up conforms to 3D CAD model. From (b) to (d) machining set-ups do not conform to 3D CAD model: e.g (b) rods moved, (c) rods moved and fixture element has been added, (d) rods do not have the correct dimensions (as defined in the 3D CAD model) and a fixture element has been added. (e) shows the 3D CAD model of the desired machining set-up.

Going back to Figure 2, it can be seen that the chain is unidirectional and does not have a feedback to control each step of the manufacturing process. Moreover, in the workshop, the operator may use dedicated software to simulate the tool trajectory according to the 3D CAD model, but does not take into account the actual machining set-up because this is a time-consuming operation as the whole machining of the part has to be simulated.

In this work, a new numerical chain is proposed to perform an automatic check of the machining set-up before the machining operation is launched. This new chain is based on computer-vision processing using a single camera. This approach can lead to fast and low-cost solutions as it makes it possible to compare the relevant parameters (CAD, CADCAM) of the numerical chain with the observed scene of the actual machining set-up, as shown in Figure 4.

In the context of collision avoidance in machine tools, some recent works have proved the interest of using computer vision for machine-tool safety, but few of them focus on set-up checking using computer vision: Ahmad *et al.* [1] developed image-processing algorithms (SET/RESET) based on a single camera in order to optimize the tool trajectory. In [2], Zhang *et al.* used a single camera to provide a 3D model of the milling tool, and in [3], Zhang *et al.* developed a stereovision system to construct the digital model based on the actual model of the machining set-up using 3D edge information. In this case, the operator has to verify the Euclidean position of each individual part in a superimposed image to check conformity. Finally, in [4] a single camera was used for a 3D reconstruction and pose estimation of the cutting tool.

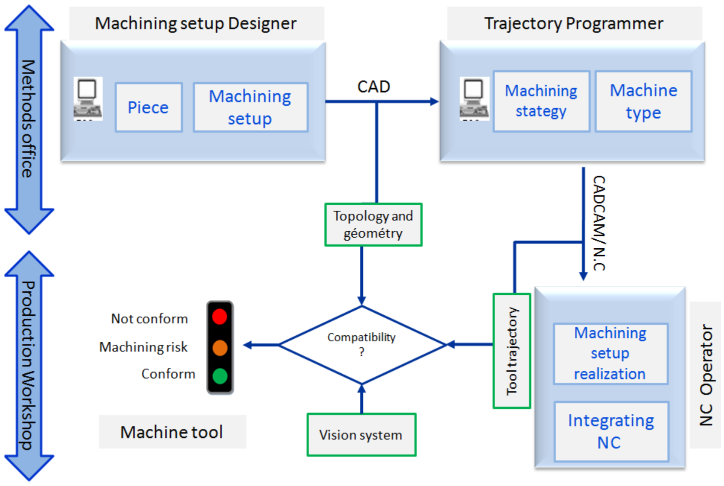


Fig. 4 Proposed numerical chain treatment of machining.

In order to compare the image of the machining set-up with the 3D CAD model of the desired machining set-up, the correspondence between 2D features extracted in the image and those of the 3D CAD model needs to be found automatically This goal can be achieved by estimating the 3D/2D pose between the camera and the 3D coordinate system attached to the 3D CAD model of the installed machining set-up. With the pose, it will be possible to project onto the image some elements extracted from the 3D CAD model (a kind of augmented reality) which will allow us to produce model-based regional segmentation of the image. Each region represents the desired location of a component in the machining set-up (part, rods, clamps, etc.) and by comparing the regions extracted for the components onto the real image, and the defined regions onto the virtual image, it will be possible to evaluate the level of conformity.

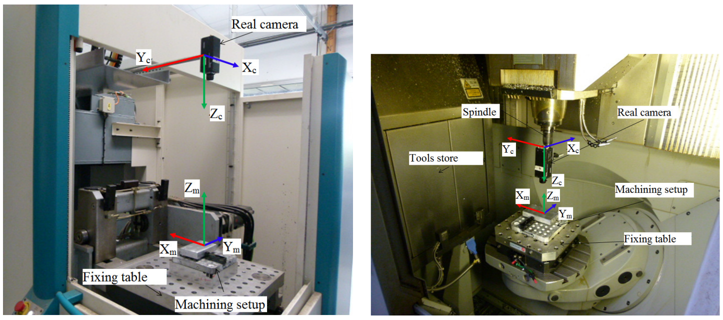
The paper is organized as follows. Section 2 outlines the vision-system set-up and presents the process of selecting the features. Section 3 describes the graph tool that is used to perform the automatic matching between the actual machining set-up and the desired one as defined in the 3D CAD model. Once feature correspondences have been defined, a 3D/2D pose is estimated. Section 4 presents the method used and provides some results. At the end of this section the relation is examined between the accuracy of the pose and some relevant parameters of the proposed method, such as the topological relation between nodes in the graphs, and the robustness of the pose using noisy images. Section 5 presents the local analysis of each subregion extracted from the images. To make a decision on the conformity of each object's region with the theoretical region, an optimal fusion classifier-based fuzzy logic system is proposed. This system is based on a skeleton-graph classifier, a boundary classifier and a surface classifier, presented in section 6. Finally, section 7 discusses the results obtained and gives some perspectives for future work to improve the method.

2 Vision-system set-up and features selection

2.1 Proposed vision system set-up

To ensure optimal imaging of the machining set-up, the camera should be installed perpendicular to it. The Z_c axis of the camera (optical axis of the camera) and the Z_m axis of the machining set-up (normal axis of the machining set-up) are parallel (see Figure 5).

In reality, two types of machine exist: machines that have a space to prepare the machining set-up, in the case of serial production, and machines which do not have space to prepare set-up. For the first situation, the camera is fixed perpendicular to the machining set-up in the preparation space, as shown in Figure 5a. For the second situation, the camera should be installed in the spindle. The camera is stored in the machine tools store, so when the operator finishes the set-up, the machine can be asked to search for the camera in the store (a line of code, similar to tool-changing code, needs to be added at the beginning of the CADCAM programme). Figure 5b shows the proposed configuration.



(a) (b)

Fig. 5 Proposed Camera/Machining set-up configuration: (a) machine has space to prepare machining set-ups, (b) machine without preparation space.

The proposed configuration guarantees that circles will be not transformed into ellipses, and it simplifies 2D/3D features matching and pose estimation.

As the aim is to be able to project onto the image some elements extracted from the 3D CAD model (augmented reality), the pose between the camera and the 3D CAD model needs to be estimated. To achieve this, some correspondences need to be established automatically between some of the primitives extracted from the image and the corresponding ones in the 3D CAD model.

2.2 Creation of a virtual image

First, a so-called ”virtual image” is computed that corresponds to the projection of the 3D CAD model onto a plane.

As shown in Figure 6, a virtual camera is positioned perpendicular to the 3D CAD model and the model is projected onto the (uv) plane, leading to the so-called “virtual image”.

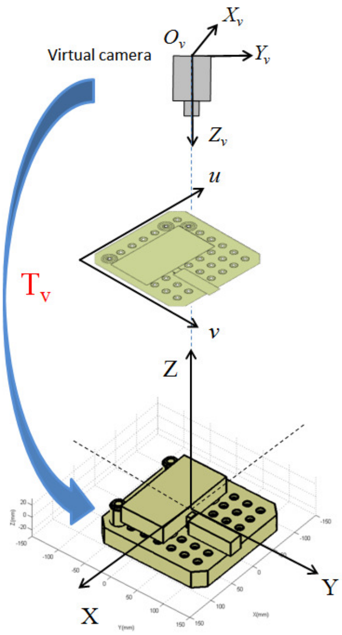


Fig. 6 Construction of a virtual image using a virtual camera.

Figure 7 shows the computed virtual image and the edges extracted from this virtual image.

2.3 Feature selection

As the machining set-up may not conform to the 3D CAD model, invariant features on the image need to be selected automatically. These need to be reliable for the 3D CAD model (i.e. they should be primitives that cannot be moved by the operator).

As shown in Figure 8, most of the pallets used in machining have several holes that are used to fix the machining set-up elements (location of clamping elements and fixing screws). Thus, circles were selected as invariant features.

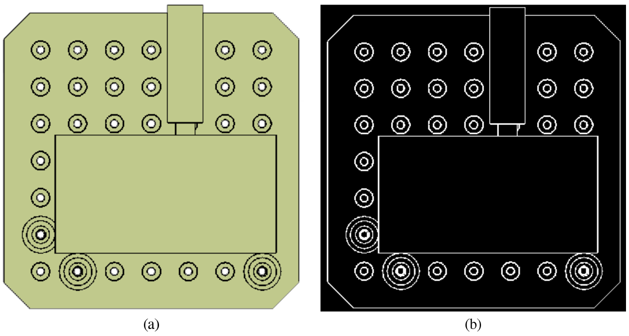


Fig. 7 Virtual image: (a) virtual image of a machining set-up, (b) edges extracted from the virtual image.

The circles were extracted from the edge image using the Hough Transform (HT).

As machining set-ups are made up of several polyhedral elements, it was also decided to select the lines that can reflect the topology of the scene. The Line Segment Detector algorithm [5] was used for this purpose. A new descriptor was proposed in order to select the relevant line features and to optimize the line intersection points that would be used to construct a topological graph. Using information from both circles and lines served to reinforce the matching process, and ensured the general compatibility of the method where machining set-ups have fewer circles and more lines, or vice versa.

In the literature, the problem of circle and line detection has been widely studied [6]. Both remain topical issues [7] [8]. Lines usually represent illumination discontinuity, and they have proved useful in many computer- vision and image-understanding applications [9]. However, circles can be used as parameters in camera-calibration processes [10]. In this section the process of feature selection is described. Firstly, to simplify the problem it will be useful to compute the edge image. A new method of image processing to compute edges from a real image has been developed [11].

2.3.1 Circle selection

Since many circle radii can be extracted from the image, a filtering process was used to retain only the relevant circles (which represent fixing screws). An attribute vector V_c was defined and is given in equation (1).

$$V_c = [\, r_{min},\ r_{max},\ r_i^c,\ x_i^c,\ y_i^c]^\top,\ \ i = 1 \ldots N \tag{1}$$

where, r_{min} and r_{max} are the interval of the radii of the circles that should be kept. r_i^c is the radius of the circle i and y_i^c, x_i^c are the coordinates of the centre.

The filtering parameter ρ_i was defined as the rate of detected circle radii between r_{min} and r_{max} and this was fixed empirically at 80%. Figure 9a shows

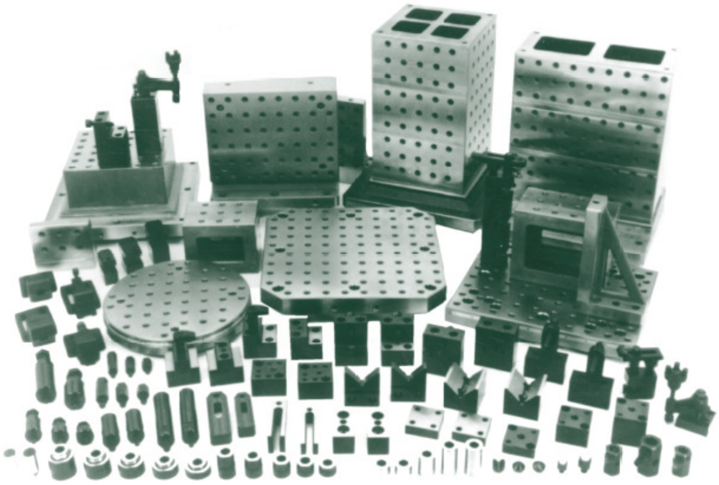


Fig. 8 Different types of pallets and clamping elements of machining set-ups.

the result of a selection of relevant circles that are designated by *yellow star* centres (x_i^c, y_i^c) in the case of the configuration shown in Figure 9.

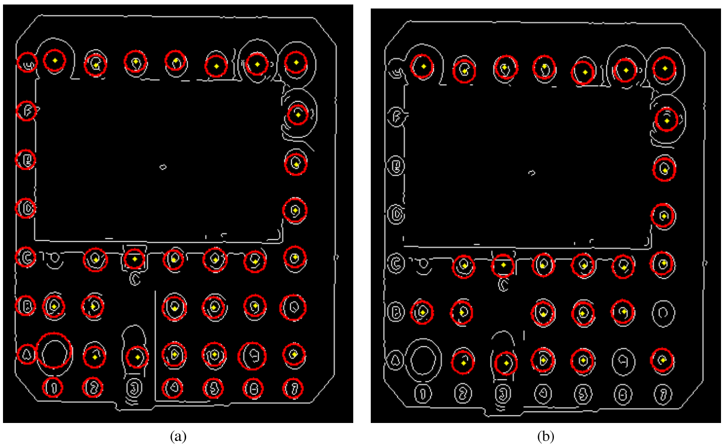


Fig. 9 (a) Detection of all circles, (b) Selection of relevant circles

2.3.2 Line selection

The discontinuities of gradient in some regions of the image may cause the detection of multiple portions of segments. In this application, this will generate many line intersections and therefore many relevant features which may be optimized for graph matching. Figure 10 shows examples designated by red ellipses.

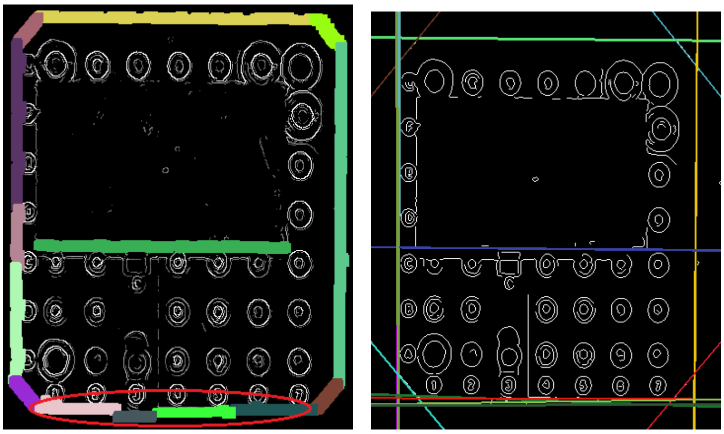

(a) (b)

Fig. 10 Line segment detection using LSD algorithm [8].

With the aim of selecting line segments, the *Line Segment detector (LSD)* algorithm proposed by Grompone von Gioi *et al.* [8] was used. The LSD aims to detect locally straight contours on images. The straight pixels are grouped together onto a rectangle. Each rectangle is characterized by its direction α, length L, width l, and as a candidate for a line segment. To validate the candidature of the rectangle support, the authors of [6,12] use an *a contrario* validation approach based on the minimization of the Number of False Alarm (NFA), as shown in equation (2). The NFA is associated with binomial distribution.

$$NFA(rect, x) = N_{test}.\mathbb{P}_{H_0}\left[k(rect, X) \geq k(rect, x)\right] \tag{2}$$

H_0 is the assumption that the image containing the rectangle bounded by the $rect$ has no pixels aligned, with N_{test} the number of tests. X is a random variable representing an image satisfying assumption H_0. P_{H_0} is used as the probability that a pixel, assuming H_0, is aligned with the rectangle $rect$. This probability is given by $p = \frac{\tau}{\pi}$ for a uniform distribution between $[0, 2\pi]$ [6]. This allows $\mathbb{P}_{H_0}\left[k(rect, X) \geq k(rect, x)\right]$ to be written in the form of a binomial distribution given by equation (3).

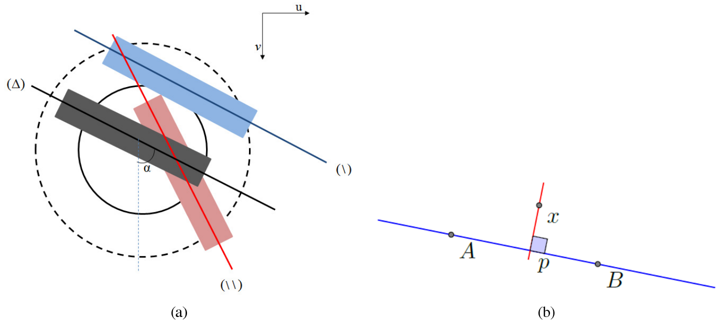
$$\mathbb{P}_{H_0}\left[k(rect, X) \geq k(rect, x)\right] = B(n(rect), k(rect, x), p) \tag{3}$$

where,

$$B(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j} \tag{4}$$

In order to optimize the number of detecting lines, a selection process was defined. As shown in Figure 11, for each line support, l represents the length of the segment, and ρ_l represents a length ratio between the selected line and the reference line, as shown in equation (5). For each line support, if it is oriented $\theta \in [\alpha - \tau_\alpha, \alpha + \tau_\alpha]$, where τ_α is fixed empirically at $\frac{\pi}{18}$, and the distance from the projected centre point P_c to the line segment support of the selected rectangle is less than $3l$, where l is the length of the rectangle, then the rectangle is merged with the selected one, and a new line support is computed using the least squares method. Figure 11 shows the principle of the proposed line selection algorithm.

$$V_l = [\, l,\ L,\ \rho_l,\ x_p,\ y_p]^\top,\ \ i = 1 \ldots M \tag{5}$$

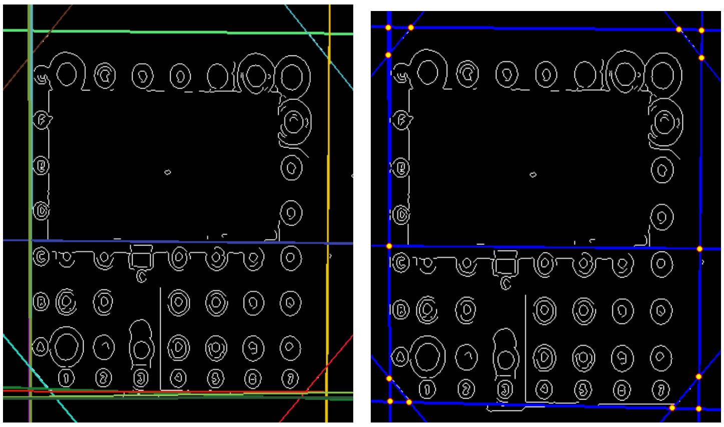


(a) (b)

Fig. 11 Principle of line selection: each line support region is fully covered by an estimated rectangle which is characterized by its direction α in relation to the v axis and its distance from the origin. In this example, the rectangle support segment [\] is merged with a rectangle support segment [Δ]. However, the rectangle support of segment [\\] is not considered, because its orientation is: $\theta_{(\backslash\backslash)} \notin [\alpha - \tau_\alpha, \alpha + \tau_\alpha]$.

Figure 12 shows the result of the selection of lines for Figure 10.

Using the vectors V_c and V_l of the features selected, a graph can be built representing the proximity information. The matching between the real graph G_r and the virtual one G_v, will provide the node correspondences between the real image of the machining set-up and the 3D CAD model.



(a)

(b)

Fig. 12 (a) Detection of all lines, (b) Selection of relevant lines and corresponding intersection points

3 Graph-based feature matching

3.1 Graphs construction

In many problems of pattern recognition [13], [14] and data mining [15], graphs have been used with great success to apply data-driven techniques in which the proximities between sets of graph nodes are widely adopted [16].

In a proximity graph, nodes are connected by edges on the sole condition that they are deemed to be close by some proximity measurement. It is this measurement that determines the type of graph that will result. Minimum spanning trees [17], Gabriel graphs [18], relative neighbourhood graphs [19], Delaunay diagrams [20] (often referred to as Delaunay triangulations), all describe different notions of vicinity between the points of the input point set. In this paper Gabriel graphs have been used. They have been proven to be good descriptors of the topological relationships of a set of points (see e.g. in the paper by David [21]). Figure 13 shows an example of graph construction for both real and virtual images of a machining set-up.

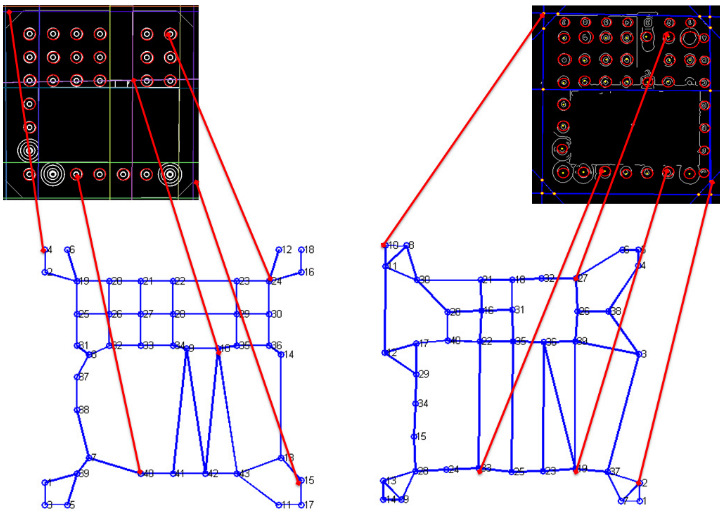


Fig. 13 Graph construction using the Gabriel graph principle for both real and virtual images.

In this study, the nodes represent the feature coordinates (such as circles or intersections of line segments). The edge represents the proximity between sets of points (nodes). The constructed graphs for both real and virtual images are shown in Figure 13.

3.2 Graph Matching

Establishing correspondences between two sets of visual features is a key problem in many computer-vision tasks, such as image classification [22] or shape matching [23]. Previous works formulate the problem as a search for the nearest neighbour [19], geometric consistency [19], or, at a high order of complexity,a combination of many similarity measurements [13], [16].

First, the graphs were normalized using equation (6). It should be noted that normalization is an important task in graph matching, because, in the creation of a virtual image, the projection of a 3D CAD model onto a (xy) plane does not take into account the real dimensions of the machining set-up (the 3D/2D pose is still unknown). Normalization allows the coordinates of both graphs to be expressed in the same chosen interval [a, b] (e.g [0, 1]).

$$\forall (x,y) \in \mathbf{R} \times \mathbf{R} \Rightarrow \begin{cases} N_x = \left(\frac{x-min(x)}{max(x)-min(x)}\right) \times (b-a) + a \\ N_y = \left(\frac{y-min(y)}{max(y)-min(y)}\right) \times (b-a) + a \end{cases} \tag{6}$$

$[a,\ b]$ is an interval of normalization and (x, y) are the centre point coordinates of the selected features.

Secondly, the issue is how to ensure the correspondence between graphs, even if they are structurally different. In other words, isomorphism cannot be used to establish correspondence because, in reality, the operator may assemble the machining set-up in a different way to the desired one, which can cause problems when checking for isomorphism. Even in the case of correct assembly of the machining set-up, isomorphism may be caused by the performances of the features detection and selection algorithm. Furthermore, uniqueness is an important piece of information, because corresponding nodes will be used to compute the 3D/2D pose estimation, which means that each node must have one and only one correspondence. To avoid these constraints, the connected level nodes and the Euclidean distance information between them were used.

The connected level nodes reflect the importance of the region in the scene, and a node that is strongly connected is a node that probably has a correspondence in the virtual image, or vice-versa. If nodes P_1 and P_1' of the real image are matched with nodes P_2 and P_2' of the theoretical image, then the geometric relation between P_1 and P_1' and that between P_2 and P_2' should be similar (no deformation). It is important to note that matched-pair nodes $(P_1,\ P_1')$ and $(P_2,\ P_2')$ should have the same class (circles or intersection of line segments). So as to ensure similarity criteria, the Normalized Gabriel Graph (NGG) is cut into sub-graphs of circles and line nodes independently with respect to the geometric relation between them. Moreover, the computed graphs G_v and G_r from the extracted points N_1 and N_2 of the virtual and real images respectively do not have the same order of complexity, and it is not assumed that $N_1 =\ N_2$ i.e, not all nodes in one graph have similar nodes in the other graph. This depends on the actual configuration of the machining set-up, and on the features-selection step. However, this problem can be solved easily without adding a significant number of additional computations. The proposed algorithm, based on topological similarity, can be expressed by two major measurement criteria: (1) compute high-level sub-graphs of real and theoretical graphs (2) find the corresponding transformation Φ between sub-graphs by minimising the neighbourhood circle expressed by the Euclidean distance between nodes.

3.2.1 Subgraph computing

For each graph, the sub-graphs can be determined once the *cut edges* are computed. The cut edge is defined as the edge linking the circle node to the line node or vice versa. This is the main idea to compute a sub-graph so as to find node correspondences. The principle of the algorithm is shown in Figure 14. This decomposition of the graphs allows us to begin matching between high-order sub-graphs to find a corresponding transformation Φ between sub-graphs.

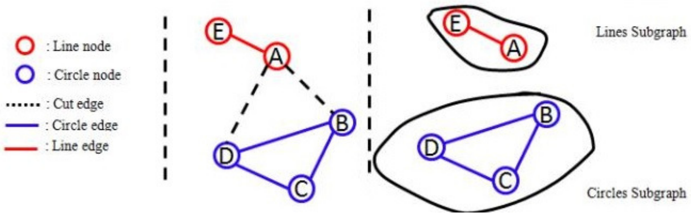


Fig. 14 Sub-graph computation: edge and node properties.

3.2.2 Transformation Φ

The sub-graphs are computed, and represent the proximity relationship between the circles and lines independently. For each sub-graph, the matching process consists primarily in finding the transformation Φ correspondence of the real graph with the virtual one. As shown in equation (7), the node correspondences are found by computing the number of nodes inside the neighbourhood circle based on the Euclidean distance for each transformation Φ.

$$S = \arg\max_{N}\{\min F(G|H, \hat{\Phi})\} \tag{7}$$

Where, Φ is the geometrical relationship that ensures a maximum number of matched nodes, N is the number of matched nodes.

As described above, the proposed method must find correspondences even if the actual machining set-up does not conform to the desired one, because anomaly detection will be carried out in the next step of local analysis. The aim of finding correspondence is to compute the 3D/2D pose and project the

3D model onto the image to achieve segmentation of the ROI (Regions Of Interest).

To overcome this problem and to determine the correspondence transformation Φ_q, which maximises equation (7), the new coordinate vectors $\overrightarrow{x_{ab}} = (\overrightarrow{P_b} - \overrightarrow{P_a})$, $\forall a, b \in \mathcal{I}$ are defined. Accordingly, a new origin is defined for graph H as the translated position of the origin to the higher-connected node P_{ref}. This is illustrated in Figure 15.

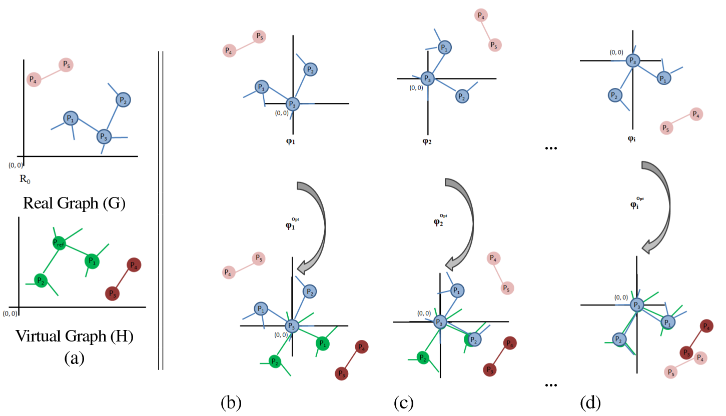


Fig. 15 Principle of landmark changing and optimal configuration ϕ finding: node P_3 of graph G and node P_{ref} of graph H are the new origin. Columns (b), (c) and (d) represent the different configurations Φ_i, and the optimal configuration is given in column (d) ϕ_i^{Opt}.

The relation between both nodes selected as the new origin has to respect the following condition:

$$||P_{ref} - P_i|| \leq \epsilon. \tag{8}$$

Where ϵ is the smallest distance between the two nodes (w_j^1, w_j^2) connected in the reference graph H. It is expressed by : $\epsilon = \dfrac{w_j^1 + w_j^2}{2}$.

Two conclusions can be drawn from this condition:

- If $||P_{ref} - P_i|| >> \epsilon$, the applied origin is too far from the optimal solution, because both nodes are situated at the extremity of the normalized graphs.
- Otherwise if $||P_{ref} - P_i|| \approx \epsilon$, the applied origin is accepted.

Respecting this condition, the optimal transformation Φ (which ensures maximal matched nodes with minimal distance), is expressed as :

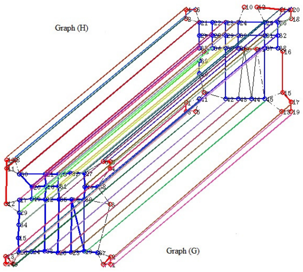
$$F(G|H, \hat{\Phi}) = \frac{1}{N_{ij}} \sum_{i,j \in \mathcal{I},\mathcal{J}} Dist(v_i, w_j | S, \Phi_q) \tag{9}$$

F is also called the neighbourhood function, for which the distance between pairs of nodes $v_i(x_i, y_i)$, $w_j(x_j, y_j)$ is given by :

$$Dist(v_i, w_j | S, \Phi_q) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{10}$$

Moreover, Φ is an affine invariance transformation, because the objects being considered here are rigid. Furthermore, both graphs G and H are normalized, so there is no problem in finding a scale factor. Since geometrical information is used to find correspondence in a relational way, Φ does not depend on any translation or scale parameter. It may be expressed as a plane rotation with four possibles rotations $k \times \frac{\pi}{2}$, with $k \in [0, 3]$. Figure 16 summarizes the parameters of equation (9). Table 16b presents the coordinates of normalized graph nodes according to the new landmark (defined by labelled nodes (16, 29), e.g grey line).

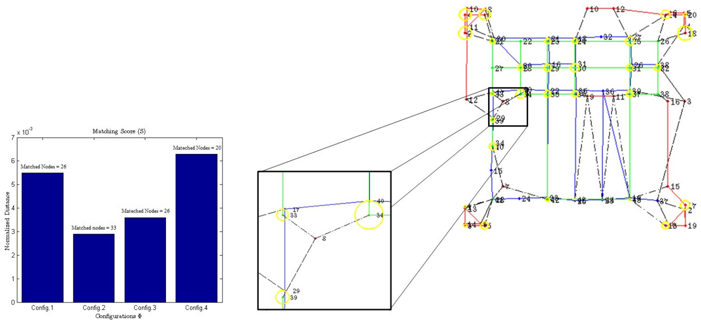
Figure 17 presents the graph-matching results achieved on several machining set-ups carried out by the operator, for the case of non-conforming machining set-ups. This figure shows that the graph matching is robust, despite the fact that the machining set-up configurations do not conform to the CAD model.



(a) : Config.Φ_2

Real Graph (G)		Virtual Graph (H)		Matched nodes label	
v_i^x	v_i^y	w_j^x	w_j^y	G Node	H Node
-0.3812	-0.6653	-0.3749	-0.6418	13	1
-0.3740	0.1715	-0.3749	0.1592	11	2
-0.3818	-0.7372	-0.3749	-0.7279	14	3
-0.3733	0.2554	-0.3749	0.2454	10	4
-0.3105	-0.7377	-0.2865	-0.7279	9	5
-0.2899	0.2538	-0.2865	0.2454	8	6
0.5463	-0.7441	0.5367	-0.7279	7	13
0.5498	0.2377	0.5367	0.2454	6	14
0.6178	-0.6719	0.6251	-0.6338	2	17
0.6133	0.1752	0.6251	0.1592	4	18
0.6182	-0.7446	0.6251	-0.7279	1	19
0.6130	0.2365	0.6251	0.2454	5	20
-0.2507	0.1190	-0.2503	0.1231	30	21
-0.0030	0.1190	0	0.1231	21	23
0.1208	0.1190	0.1238	0.1231	18	24
0.3685	0.1264	0.3741	0.1231	27	25
-0.1299	-0.0050	-0.1234	0	20	28
0	0	0	0	16	29
0.1208	0.0025	0.1238	0	31	30
0.3745	-0.0025	0.3741	0	26	31
0.4953	-0.0025	0.5005	0	38	32
-0.2537	-0.1289	-0.2503	-0.1205	17	33
-0.1299	-0.1165	-0.1234	-0.1205	40	34
-0.0060	-0.1215	0	-0.1205	22	35
0.1238	-0.1215	0.1238	-0.1205	35	36
0.3655	-0.1215	0.3741	-0.1205	39	37
-0.2537	-0.2479	-0.2503	-0.2410	29	39
-0.2567	-0.3644	-0.2503	-0.3640	34	40
-0.2567	-0.6248	-0.2503	-0.6076	28	41
-0.0091	-0.6148	0	-0.6076	33	42
0.1178	-0.6297	0.1238	-0.6076	25	43
0.2416	-0.6272	0.2503	-0.6076	23	44
0.3655	-0.6173	0.3741	-0.6076	19	45

(b)



(c)

(d)

Fig. 16 Real (G) and theoretical (H) graph matching result: (a) Optimal configuration Φ_2. (b) Table of matching labelled nodes; grey row indicates the selected origin nodes. (c) Bar diagram for different configurations Φ_q. Config. #1 corresponding to Φ_0 for which rotation plane equal to zero. Config. #2 corresponding to plane rotation $\Phi_1 = \pi/2$. This configuration gives the best matching result, e.g maximum number of matched nodes with minimum distance. Config. #3 corresponding to plane rotation $\Phi_2 = \pi$. Finally, config. #4 corresponding to plane rotation $\Phi_2 = 3\pi/2$. (d) finding the nearest node. The zoom in shows the nearest circle, where the radius is given in equation (10).

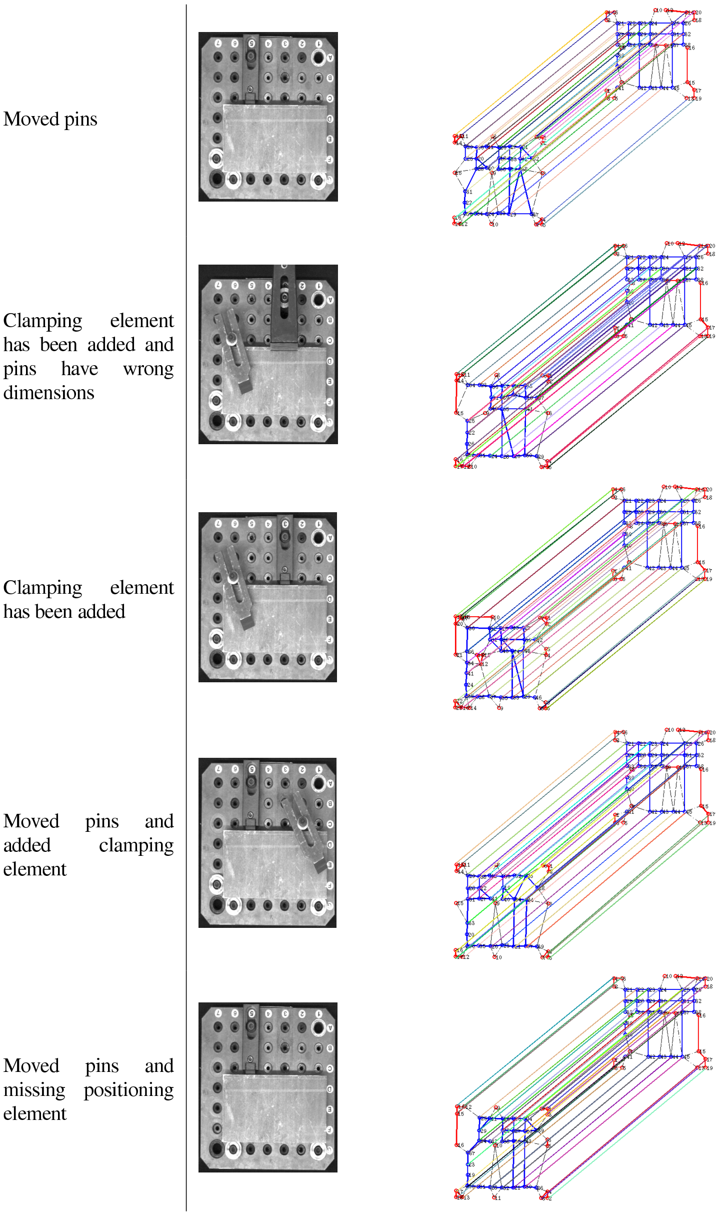

Moved pins
Clamping element has been added and pins have wrong dimensions
Clamping element has been added
Moved pins and added clamping element
Moved pins and missing positioning element

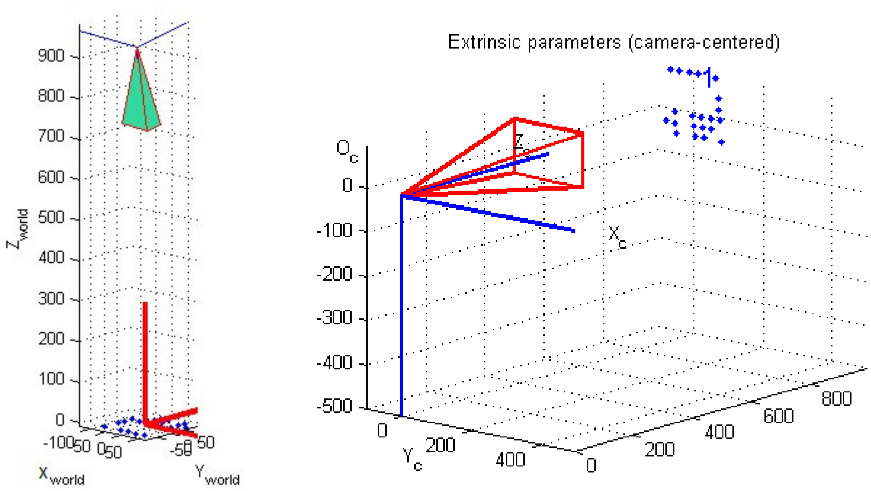
Fig. 17 Automatic graph matching of different non-conforming configurations of machining set-up.

4 Pose estimation and results

4.1 3D/2D Pose estimation

The matching of the graph nodes provides the correspondences between the 2D features extracted in the image and their correspondences in the 3D CAD model.

Using these matches, the 3D/2D pose (rotation and translation) between the 3D CAD model and the 2D image can be estimated and given in $[R|t]$ matrix. The camera was calibrated off-line and the intrinsic parameters were determined and given in matrix [A] using the Matlab® Camera Calibration Toolbox [24]. The projection matrix [P] is expressed as $P = [A].[R|t]$, where A stands for the camera intrinsic parameters and R and t stand for the extrinsic parameters. The $[R|t]$ matrix is built from the rotation matrix R and the translation vector t. Figure 18 shows the results of automatic computation of pose.



(a)

(b)

Fig. 18 Estimation of intrinsic parameters and pose of the camera relative to the machining assembly, in accordance with (a) the pose of the camera expressed relative to the benchmark machining assembly given by the CAD model, and (b) the camera pose expressed relative to the reference of the camera.

4.2 Attributed Relational Graph (ARG)

The topological organization of the elements constituting the machining assembly in its numerical CAD definition is expressed by an Attributed Relational Graph (ARG). The ARG is calculated firstly from the CAD model. In the CAD model each element is defined by its dimensions and location. The *ARG* computed from the CAD model is projected onto the "real" image after calculating the parameters of the projection matrix $[P]$. The edges between the nodes representing the neighbourhood are measured on the CAD model. Each node in the ARG is characterized by a feature vector containing the dimension of the object, thereby delimiting the region of interest. Figure 19 shows an example.

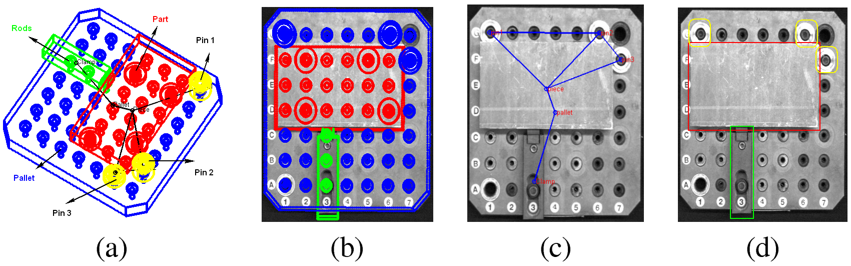
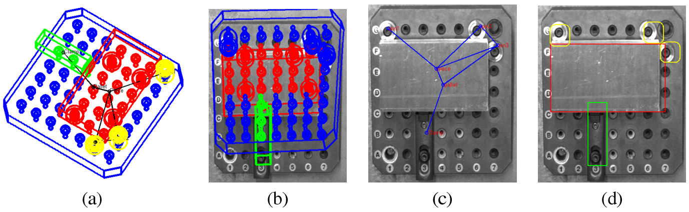


Fig. 19 Region-segmentation-based model: (a) computed attributed relational graph from 3D CAD model ARG_{ref}, (b) projection of 3D CAD model onto image of machining set-up, e.g augmented reality, (c) Computed ARG_{real} from pose estimation. (d) region segmentation result.

The aim of using ARG is to localize the regions of interest containing the clamping elements and the part, after calculating the pose in relation to their CAD definition. This strategy of using ARG allows us to detect any eventual anomaly in the calculation of the pose before proceeding to local analysis of the regions of interest. Indeed, it proves to be robust even if the actual machining set-up does not conform to the reference image given by the CAD model. Figure 20 shows an example where the estimated pose does not conform. In this example, the machining set-up is inclined so that the assumption regarding the relative position of the camera and the machining set-up is not respected (the OZ_c axis of the camera and the OZ_m axis of the machining set-up should be parallel or, in an ideal case, superposed ; axes are described in Figure 5). This anomaly can be checked using the definition of the adjacency matrix of ARG. In fact, the edge between the node of "Pin 3" and the node of "Pallet" in ARG_{ref} (see Figure 19) was not defined (expressed by 0 in adjacency matrix) which is not correct in the projected ARG_{real}. In this case, the system should stop computing and give information to the operator about the actual machining set-up configuration.

If the anomaly is detected, the system should stop computing and give the information to the operator that there is a problem in the set-up installation.

Figure 21 shows the projection of ARG_{th} and image segmentation results for Norelem©, Semelle© and Minimanche© machining set-ups respectively in the case of conforming and non-conforming configurations.



	Clamp	Pallet	Piece	Pin1	Pin2	Pin3
Clamp	0	1	0	0	0	0
Pallet	1	0	1	0	0	0
Piece	0	1	0	1	1	1
Pin1	0	0	1	0	0	0
Pin2	0	0	1	0	0	1
Pin3	0	0	1	0	1	0

(e)

	Clamp	Pallet	Piece	Pin1	Pin2	Pin3
Clamp	0	1	0	0	0	0
Pallet	1	0	1	0	0	1
Piece	0	1	0	1	1	1
Pin1	0	0	1	0	0	0
Pin2	0	0	1	0	0	1
Pin3	0	1	1	0	1	0

(f)

Fig. 20 Anomaly detection in pose estimation from the ARG: (a) Graph ARG_{th} according to its definition in CAD model, (b) projection of digital CAD model onto image, (c) graph ARG_r calculated after pose estimation, (d) image segmentation and extraction of regions of interest, (e) adjacency matrix of the graph ARG_{th} according to the CAD definition, (f) adjacency matrix of the graph ARG_r. The red boxes in the matrix (f) represent the anomaly described by defining a new neighbourhood that is not defined in ARG_{th}.

4.3 3D/2D Pose evaluation

Two principal tests were conducted to study the robustness of the proposed method with regard to pose estimation. The first test concerned the study of the performances of the feature detection and selection algorithms and the pose accuracy. To evaluate the performances, the CAD edge image was chosen as the reference image and at each iteration i a Gaussian noise was applied with σ_i varied at each iteration (starting with $\sigma_1 = 1$) and $\mu = 0$. Figure 22 shows the results. It can be deduced that if $\sigma_i \leqslant 10$ pixels the pose accuracy is about $\leqslant 3$ pixels, otherwise the pose precision curve increases exponentially.

The second test concerns the study of the relationship between the spatial distribution of the nodes and the accuracy of the pose. Haralick *et al.* [25], Horaud [26] and recently Hartley and Zisserman [10] have studied the relation between pose accuracy and spatial distribution of corresponding nodes (and their number, for 3, 4 and ≥ 5 point correspondences). In this study, it can be deduced from Figure 23 that if the nodes are aligned, the pose accuracy is low, because the translation estimation is false. However, if most of the nodes are unaligned, then the pose accuracy is less than 3 pixels (accepted accuracy in this application). The bar diagrams of Figure 23a and Figure 23b show the 29 configurations that have been randomly generated and studied. Figure 23c shows some graph configurations that correspond to two extreme situations: for Config. #2 and Config. #29 the accuracy of the pose estimation is quite good (only a few pixels) whereas for Config. #10 and Config. #29 the accuracy is quite bad (several tens of pixels).

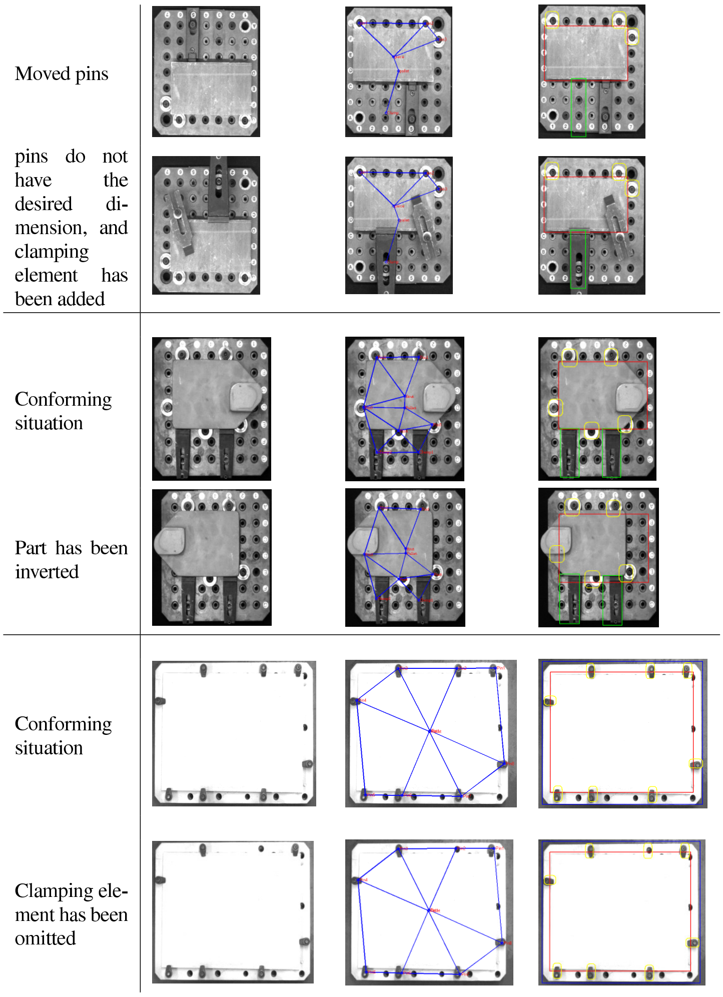

Moved pins
pins do not have the desired dimension, and clamping element has been added
Conforming situation
Part has been inverted
Conforming situation
Clamping element has been omitted

Fig. 21 Region segmentation for different conforming and non-conforming configurations of Norelem© machining set-up, Semelle© machining set-up and Minimanche© machining set-up, respectively.

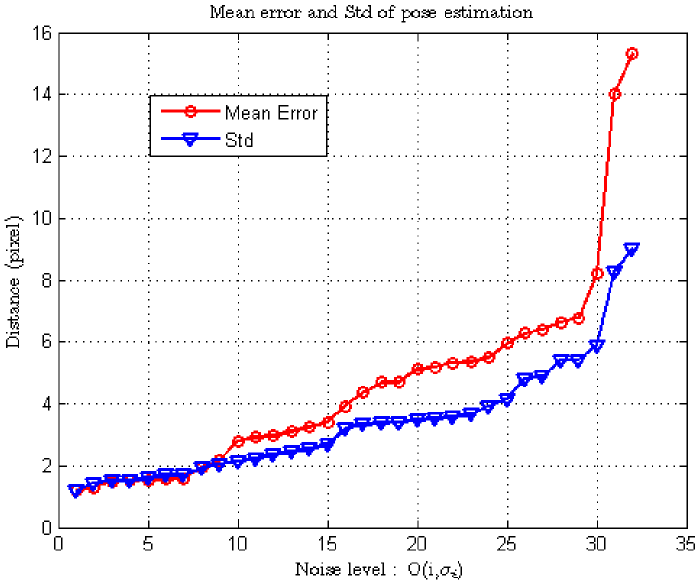
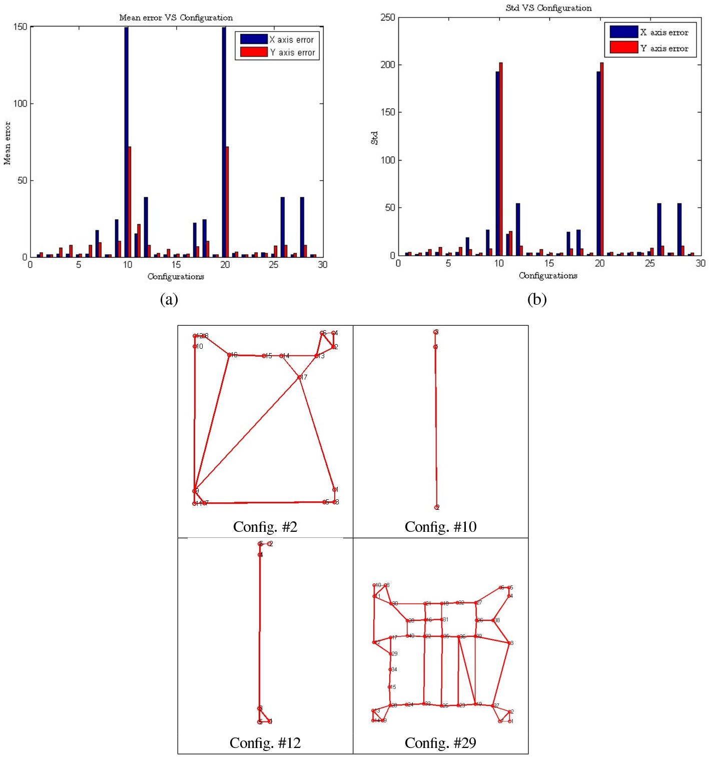


Fig. 22 3D/2D pose evaluation with coordinates of nodes (mean error and standard deviation (Std)).



(a)

(b)

(c)

Fig. 23 Pose evaluation with spatial relationship between nodes (29 configurations have been randomly generated and studied) : (a) mean error in pixels along the X and Y image axes, (b) mean variance in pixels along the X and Y image axes, (c) some graph configurations. Config. #29 presents the configuration in which the mean error is minimal, with $Mean(x, y) = (1.2063, 2.3697)$ and standard deviation $std(x, y) = (1.3732, 1.6096)$.

5 Local analysis

In order to analyse the content of each image region delimited by the result of the pose estimation, the object must be segmented from the background. To this end, repeat performances were tested for two algorithms, reputedly the highest performing ones, the *Ncut* algorithm proposed by [27], based on the spatial segmentation of the image using gradient information, and secondly, the *AGGMM* algorithm, based on the modelling of a grey-scale image histogram proposed by [28]. The results are given in Figure 24. These results were evaluated using a confusion matrix (see Table 1).

		Reference (%)	
		Object	Background
Calculated (%)	Object	True Positive (TP)	False Positive (FP)
	Background	True Negative (TN)	False Negative (FN)

Table 1 Confusion matrix used in Figure 24.

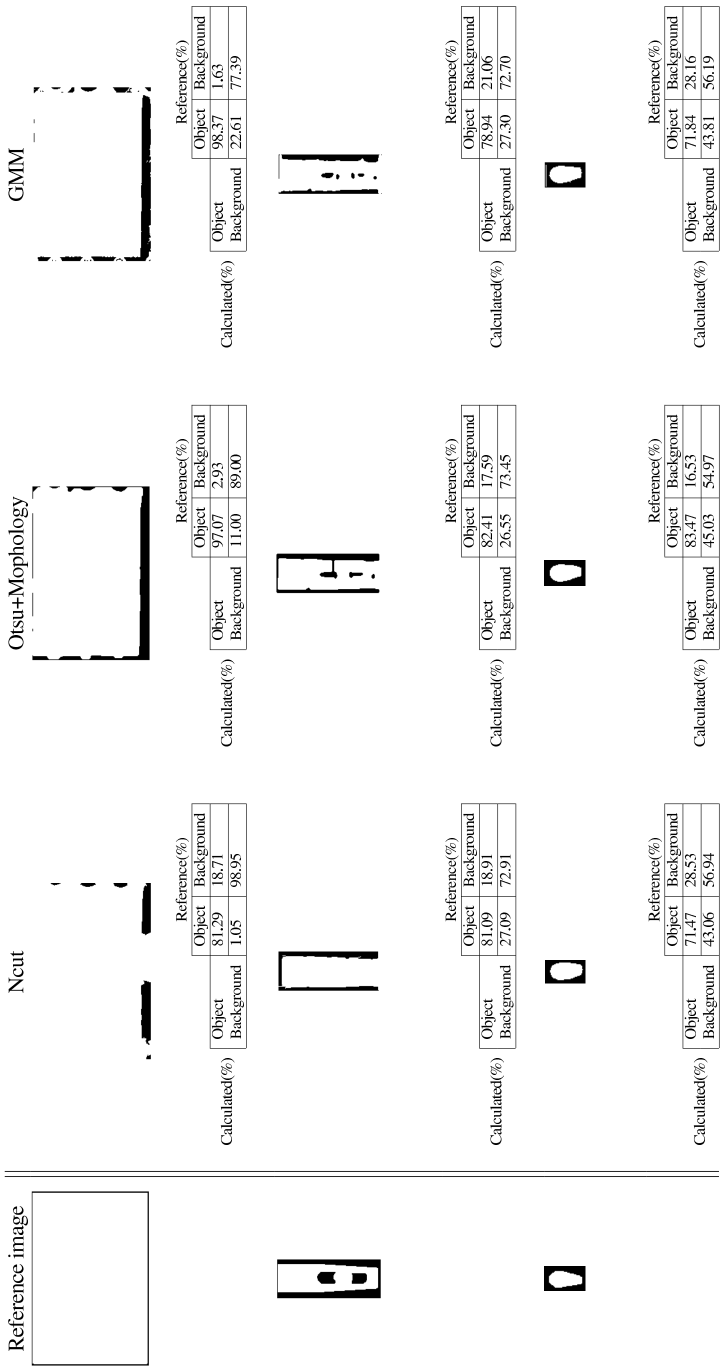
The rate of recognition in both algorithms was more than 80% on average. Graphics Processing Unit (GPU) time is also an important factor with regard to this application. It was therefore tested by a standard method, Otsu's algorithm [29], using the mathematical morphology technique. The aim of the mathematical morphology operation is to recreate the pattern of an object as close to the reference as possible. This is important because there are polyhedrons, cylindrical and spherical shapes among the clamping elements in the machining set-up. The step-by-step results of segmentation in the case of a part for a Semelle$^{\copyright}$ machining set-up are given in Figure 25.

It can be seen that the rate of recognition is still more than 80%. However, the interest of using Otsu's method is that the GPU time computation is about 2 to 3 time shorter than with the *Ncut* and *AGGMM* methods.

Many pattern-recognition applications use a skeleton to describe complex objects or for retrieval in databases, while others use boundaries and surface. In this application, one of key ways to define a highly discriminant object descriptor is to combine descriptors, such as skeleton, boundary and surface descriptors.

5.1 Object pattern characterization

A lot of research effort has been devoted to finding and proposing classifiers by selecting the most relevant features for a given object. Generally, the best descriptor will depend on the application, being necessarily more than one descriptor in order to detect all relevant features of the shape. This proposal focuses on the use of three descriptors, a skeleton-graph descriptor (SGD), a boundary descriptor (BD) and a surface descriptor (SD). A classifier of each of them is trained to compute scores (outputs) and combine the output of



		Reference(%)	
		Object	Background
Calculated(%)	Object	81.29	18.71
	Background	1.05	98.95

		Reference(%)	
		Object	Background
Calculated(%)	Object	97.07	2.93
	Background	11.00	89.00

		Reference(%)	
		Object	Background
Calculated(%)	Object	98.37	1.63
	Background	22.61	77.39

		Reference(%)	
		Object	Background
Calculated(%)	Object	81.09	18.91
	Background	27.09	72.91

		Reference(%)	
		Object	Background
Calculated(%)	Object	82.41	17.59
	Background	26.55	73.45

		Reference(%)	
		Object	Background
Calculated(%)	Object	78.94	21.06
	Background	27.30	72.70

		Reference(%)	
		Object	Background
Calculated(%)	Object	71.47	28.53
	Background	43.06	56.94

		Reference(%)	
		Object	Background
Calculated(%)	Object	83.47	16.53
	Background	45.03	54.97

		Reference(%)	
		Object	Background
Calculated(%)	Object	71.84	28.16
	Background	43.81	56.19

Fig. 24 Comparison of tested methods of region-based segmentation.

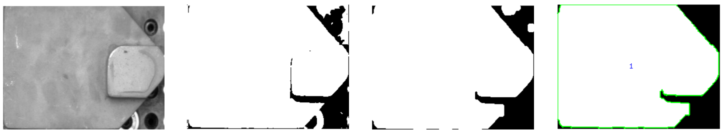


Fig. 25 Result of object segmentation from a background.

all classifiers in a fuzzy expert system, so that the overall performance increases. The next paragraph defines the descriptors and the proposed strategy to compute a classification score.

5.1.1 Skeleton-Graph Descriptor (SGD)

The *SGD* descriptor has been widely used for pattern-recognition applications. It can integrate the geometric and topological features of the object. However, it is a challenging task to automatically recognize objects using only their skeletons, due mainly to the skeleton's sensitivity to boundary deformation [30], [31] and also to the similarity of the skeletons for two different objects. Skeleton-based recognition methods are usually based on a graph or tree representation of the skeleton. Before describing the proposed method, some relations will be defined between the original skeletons and the computed graphs for matching.

Definition 1 A skeleton point having only one adjacent point is an endpoint. A skeleton point having two or more adjacent points is a junction point. The endpoint in the skeleton graph is called the end node, and the junction point in the skeleton graph is called a junction node.

Definition 2 The edge between an endpoint and a junction point is called the cut edge.

5.1.2 Matching the skeleton graphs and classification

Following the definitions given above, the skeleton graphs matching problem can be defined as the determination of a one-to-one node correspondence according to the kind of node (e.g. end-nodes and junction-nodes). The dissimilarity value between two nodes v_i and w_j of the same kind is given by the minimal distance between them. Nodes are matched using the same method as that proposed in section 4.1. Using the same formula as in equation (7), the advantage here is that the 3D/2D pose is estimated and the object regions are located. In other words, the translation vector (T), the rotation matrix ($\varPhi$) and the scale factor ($\varUpsilon$) are known, and therefore equation (7) can be written as:

$$S(G|H) = \min_{i,j \in \mathcal{I},\mathcal{J}} (Dist(v_i, w_j | \varUpsilon, \varPhi, T)) \tag{11}$$

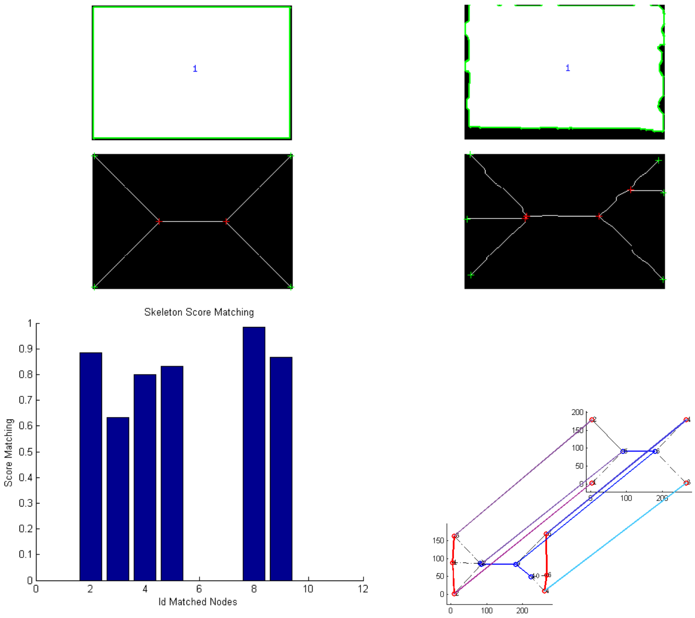


Fig. 26 Skeleton graph matching in the case of a part in a Norelem© machining set-up.

In Figure 26, it can be seen that labelled nodes 1, 6 and 7 have not been matched, therefore they do not affect the score calculation. The matching score is given as the average of matched nodes. In this case, the score $S_{Skel} = 0.64$.

5.1.3 Boundary and Surface Descriptors

The boundary descriptor is computed from the edge image. The boundaries are traced and elementary descriptors extracted at every equidistant point along the boundary. In reality, boundary descriptors depend on two independent parameters, such as pose-estimation accuracy and edge-detection algorithm performance. For the first parameter, pose estimation, there is an average error of about 3 pixels. To compute a score S_{edge}, a 5×5 pixel cell of a patch centred at p_{ref} is used to scan the region in the real edge image corresponding to the position of p_{ref} coordinates in the theoretical edge image. The score S_{edge} is the proportion of foreground (boundary) pixels in the theoretical image within the corresponding region of the local patch extracted around the boundary of the real image. This descriptor is applied to several real and theoretical images of the machining set-up clamps.

As regards the surface descriptor, it is simply the ratio between the real surface (S^r_{Area}) and the theoretical surface (S^t_{Area}), as shown in Figure 27. Equation (12) presents the ratio.

$$S_{Surf} = \begin{cases} \dfrac{S^r_{Area}}{S^t_{Area}} & \text{if } S^r_{Area} \leq S^t_{Area} \\ \\ \dfrac{S^t_{Area}}{S^r_{Area}} & \text{else} \end{cases} \tag{12}$$

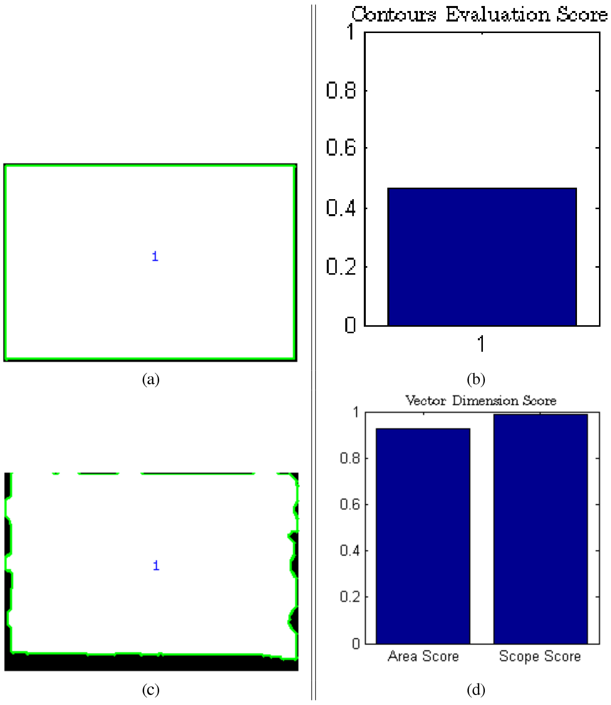


Fig. 27 Boundary and surface scores. (a) reference image, (b) scores for reference image, (c) real image and (d) scores for real image.

6 Decision

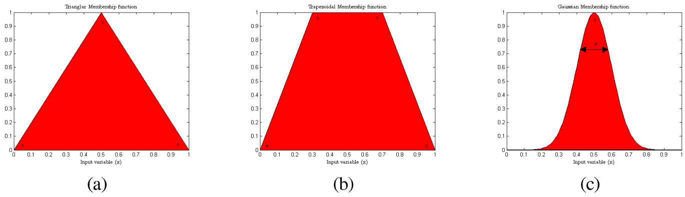
In similar situations of uncertainly, the problem of object recognition may be treated by probabilistic or possibilistic methods. Often, probabilistic methods seek to model the intrinsic behaviour of the system, using the mathematical relationship between random variables. However, in some cases, integrating human expertise in the combination of classifiers (inference step in fuzzy logic system) is an important step towards automating the decision process.

Combining the outputs of classifiers has been one of the strategies used to improve classification rates in general. Some of the most common approaches can be explained using the Bayes formula [32] for eye modelling and [33] for object tracking. In this application the problem of the combination of classifiers uses a non-Bayesian probabilistic framework. In the paper by Terrades *et al.* [34] the authors demonstrated the optimal combination of classifiers in a non-Bayesian framework. The proposed solution in this paper uses a fuzzy system reasoning method. There are two principal reasons for this choice:

- The fuzzy logic system approach reflects the human side of decision-making and the uncertain nature of the information provided in different steps of the processing chain, which allows us to examine the non-linear relationship between the calculated scores.
- The fuzzy logic system may effectively replace a complication model, such as hierarchical decision, because it is difficult to establish the relationship between descriptors.

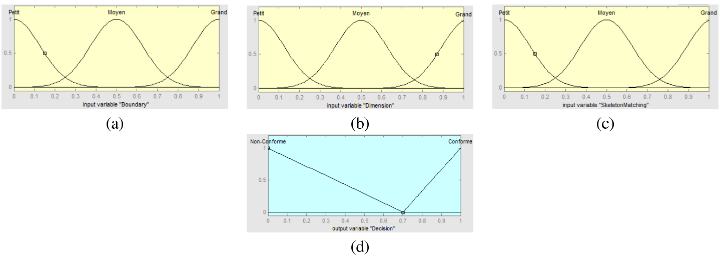
6.1 Set definition

The choice of the form of fuzzy set depends on the complexity of the problem and the variation ranges of the variables. There are several ways to model fuzzy sets, e.g triangular, trapezoidal or Gaussian. Both triangular and trapezoidal sets have a discontinuous characteristic within the Gaussian set. In Figure 28, the triangular form tends to cut through its maximum point marked by (b), resulting in two parts (a, b) and (b, c). The trapezoidal form tends to allocate its maximum over the range (b, c), which gives the three parts (a, b), (b, c) and (c, d). This can be interpreted by a spreading of the uncertainty for near one values. The Gaussian form is characterized by its continuity and it can combine the properties of both sets, triangular and trapezoidal and so give a continuous form to small uncertainty values. This characteristic is dependent on the choice of mean (μ) and variance (σ) of the Gaussian set. A Gaussian form is used to define the input variables of the fuzzy system, and the triangular form to define the output variables. Furthermore, two linguistic variables are defined, as "conforming" and "not conforming" for the "Decision" set, in which "not conforming" is the range [0, 0.7] and "conforming" is]0.7, 1].



(a) (b) (c)

Fig. 28 Definition of set forms. (a) Triangular membership form, (b) Trapezoidal membership form and (c) Gaussian membership form.



(a) (b) (c)

(d)

Fig. 29 Decision Sets. (a) input variable “Boundary”, (b) input variable “Dimension”, (c) input variable “Skeleton matching” and (d) output variable “Decision”.

6.2 Inference and decision

The defuzzification process consists in transforming linguistic variables into digital variables. In the first step, the variables language must be merged with the fuzzy logic operators and rules previously defined. The logical ”AND” operation was chosen for this task. For example, one fuzzy rule which was defined is:

If S_{Skel} is high and S_{CE} is high and S_{DIM} then the object conforms.

The second step consists in finding the quantitative value based on the degree of membership of the linguistic variables. There are several defuzzification techniques, including the low of the maximum value (LOM), the end of the maximum value (SOM), the area average value of the maximum (MOM), or the center of gravity (Centroid).

There are two known methods of inference in the defuzzification process: the so called Mamdani defuzzification system and the Sugeno method [35]. Both methods differ in computing the inference output and aggregation rules.

The inference method proposed by Mamdani consists in calculating the polygon including the membership of the input variable values. The height of the polygon is defined by the largest membership value. The aggregation of this variable is calculated by one of the defuzzification techniques presented previously, i.e MOM, LOM, SOM, and the center of gravity method. The method of center of gravity data is used in equation (13).

$$CG_{Mamdani} = \frac{\sum_{x=a}^{b} \mu_A(x).x}{\sum_{x=a}^{b} \mu_A(x)} \tag{13}$$

where x is the variable, A is the set and $\mu_A(x)$ stands for the mean of the area.

The method proposed by Sugeno relies on modelling the fuzzy set in a peak. The amplitude of the peak is given by the membership value of the variable x to the fuzzy set in question. The result of aggregation uses the weighted average method, given by equation (14).

$$CG_{Sugeno} = \frac{\sum_i \mu_A(k_i).k_i}{\sum_i \mu_A(k_i)} \tag{14}$$

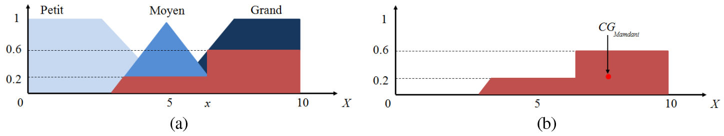


Fig. 30 Mamdani inference principle of aggregation rules and output (a) calculating membership rate, (b) calculation and inference of the value of the output.

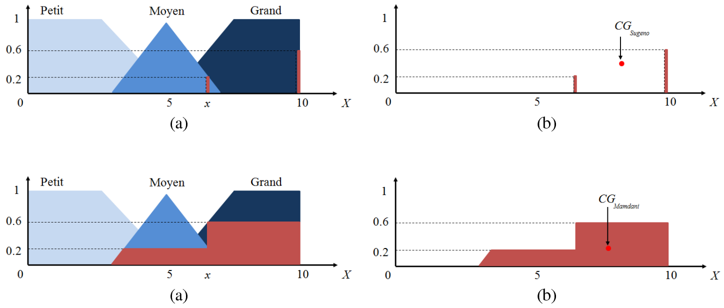


Fig. 31 Sugeno inference and output sets aggregation, (a) calculation of membership rate, (b) inference and given output value.

Two methods of defuzzification, Mamdani and Sugeno, have been tested. No difference was noted in the decision, because the shapes of the fuzzy sets defining the output variable ”Decision” are separated and do not have an intersection area, or area of confusion. In addition they are, respectively, centred on zero for all ”non-conforming” and on one for ”conforming”, which means that the result of the decision is often greater than 0.8 for the ”conforming” case and less than 0.2 for the case of ”non-conforming”.

7 Conclusion and perspectives

The aim of this work was to develop a vision system for automatic verification of machining fixtures. This would increase the productivity of machining companys, by protecting machines against the collision of the tool with the clamping elements.

The specification requires taking into account three main parameters, CPU-time processing, system cost and simplicity, and finally the dimensions of the control system to be installed within the machine. The solution chosen was to use monocular vision. The use of a monocular system could meet the requirements of the specifications. As concerns the computing time, operating a single image is much more practical than the treatment of two images of a binocular vision or two acquisitions of a single camera with two different points of view. The response time was less than 1 min on average for three tested machining set-ups (Norelem©, Semelle© and Minimanche©). As regards the system cost, the use of a single camera is obviously cheaper and less bulky than a three-dimensional scanning system. Finally, the system will be commercialised at around 10% of the value of the spindle.

The use of a 3D sensor without contact, as well as being inexpensive, gives a measurement precision of one tenth of a millimetre and would make it possible to obtain a more easily exploitable segmentation, in particular for the supernumerary parts. It would also provide immediate 3D information about the parts, which would ensure a better recognition of the objects and would lead to improved confrontation of the topology of the scene with the tool trajectory. The initial plan was to use a Kinect sensor, but its optical system is not compatible with the dimensions of the field of observation of the fixtures. Monocular systems of 3D acquisition , based, for example, on techniques of shape from shading, were also examined, but they do not give good results on abrupt discontinuities of depth [36]. Depth was also studied by means of defocus methods [37], but for the moment these methods do not reach the desired precision. Finally, the idea of integrating a programmable camera equipped with a FPGA and DSP for use in an industrial context, is currently being evaluated.

Acknowledgements The Conseil Régional Midi-Pyrénées, the CUFR JF Champollion, the IUT of Figeac and the Ecole des Mines of Albi are gratefully acknowledged for their support.

References

1. R. Ahmad, S. Tichadou, J.-Y. Hascoët, Integration of vision based image processing for multi-axis CNC machine tool safe and efficient trajectory generation, International Journal of Machine Engineering 10 (4).
2. X. Zhang, W.-M. Tsang, Y. Kazuo, M. Masahiko, A study on automatic on-machine inspection system for 3D modeling and measurement of cutting tools, Journal of Intelligent Manufacturing.

3. X. Zhang, X. Tian, Y. Kazuo, On-machine 3D vision system for machining setup modeling, The International Journal of Advanced Manufacturing Technology 48 (1-4) (2009) 251–265.
4. X. Zhang, X. Tian, Y. Kazuo, F. Makoto, 3D reconstruction and pose determination of the cutting tool from a single view, in: ISVC (2)'07, 2007, pp. 377–386.
5. V. G. Rafael Grompone, J. Jeremie, M. Jean-Michel, R. Gregory, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Analysis and Machine Intelligence 32 (2010) 722–732.
6. J. Burns, A. Hanson, E. Riseman, Extracting straight lines, IEEE Trans. Pattern Analysis and Machine Intelligence 8 (4) (1986) 425–455.
7. F. Deschenes, D. Ziou, M.-F. Auclair-Fortier, Detection of lines, line junctions and line terminations, International Journal of Remote Sensing 25 (3) (2004) 511–535.
8. R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Analysis and Machine Intelligence 32 (4) (2010) 722–732.
9. G. Papari, N. Petkov, W. C. Regli, Edge and line oriented contour detection: State of the art, Image and Vision Computing 29 (2011) 79–103.
10. R. I. Hartley, A. Zisserman, Multiple view geometry in computer vision.
11. B. Karabagli, T. Simon, J.-J. Orteu, Détermination automatique du seuil de binarisation des modules des gradients par modélisation de leur histogramme, in: Proceedings of ORASIS 2011, Praz-sur-Arly, France, 2011, pp. 2–8.
12. J. Beveridge, C. Graves, C. Lesher, Some lessons learned from coding the burns line extraction algorithm in the Darpa image understanding environment, Technical Report CS-96-125, Computer Science Dept. Colorado State Univ.
13. O. Duchenne, F. Bach, I.-S. Kweon, J. Ponce, A tensor-based algorithm for high-order graph matching, IEEE Trans. Pattern Analysis and Machine Intelligence. 33 (Dec. 2011) (2011) 2383–2395.
14. M. Gori, M. Maggini, L. Sarti, Exact and approximate graph matching using random walks, IEEE Trans. Pattern Analysis and Machine. Intelligence 27 (7) (2005) 1100–1111.
15. Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 15 (11) (1993) 1101–1113.
16. M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: International Conference of Computer Vision (ICCV), Vol. 2, 2005, pp. 1482 – 1489.
17. W. Grimson, T. Lozano-Pérez, Localizing overlapping parts by searching the interpretation tree, IEEE Trans. Pattern Analysis and Machine Intelligence, 9 (4) (1987) 469–482.
18. R. K. Gabriel, R. R. Sokal, A new statistical approach to geographic variation analysis, Systematic Zoology 18 (3) (1969) 259–278.
19. J. Zhang, A. Berg, M. Maire, J. Malik, SVM-KNN discriminative nearest neighbor classification for visual category recognition, Proc. IEEE Conf. Computer Vision and Pattern Recognition,.
20. B. Delaunay, Sur la sphère vide. A la mémoire de Georges Voronoï, Bulletin de l'Académie des Sciences de l'URSS (6) (1934) 793–800.
21. G. Liotta, Low degree algorithms for computing and checking gabriel graphs (1996).
22. J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, Int. J. Comput. Vision 73 (2) (2007) 213–238.
23. R. Zass, A. Shashua, Probabilistic graph and hypergraph matching, 2008, pp. 1–8.
24. J.-Y. Bouguet, Camera calibration toolbox for matlab (mar 2004).
25. R. Haralick, C. Lee, K. Ottenberg, M. Nôlle, Analysis and solutions of the three point perspective pose estimation problem, 1991, pp. 592–598.
26. R. Horaud, O. Monga, Vision par ordinateur, outils fondamentaux, 1993.
27. J. Shi, M. Jitendra, Normalized cuts and image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 22(8) (2000) 888–905.
28. N. Nacereddine, S. Tabbone, D. Ziou, L. Hamami, L'algorithme EM et le modèle de mélanges de gaussiennes généralisées pour la segmentation d'images. Application au contrôle des joints soudés par radiographie, Conf. Traitement et Analyse de l'Information : Méthodes et Applications (2009) 217–222.

29. N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66.
30. C. Arcelli, K.-M. Lam, W.-C. Siu, Extraction of the euclidean skeleton based on a connectivity criterion, Pattern Recognition 36 (1985) 721–729.
31. X. Bai, L. Latecki, W.-Y. Liu, Skeleton pruning by contour partitioning with discrete curve evolution, IEEE Trans. Pattern Analysis and Machine Intelligence 29(3) (2007) 449–462.
32. N. Bouguila, D. Ziou, R. I-.Hammoud, A bayesian non-gaussian mixture analysis: Application to eye modeling, Conf. Computer Vision and Pattern Recognition, 2007. (June 2007) 1–8.
33. N. Petrovič, L. Jovanov, A. Pizurica, P. , Wilfried, Object tracking using naive bayesian classifiers 5259 (2008) 775–784.
34. O. Terrades, E. Valveny, S. Tabbone, Optimal classifier fusion in a non-bayesian probabilistic framework, Conf. Computer Vision and Pattern Recognition, 2007. 31(9) (Sept. 2009) 1630–1644.
35. T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on System Man and Cybernetics (15) (1985) 116–132.
36. B. Karabagli, M.-F. Auclair-Fortier, T. Simon, J.-J. Orteu, Depth map estimation from a single RGB image: Principle and calibration, Séjour à l'Université de Sherbrooke, Québec, présenté au congrès de l'Ecole Doctorale EDSYS, Tarbes (2013).
37. P. Daniel, Peut-on extraire le relief d'une seule image ?, Ph.D. thesis, Institut de Recherche en Informatique de Toulouse (28 Janvier 2000).