

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Artificial Vision for Manufacturing System

---

Mejia Ugalde Mario and Mejia Ugalde Ignacio

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/64611>

---

## Abstract

Artificial vision for manufacturing is a very important step when the process is automatic, so to improve productivity with high quality, this investigation presents a new method for manufacturing by applying artificial intelligence for tool selection; in the proposed method, a camera takes a picture and artificial program is applied to process the image and thus generate it to the size of the cutting tool. This chapter describes the development of a new method by programming using artificial vision to select the cutting tool. The experimental results show that the combination of artificial vision and programming is capable of selecting the correct tool.

**Keywords:** artificial vision, cutting tool, artificial intelligence, image, manufacture

---

## 1. Introduction

In recent years, artificial vision is impacting the manufacturing process due to the mass adoption in different lines of research such as robotics [1, 2], machining [3, 4], and automotive [5], among others. However, numerous variables affect the machining process, for example, tool materials, total error compensation, concepts of autonomous manufacturing, and process condition monitoring, among others. On the other hand, the computer vision for cutting condition effects as roughness (presented by Sarma et al. [6]), the prediction of cutting conditions (by Gadelmawla et al. [7]), and path generation are included for milling Computerized Numerical Control (CNC) machines (by Eladawi et al. [3]).

At present, the development of a system capable of recognizing complex features for tool selection in commercial computer-aided manufacturing (CAM) transfers this task to the workers who do it based on their own experience, ability, and knowledge. The literature on

---

automatic tool selection using image processing is minimal; on the other hand, simple and easy to implement algorithms are the core of this method [8].

## 2. Artificial vision in manufacture

This research presents a new technique based on artificial vision using image processing for automatic tool selection in operations of milling-turning manufacture.

The proposed method starts recognizing an image taken by a photographic camera. Then, different techniques of image processing are applied, such as binarization to convert the image of gray scale to black and white; after segmentation to reduce the contour or increase it if necessary; directional morphology to determine the magnitude, position, and direction of edge of the part; and finally, a structural element with the shape of the tool is moved to generate the trajectory and dimensions of the cutting tool [9].

## 3. Artificial vision in lathe

This research consists of eight steps for machining 2D objects mainly for two-axis lathe machines: in step 1, an image designed in CAD or taken from a camera is extracted and saved to a DXF file; the next step (step 2) consists of transforming the DXF file to the BMP file (image); in step 3, a binarization is applied to convert the image to grayscale; step 4 consists of labeling to separate the object of the picture; in step 5, the perimeter of the piece is obtained; in step 6, a partial derivative to obtain the gradient in the edge of the piece is applied; in step 7, a structure element with the shape of the tool is moved in the image; and finally, in steps 8 and 9, a condition is applied to determine if there is no intersection between the selected tool and the object to generate the selection of the tool and the trajectory [9].

The method for automatic selection of the tool consists in identifying when an element structure with the shape of the tool travels in each pixel on the edge of the part in the image. If there is an intersection between the cutting tool and the workpiece, then the diameter of the cutting tool is automatically changed until there is no intersection. These steps and the previous steps are shown in **Figure 1**, as well as in the following steps [9]:

### Step 1. Generation of DXF file starting with a picture

The DXF file contains information of the piece inside the image (**Figure 2b**), while the CAD file is generated in 3D but it is exported in 2D (**Figure 2a**) [9].

### Step 2. File transformations (DXF to BMP)

When different transformations are applied, it is necessary to know the dimensions of the piece and keep the precision nearest between the part in physical (length in mm) and the image (length in pixels), an analysis applied is shown in **Table 1** [9].

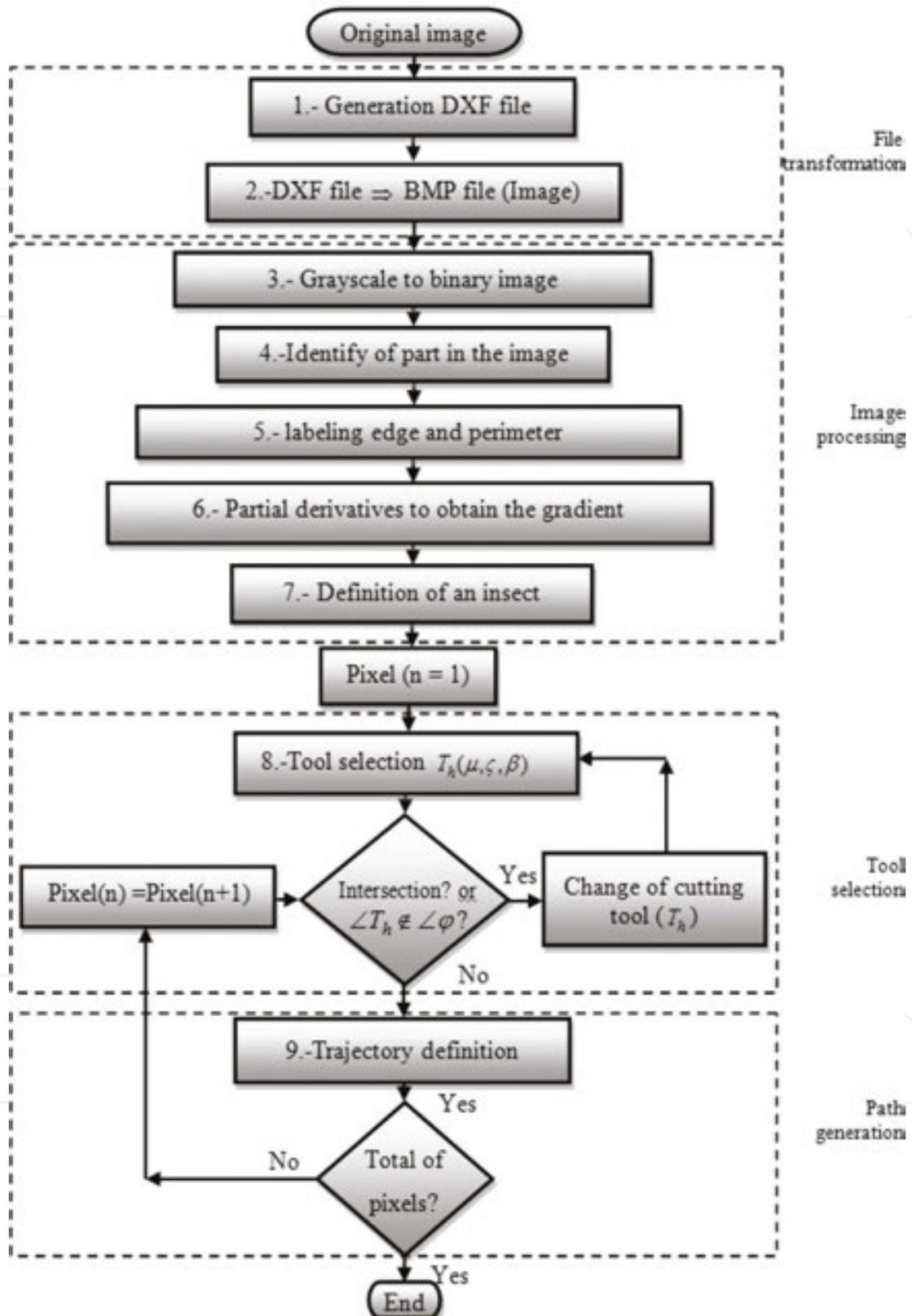


Figure 1. General diagram of the automatic cutting tool selection.

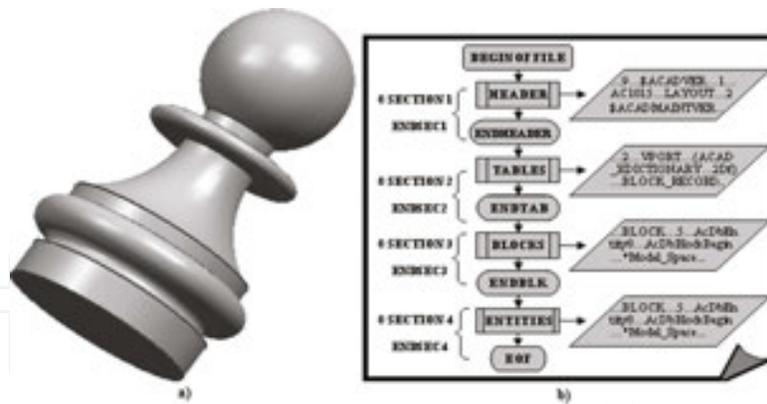


Figure 2. Principal file, (a) design from CAD and (b) DXF file [9].

Resolution	Scale	mm	Length (pixels)	Height (pixels)
800 × 600	4:3	1	15	11
3200 × 2048	25:16	1	62	40
5120 × 4096	5:4	1	100	80
7680 × 4800	16:10	1	151	94
10,000 × 10,000	1:1	1	196	196
50,800 × 50,800	1:1	1	1000	1000

Table 1. The resolution of this system.

The calibration value is also the resolution of this system and is approximately 1 μm. A method of data extraction [6] is applied to convert the DXF file to BMP file, as shown in Figure 3(a) and (b). The same distance is then automatically determined in pixels in the image, and a calibration value is obtained by dividing the distance in millimeters by the distance in pixels to obtain a relation of mm per pixel:

$$Relation = \frac{distance_{mm}}{distance_{pixel}} \quad (1)$$

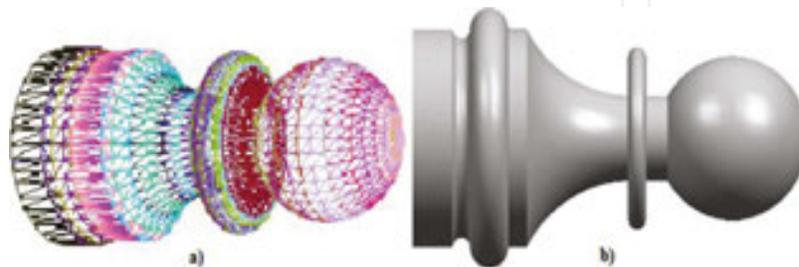


Figure 3. Files, (a) design without texture (Wire file) and (b) image (BMP file) [9].

All the dimensions are known in the DXF file. In order to find the directional gradient of the edge, image processing is applied.

### Step 3. Grayscale to binary image

The workpiece  $\lambda^\tau$  is a subset  $\lambda^\tau \subset \mathbf{f}$  of the image  $\mathbf{f}$ , where it represents a matrix with  $m$  pixels to be stored in an array  $f$  in the following way:

$$\lambda^\tau \subset \mathbf{f} \Rightarrow \sum_{\tau=1}^m (\lambda^\tau = \mathbf{f}) \quad (2)$$

where  $m$  is the total number of pixels in the image and  $\tau$  is the region of the image. After starting with the image that is shown in **Figure 3(b)**, a transformation (thresholding) is applied to convert the image of gray scale to binary using Eq. (3):

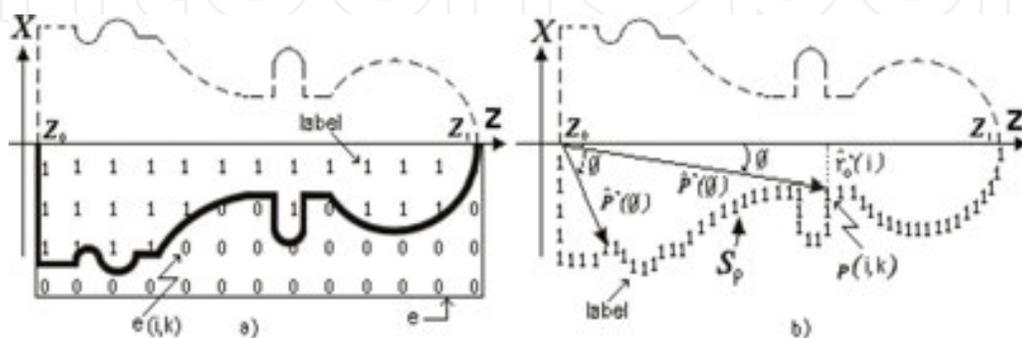
$$\sum_{i=1}^n \sum_{k=1}^m \{ \text{if } \mathbf{f}(i,k) > 128 \text{ then } \mathbf{f}^B(i,k) = 1 \text{ else } \mathbf{f}^B(i,k) = 0 \} \quad (3)$$

Each pixel has a value of 255 representing the largest tonality (white) and 0 when the minimum (black) cannot work with this whole range of colors.  $\mathbf{f}(i,k)|_n^m$  (image in gray scale 0 multicolor) is needed only for two  $\mathbf{f}^B(i,k)|_n^m$  (binary image), one for the piece (black) and the other for the rest of the image (white) [9].

### Step 4. Labeling of the workpiece into of the image

**Figure 4(a)** shows the labeling of the image and the piece  $e^{(i,k)}|_n^m$  (Eq. (4)). Be an image  $\mathbf{f}_k \subseteq \mathbf{f}(i,k)|_n^m$  white and black in the binary space  $\mathbf{f}^B$ , where

$$\mathbf{e} = \bigcup_k \{ \forall \mathbf{f}_k \in \mathbf{f}^B \mid \langle \text{if } \mathbf{f}_k = 0 \text{ then } x_k = 0 \rangle \vee \langle \text{if } \mathbf{f}_k = 1 \text{ then } x_k = 1 \rangle \} \quad (4)$$



**Figure 4.** Labeling definition, (a) labeling of part and (b) labeling of perimeter [9].

Or it can also be written as Eq. (5):

$$\sum_{k=\frac{m}{2}}^m \sum_{i=0}^n \{ \mathbf{e}(i,k) = [ \text{if } \mathbf{f}(i,k) = 1 \text{ then } \mathbf{f}_e(i,k) = 1 ] \cup [ \text{if } \mathbf{f}(i,k) = 0 \text{ then } \mathbf{f}_e(i,k) = 0 ] \} \quad (5)$$

where  $\mathbf{f}_e(i,k)|_n^m$  is the labeling image saved in the image of the exit  $\mathbf{e}(i,k)|_n^m$  of Eq. (5), taking half of the image  $k = \frac{m}{2}$  on the axis  $Z$ . The directional classification of the edge for half of the image in the point  $p(x,z)$  is presented in **Figure 3(b)**.

**Step 5. Labeling edge and perimeter**

The perimeter ( $P(\theta)$ ) can be obtained from the edge ( $|\nabla f|$ ) of the image ( $\mathbf{f}(i,k)|_n^m$ ) by Eq. (6):

$$\hat{P}(\theta) = 2 \int_{z_0}^{z_1} \sqrt{1 + \hat{q}'^2} dz = \int_0^{2\pi} \sqrt{\hat{r}_o'^2 + \hat{r}_o'^2} d\theta \quad (6)$$

The vector  $\hat{q}'(i,k)$  represents the partial derivative of the piece in the image  $f(i,k)$  with respect to a frame of reference ( $\hat{r}_0(i)$  and  $\hat{r}_0'(k)$  are the radii) for each pixel ( $p(i,k)$ ) (see **Figure 4(b)**).

Another way to calculate an integral is the arithmetic sum of each pixel of the image represented with the symbol ( $\Sigma \mathbf{f}(i,k)|_n^m$ ) for a frame of reference (of  $z_0$  to  $z_1$ ), with the aim of finding the perimeter  $S_p$  [10] that represents the labeled edge, as shown in Eqs. (7) and (8) [9]:

$$\sum_{N=0}^{\infty} \{ (\mathbf{E}_N(i,k) \leftarrow \{ \mathbf{p}(z, P(z)), \{ N = N + 1 \vee z = N \} \} ) \} \quad (7)$$

$$S_p = \sum_{k=\frac{m}{2}}^{m-2} \sum_{i=1}^{n-1} \left\{ \prod_{N=1}^{m-2} \langle \text{if } \mathbf{E}_N(i,k) = 1 \text{ then } \mathbf{f}(i,k) \leftarrow \mathbf{E}_N(i,k), \text{ else } \mathbf{f}(i,k) \leftarrow \mathbf{f}(i,k) \rangle \right\} \quad (8)$$

**Figure 4(b)** shows the labeling of the perimeter  $S_p$  in the edge of the piece for each point  $p(i,k)$ .

**Step 6. Using partial derivatives to obtain the gradient**

There are several methods to obtain the gradient (variations in intensity of pixels) (Eq. (9)), but among them the Sobel method is selected for its computational easeness is shown in Eqs. (10) and (11) [11], the gradient generates directional vectors, orientation of the piece, and direction of the contour of the workpiece:

$$\nabla f = \text{gradient } \mathbf{f} = \begin{cases} \mathbf{G}_x = \frac{\partial}{\partial x} \mathbf{f}(x, z) = \frac{\partial}{\partial x} \hat{\mathbf{i}} = \frac{\partial}{\partial x} \hat{\mathbf{u}}_1, \\ \mathbf{G}_z = \frac{\partial}{\partial z} \mathbf{f}(x, z) = \frac{\partial}{\partial z} \hat{\mathbf{k}} = \frac{\partial}{\partial z} \hat{\mathbf{u}}_2, \end{cases} \quad (9)$$

Therefore,

$$\mathbf{G}_x = \sum_{i=1}^{n-1} \sum_{k=1}^{m-2} \left\{ \text{Edge}_x(i, k) = \sum_{z=-1}^1 \sum_{x=-1}^1 \mathbf{e}(x, z) \mathbf{M}_x(z+2, x+2) \right\} \quad (10)$$

$$\mathbf{G}_z = \sum_{i=1}^{n-1} \sum_{k=1}^{m-2} \left\{ \text{Edge}_z(i, k) = \sum_{z=-1}^1 \sum_{x=-1}^1 \mathbf{e}(x, z) \mathbf{M}_z(z+2, x+2) \right\} \quad (11)$$

Here,  $\mathbf{M}_x$  and  $\mathbf{M}_z$  are the transformation matrices to generate the edge applied to the image. The vector gradient represents the maximum change of intensity for the point  $p(i, k)$ .

The magnitude ( $|\nabla f|$ ) and direction ( $\angle \nabla f$ ) are given by Eqs. (12) and (13), respectively,

$$|\nabla f| = \sqrt{G_x^2 + G_z^2} \quad (12)$$

$$\angle \nabla f = \tan^{-1} \frac{G_z}{G_x} \quad (13)$$

Figure 5(a) illustrates the vector direction of the edge of the part sample.

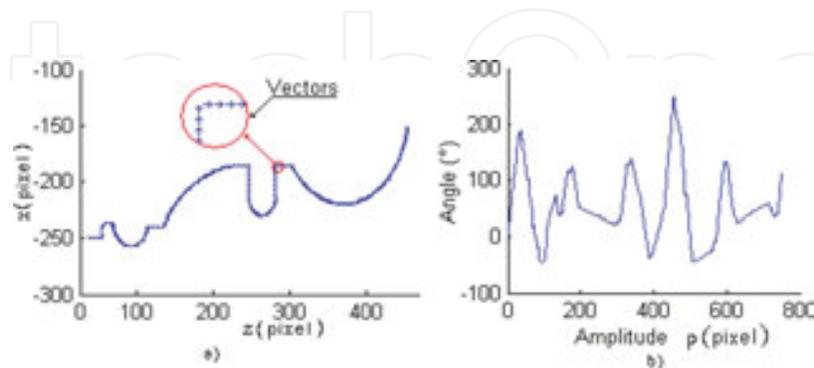


Figure 5. Vectors of part, (a) direction vector of part and (b) magnitude of edge direction [9].

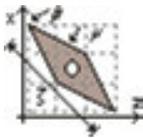
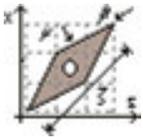
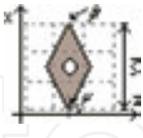
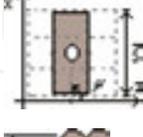
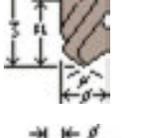
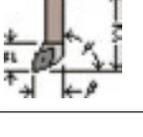
Figure 5(b) shows the directional angles according to the trajectory of the edge, when the gradient is applied to binary image in the part.

The magnitude and direction of the edge are saved in the variable  $Maq(i, k)$  using Eq. (14) to generate the dimension of structural element:

$$\sum_{k=m}^{\frac{m}{2}-1} \sum_{i=1}^{n-1} \left\{ \text{if exist edge then } \left\{ \mathbf{Maq}(i, k) \leftarrow \begin{cases} |\nabla f|(i, k) \text{ "magnitude" } \\ \angle \nabla f(i, k) \text{ "direction" } \end{cases} \right\} \right\} \quad (14)$$

### Step 7. Automatic tool selections

The workspace is considered according to the size of the piece but it is necessary to consider other collisions as the turret with the part, these will be considered in future issues [10]. **Table 2** shows the common cutting tool, the dimensions of the cutting tool are obtained from manuals and introduced into the software to be converted into structural elements with the length, orientation, and radius of the nose, this part is shown in **Figure 6(a)**. Among the common dimensions are those of 15°, 35°, and 45°, with 9525 mm requiring 36 pixels, 2540 requiring a mesh of 25.5 square units of pixels, and the structural element requiring 414 pixels, see **Figure 6(a)** [9].

Tool for turning			Type	$\mu(^{\circ})$	Size (mm)			
Name	Letter	Shape			$\zeta$	$\beta$	FL	$\varphi$
1. Right	R		External	35	19.049	0.406	-	-
			Internal	125			-	-
2. Left	L		External	197	19.0496	0.406	-	-
			Internal	19			-	-
3. Neuter	N		External	90	12.699	0.507	-	-
			Internal	93			-	-
4. Knife	K		External	90	12.699	-	-	-
			Internal	95			-	-
5. Drilling	D		External	60	44.449	-	38.099	12.699
			Internal					
6. Holder	H		S16R PCLNR-L 09	95	71.118	20.141	20.319	19.989

**Table 2.** Common inserts (structural element).

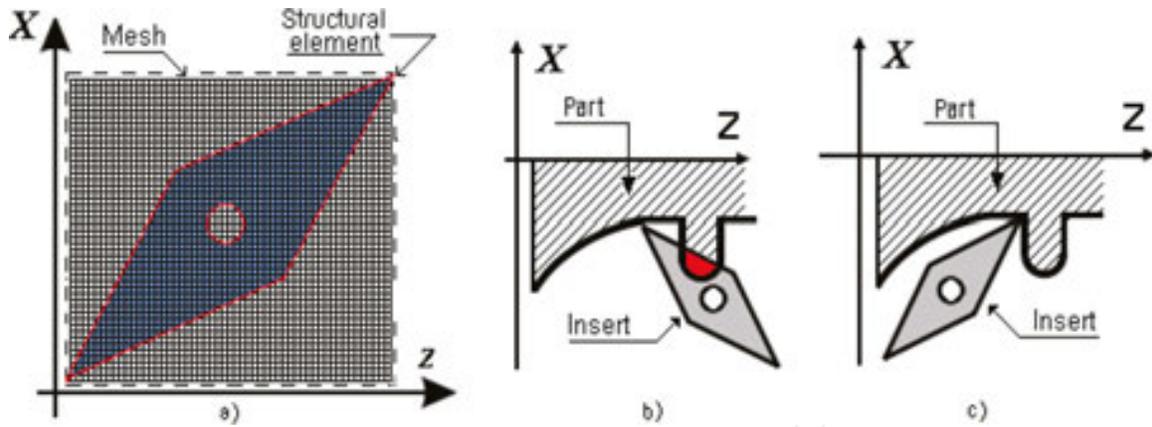


Figure 6. (a) Structural element for right insert, (b) incorrect insert and (c) correct insert [9].

There are several conditions for machining between them and it is necessary to determine if the cut is internal or external, and whether it is an insert right ( $180^\circ \leq \angle \nabla f < 270^\circ$ ) or left ( $270^\circ \leq \angle \nabla f < 360^\circ$ ). The common are those that are situated in the last two quadrants, the first two quadrants are generally used for internal machining. Eq. (15) generates the movements of the structural element through the image; the angle marks the start of the selection of the cutting tool in machining  $\text{Maq}(i, k) = 1$  [9]:

$$\sum_{k=m}^{\frac{m}{2}-1} \sum_{i=1}^{n-1} \{ \text{if } \text{Maq}(i, k) = 1 \text{ then select insert } \{ \mu(i, k) \leftarrow \angle \nabla f (\text{angle of edge}) \} \} \quad (15)$$

where  $\mu$  is the dimension for cutting tool presented in Eq. (16):

$$\sum_{k=m}^{\frac{m}{2}-1} \sum_{i=1}^{n-1} \left\{ \text{if } \text{Maq}(i, k) = 1 \text{ then } \mu(i, k) \text{ is } \begin{cases} \text{left insert, if } 17.5^\circ \leq \mu(i, k) < 45^\circ \\ \text{right insert, if } 107^\circ \leq \mu(i, k) < 180^\circ \\ \text{neuter insert, if } \mu(i, k) = 90^\circ \\ \text{knife, if } \mu(i, k) = 90^\circ (\text{up and down}) \end{cases} \right\} \quad (16)$$

Table 2 shows the common inserts that are generated in structural elements. Figure 6(a) illustrates a structural element generated with pixels to be moved through the contour so that the contour matches the nose radius of the tool. The number of pixels in a line according to the used resolution of  $800 \times 600$  is five pixels; the evaluation is taken from reference [9]. Second, the pixels that are represented the line in Figure 6(a) are used only to demonstrate the method; the pixels processed are the inner using centroid method.

When a structural element passes through the edge of the workpiece, it is necessary to know if this insert is correct, free of intersection, if there is intersection, and if another insert is treated. The structural element (Figure 6b) is displaced through all the images, when a structural element (Figure 6b) collides (Figure 6c), other structural element is chosen automatically, this process is determined in the software itself. The technique for automatic tool selection may be

to change the radius, length, or angle of the insert according to the inserts, obtained from handbooks [9].

After the correct holders and inserts have been selected to work on each area, many changes in cutting tools are probably required. The number of changes can be reduced by ordering the changes with the next procedure (Eq. (17)):

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-2} \left\{ \begin{array}{l} \text{if } \left\{ (S_P(i, j) = 1) \text{ and } \sum_{h=-1}^1 \sum_{\psi=-1}^1 \lambda[i + \psi][j + 1] = 1 \right\} \\ \text{then } \mathbf{Tray}(i, j) \leftarrow \lambda[i + \psi][j + 1] \end{array} \right\} \quad (17)$$

To determine the tool path, each pixel should share information with eight neighbors and follow the path of the perimeter (edge  $S_P$ ) in the mesh  $\lambda[i + \psi][k + 1]$ .

### Step 8. Generation of trajectory

Figure 7(a) illustrates the method of zig using one way, but there are other methods of creating paths such as zigzag of two paths (back and forth), zig with contour, which follows only the path of the edge of the piece widely used for finishing and roughing at the same time; the follow-periphery that has the function to be used only for finishing; the trochoidal profile is used in special cases when the piece has peaks, elevations, or very long inclinations; and on the other hand, the generation of trajectory is used only for rough cutting. To generate the trajectories, a structural element with the shape of the cutting tool with angle  $\mu$ , length  $s$ , and nose angle  $\beta$  is displaced through all the images from right to left.

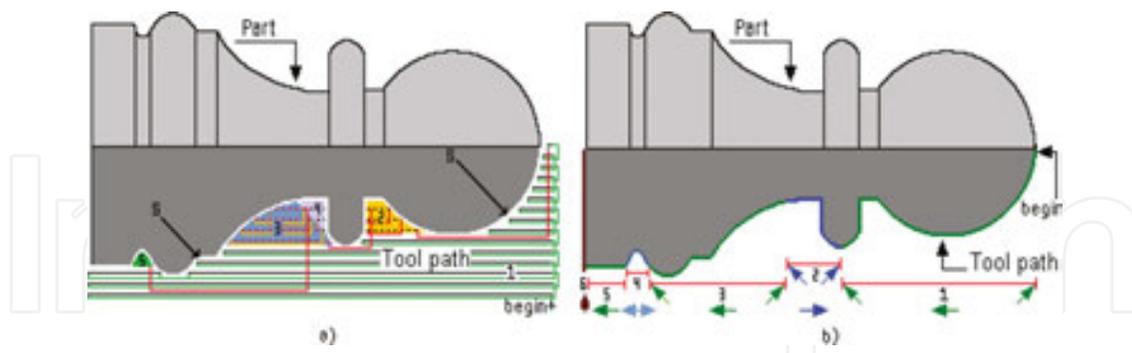


Figure 7. Tool selection (final piece), (a) rough turn (zig) and (b) finish turn (zig with contour) [9].

Finally, to move the tool in the image, it is necessary to use a counter ( $k = 0, i = i + 1$ ) with the total number of pixels  $m$ . This method is obtained by Eq. (18) and is displayed in Figure 7(a) [9]:

$$\sum_{k=m}^{\frac{m}{2}-1} \sum_{i=1}^{n-1} \left\{ \mathbf{Move\_Tray}(i, k) = f(i, k); \left\{ \text{if } f(i, k) = \mathbf{Tray}(i, k) = 1 \text{ then } k = k + 1 \right\} \right\} \quad (18)$$

Machining time is given by Eq. (19):

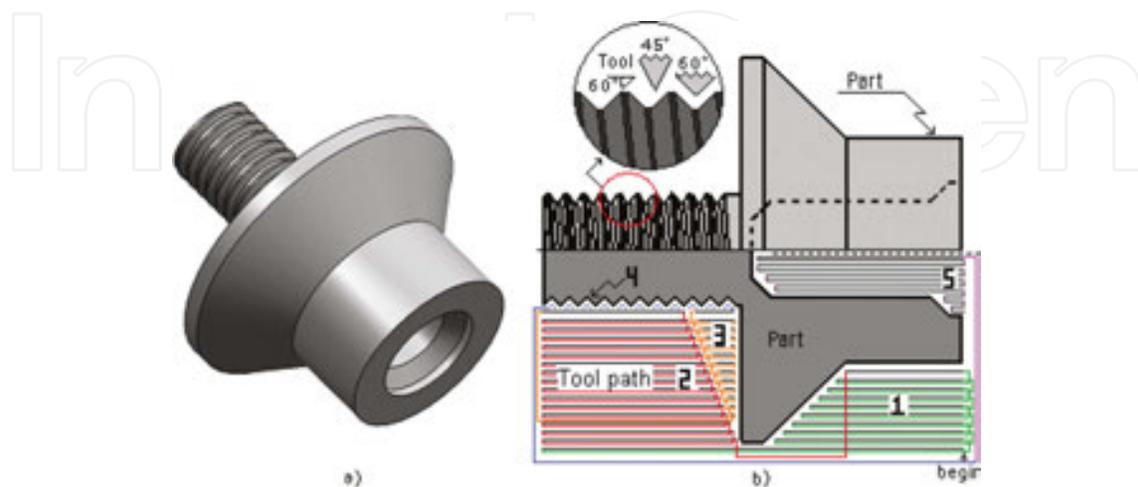
$$T = \sum_{i=1}^n \frac{L_i}{v_i} \quad (19)$$

#### 4. Results in lathe

With the proposed method, an image can be directly taken by a camera or other devices for automatic tool selection. The use of an image generated from DXF file is to compare the structural elements (part and tool) in pixels using PI; the original dimensions from DXF file are used to compare them if the design comes from CAD file. This new method, which reduces the complex, complicated, and difficult to understand mathematical algorithms, shows an easy simulated viewing for selecting cutting tool. To show the method for automatic selection of the tool, two pieces were used, the results are presented in **Table 2**. The generation of trajectories is shown in **Figure 7(a)** and **(b)**, and the selection of the tool in **Table 3** for the first piece.

Rough turn		Finish turn	
Area	Type	Line	Type
1	L	1	L
2	L	2	R
3	L	3	L
4	R	4	N
5	N	5	R
6	-	6	K

**Table 3.** Selected inserts for part of **Figure 7**.



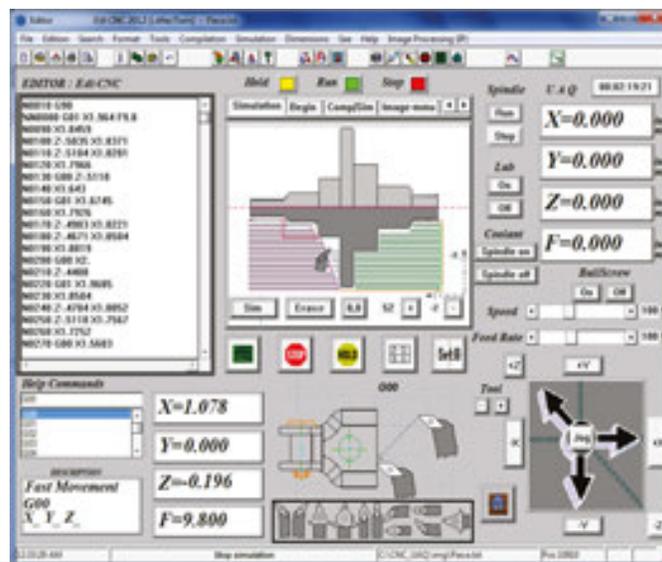
**Figure 8.** Tool selection, (a) final piece and (b) thread and rough turn (zig) [9].

**Table 3** shows the results of the selected inserts by applying the proposed method. Where the numbers represent the area for each insert, and L, R, or N represents the types of inserts more common than before which was proposed relative to its size.

The second example is presented in **Figure 8**. The selected cutting tools are presented in **Table 4**, where an existing thread machining is represented by the number four and an internal machining is labeled number five.

Rough turn		Finish turn	
Area	Type	Line	Type
1	L	1	L
2	L	2	-
3	R	3	R
4	M12_x_1.25	4	-
5	D/L	5	D/L

**Table 4.** Cutting tool selection for second example.



**Figure 9.** Final software for tool selection [9].

The developed software is shown in **Figure 9**. It was developed in Microsoft Visual C++ 2010 and tested in 2D images generated in any CAD software and exported to DXF file.

## 5. Conclusions

In this work, a new method for automatic tool selection using techniques of image processing for computer numerical control lathe machines has been presented. The proposed methodology has been implemented in images generated by CAD software and exported to DXF file. The resolution of this system is determined as approximately  $1 \mu m$ .

From this, it is clear that this methodology can be implemented in commercial CAM software in order to systematize the lathe CNC process. The novelty of this article is the use of image processing to automatically generate the selection of the cutting tool, the design of the part can be from a photograph taken from a camera or other device or directly from a CAD.

## 6. Artificial vision in mill

A 2.5D solid model can be defined as a cut with a series of 2D tool paths at different Z slices of a 3D solid model. Nowadays, a large percentage of pieces used in the industry have a shape of contours, where the base face is a plane denominated by 2.5 axis. Automatic tool selection in milling operation is one of the important steps of process planning. Moreover, at present, the commercial computer-aided manufacturing software transfers this task to the worker who does it based on its own experience, ability, and knowledge. Considerable articles have reported about the variables that affect the milling process such as material piece selection [7, 12], tool selection [7], cutting conditions [7], tool materials [12], tool sequence [13], cutting fluids tool selection [12], tool path [6], and control and identification, among others.

The tool selection, task commonly made by a human operator, is an important aspect in machining processes, since if the tool selected is incorrect, it can produce dimensional errors in workpiece, possible crashes, and consequently, reject the piece. In this way, some researches based their principles on the selection of the tool for machining 2.5D parts as in the case of Ahmad et al. [13], who present an optimization algorithm for the problem of tool sequence [8]. Or the method proposed by Lim et al. [14] who used experimental algorithms using mathematical Boolean to determine the optimal setoftools in pockets with the integration of CAD/CAM.

In the same way, Hemant et al. [15] developed an algorithm for the tool selection used by Veeramani tool, although using dynamic programming, human intervention is necessary because of its mathematical complexity [4]. The tool selection using image processing based on the shape is a new method in the literature, the core of this research is simple and easy to implement algorithms [8].

Although the image processing plays a very important role when the images are manipulated inside the morphology (opening operation), the main research is the defects by Tunák et al. [16], the cutting conditions by Sarma et al. [7] and Gadelmawla et al. [3], and the path generation by Eladawi et al. [6, 8].

The contribution of this research is the easy way to simulate the cutting path, automatic tool selection, easy method for 2.5 axis machining in operation of milling, reduction of errors, and not requiring prior knowledge. The novelty of this work is the use of morphology based on the image processing applied for the automatic tool selection. The advantages of this method are: easy to implement, eliminates the decisions and errors when the tool is selected [8].

## 7. Introductions

Application of directional morphology in the tool and the workpiece with the sole aim of finding the selection of the cutters for 2.5-axis machining in the bibliography is a new proposal. The method begins by taking the mesh of a 3D object exported to a DXF file (Mejia et al. [8]) or an image taken by a camera. To manipulate this file, it is necessary to extract the coordinates, lengths, and positions of the pixels generated on the frontier of the piece on the surface. The pixels are discretized on the edge of the piece to obtain normal vectors and stored in a text file. The file data gives the necessary information about the orientation of tool when it passes through that place. The dimensions of the tools are stored and discretized; these were extracted from the handbook more common in the industry. The resolution of the system depends on the size of the tool and the complexity of the piece. To improve the algorithm, it is necessary to reduce the size of the pixels, after the pixels of the tool are displaced on the frontier of the piece. Each movement of the tool is inspected if there is no collision.

The image processing techniques used are: (a) morphological operations: using erosion and dilation [17, 18], these methods reduce or increase the contour of the workpiece; (b) binary image: the pixels that contain a gray scale  $f(i,j)$  are converted to black and white  $f^B(i,j)$ ; (c) direction vectors of piece: Sobel model is the common operator because it has better performance and is easy to use [19]; (d) extracting piece: the common method is labeling (scan mask) to register images; (e) Software CAD-CAM-CAE: these are the systems that graphics, designs, and simulates, respectively; (f) 2.5D models: the images in the present article are 2.5D model. These models are from a single 2D image having a manual axis that is the axis Z; and (g) structural elements: the cutting tool is represented in pixels, because a greater number of pixels to represent the cutting tool improves accuracy [8].

The work published by Bithika and Asit [18] applies mathematical morphology to detect manufacturing defects. It is an example of the application of image processing, using such techniques for measuring the effect of cutting speed on the surface roughness, which is another example that provides a new strategy in manufacturing, as presented by Sarma et al. [7].

## 8. Methodology

This research consists of six steps for machining 2D and 2.5D objects mainly for 2.5-axis lathe machines, an image designed in CAD or taken from a camera is extracted and saved to a DXF file, after transforming the DXF file to the BMP file (image), a binarization is applied to convert the image to grayscale, the labeling to separate the object of the picture, the perimeter of the piece is obtained, a partial derivative to obtain the gradient in the edge of the piece is applied, in step seven, a structure element with the shape of the tool is moved in the image, finally, a condition is applied to determine if there is no intersection, using mathematical morphology, especially erosion and dilation to generate automatic tool selection and tool path as it is shown in **Figure 10** [8].

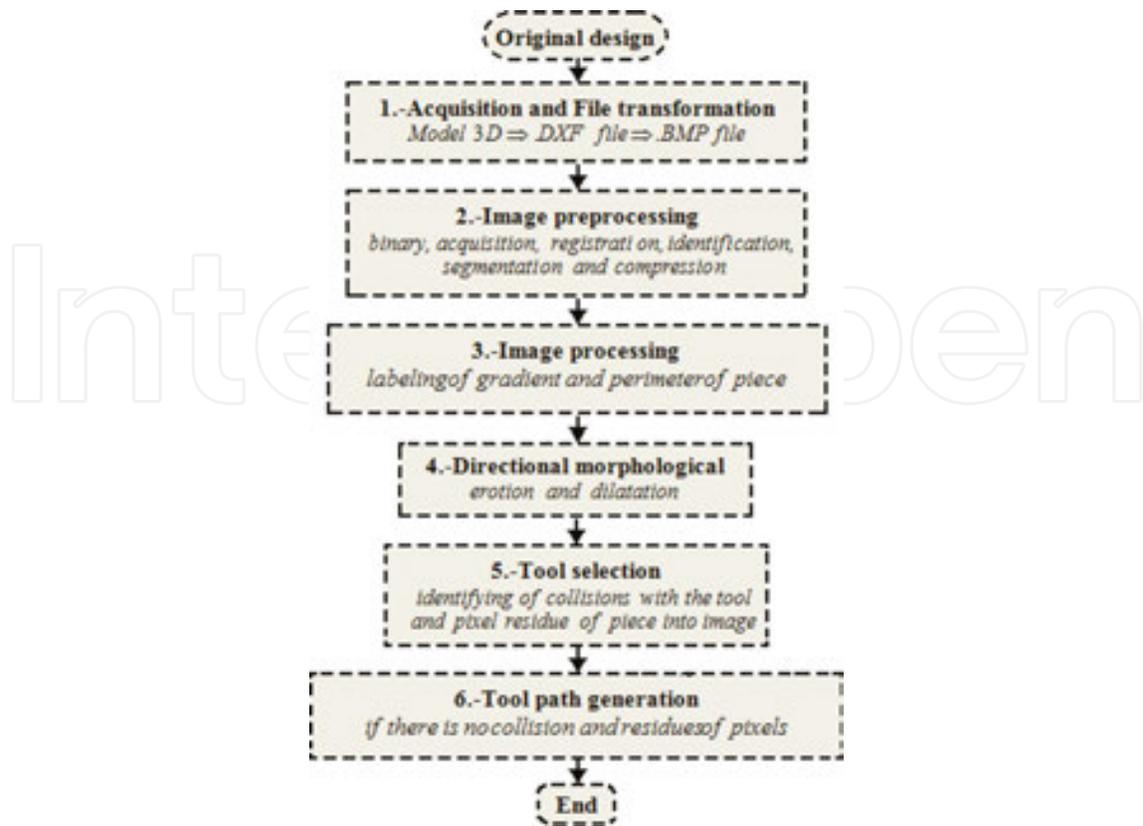


Figure 10. General diagram of the automatic cutting tool selection [8].

### Step 1. Generation of DXF file starting with a picture

The DXF file (Figure 11b) contains information of the piece inside the image (Figure 11a), the CAD file is generated in 3D but it is exported in 2D.

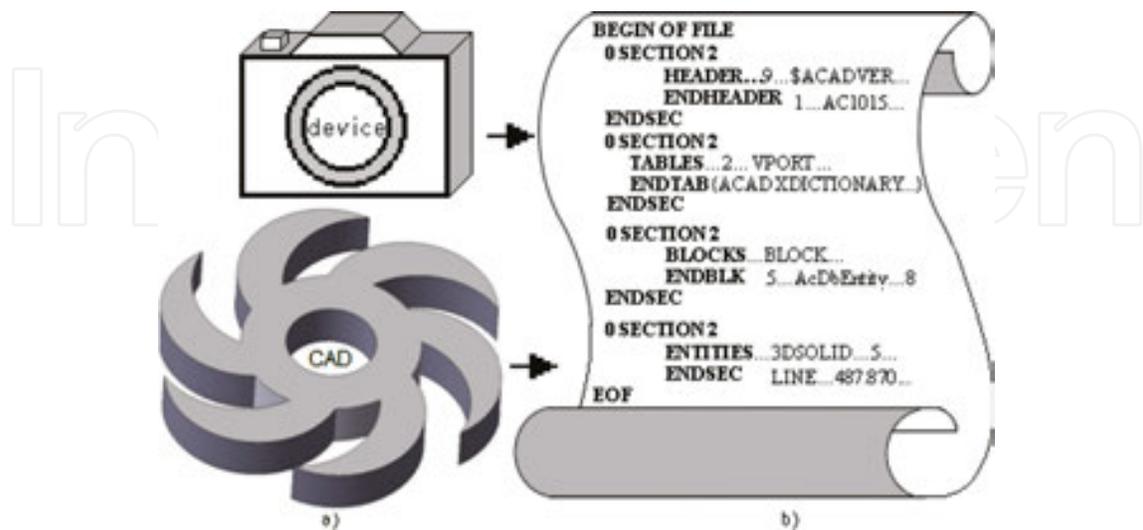
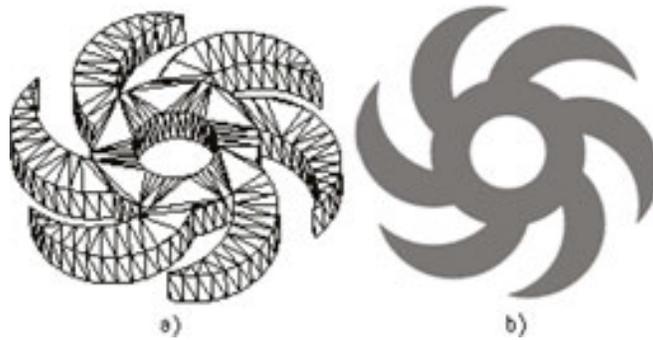


Figure 11. Original files, (a) design from CAD or camera and (b) DXF file [8].

When different transformations are applied (DXF to BMP), it is necessary to know the dimensions of the piece and keep the precision nearest between the part in physical (length in mm) and the image (length in pixels), an analysis is applied, the calibration value of this system is in micrometer, the adjustment parameters for 1 mm of 800×600 (scale 4:3, 15×11 pixels) and 50,800×50,800 (scale 1:1, 1000×1000 pixels) [8].

The archive DXF file contains the necessary information about the part that is necessary when the DXF file is converted to a BMP file, as shown in **Figure 12(a)** and **(b)**. The calibration is the relation between the distance in millimeters ( $\text{distance}_{\text{mm}}$ ) and the relation of millimeters per pixel (Relation (mm per pixel)).



**Figure 12.** Files, (a) design without texture (Wire file) and (b) image (BMP file) [8].

**Step 2. Image preprocessing**

Binary image: a binary image ( $(\mathbf{f}^B(i, j)|_n^m)$ ) in 2.5D is a subset  $\chi_D$  of  $\mathfrak{R}^3$  if  $\{0, 1\}^{2.5}$  represents the set of functions as  $\chi_D \subset \mathfrak{R}^3$  in the set  $\{0, 1\}$ , so that any binary image can be represented by a characteristic function  $\chi_D: \mathfrak{R}^3 \rightarrow \{0, 1\}$ . After starting with the image, a transformation (thresholding) is applied to convert the image of gray scale to binary using

$$\sum_{i=1}^n \sum_{k=1}^m \left\{ \text{if } \left( \begin{array}{l} \mathbf{f}(i, j) > T(\text{object pixels}) \Rightarrow A_1 = \mathbf{f}(i, j) \\ \mathbf{f}(i, j) \leq T(\text{background pixels}) \Rightarrow A_2 = \mathbf{f}(i, j) \end{array} \right) \text{ then } \mathbf{f}^B(i, j) \right\} \quad (20)$$

where  $\mathbf{f}^B(i, j)$  is equal to

$$\sum_{i=1}^n \sum_{k=1}^m \left\{ \mathbf{f}^B(i, j) = \frac{\text{Average value of } A_1 + \text{Average value of } A_2}{2} \right\} \quad (21)$$

where  $\mathbf{f}(i, k)|_n^m$  is the image in gray scale (image matrix of  $n \times m$  pixels) and  $f^B(i, j)|_n^m$  is the result of the transformation of gray scale into a binary image.

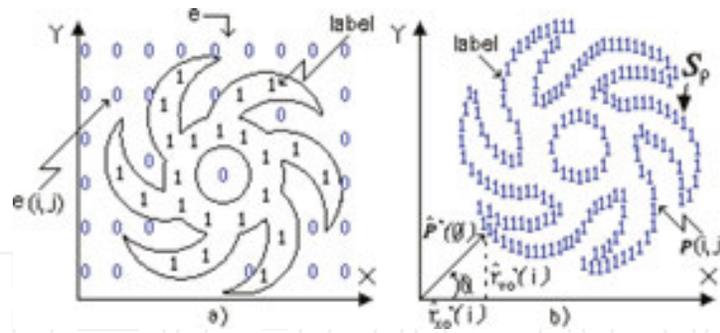


Figure 13. Labeling definition, (a) labeling of piece and (b) labeling of perimeter [8].

### Step 3. Image processing

Detection and labeling of gradient and perimeter in the edge of the piece: The following step is the binary label, or limitation, of the piece  $e(i,k)_n^m$ , see **Figure 13(a)** (Eq. (22)). Be an image  $f_k \subseteq f(i,k)_n^m$  white and black in the binary space  $f^B$

$$\sum_{j=1}^m \sum_{i=1}^n \left\{ e(i,j) = \begin{cases} f(i,j) = 1 \Rightarrow f_e(i,j) = 1 \\ f(i,j) = 0 \Rightarrow f_e(i,j) = 0 \end{cases} \right\} \quad (22)$$

where  $(f(i,j)_n^m)$  is the labeling image saved in the image of exit  $e(i,j)_n^m$  of Eq. (22). The directional classification of the edge for half of the image in the point  $p(x,y)$  is presented in **Figure 13(b)**.

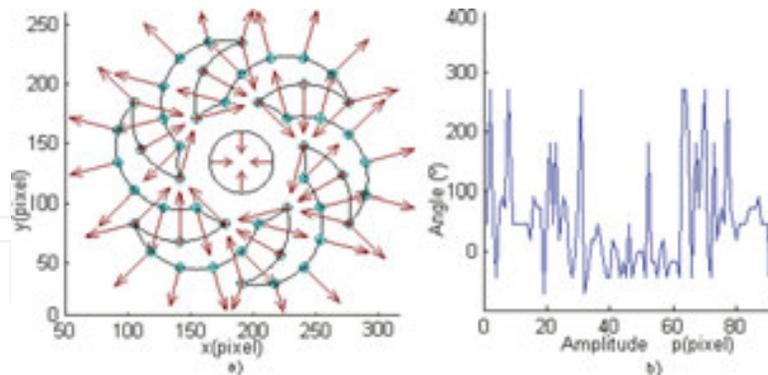


Figure 14. Vectors of piece, (a) direction vector of piece and (b) magnitude of edge direction [8].

Perimeter of the piece: The perimeter  $(\hat{P}(\theta))$  can be found by applying an edge  $(|\nabla f|)$  into the image  $(f(i,j)_n^m)$  to provide information about the shape of the object, and the labeled perimeter  $S_p$  can be defined as the position vector for the edge. **Figure 14(b)** presents the result of the trajectory of the labeled perimeter  $S_p$  represented with label "1" in the point  $p(i,j)$  of the image.

**Figure 14(a) and (b)** show the direction vectors in the contour of the workpiece when image processing is applied.

The magnitude ( $|\nabla f|(i, j)$ ) and direction ( $\angle \nabla f(i, j)$ ) of the labeling gradient and perimeter ( $\angle \nabla f(i, j)$ ) of the edge are saved in the variable  $\mathbf{Maq}(i, j)$  to generate the dimension of structural element:

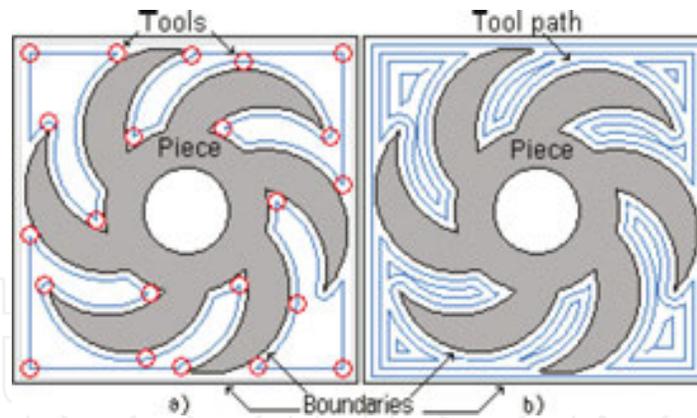
$$\left\{ \forall \mathcal{S}_p(i, j) \in f(x, y) : \exists \left( \begin{matrix} |\nabla f|(i, j) \wedge \\ \angle \nabla f(i, j) \end{matrix} \right) \subset \mathbf{Maq}(i, j) \right\} \quad (23)$$

Therefore,

$$\mathbf{Maq}(i, j) = \int_{j=1}^{j=m-1} \int_{i=1}^{i=n-1} \left[ (|\nabla f|(i, j) \wedge \angle \nabla f(i, j)) \right] dx dy \quad (24)$$

#### Step 4. Directional morphology

Directional morphologies to generate size of tool, erosion, and dilation are fundamental operations in morphological image processing, defined for binary images. To grayscale images and to complete lattices, dilation ( $\delta$ ) and erosion ( $\epsilon$ ) are formed by a structural element.



**Figure 15.** Directional morphology, (a) tool size dilation and (b) tool path dilation [8].

Dilation allows thinning of the contour of the part in the image  $\mathbf{f}(i, j)$ , using a structural element  $b(i, j)$ , which is developed in Eq. (25), see **Figure 15**:

$$(A \oplus B)(w, q) = \max \{ \mathbf{f}(w - i, q - j) + b(i, j) \} \quad (25)$$

Such that  $(w-i, q-j) \in D_A, (i, j) \in D_B$

With  $D$  functions with respect to  $A$  and  $B$  [4].

Eqs. (26) and (27) represent fundamental operations of dilation  $A \oplus B$ , where  $A$  contains all pixels and  $B$  is a structural element of cutting tool with  $\hat{B}$  reflected in  $B$  a variation of  $s$ :

$$A \oplus B = \left\{ s \mid (\hat{B})_s \cap A \neq \Phi \right\} \quad (26)$$

Equivalently,

$$A \oplus B = \left\{ s \mid ((\hat{B})_s \cap A) \subseteq A \right\} \quad (27)$$

Dilation is a joining of the translations of one picture for each pixel of an image  $B$ , called structural element as

$$A \oplus B = \bigcup_{b_i \in B} \mathbf{A}_{b_i} \quad (28)$$

Therefore

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ \text{if} \left[ \begin{array}{l} ((f(i,j) == 0)) \delta \delta ((f(i,j-1) == 1) \mid \mid f(i- \\ 1,j) == 1 \mid \mid f(i,j+1) == 1 \mid \mid f(i+1,j) == 1) \end{array} \right] \Rightarrow f(i,j) \leftarrow 1 \right\} \quad (29)$$

$$(A \ominus B)(w,q) = \min \{ \mathbf{f}(w+i,q+j) - b(i,j) \} \quad (30)$$

Such that

$$(w+i,q+j) \in D_A, (i,j) \in D_B \quad (31)$$

With  $D$  functions with respect to  $A$  and  $B$  [8].

Eqs. (32) and (33) represent fundamental operations of erosion  $(A \ominus B)$ , and inverse operation to dilation:

$$(A \ominus B) = \left\{ s \mid (B)_s \subseteq A \right\} \quad (32)$$

$$(A \ominus B) = \{p \mid B_p \subseteq A\} \tag{33}$$

The set of edges reduction, elimination of white dots, and the expansion of the small black dots of an image B are called erosion:

$$A \ominus B = \bigcap_{b_i \in B} A_{b_i} \tag{34}$$

Therefore

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ \text{if} \left[ \begin{aligned} &((f(i,j) == 1)) \delta \delta ((f(i,j-1) == 0) \vee f(i- \\ &1,j) == 0 \vee f(i,j+1) == 0 \vee f(i+1,j) == 0) \Rightarrow f(i,j) \Leftarrow 0 \end{aligned} \right. \right\} \tag{35}$$

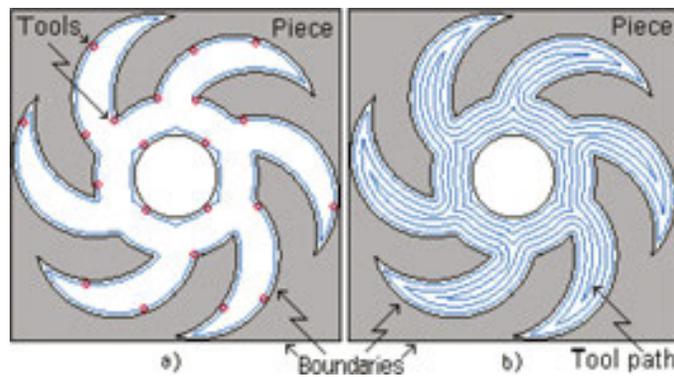


Figure 16. Directional morphology, (a) tool size erosion and (b) tool path erosion [8].

Modifying Eqs. (29) and (35) of directional morphology to generate the structural element of insert and tool path using the workspace boundaries, edge, and the perimeter function of the piece generating Eq. (39) and as results provided by Figures 16 and 8:

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ SE_1(i,j) = \sum_{a=-2}^2 (i,j+a) \right\} \tag{36}$$

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ SE_2(i,j) = \sum_{b=-2}^2 (i+b,j) \right\} \tag{37}$$

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ \sum_{h=n-4}^n \left( SE_{n-2}(i,j) = \sum_{c=-1}^1 \sum_{d=-1}^1 (i+c,j+d) \right) \right\} \tag{38}$$

$$SE = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \left\{ \{SE_1(i, j)\} \cup \{SE_2(i, j)\} \cup \dots \cup \left\{ \sum_{h=n-4}^n (SE_{n-2}(i, j)) \right\} \right\} \quad (39)$$

$$if \{ (SE_{n-2}) == ((SE_1) or (SE_2)) \} then eliminate (SE_{n-2}) \} \quad (40)$$

where  $(SE_1)$  is the first piece of structural element and  $(SE_1)$  is given by

$$SE_1 = \sum_{a=-2}^2 (i, j + a) = (i, j - 2) \cup (i, j - 1) \cup (i, j) \cup (i, j + 1) \cup (i, j + 2) \quad (41)$$

### Step 5. Automatic tool selections

Table 5 shows the most common cutting tool. The dimensions of the cutting tool are obtained from manuals and introduced into the software to be converted into structural elements with the length, orientation, and radius of the nose, this part is shown in Figure 16(a). Among the common dimensions are those of 15°, 35°, and 45°, with 9525 mm needing 36 pixels, 2540 requiring a mesh of 25.5 square units of pixels, and the structural element having 414 pixels, see Figure 17(a) [8].

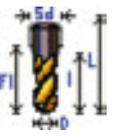
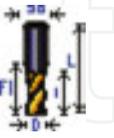
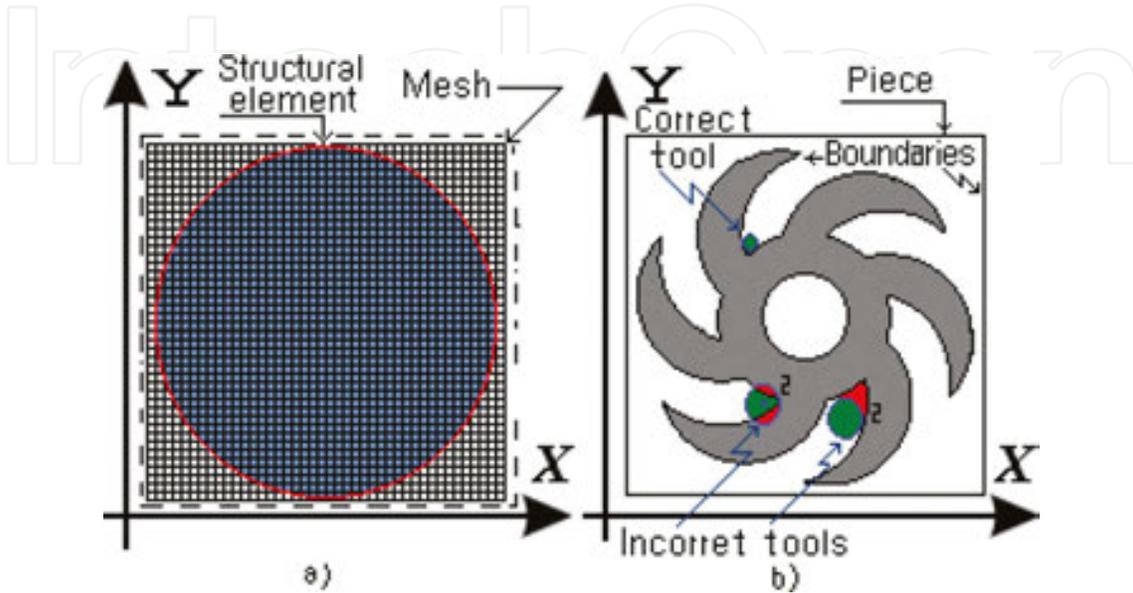
Tool for milling		Type	Size (mm)						
Name	Shape		<i>D</i>	<i>d</i>	<i>L</i>	<i>l</i>	<i>FI</i>	<i>Sd</i>	$\phi(^{\circ})$
Ballnose		Milling	0.016	0.016	40.149	15.181	20.417	0.016	0.009
			0.200	0.199	20.149	15.181	17.417	0.210	0.099
Endmill		Drilling	0.150	0.015	20.149	14.181	16.417	0.015	0.749
Conic		Milling	0.099	0.099	25.149	14.181	20.417	0.100	0.049
			0.049	0.049	15.149	7.181	12.417	0.050	0.025
Corner			0.009	0.005	10.149	5.181	7.4173	0.010	0.005
Dovetail			0.309	0.309	30.000	19.999	22.999	0.400	0.200
Chamber			0.349	0.349	40.149	15.181	20.417	0.350	0.210
Counter		Facing	0.399	0.399	45.149	10.181	15.417	0.380	0.230
Holder			0.599	0.599	25.149	10.181	15.417	0.390	0.250

Table 5. Common cutting tool (structural element) used.

The cutting tools are shown in **Table 5**, where  $D$  is the diameter,  $d$  is the tip diameter,  $L$  is the major length,  $l$  is the minor length,  $Fl$  is the usable length,  $Sd$  is presetting, and  $\varphi$  is the corner radius of the cutting tool. The tool is generated from handbooks; although, a real insert as the drilling cutting tool with a diameter of 5.56 mm ( $D$ , 21 pixels) requires a structural element as a tool of  $\pi(20)^2$  pixels. For more details see **Figure 17(a)**.



**Figure 17.** Tool selection, (a) structural element and (b) incorrect tool and correct tool [8].

The conditions for machining to determine the cutting tool are given by Eq. (42) that generates the movements of the structural element through the image, and the angle labels the start of the selection of the cutting tool in machining  $M(i,k) = 1$  [9].

$$\sum_{j=m}^{m-1} \sum_{i=1}^{n-1} \{ \text{if } M(i,j) = 1 \text{ then select insert } \{ \mu(i,j) \leftarrow \text{diameter} \} \} \tag{42}$$

The structural element (**Figure 17a**) is designed to create a trajectory using Eq. (42). To check the intersection (**Figure 17b**), it is necessary that the pixels have the same coordinates. The design of the software to create the automatic tool selection is developed in Microsoft Visual C++ 2010, generating a matrix with coordinates  $(i, j)$  of the image.

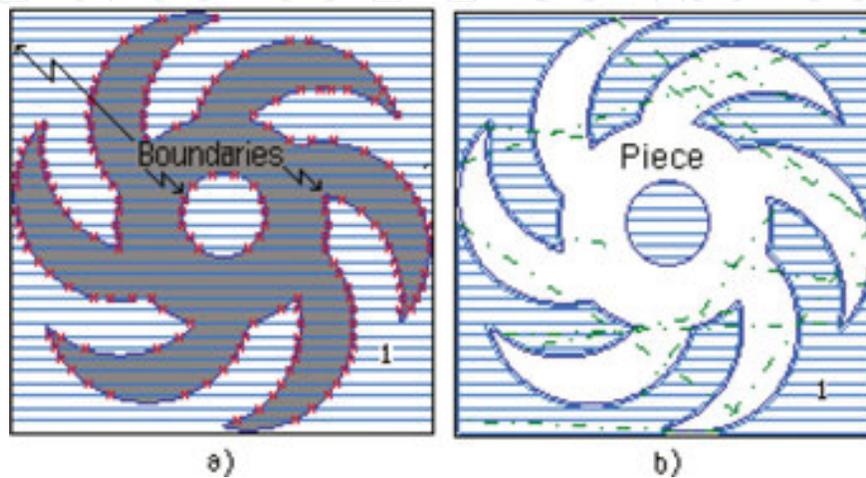
After the correct holders and inserts have been selected to work in each area, many changes in cutting tools are probably required. The number of changes can be reduced by ordering the changes with the next procedure:

$$\sum_{i=1}^{n-1} \sum_{j=1}^{m-2} \left\{ \text{if } \left\{ (S_p(i,j) = 1) \text{ and } \sum_{h=-1}^1 \sum_{\psi=-1}^1 \lambda[i+\psi][j+1] = 1 \right\} \right. \\ \left. \text{then Tray}(i,j) \leftarrow \lambda[i+\psi][j+1] \right\} \tag{43}$$

To determine the tool path, each pixel should share information with eight neighbors and follow the path of the perimeter (edge  $S_p$ ) in the mesh  $\lambda[i + \psi][k + 1]$ .

### Step 6. Generation of trajectory

In the present article, zig and zig with contour were developments to generate the trajectories. A structural element with the shape of the cutting tool, with diameter  $D$  and longitude, is displaced through all images from right to left. If there is no intersection between the structural element and the edge of the piece, other tool is selected [8].



**Figure 18.** Tool selection (final piece), (a) boundaries identification and (b) rough milling (zig) [8].

Finally, to move the tool in the image, it is necessary to use a counter ( $k = 0, i = i + 1$ ) with the total number of pixels  $m$ . This method is obtained by Eq. (44) and is displayed in **Figure 18(a)** and **(b)** [8]:

$$\sum_{j=m}^{m-1} \sum_{i=1}^{n-1} \{ \mathbf{Maq}(i, j) \leftarrow \mathbf{f}(i, j); \{ \text{if } \mathbf{f}(i, j) = \mathbf{Tray}(i, j) = 1 \text{ then } j = j + 1 \} \} \quad (44)$$

## 9. Results in mill

A new method is presented for tool selection using directional morphology. To validate the method, three examples were proposed.

**Figure 19(a)** and **(b)** depict machining with tools CB and C with labels “4” and “3,” respectively. **Figure 20** shows four selected tools, using pocket in zigzag with contour, applying dilation because the machining is external. After this, in **Figure 19(c)**, the machined area (labeled with “2”) can be seen. In the same way, the piece finally machined with a B tool is shown in **Figure 19(d)**. **Figure 19(e)** depicts the final workpiece obtained. In **Table 6**, parameters of cutting tool selected by the proposed method can be seen.

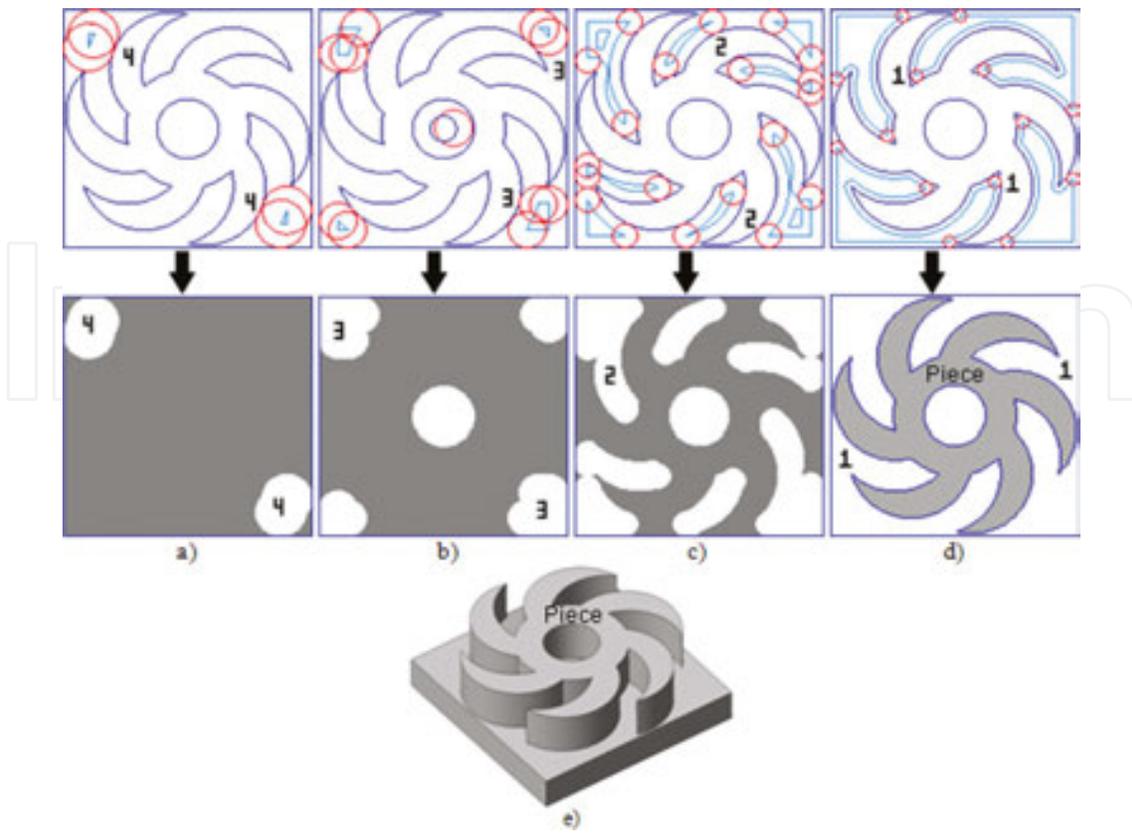


Figure 19. Tool selection for first piece based in residues of pixels [8].

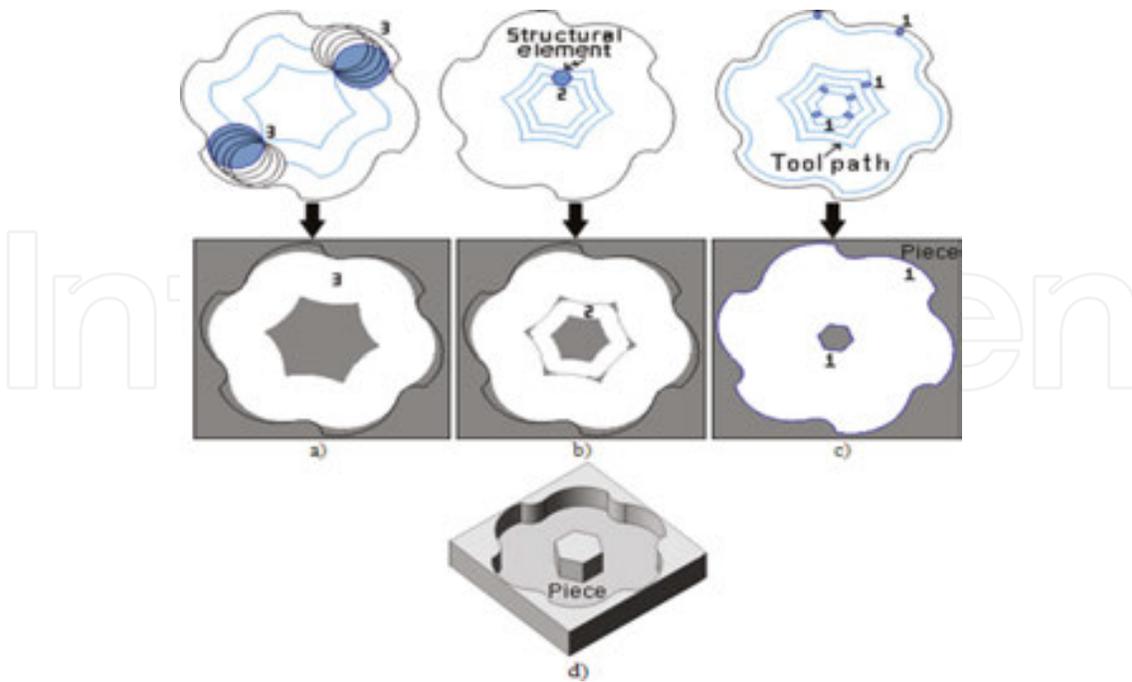


Figure 20. Tool selection for second piece using zigzag with contour [8].

Rough milling			Finish milling
Area	Type	D (mm)	D (mm)
1	B	–	0.0157
2	C	0.1500	–
3	C	0.3499	–
4	CB	0.3999	–

**Table 6.** Selected tools for the first piece.

The second piece is shown in **Figure 20**. In this case, the piece is machined using zigzag with contour and boundaries. An erosion operation, to automatically select the tool, applied to the surface to remove pixels is internal by the geometry desired. The three contours and approximations of machining with different tools marked with labels “3,” “2,” and “1” are depicted in **Figure 20(a)**, **(b)**, and **(c)**, respectively. Finally, **Figure 20(d)** depicts the machined piece. **Table 7** shows the parameters of the three different tools selected by the method (see **Table 8**).

**Figure 21** shows the rough mill with the last piece using zig with contour. **Figure 21(a)–(c)** with their respective tool selections. **Figure 21(d)** shows the finish milling and **Figure 21(e)** the final piece.

Rough milling			Finish milling
Area	Type	D (mm)	D (mm)
1	B	–	0.0157
2	C	0.2000	–
3	C	0.3999	–

**Table 7.** Selected tools for second piece.

Rough milling			Finish milling
Area	Type	D (mm)	D (mm)
1	B	–	0.0157
2	C	0.0499	–
3	C	0.1500	–
4	D	0.5999	–

**Table 8.** Selected tools for third example.

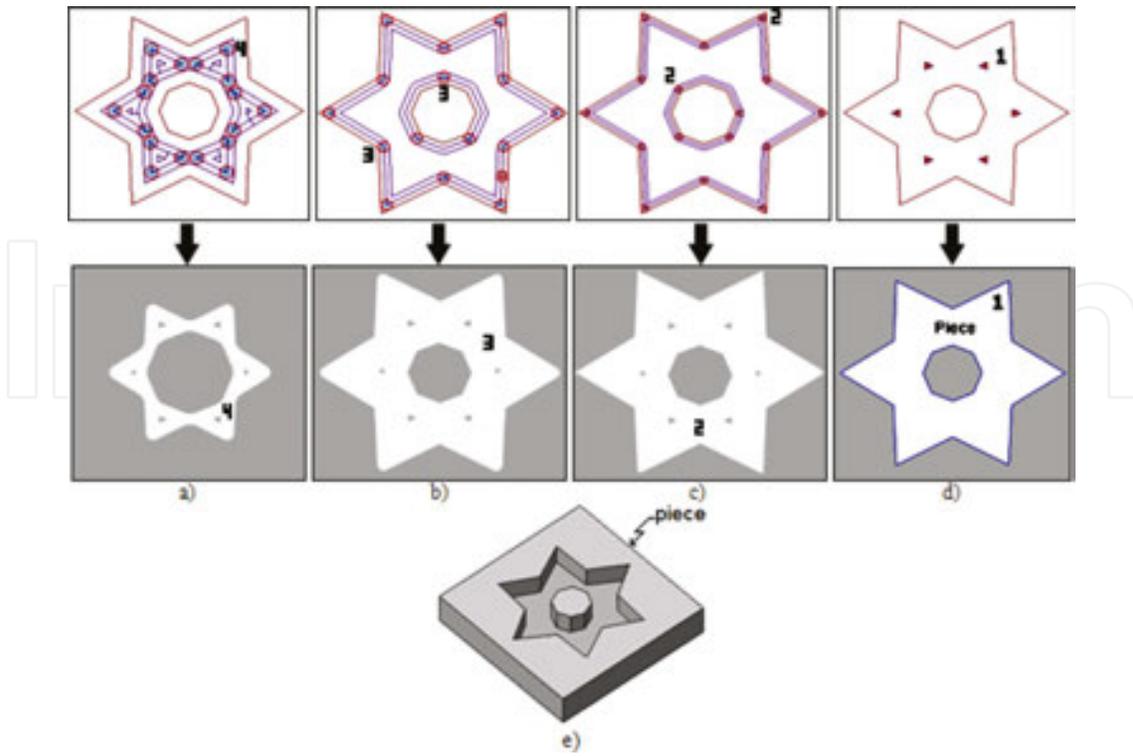


Figure 21. Tool selection for third piece (rough mill using zig with contour) [8].

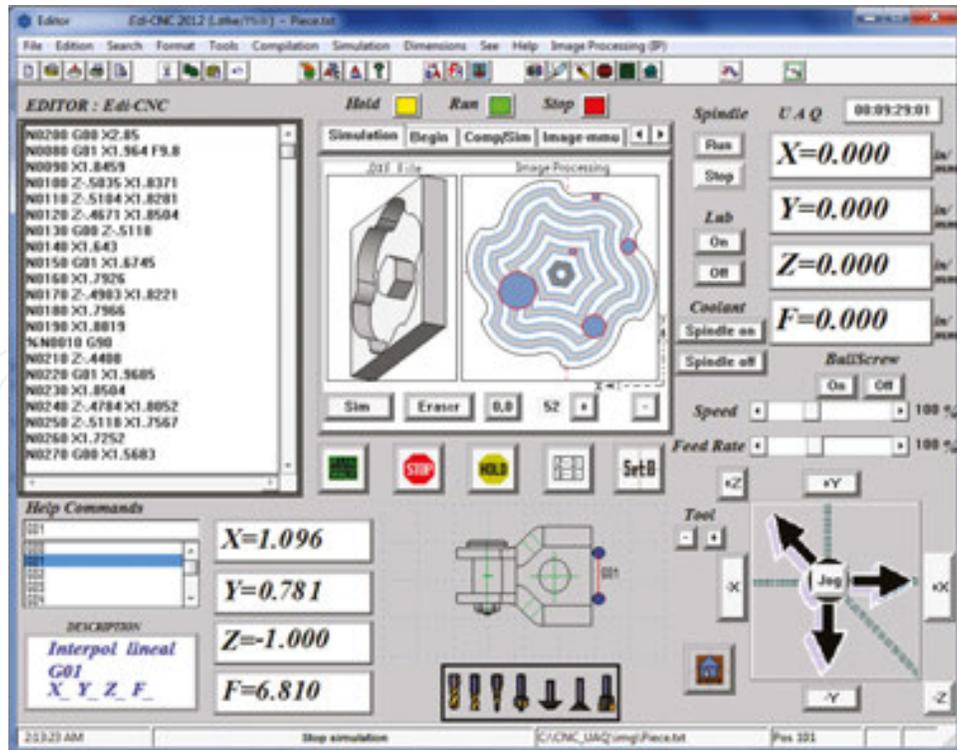


Figure 22. Final software for automatic tool selection [8].

**Figure 22** shows the software developed in Microsoft Visual C ++ 2010 for the automatic tool selection.

## 10. Conclusions of mill

Image processing for manufacturing three-dimensional models that require movements in three axes based on the directional morphology to detect collisions when the workpiece and the tool are moving is a new method in the literature. The automatic selection of cutting tool and the generation of tool paths to manufacture pieces in three-axis machining have been presented. The advantages of this method are fast and easy-to-implement programming, the proposed method can correctly select cutting tool; traditional methods of morphology as dilation and erosion in conjunction with edge piece to create tool dimensions and automatic tool selection are used. In order to diminish the possible error in boundary obtaining, a DXF file was used to compare and correct it when an image is utilized with a resolution of 1  $\mu\text{m}$ .

## Author details

Mejia Ugalde Mario\* and Mejia Ugalde Ignacio

\*Address all correspondence to: [mmu92@hotmail.com](mailto:mmu92@hotmail.com)

Engineering Faculty, Autonomous University of Queretaro, San Juan of Rio, Queretaro, Mexico

## References

- [1] Perez Paina G., Araguas G., Gaydou D., Steiner G., and Rafael Canali L., RoMAA-II, an open architecture mobile robot. *Lat Am Trans IEEE (Revista IEEE Am Lat)*. 2014; 12(5): 915–921.
- [2] Jimenez Moreno R. and Brito M., Path planning for a mobile robot in a 3D environment. In *2014 IEEE Biennial Congress of Argentina (ARGENCON)*, 2014; 125–129.
- [3] Gadelmawla E.S., Eladawi A.E., Abouelatta O.B., et al., Application of computer vision for the prediction of cutting conditions in milling operations. , Part B: *J Eng Manuf*. 2009; 223: 791–800.
- [4] Al-Kindi G. and Zughaer H., An approach to improved CNC machining using vision-based system. *Mater Manuf Process*. 2012; 27(7): 765–774.

- [5] Boix E., Ferrer A., Fernandez S., and Sellart X., Vehicle guiding system through image processing in crash and misuse tests. In Proc SAE-China Congress 2014: Selected Papers. 2015; 411–424.
- [6] Eladawi A.E., Gadelmawla E.S., Elewa I.M., et al., An application of computer vision for programming computer numerical control machines. Proc IMechE, Part B: J Eng Manuf. 2003; 217: 1315–1324.
- [7] Sarma P.M.M.S., Karunamoorthy L., and Palanikumar K., Surface roughness parameters evaluation in machining GFRP composites by PCD tool using digital image processing. J Reinf Plast Compos. 2009; 28: 1567–1585.
- [8] Mejia-Ugalde M., Trejo-Hernandez M., Dominguez-Gonzalez A., Osornio-Rios R.A., and Benitez-Rangel J.P., Directional morphological approaches from image processing applied to automatic tool selection in computer numerical control milling machine. Proc IMechE, Part B: J Eng Manuf. 2013; 227: 1607–1619
- [9] Mejia-Ugalde M., Dominguez-Gonzalez A., Trejo-Hernandez M., Morales-Hernandez L.A., and Benitez-Rangel J.P., New approach for automatic tool selection in computer numerically controlled lathe by applying image processing. Proc IMechE, Part B: J Eng Manuf. 2012; 226: 1298–1308.
- [10] Mejia-Ugalde M., Dominguez-Gonzalez A., Trejo-Hernandez M., Morales-Hernandez L., Osornio-Rios R., and Benitez-Rangel J., Triangulation intersection approach from Poissons equation applied to automatic tool selection in computer numerical control mill-lathe. Proc IMechE, Part B: J Eng Manuf. 2014; 230: 722–731.
- [11] Chen T.H., Chang W.T., Shen P.H., and Tarng Y.S., Examining the profile accuracy of grinding wheels used for microdrill fluting by an image-based contour matching method. Proc IMechE, Part B: J Eng Manuf. 2010; 224: 899.
- [12] Siller H.R., Vila C., Rodriguez C.A., and Abellan J.V., Study of face milling of hardened AISI D3 steel with a special design of carbide tools. Int J Adv Manuf Technol. 2009; 40: 12–25.
- [13] Ahmad Z., Rahmani K., and D'Souza R.M., Applications of genetic algorithms in process planning: tool sequence selection for 2.5-axis pocket machining. J Intell Manuf. 2010; 21: 461–470.
- [14] Lim T., Corney J., Ritchie J.M., and Clark D.E.R., Optimizing tool selection. Int JProd Res. 2001; 39: 1239–1256.
- [15] Ramaswami H., Shaw R.S., and Anand S., Selection of optimal set of cutting tools for machining of polygonal pockets with islands. Int J Adv Manuf Technol. 2011; 53: 963–977.
- [16] Maros T., Vladimír B., and Testik M.C., Monitoring chenille yarn defects using image processing with control charts. Tex Res J. 2011; 81: 1344–1353.

- [17] Mallik-Goswami B. and Datta A.K., Detecting defects in fabric with laser-based morphological image processing. *TexRes J.* 2000; 70: 758–762.
- [18] Li J., Huang P., Wang X., et al. Image edge detection based on beamlet transform. *J Syst Eng Electron.* 2009; 20: 1–5.
- [19] Tufoi M., Vela I., Marta C., et al. Optimization of withdrawing cylinder at vertical continuous casting of steel using CAD and CAE. *Int J Mech.* 2011; 5: 10–18.

IntechOpen

IntechOpen

