Chapter 4

Automatic localisation of broken inserts in edge profile milling heads

4.1. Dataset

To the best of our knowledge, there are no publicly available image datasets of milling cutting heads in the literature. For this reason, we created a new dataset with ground truth and we published it on-line. It is made up of 144 images of an edge profile cutting head used in a computer numerical control (CNC) milling machine. We set up a capturing system as shown in Fig. 4.1. The head tool, with cylindrical shape, contains 30 inserts in total from which 7 to 10 inserts are seen in each image of the dataset. The inserts are arranged in 6 groups of 5 inserts diagonally positioned along the axial direction of the tool perimeter, as seen in Fig. 4.2. The last insert of each group is vertically aligned with the first insert of the following group. It gives a total of 24 different positions along the radial perimeter of the tool head in which at least one insert is aligned with the camera in intervals of 15°. Therefore, the same insert is captured in several images (between 7 and 9) under different poses as the head tool rotates, Fig. 4.3. The evaluation of inserts is planned to be performed during the resting state of the milling head tool between the processing of two metallic plates. The described capturing system can be set up at that resting position.

We created the dataset following an iterative process. We mounted 30 inserts in the head tool and took 24 images of the head tool in different orientations that differ by 15°. We repeat this process for 6 times, where each time we use a different set of inserts, thus collecting (6 × 24 =) 144 images that contain (6 × 30 =) 180 unique inserts, of which 19 are broken and 161 are unbroken. All inserts that we used to create this dataset were taken after some milling operations by the same machine.

We used a monochrome camera Genie M1280 1/3” with pixel size of 3.75 μm, active resolution of 1280 × 960 pixels and fixed C-mount lent AZURE-2514MM with a focal length of 25 mm and 2/3” format. The two compact bar shape structures with high red intensity LED arrays BDBL-R(IR)82/16H were used to enhance the image capturing capability and intensified the lighting on the edges. The milling
4. Automatic localisation of broken inserts in edge profile milling heads

Machine that we used to create our dataset does not use oils, lubricants or other kind of substances that can cause a filthy tool.

Together with the dataset, we provide the corresponding ground truth masks of all ideal cutting edges along with the labels of the state of the inserts (broken or unbroken). Moreover, we labelled each distinct insert, by giving them unique identification numbers. In Fig. 4.3 we show three consecutive images that contain the same inserts (with the same identification number) in different locations and poses due to the rotation of the milling head in steps of $15^\circ$.
4.2. Automatic localisation of inserts and cutting edges using image processing

Figure 4.3: In the first row, the numbers indicate the ground truth labels of each cutting edge along three consecutive images of the dataset. Consecutive images are taken by rotating the milling head by 15°. A cutting edge present in different images is labelled with the same number. In the second row, ground truth circle masks located at the centres of the screws. The white circles approximately cover the screws of the image.

Furthermore, we create a ground truth localising the centres of all inserts. Only complete inserts have been considered for creating this ground truth, discarding partly visible inserts. The ground truth consists of two parts. First, a list of coordinates for each central point of the screws that fasten the inserts. Secondly, for each image of the dataset we provide a mask image with circles of radii 40 pixels centred at the previous coordinates, Fig. 4.3 second row. A radius of 40 pixels covers about a whole screw for all the images.

4.2. Automatic localisation of inserts and cutting edges using image processing

In this section we propose a methodology for the automatic detection of a region of interest (ROI) around the cutting edges of inserts that can be used to evaluate their wear state at a later stage.
4.2.1. Method

The localisation that we propose is done in two steps. First, we detect the screws of the inserts and use them as reference points, and then we localise the cutting edges. In order to improve the quality of the images and facilitate the detection of edges, we apply the contrast-limited adaptive histogram equalization (CLAHE) method Zuiderveld (1994). Figure 4.4 shows a schema with all the steps in the proposed methodology. Below we elaborate each one of them.

**Detection of inserts**

The screw that fastens each insert has a distinctive circular shape. We use a circular Hough transform (CHT) to detect circles with radii between 20 and 40 pixels,
4.2. Automatic localisation of inserts and cutting edges using image processing

Figure 4.5: First row: In blue, detected circles by CHT. The circles are drawn with the detected radii and positioned around coordinates that have local maximum values. In yellow, cropped areas around the centre of the detected circles that contain a whole insert. Second row: Accumulator arrays obtained with CHT on the three images in top row.

because this is the size in which a screw appears on the images of size 1280×960 pixels. For the CHT, we use a two-stage algorithm to compute the accumulator array [Atherton and Kerbyson (1999), Yuen et al. (1989)]. In the bottom row of Fig. 4.5 we show the CHT accumulator arrays for the images in the top row. By means of experiments, we set the sensitivity parameter of the CHT accumulator array to 0.85. The range of the sensitivity parameter is [0, 1], as you increase the sensitivity factor, more circular objects are detected. Figure 4.5 shows examples in which the detected circles are marked in blue. Screws that appear in the left and right peripheries of the image are usually missed due to their elliptical shape. This does not pose a problem because the same insert is seen in different positions in the previous or next images.

We crop a rectangular area of size 205×205 pixels centred on a detected screw, the chosen dimensions are just enough to contain the whole insert. We then use this cropped area to identify the cutting edge. Figure 4.5 shows examples of cropped areas marked with yellow squares.

Localisation of cutting edges

Inserts have a rhomboid shape formed by two nearly vertical (± 22°) and two nearly horizontal (± 20°) line segments (Fig. 4.6). First we use Canny’s method [Canny (1986)] to detect edges in a cropped area
4. Automatic localisation of broken inserts in edge profile milling heads

Figure 4.6: (a) Automatically detected lines that form the rhomboid shape of an insert. (b) Hough transform of the image in (a). (c) Hough transform for nearly vertical lines ($\pm 22^\circ$). The black rectangle indicates the position of the largest peak ($\theta = -8^\circ$ and $\rho = 168$). (d) Hough transform for vertical lines with slope ($-8 \pm 5^\circ$). Black rectangles superimposed to the hough transform indicate two peaks that are greater than a fraction 0.75 of the maximum: ($\rho_1 = 7, \theta_1 = -9^\circ$) and ($\rho_2 = 168, \theta_2 = -8^\circ$).

Then, we apply a standard Hough transform (SHT) (Hough, 1962) to the edge image in order to detect lines (Fig. 4.6(b-d)).

We look for the strongest vertical line segment which is represented as the highest value of peaks in the Hough transform matrix. Then, we look for line segments with peak values greater than a fraction 0.75 of the maximum peak value and with slopes in a range of $\pm 5^\circ$ with respect to the slope of the strongest nearly vertical line. In Fig. 4.6b we show the Hough transform of the cropped area shown in Fig. 4.6a. We consider the strongest nearly vertical line segment which is at least 47 pixels to the left of the center as the left edge of the insert. This detected line segment is considered as a full line and it is drawn in magenta in Fig. 4.7d. In this way, we avoid possible detection of lines around the screw area.

Similarly, we look for two horizontal line segments above and below the screw. In this case, the minimum distance from the line to the centre is set to 52 pixels and the range of possible slopes is $\pm 11^\circ$ with respect to the slope of the strongest horizontal line. The top and bottom detected lines are shown in yellow and cyan respectively in Fig. 4.7d. The points where the horizontal lines intersect with the left vertical line define the two ends of the cutting edge segment. These points are marked as dark blue dots in Fig. 4.7d. The localised cutting edges in these examples
4.2. Automatic localisation of inserts and cutting edges using image processing

Figure 4.7: (a) Cropped areas containing inserts. (b) Canny edge maps. (c) Detection of (nearly) vertical and (nearly) horizontal lines. (d) Blue spots indicate the intersections between the two horizontal lines and the left vertical line. Lines obtained by symmetry are the following. Second row: top horizontal line; third row: left vertical line; forth row: left vertical line and bottom horizontal line; fifth row: left vertical line and top horizontal line. (e) Detected cutting edges.

The fact that we have a controlled environment (fixed camera and fixed resting position of the head tool), this set of parameters is fixed only once for every given milling machine.

If the left line segment or any of the horizontal segments are not detected, we use symmetry to determine the missing lines. For instance, if the vertical line on the left of the screw is not detected but the one on the right is detected, we reconstruct the left line by rotating by 180 degrees the left line around the center of the concerned area. The bottom three examples in Fig. 4.7 show this situation.
Finally, we define a ROI by dilating the detected cutting edge segment with a square structuring element of 10 pixels radius. In Fig. 4.8, we show the cutting edge segments and ROIs localised by the proposed method for images containing inserts with different wear state. Notably is the fact that the proposed method can generalise the localisation of the cutting edge even in cases of worn or broken inserts.

4.2.2. Experiments and results

For each of the input images, we determine a set of ROIs around the identified cutting edges using the method described in Section 4.2.1. If the ground truth of a cutting edge lies completely in a ROI, we count that ROI as a hit and when it does not lie within any of the determined ROIs, the hit score is 0. If the ground truth overlaps a ROI, the hit score is equal to the fraction of the ground truth segment that lies inside a ROI. Some examples can be observed in Fig. 4.9.

Every insert is detected in at least one of the 144 images. Moreover, whenever an insert is detected, the corresponding cutting edge on the left side is also always determined. We measure the accuracy of the method as an average of the partial scores for the individual cutting edges. Using this protocol, we obtain an accuracy measure of 99.61%.

Results can be improved by increasing the width of the structuring element in the final dilation stage. With a square structuring element of radius 34 pixels we
4.2. Automatic localisation of inserts and cutting edges using image processing

Figure 4.9: The green quadrilaterals are the ROIs detected by the proposed method and the red lines represent the ground truth for the cutting edges. The accuracy scores of the inserts are indicated in white font. (b) Example of a cutting edge that is not completely contained within the detected ROI. The accuracy of 0.9481 is the fraction of pixels of the cutting edge that lie within the detected ROI.

Figure 4.10: Red line segments define the ground truth and green quadrilaterals define the detected ROIs with a morphological dilation operation of 34 pixel radius square structuring element.

achieve 100% accuracy. Figure 4.10 shows examples of the resulting ROIs.

4.2.3. Discussion

To the best of our knowledge the proposed approach is the first one automatically localises multiple inserts and cutting edges in an edge profile milling head. Parameters have been computed in order that they can generalise for every insert
at any position in the milling head tool given the geometry of the head tool and the
arrangement of the capturing system. For a specific milling machine, parameters
can be easily estimated and then no further need of adjustment is needed.

We achieve an accuracy of 99.61% for the detection of cutting edges. This is
achieved by dilating the automatically detected line using a square structuring ele-
ments of 20 pixel side. When the quadrilateral is 68 pixels wide, the accuracy reaches
100%. In future works, the ROIs defined around the detected cutting edges can be
used for further evaluation of the wear state of the cutting edges.

Furthermore, the proposed method can be used for different milling heads con-
taining polygonal inserts fastened by screws, a design which is typical in edge
milling machines. We implemented the proposed approach in Matlab and ran all
experiments on a personal computer with a 2 GHz processor and 8 GB RAM. It
takes less than 1.5 seconds to process all the steps on one image it takes about 1
minute to capture and process the 24 images taken to the head tool. This milling
head tools are resting between 5 to 30 minutes, so the implementation reaches real
time performing.

4.3. Classification of inserts as broken or unbroken

In this section we present a method for the classification of inserts as broken or
unbroken by analysing the cutting edges that have been already localised in Section 4.2.1.

4.3.1. Method

In the method that we propose we first localise cutting edges in a given im-
age, and then we classify every cutting edge as broken or unbroken. From the image
analysis point of view, an unbroken insert is one which has a straight cutting edge
(Fig. 4.11a), while a broken insert has a curved or uneven cutting edge (Fig. 4.11b).

Figure 4.12 presents a schema that shows the proposed methodology. First we
localise the inserts and the cutting edges and then, we evaluate the inserts using a
three-stage method: applying an edge preserving smoothing filter, computing the
gradient for each edge and finally using geometrical properties of the edge to assess
its state. Below we elaborate on each of these steps.

Detection of inserts and localisation of ideal cutting edges

We use the algorithm introduced in Section 4.2 to detect inserts and localise
the respective ideal cutting edges. For each localised insert, we consider a set \( I \)
4.3. Classification of inserts as broken or unbroken

Figure 4.11: (a) In green the real cutting edge of an intact insert. The white cross marks the centre of the detected screw. (b) In green the real cutting edge of a broken insert. In red the ideal cutting edge. All markers are manually drawn.

![Diagram showing the methodology of insert detection and classification.](image)

Figure 4.12: Outline of the proposed methodology.

Of Cartesian coordinates that form the ideal cutting edge,

\[ I = \{(x_t, y_t) \mid t = 1...u\} \]  \hspace{1cm} (4.1)
where \( u \) is the number of locations of the ideal cutting edge of a localised insert.

We determine a region of interest (ROI) from the ideal cutting edge and the horizontal edges that are detected by the algorithm in Section 4.2. In Fig. 4.13a we show examples of ROIs in broken and unbroken inserts. A ROI is determined by considering two parallel lines to the ideal cutting edge, one 3 pixels to the left and the other one to the right with a distance of 0.7 times the space between the ideal cutting edge and the centre of the screw. Moreover, we consider a parallel line to the top edge 3 pixels towards the bottom and a parallel line to the bottom edge 3 pixels towards the top. From the resulting quadrilateral, we remove a segment from a circle (with a radius of 45 pixels) around the centre of the screw that coincides with the quadrilateral. Such a ROI is sufficient to evaluate the state of a cutting edge while ignoring possibly worn edges coming from the top or bottom parts of the insert as well as ignoring any texture coming from the screw. In the end, we consider a rectangle around the ROI with a 3-pixel width boundary and use it to crop the corresponding part of the image that contains the ROI, Fig. 4.13b. We also consider a mask defining the ROI in such a rectangular area, Fig. 4.13c.

Detection of real cutting edges

The heterogeneous texture and the low contrast of the insert with respect to the head tool make the detection of the real cutting edge an arduous task. If an edge detector is applied directly to the cropped images, many edges apart from the cutting edge would be recognised. In order to enhance the edge contrast, we apply the edge-preserving smoothing filter of Gastal and Oliveira (2011) to the cropped region. We choose this approach due to its efficiency and its good performance. This filtering method smooths the heterogeneous texture of the insert and of the background but preserves the edges of the insert. The images in Fig. 4.13d show examples of the output of this algorithm.

Afterwards, we apply Canny’s method Canny (1986) to find edges by looking for local maxima of the gradient on the filtered region. Other edge detectors, such as the ones based on Sobel, Prewitt, Roberts and LoG, performed worse. Canny’s algorithm computes the gradient after applying a Gaussian filter that reduces noise. Non-maximal suppression is applied to thin the edge. This is followed by hysteresis thresholding which uses a low and a high threshold in order to keep the strong edges (above the high threshold) and only the weak edges (with a value between the low and high threshold) that are connected to any of the strong ones. We show examples of Canny’s gradient magnitude and binary edge maps in Fig. 4.13e-f). Finally, we only consider the edges within the ROI (Fig. 4.13g). For each localised insert, we define a set \( R \) of 3-tuples that represent the Cartesian coordinates \((x_q, y_q)\)
4.3. Classification of inserts as broken or unbroken

Figure 4.13: (a) In yellow, ideal, top and bottom edges. In red, definition of the ROI. In green, rectangle to crop. (b) Cropped region. (c) Mask defining the ROI in a cropped region. (d) Edge-preserving smoothed region. (e) Gradient magnitude map. (f) Edge map. (g) Result of multiplying the edge map by the mask. (h) $R$ set in white and $I$ set overlaid in red. The two top inserts are unbroken while the two bottom ones are broken.

and the corresponding gradient magnitude value $g_q$ of each location in the real cutting edge:

$$R = \{(x_q, y_q, g_q) \mid q = 1...v\}$$

(4.2)

where $v$ is the number of locations of the real cutting edge of the localised insert.

Measurement of deviations between real and ideal cutting edges

For a pair of coordinates $(x_t, y_t)$ in the ideal set of edges $I$, we determine a set $P_t$ of coordinates $(\hat{x}, \hat{y})$ and the corresponding gradient magnitudes $\hat{g}$ from the set of real edges $R$ such that $(\hat{x}, \hat{y})$ lie on a line that passes through $(x_t, y_t)$. The slope $m$ of this line is the gradient of the top edge.
4. Automatic localisation of broken inserts in edge profile milling heads

\[ P_t = \{ (\hat{x}, \hat{y}, \hat{g}) \mid \hat{y} = m(\hat{x} - x_t) + y_t, (\hat{x}, \hat{y}, \hat{g}) \in R, (x_t, y_t) \in I \} \quad (4.3) \]

Examples of such lines are marked in blue in Fig. 4.14a. Next, we denote by \( E_t \) the set of Euclidean distances from \((x_t, y_t)\) to all coordinates in the set \( P_t \):

\[ E_t = \left\{ \sqrt{(x_t - \hat{x}_p)^2 + (y_t - \hat{y}_p)^2} \mid (x_t, y_t) \in I, \forall (\hat{x}_p, \hat{y}_p) \in P_t \right\} \quad (4.4) \]

\( E_t \) could be an empty set. Let \( D \) be the set of minimum distances of \( E_t \) for each point \( t \) in \( I \). \( D \) represents the minimum deviations between the ideal and real edges.

\[ D = \{ \min(E_t) \mid t = 1...|I| \} \quad (4.5) \]

Let \( G \) be the set of gradient magnitudes of the points in \( P_t \) with the minimum distance in \( E_t \) for each point in \( I \).

\[ G = \{ g_i \mid g_i \in P_t, i = \arg\min(E_t), t = 1...|I| \} \quad (4.6) \]

In Fig. 4.14 we plot the values of the sets \( D \) and \( G \).

We remove abnormal deviations that are usually caused by texture on the surface of the insert rather than by the cutting edge. For example, Fig. 4.14(e) presents two such abnormal deviations (spikes) at the beginning and end of the set \( D \). We denote by \( f_{A,N}(D, t) \) a function that evaluates a neighbourhood of width \( A \) within the given set \( D \) centred at point \( t \):

\[ f_{A,N}(D, t) = \begin{cases} D_t & \text{if } D_t \leq N \times \text{median}_{j=-A}^{A}(D_{t+j}) \\ \emptyset & \text{otherwise} \end{cases} \quad (4.7) \]

This function returns the element \( D_t \) if \( D_t \) is higher than a fraction \( N \) of the median within a local window of width \( A \), otherwise it returns \( \emptyset \). We define a set \( D' \) which is formed by applying the function \( f \) two consecutive times in order to remove spikes with a length of at most 3 points.

\[ D' = \{ f_{A_2,N_2}(f_{A_1,N_1}(D, t), t) \mid \forall t \in D \} \quad (4.8) \]

The first insert in Fig. 4.14 shows a typical problem in this application. The lower part of its cutting edge has low contrast and as a result the corresponding edge points have very low gradient magnitudes. We are only interested in evaluating the parts along the cutting edge that have high contrast because they are more reliable. Formally, we define a new set \( D'' \) whose elements are copied from the set \( D' \) when the corresponding edge points have gradient magnitudes higher than a threshold \( B \), otherwise they are set to \( \emptyset \).
4.3. Classification of inserts as broken or unbroken

Figure 4.14: Two examples of deviation and gradient magnitude computation. (a-e) and (f-j) correspond to the second and third rows in Fig. 4.13 respectively. (a and f) Edge maps. (b and g) Gradient magnitude maps. (c and h) In white the edge maps, overlapped in red the detected ideal cutting edge and in blue examples of analysed lines $\hat{y}$ as in Eq. 4.3. (d and i) In blue deviations along the detected ideal cutting edge $D$, in green deviations after spike and low contrast elimination $D''$ and in magenta deviations after mean filtering $D'''$. (e-h) In dark green gradient magnitudes along the detected ideal cutting edge and in red an example of threshold $B = 0.2$. 
4. Automatic localisation of broken inserts in edge profile milling heads

\[ D'' = \{ g_t \geq B \rightarrow d_t \wedge g_t < B \rightarrow \emptyset \mid d_t \in D', \forall g_t \in G \} \]  \hspace{1cm} (4.9)

In order to ensure that an insert is broken, the deviation should be sufficiently high along a region of the cutting edge and not just in one isolated pixel. We apply a mean filter of window with a width \( C \) and subsequently take the maximum deviation \( \bar{d} \) of the cutting edge.

\[ \bar{d} = \max \left\{ \frac{1}{2C + 1} \sum_{j=-C}^{C} (d_{t+j}) \mid \forall d_t \in D'' \right\} \]  \hspace{1cm} (4.10)

Moreover, we also compute the mean gradient magnitude \( \bar{g} \) along the cutting edge.

\[ \bar{g} = \frac{1}{|G|} \sum_{t=1}^{G} g_t, \forall g_t \in G \]  \hspace{1cm} (4.11)

As a result every localised insert is represent by the two parameter values \( \bar{d} \) and \( \bar{g} \).

**Classification of inserts**

We remind the reader that the same insert is detected in several images under different poses. In this work, the correspondences of the same insert in multiple images is manually labelled. In the Section 4.3.3 we provide a suggestion how the correspondence issue can be implemented automatically. For each insert we compute the maximum deviation \( \bar{d} \) and the mean gradient \( \bar{g} \) for every image where it is detected.

We classify an insert as broken if the image with the highest mean gradient magnitude \( \bar{g} \) along the cutting edge has a maximum deviation \( \bar{d} \) higher than a threshold \( T \), or if the maximum deviations of at least two images (irrespective of the mean gradient magnitude) are greater than \( T \). Otherwise we classify the insert as unbroken. Formally, we define the classification function \( z \) as:

\[ z(e) = \begin{cases} 1 & \text{if } (\bar{d}_{\text{argmax}_{n=1...r} (\bar{g}_n)} > T) \vee (\sum_{h=1}^{r} (\bar{d}_h > T)) \geq 2 \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (4.12)

where \( r \) is the number of images where the same insert \( e \) is detected.

**4.3.2. Experiments and results**

We used Matlab in a personal computer with a 2GHz processor and 8GB RAM. The complete process to identify broken inserts in a head tool with 30 inserts takes
less than 3 minutes. This is sufficient for the application at hand because according to the consulted experts the milling tool head stays in a resting position between 5 and 30 minutes, during which the milled plate is replaced by a new one.

Our dataset is skewed with 19 broken inserts and 161 unbroken ones. We refer to the broken inserts as the positive class and the unbroken as the negative class. Therefore, a true positive (TP) is a broken insert classified as broken; a false positive (FP) is an unbroken insert classified as broken and a false negative (FN) is a broken insert classified as unbroken. We compute the precision $P = \frac{TP}{TP + FP}$, recall $R = \frac{TP}{TP + FN}$ and their harmonic mean $F = \frac{2PR}{P + R}$ for a set of thresholds $T \in \{5, 5.01, \ldots, 8\}$ used in the classification function, and obtain a $P−R$ curve. We consider the best pair $(P, R)$, the one that contributes to the maximum harmonic mean.

We apply a repeated random sub-sampling validation where in each run we randomly (stratified sampling) split the dataset into training (70%) and validation (30%) sub sets. For each such split, we use the training data to determine the set of parameters $(A_1, N_1, A_2, N_2, B, C)$ that achieves the global maximum harmonic mean $F$. This is obtained by applying a grid search on $A_1 \in \{3, 5, 7\}$, $N_1 \in \{1, 1.5, 2\}$, $A_2 \in \{3, 5, 7\}$, $N_2 \in \{1, 1.25, 1.5\}$, $B \in \{0.18, 0.2, 0.22\}$ and $C \in \{3, 5, 7\}$ and computing the maximum harmonic mean for each combination. If several combinations of parameters yield the same harmonic mean, we take a random one. The determined set of parameters is then used to evaluate the validation set. We repeat this process 20 times and finally we average the results obtained from the validation sets. We obtain an average harmonic mean $F = 0.9143(\pm 0.079)$ with a precision $P = 0.9661(\pm 0.073)$ and a recall $R = 0.8821(\pm 0.134)$. The most repeated (6 out of 20 runs) set of parameters in the training is $(A_1 = 5, N_1 = 1.5, A_2 = 3, N_2 = 1, B = 5, C = 0.2)$. When we evaluate the entire dataset with these parameter values we achieve precision $P = 1$ and recall $R = 0.95$ for the maximum harmonic mean $F = 0.9744$.

4.3.3. Discussion

We performed an effective classification of the inserts according to the state of their cutting edges as broken and unbroken. The high performance results that we achieved demonstrate the effectiveness of the proposed approach and suggest that this system can be applied in production. The performance can be further improved by using more appropriate illumination conditions and better quality of the lenses in order to obtain higher contrast between inserts and background.

Typically, an insert appears in 7 to 10 images in different positions and poses. In this work, the ground truth contains the identification numbers of the inserts in all images. This means, that an insert that appears multiple times is manually given the same identification number. Alternatively, the approximate position of inserts in
the consecutive images can be inferred from the radius of the head tool cylinder, the distance of the fixed camera from head tool and the degrees of rotation. In this way, after automatically detecting the positions of inserts we can automatically determine the correspondences (labelling) according to the expected positions.

In this work we are concerned with detecting broken inserts as it is the most critical evaluation for the stability of the milling head tool. In future, we will also evaluate the wear of inserts in order to detect the weak ones as early as possible. Moreover, we would also like to compare the performance of different image acquisition methods.

In addition, the proposed methodology can be set up for different machining heads that contain polygonal inserts fastened by screws, a typical design in milling machines.

4.4. Automatic localisation of inserts using COSFIRE

In this section we propose a method for the localisation of inserts based on COSFIRE filters. This approach considers independently each image of the dataset and can be automatically configured regardless of the appearance of the inserts. This trainable approach is more versatile and generic than previous works on the topic, as it is not based on, and for that reason does not require, any a priori domain knowledge.

4.4.1. Method

Overview

In order to detect a particular object in an image, COSFIRE filters are first configured by using some training patterns, also referred to as prototypes. We obtain a prototype pattern by extracting a delimited area —region of interest (ROI)— containing one of the inserts in a representative image, Fig. 4.15.

COSFIRE filters, Azzopardi and Petkov (2013c), combine the responses of 2D Gabor filters at specific locations around a given point. Gabor filters, Petkov and Wieling (2008), are configured by establishing their characteristic directions and the locations at which their responses are taken. Consequently, the resulting COSFIRE filter only responds to inserts similar in local spatial arrangement to that in the ROI. In this case, the most characteristic edges are found on the sides of the insert, around the screw and on the top right crack, Fig. 4.15.
4.4. Automatic localisation of inserts using COSFIRE

Figure 4.15: Selection of the region of interest: (a) Input image. (b) Prototypical insert. (c) Selection of the ROI. (d) Mask that mark out the area of the ROI.

Gabor filters

The real Gabor function $h_{\lambda,\theta}(x,y)$ for a given wavelength $\lambda$ and orientation $\theta$ is defined as:

$$h_{\lambda,\theta}(x,y) = e^{\left(-\frac{u^2 + v^2}{2\sigma^2}\right)} \cos\left(2\pi \frac{u}{\lambda} + \zeta\right)$$  \hspace{1cm} (4.13)

$$u = x \cos (\theta) + y \sin (\theta)$$  \hspace{1cm} (4.14)

$$v = -x \sin (\theta) + y \cos (\theta)$$  \hspace{1cm} (4.15)

where $\gamma = 0.3$ is the aspect ratio that specifies the ellipticity of the support of the Gabor function; $\sigma$ determines the size of the support; and $\zeta = \pi/2$ is the phase offset that determines the symmetric or antisymmetric shape of the Gabor function.

We denote by $g_{\lambda,\theta}(x,y)$ the response of a Gabor filter to a grayscale input image $I$:

\footnote{For more details about Gabor filters and the use of their parameters such as the aspect ratio or the standard deviation of the Gaussian envelope, we refer the reader to Grigorescu et al. (2003a, 2002); Kruizinga and Petkov (1999); Petkov (1995); Petkov and Kruizinga (1997); Petkov and Westenberg (2001).}
Gabor functions are normalized in such a way so all positive values sum up to 1 whereas all negative values sum up to -1. In this way, the response to an image of constant intensity is 0 even for symmetrical filters (ζ = {0, π}) and the largest response to a line of width \( w \) is achieved using a symmetrical filter (ζ = {0, π}) with \( λ = 2w \).

**Configuration of COSFIRE filters**

A COSFIRE filter is configured by determining the geometrical properties of the lines and edges in the neighbourhood of a specified point of interest, which in this case is the centre of a screw. The neighbourhood is defined by a set of circles of given radii. We first superimpose the responses of a bank of Gabor filters with one scale (\( λ = 6 \)) and 16 orientations (\( θ = \{0, π/8, \ldots\} \)). For each local maximum Gabor response along these circles we consider the Gabor filters that give a response greater than a fraction \( t_2 \) of the maximum Gabor response at that position. Then, we create a 4-tuple \( (λ, θ, ρ, φ) \) for every Gabor filter that satisfies the mentioned criteria: the wavelength \( λ \) and orientation \( θ \) of the Gabor filter define the characteristics of the concerned Gabor filter, while the distance \( ρ \) and polar angle \( φ \) define the position with respect to the center.

We denote by \( S_f \) a COSFIRE filter with a set of 4-tuples \( (λ_i, θ_i, ρ_i, φ_i) \) that characterize the properties of contour parts:

\[
S_f = \{(λ_i, θ_i, ρ_i, φ_i) \mid i = 1, \ldots, n_f\}
\]  

(4.17)

The subscript \( f \) stands for the feature (in this case an insert) around the point of interest (ROI) and \( n_f \) stands for the number of involved contour parts.

For the ROI shown in Fig. 4.15, taking 25 equally spaced radii from 0 to 150 (the half diagonal of the ROI), this method results in a COSFIRE filter with 127 tuples. Fig. 4.16 illustrates the consideration of Gabor responses for the circle with radius \( ρ = 107 \). Along this circle the automatic configuration determines 3 tuples; one for each point \( a, b \) and \( c \) with parameter values specified in the set shown in Table 4.1. The third tuple \( (λ_3 = 6, θ_3 = π, ρ_3 = 107, φ_3 = 4.08) \) describes a contour part with a wavelength of \( (λ_3 = 6) \) and an orientation of \( θ_3 = π \), therefore it is a vertical contour part, that can be detected by a Gabor filter with preferred wavelength \( λ_3 = 6 \) and orientation \( θ_3 = π \), at a position of \( ρ_3 = 107 \) pixels to the bottom-left \( (φ_3 = 1.29π) \) from the support center of the filter. This location is marked by the label ‘c’ in Fig. 4.16a. In Fig. 4.16c we illustrate the structure of the resulting COSFIRE filter with 127 tuples.
4.4. Automatic localisation of inserts using COSFIRE

Figure 4.16: Configuration of a COSFIRE filter: (a) Superposition of the response maps of a bank of Gabor filters. The white cross indicates the point of interest and the white circle represents the locations of the Gabor responses considered around the point of interest for a given radius $\rho$, here $\rho = 107$. The gray-level of a pixel represents the maximum value superposition of the responses of a bank of symmetric Gabor filters ($\lambda = 6, \theta = \{ \frac{\pi}{8}, i = 0...7 \}$ and $\zeta = \pi/2$) at that position. (b) The maximum Gabor responses along the depicted circle in (a). The three local maxima in the plot are respectively labelled and marked with magenta dots in (a). (c) Structure of the COSFIRE filter. Each of the ellipses represent a tuple of the set of contour parts. Their size and orientation represent the scale $\lambda$ and orientation $\theta$ parameters of the Gabor filters. This filter is configured to detect the spatial local arrangement of 127 contour parts. The green enumerated ellipses represent the three contour parts found for $\rho = 107$ described in Table 4.1: ellipse 1 corresponds to the local maximum $a$, ellipses 2 to $b$ and ellipse 3 to $c$. The bright blobs are intensity maps of the Gaussian functions that are used in the application step for blurring the responses of the Gabor filters. The blurring step is explained in more detail in Section 4.4.1.4.

Table 4.1: Set of tuples that describe the contour parts of the prototype shown in Fig. 4.15 for a circle with radius $\rho = 107$.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\lambda_i$</th>
<th>$\theta_i$</th>
<th>$\rho_i$</th>
<th>$\phi_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>$15\pi/8$</td>
<td>107</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>$5\pi/8$</td>
<td>107</td>
<td>1.22</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>$\pi$</td>
<td>107</td>
<td>4.08</td>
</tr>
</tbody>
</table>

Application of COSFIRE filters to milling head images

A COSFIRE filter is applied by computing the Gabor filters defined in the set of tuples. Then, for each position in an image, we combine the Gabor responses whose locations are specified by the polar coordinates in the set of tuples, and combine them with a multivariate output function.

**Blurring and shifting.** Before computing the output function of a COSFIRE filter, we first blur the Gabor responses in order to allow for some spatial tolerance of the involved contour parts. The blurring consists of a convolution of the Gabor responses with a rotationally symmetric Gaussian lowpass filter $G_{\sigma}(x, y)$ with standard deviation $\sigma$. The standard deviation is a linear function of the distance $\rho$ from
the centre of the COSFIRE filter:

$$\sigma = \sigma_0 + \alpha \rho$$  \hspace{1cm} (4.18)

We use $\sigma_0 = 0.67$ and $\alpha = 0.04$. The visual system of the brain inspired the choice of the linear function in Eq. 4.18 as explained in (Azzopardi and Petkov, 2013c). The blurred response for the tuple $(\lambda_i, \theta_i, \rho_i, \phi_i)$ is defined as:

$$b_{\lambda_i, \theta_i, \rho_i}(x, y) = g_{\lambda_i, \theta_i}(x, y) * G_{\sigma_i}(x, y)$$  \hspace{1cm} (4.19)

Instead of retrieving the Gabor responses using the polar coordinates specified in tuples of the filter with respect to each pixel in the image, we shift the blurred responses of each Gabor filter by a distance of $\rho_i$ in the opposite direction to $\phi_i$. In polar coordinates, we can express this as $(\rho_i, \phi_i + \pi)$, whereas in Cartesian coordinates it is described as an increment $(\Delta x_i, \Delta y_i)$ where $\Delta x_i = -\rho_i \cos \phi_i$ and $\Delta y_i = -\rho_i \sin \phi_i$. We denote by $s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y)$ the blurred and shifted response of the Gabor filter specified by the tuple $(\lambda_i, \theta_i, \rho_i, \phi_i)$ in the set $S_f$:

$$s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) = b_{\lambda_i, \theta_i, \rho_i}(x - \Delta x_i, y - \Delta y_i)$$  \hspace{1cm} (4.20)

where $-3\sigma \leq x, y \leq 3\sigma$.

**Response of a COSFIRE filter.** In the work published in Azzopardi and Petkov (2013c) the response of a COSFIRE filter is defined as the geometric mean of all blurred and shifted responses of the involved Gabor filters as defined in Eq. 4.21. This is a hard AND-type function as the absence of only one of the preferred contour parts suppresses completely the response of the COSFIRE filter, onwards named Hard Geometric Mean (HGM). Here, we experiment with two other softer output functions, namely Arithmetic Mean (AM) and Soft Geometric Mean (SGM), defined in Eq. 4.22 and Eq. 4.23 respectively.

$$r_{S_f}(x, y) \overset{\text{def}}{=} \left| \left( \prod_{i=1}^{\left|S_f\right|} s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \right)^{1/\left|S_f\right|} \right|_{t_3}$$  \hspace{1cm} (4.21)

$$r_{S_f}(x, y) \overset{\text{def}}{=} \left| \left( \sum_{i=1}^{\left|S_f\right|} s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) \right) \right|_{t_3}$$  \hspace{1cm} (4.22)

$$r_{S_f}(x, y) \overset{\text{def}}{=} \left| \left( \prod_{i=1}^{\left|S_f\right|} (s_{\lambda_i, \theta_i, \rho_i, \phi_i}(x, y) + \epsilon) \right)^{1/\left|S_f\right|} \right|_{t_3}$$  \hspace{1cm} (4.23)

where $|.|_{t_3}$ means that the response is thresholded at a fraction $t_3$ of the maximum.
4.4. Automatic localisation of inserts using COSFIRE

across all coordinates \((x, y)\). The parameter \(\epsilon\) in Eq. 4.23 is a very small value in order to avoid complete suppression by non-present contour parts. In this work, we set \(\epsilon = 10^{-6}\). In this way a COSFIRE filter that uses an SGM output function always gives a response greater than zero. As for the AM metric, the lack of presence of a contour part has a lower effect in the response of the COSFIRE filter than SGM or HGM.

From the COSFIRE response map \(r_{Sf}(x, y)\), we first choose the local maxima points by considering neighbourhoods of 8 pixels. Then, if two local maxima points are within a Euclidean distance of 200 pixels, we only keep the point with the strongest response. Due to the shape of the milling cutting head and the conditions of the image capture, inserts are always separated by at least 200 pixels. We call these points, positive response points.

Figure 4.17 shows the whole process of edges detection. In this example, the COSFIRE filter is applied with the SGM function and has 127 tuples. Each blurred and shifted response corresponds to each of the 127 contour parts found in the configuration. The filter responds in locations where there is an identical or similar pattern to the prototypical insert. In this example, the maximum response is reached in the center of the prototype insert that was used to configure this COSFIRE filter and the other four local maxima points correspond to inserts that are similar to the prototypical insert.

4.4.2. Experiments

The dataset is split in two subsets, training and test. The training set is formed by the images of the dataset separated by 13 snapshots with numbers 0001, 0014, 0028, 0042, 0056, 0070, 0084, 0098, 0112 and 0126. The other 134 images form the test set.

We configure filters in an iterative process by using inserts from the training images. We configure a filter \(S_{f1}\) for prototype \(f_1\), shown in Fig. 4.18a. Then, we apply this filter to all the images in the training set. We set the value of \(t_3\) to produce the highest number of correctly detected inserts and no false positives, therefore achieving 100% precision. Figure 4.19 shows the inserts found with functions AM, HGM and SGM using the filter for prototype \(f_1\). Threshold \(t_3\) is set to 0.283, 0.044 and 0.119 for AM, HGM and SGM detecting 9, 35 and 37 correct inserts respectively. In total, there are 86 inserts in the 10 training images. Thus, this single COSFIRE filter detects 43.02% of the inserts using SGM, and no false positives.

In the second iteration, we randomly choose one of the inserts that was not detected by the first filter \(S_{f1}\) and we call it prototype \(f_2\). We use this prototype to configure a second COSFIRE filter \(S_{f2}\). Then, we apply this filter to the 10 images of the training set and determine the \(t_3\) parameter values that achieve 100% precision. Filter \(S_{f2}\) detects an amount of inserts, some already detected by filter \(S_{f1}\) and some
4. Automatic localisation of broken inserts in edge profile milling heads

Figure 4.17: (a) Input image. We show just part of the input image for better visualization. The framed area shows (top) the enlarged pattern of interest selected for the configuration and (bottom) the structure of the COSFIRE filter that was configured for this pattern. The contour parts found at $\rho = 107$ whose application is shown in this figure are numbered and marked in green color. (b) Each contour part of the prototype pattern is detected by the response of an antisymmetric Gabor filter with preferred values of wavelength $\lambda_i$ and orientation $\theta_i$. In this case, we need a Gabor filter to detect each of the contour parts. In general, contour parts with the same pair of values $(\lambda_i, \theta_i)$ are detected by the same Gabor filter. (c) The response $g_{\lambda_i, \theta_i}(x, y)$ is then blurred and later shifted by $(\rho_i \phi_i + \pi)$ in polar coordinates. (d) Finally, the output of the COSFIRE filter is computed by the thresholded soft geometric mean of all the contour part responses, for this example $t_3 = 0.15$. The five local maxima in the output of the COSFIRE filter correspond to the configured insert and four other similar inserts in the input image. The red ‘$\times$’ marker indicates the location of the specified point of interest.

new detections. For example, $S_{f_2}$ with SGM correctly detects 14 inserts, of which 4 coincide with the inserts detected by $S_{f_1}$, and 10 are newly detected ones. At this point, we have detected a total of 47 inserts out of 86.

The process successively continues until all the 86 inserts in the training set are detected. We configure a total of 19 filters for HGM, from the prototypes shown in Fig. 4.18, for yielding 100% precision at 100% recall, only the first 17 filters are necessary when using SGM. The number of filters needed for each output function
4.4. Automatic localisation of inserts using COSFIRE

Figure 4.18: A set of 19 prototypical inserts. The whole set was needed to detect all inserts of the training set with 100\% precision and 100\% recall with HGM function. The first 17 filters were needed when using SGM.

are reported in Table 4.2.

The set of configured COSFIRE filters is applied to the test set where results are computed in terms of precision, recall and their harmonic mean, also known as F-Score:

\[
F_{\text{Score}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

(4.24)

Recall is the percentage of true inserts that are successfully detected, \( \text{Recall} = \frac{TP}{TP + FN} \). Precision is the percentage of correctly detected inserts from all positive response points, \( \text{Precision} = \frac{TP}{TP + FP} \). \( TP, FP \) and \( FN \) stand for true positives, false positives and false negatives, respectively.

4.4.3. Results

We evaluated the performance of the detection of inserts by a set of COSFIRE filters and we compared results using different output functions. Results are shown in Table 4.2. With AM, 24 COSFIRE filters were configured and applied to the test set yielding an F-Score of 79.83\%. A set of 19 filters was configured for HGM reaching a F-Score of 89.76\%. SGM required only 17 filters and it achieved 88.89\% F-Score.
Figure 4.19: Results of applying the filter configured for prototype $f_1$ to the training set. Detected inserts are marked with a white rectangle. Above the rectangle, a colored square indicates by which output function the insert was found. Red denotes SGM, yellow HGM and green AM.

We can conclude that the output functions based on geometric mean are more appropriate than arithmetic mean for detecting inserts.

Besides, the number of configured filters affects detection rates as shown in
Table 4.2: Results in terms of number of configured COSFIRE filters, precision, recall and F-Score for the different output functions evaluated: Arithmetic Mean (AM), Hard Geometric Mean (HGM), Soft Geometric Mean (SGM), SGM when configuring the same 19 COSFIRE filters than for HGM (SGM$_{19}$).

<table>
<thead>
<tr>
<th></th>
<th>AM</th>
<th>HGM</th>
<th>SGM</th>
<th>SGM$_{19}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of filters</td>
<td>24</td>
<td>19</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>81.77</td>
<td>92.62</td>
<td>92.25</td>
<td>92.39</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>78.03</td>
<td>87.08</td>
<td>85.76</td>
<td>87.52</td>
</tr>
<tr>
<td>F-Score (%)</td>
<td>79.83</td>
<td>89.76</td>
<td>88.89</td>
<td>89.89</td>
</tr>
</tbody>
</table>

Azzopardi and Petkov (2013a). They proved that the performance results change with a different number of such filters, for their application, harmonic mean increased when increasing the number of configured filters up until 6 and then it progressively decreased. In order to compare the output functions SGM and HGM, we used the 19 COSFIRE filters that were configured with the HGM method and applied them with the SGM output function. In this experiment we obtained an F-Score of 89.89%, which is better than the F-Score of 89.76% (improvement of 1.12%) that we achieved with the same 19 filters but using the HGM function.

Although COSFIRE filters can achieve tolerance to rotation, scale and reflection Azzopardi and Petkov (2013c), in this application we did not apply any invariances to such geometrical transformations.

Changing the values of the parameter $t_3$ reaches different performance results. Increasing the value of $t_3$ causes an increase of precision and a decrease of recall. For each COSFIRE filter, we added to (or subtracted from) the corresponding learned threshold value $t_3$ an offset value in steps of $0.01t_1$. For all the studied function outputs, the maximum F-Score was reached at values of the threshold parameter $t_3$ with 0 offset (Fig. 4.20). Thus, the configured values of threshold $t_3$ at the training set are proven to be the best threshold values also for the test set.

4.4.4. Discussion

In the literature of machine vision, there are three families of approaches that are typically used for the detection of patterns of interest in images.

The first family of solutions are methods based on keypoint descriptors, such as SIFT Lowe (2004), SURF Bay et al. (2008), HOG Dalal and Triggs (2005b), CCS Jacobson et al. (2007). We attempted to use that approach for our application (data not shown), but resulted in lower performance. It is our belief that the reason for this is that, in our case of study, the information lies within the shape and contour of the object, rather than in its texture. Methods based on keypoint descriptors are more suitable for textured surfaces.
Figure 4.20: Precision-recall curves obtained for each of the studied metric functions: Arithmetic Mean (AM), Hard Geometric Mean (HGM), SGM when configuring the same 19 COSFIRE filters than for HGM (SGM19) and HGM with rotation invariance (HGMr). For each plot, the threshold values of parameters $t_3$ are varied by adding the same proportional offset value, ranging between $-0.05t_3$ to $0.05t_3$ in intervals of $0.01t_3$, to the corresponding learned threshold values. Precision increases and recall decreases with an increasing offset value. The F-Score reaches the maximum values for each plot at the original offset value $t_3$ with 0 offset.

The second family consists of those methods based on template matching. Template-matching methods use a set of typical image patterns or templates to determine similarities of an inspection image to a particular pattern in order to make classification decisions in automated visual inspection [Sun et al. (2012)]. A previous work of the authors [Aller-Álvarez et al. (2015)] applied template matching to this problem and obtained lower performance (F-Score=86%, precision 82% and recall 89% on the same dataset) than those obtained with the approach reported in this paper. In that work, first the authors preprocessed the images by applying Canny’s algorithm to the input image followed by a dilation of the edge map with a flat diamond-shaped structuring element of size 1 pixel from the centre of the structuring element to the points of the diamond. Then, they performed a normalized cross-correlation to measure the correspondence between each template, manually selected by the user and the considered window in the input image. The response of the template matching was considered as the two best correspondences per input image and template. The same test and training sets as in this work were used for obtaining the experimental results.

The third family of solutions are those that use domain knowledge. For instance, in this particular application we know that an insert is made of a circular screw surrounded by a rhomboid shape. We attempted this approach and obtained good
results [Fernández-Robles et al. (2015)]. In that case, the presence of a screw allowed the identification of the insert by means of detecting its circular contour.

The approach reported in the present section is far more versatile as it can also be applied to identify any tool or part without using domain knowledge. This is particularly important in other machine vision applications with objects of interests that might be very different than the inserts in the concerned application.

4.5. Conclusions

The contributions of the work presented in this chapter are four-fold. First, we described a method for the localisation of inserts with independence among images. Second, the approach that we proposed for the localisation of cutting edges in milling machines is highly effective and efficient. Its output is a set of regions surrounding cutting edges, which can be used as input to other methods that perform quality assessment of the edges. Third, we achieved an effective classification of the inserts with respect to the state of their cutting edges as broken and unbroken. Fourth, we presented a dataset of 144 images of a rotating edge milling cutting head that contains 30 inserts, analysing 180 inserts in total. It contains the ground truth information about the locations of the cutting edges, the locations of the centres of the inserts and broken inserts are labelled by experts. We made our dataset publicly available.

To our knowledge, this is the first automatic solution for the identification of broken inserts in edge profile milling heads. The presented system can be set up on-line and it can be applied while the milling head is in a resting position without delaying any machining operations. This system highly reduces the risk of head tool collapse, which is very expensive and time consuming to replace.