MM Science Journal | www.mmscience.eu
ISSN 1803-1269 (Print) | ISSN 1805-0476 (Online)
Special Issue | HSM 2023
17th International Conference on High Speed Machining
October 25-28, 2023, Nanjing, China
DOI: 10.17973/MMSJ.2023_11_2023124

HSM2023-00061

DIGITAL GEOMETRY GENERATION OF HIGH PRECISION BROACHING TOOL CUTTING EDGES THROUGH IMAGE PROCESSING ALGORITHM

Z. Gabos[1,2]*, D. Plakhotnik[3,4], Z. Dombovari[1,2]
[1]Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3. H-1111 Budapest, Hungary
[2]MTA-BME Lendület Machine Tool Vibration Research Group, Department of Applied Mechanics, Budapest University of Technology and Economics, Budapest H-1521
[3]ModuleWorks GmbH, Aachen, Germany
[4]University of Twente, Enschede, the Netherlands
*Corresponding author; e-mail: zoltan.gabos@mm.bme.hu

Abstract
In this paper an efficient geometry description algorithm is proposed that reads the cutting-edge points of an arbitrary broaching tool from the provided model in a pixelated format which is surely available in any engineering environment. The created geometry can be scaled arbitrarily and used for simulations, ensuring adaptability due to its point cloud like format. High precision machining requires large resolution (microns) that often leads to unmanageable data storage needs with increased calculation time. To overcome these problems, the proposed algorithm is built around a pseudo arc-length-like continuation of the cutting-edge pixels, which works on broaching tools due to the cutting-edge geometry is free from any loops or isolas. This method provides all the important data including the high-resolution edge sections, related chip thicknesses and the normal directions for dynamic simulations and manufacturing.

Keywords:
Broaching, High precision machining, Big data, Image processing

1 INTRODUCTION

Broaching is a high-productivity and high-precision machining procedure for workpieces with repeated intricate features [Arrazola 2020]. Although broaching is an expensive and relatively slow material removing process in terms of the involved speeds, it is still the most effective and frequently used method for component feature creation with complex geometry. Usually, broaching is one of the last steps of the manufacturing cycle used for finishing resulting in functioning surfaces [Brinksmeier 2012]. Note that, any malfunction in the process very likely lead to the destruction of the, most of the cases, high-valued components involved. To ensure a safe procedure and low waste, prior simulations of chip load, tool stress distribution and even machining dynamics of the broaching process might be recommended. For the proper mechanical description of the broaching process, a high-resolution geometric description of the cutting edges and chip thicknesses is necessary.

Geometric description of high-precision broaching tool edges is often defined through some kind of concatenated spline or Bézier description commonly used in computer-aided design (CAD) software [Hosseini 2013, Vogtel 2015]. Although these parametric curves provide the ideal description of the edges, saving computational costs and keeping data storage at a minimum, manufacturers are usually reluctant to provide such a description due to obvious intellectual property rights (IPR) related to the finished geometry of the often-high value workpieces involved. Another issue is that the extraction of chip thicknesses between subsequent edges, considering even multi-edge relations, needs consistent normalized parametrization which is far from the capability of a conventional CAD environment.

The usual method for tool geometry generation is the offset of the initial workpiece surface, which is given in a parametrized form [Hosseini 2013]. The problems with this method are the nonuniform offset of a real broaching tool [Vogtel 2015], and the parametrization of the subsequent connected cutting-edge geometries between the teeth is also not uniform. Avoiding the creation of isolas also must be addressed [Choi 1999, Liu 2007]. These are important measures for the proper mechanistic modelling of the broaching procedure leading to load and vibration predictions.

This paper introduces an image processing algorithm-based geometric description of broaching tools built around a pixelated data format to ensure adaptivity for simulations and manufacturing. The steps of the methodology are based on a pseudo arc-length-like continuation (PA) of the edge points from imported images. To ensure low calculation costs, the separate pictures of the cutting edges (e.g. exported from CAD software) are converted into a reduced sparse data format. After the detection of the cutting-edge pixels, the recognized geometry can be used for the relevant chip thickness extraction. All the necessary parameters are calculated and stored in a convenient matrix form which can be arbitrarily used between manufacturers for performing engineering predictions or to be included in digital-twin applications [Zhang 2019]. We note that this image processing algorithm is a built-in part of a self-developed dynamic simulation software created in MATLAB environment.

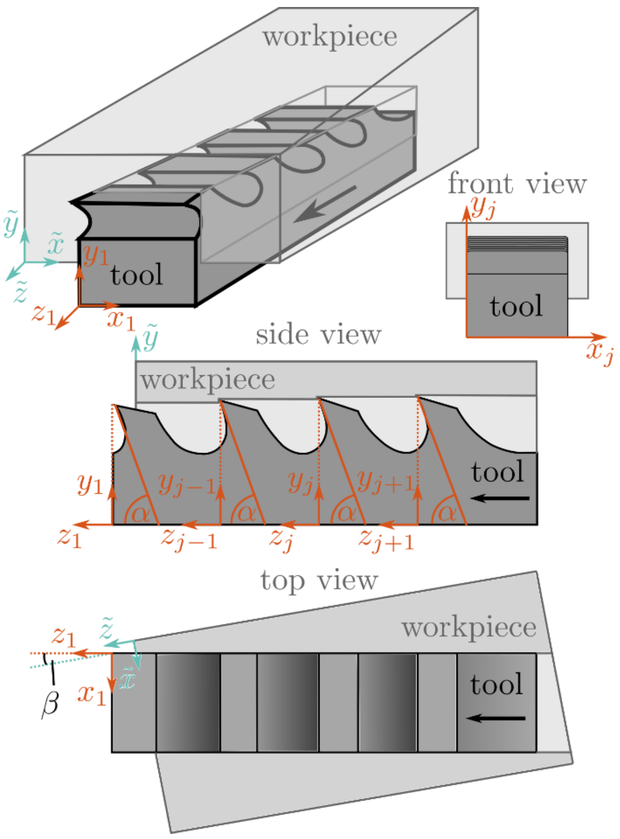


Fig. 1: Example broaching tool and its orientation relative to the workpiece. The angles α and β are the rotation of the cutting edge into the normal plane of the motion direction and the rotation of the tool relative to the workpiece.

2 BROACHING TOOL GEOMETRY

Broaching tool geometries can be very distinct from features to features and even between manufacturing steps. Similar products, manufacturers tend to use unique evolution of geometries and their description, which causes a lack of uniformity when it comes to studying broaching. Therefore, the goal of this section is to introduce a uniformly useable data structure for the tool geometry, which can be used in both simulations and on online manufacturing aiding Industry 4.0 solutions [Oztemel 2020]. Figure 1 is a visual representation of an example broaching tool and its orientation relative to the workpiece.

First, the coordinate system of the broaching geometry has to be defined, which is repeatable for arbitrary tool geometries. Every cutting-edge (tooth) has its local coordinate system (x_j, y_j, z_j), which is a projection to the normal plane of the feed direction. The angle between this normal plane and the actual cutting-edge is α as Fig. 1 shows. In addition, the workpiece can be also rotated relative to the broaching tool, as noted by the letter β. In Fig. 1, this rotation occurs around the axis y, thus $y \,||\, y_i$.

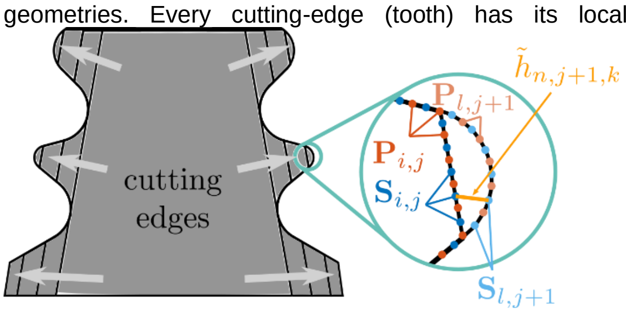


Fig. 2: Example broaching tool cutting edge geometry and its point cloud description, where the angle of the trapezoid sides is arbitrarily chosen to be around 10°. The points from the image processing algorithm and their middle points are $\mathbf{P}_{i,j}$ and $\mathbf{S}_{i,j}$ respectively. The direct chip thicknesses are noted as $\tilde{h}_{l,j+1,k}$.

Here, the β angle is considered to be zero, due to simplicity. An arbitrary β angle can be reached by the rotation of the stored dataset later. Notice, that the cutting-edge local coordinate systems are moving relative to the workpiece coordinate system. Also note that the edge planes are parallel to each other, given the edge angle α is constant.

The parametrization of the cutting edges is defined and stored in the projected (x_j, y_j, z_j) coordinate system of every broaching tool tooth. Intuitively, the stored data is the captured front view of the individual cutting edges. The visual representation of the cutting-edge parametrization can be seen in Fig. 2, where a general broaching tool is presented with an arbitrary geometry often referred to as firtree.

First, the cutting-edge has to be discretized into small edge segments for the parametrization of the teeth geometry. These edge segments are stored in a matrix form, where the first point coordinates of the line segments are described as $\mathbf{P}_{0,i,j}$, and the endpoint coordinates as $\mathbf{P}_{1,i,j}$. From here, the middle points can be calculated as

$$\mathbf{S}_{i,j} = \frac{\mathbf{P}_{1,i,j}+\mathbf{P}_{0,i,j}}{2}, \tag{1}$$

where $\mathbf{S}_{i,j}$ is the collection of middle points on the jth edge and the index i denotes the individual points. The determination of the middle points is important for the calculation of the cutting-edge segment normal $\mathbf{n}_{i,j}$. If the point cloud $\mathbf{S}_{i,j}$ is traced in a clockwise direction, then the calculation of the edge segment normal is formulated as

$$\mathbf{n}_{i,j} = \mathbf{e}_z \times \frac{\mathbf{P}_{1,i,j}-\mathbf{P}_{0,i,j}}{|\mathbf{P}_{1,i,j}-\mathbf{P}_{0,i,j}|},\ \mathbf{e}_z = \begin{bmatrix}0\\0\\1\end{bmatrix}, \tag{2}$$

where $\mathbf{e}_z$ is the rotation vector which results in outward-pointing normal vectors. Contrary, counterclockwise tracing requires multiplication with $-\mathbf{e}_z$ to ensure an outward-pointing vector field.

One of the most important parameters is the chip thickness along the cutting-edge since it directly affects the finished surface. To calculate this parameter at every edge point, first, the consecutive cutting segments have to be paired from tooth to tooth. Meaning, that the points on the current cutting-edge are connected to individual points on the previous edges. This is not a simple task to achieve, since

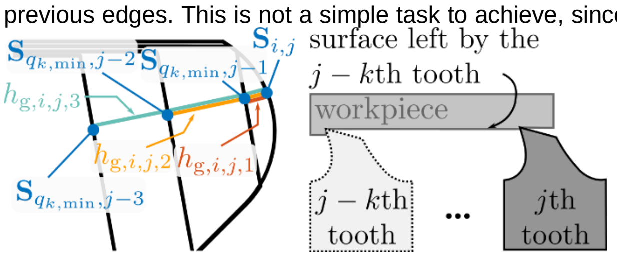


Fig. 3: Visual representation of the chip thickness matrix (left) and the path that is left by the last edge segment in contact (right).

there is a different number of points on each cutting-edge. There are points on the previous edge that connects to multiple edge points on the current edge. To create the pairing between two edge segments, the minimum distance has to be found between points on the curves as

$$\min_q \left|\mathbf{n}_{i,j}(\mathbf{S}_{i,j}-\mathbf{S}_{q,j-k})\right|, \; k = 1,2\ldots,j-1, \qquad (3)$$

where $q = 1,2,\ldots,i_{j-k}$ and the index $q_{k,\min}$ refers to the closest point on the kth previous edge to the ith point on the current edge j. From here, the geometric chip thickness is roughly

$$h_{\mathrm{g},i,j,k} = \mathbf{n}_{i,j}(\mathbf{S}_{i,j}-\mathbf{S}_{q_{k\min},j-k}), \qquad (4)$$

where the indices are the notation of an individual edge segment i on the current edge j.

This signed nominal chip thickness $h_{\mathrm{g},i,j,k}$ is positive if the edge segment is cutting in relation to the edge segments k teeth before. In order to have the real decisive nominal chip thickness the following has to be performed according to [Dombovari 2010, Wang 1996]

$$h_{\mathrm{g},i,j} = \min_l h_{\mathrm{g},i,j,l}. \qquad (5)$$

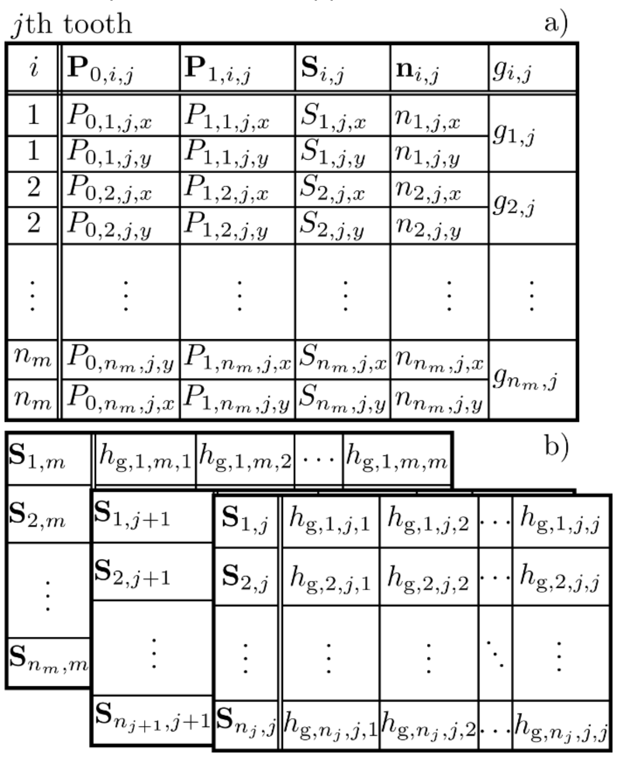
This is now sufficient to describe the connection between every consecutive edge-segments even if it has contact loss during operation. If the edge misses the workpiece, we talk about missed-cuts [Dombovari 2010]. Notice, that multiple edges-segments can be in a contact loss state consecutively due to concave geometry segments, edge breakages, or in extreme cases of vibration resulting in the flyover effect [Iklodi 2022]. Therefore, to simulate the static and dynamic behaviour of a broaching process, the whole chip thickness history of the current edge point has to be traced back to the initial surface.

In addition, the animation of the final surface also requires the geometric chip thickness history with $h_{\mathrm{g},i,j,k}$. The visual representation of $h_{\mathrm{g},i,j,k}$ and its importance is shown in Fig. 3.

Finally, the real (realized) chip thickness can be determined on which the cutting force calculation will be based on in the following way, by accepting only the positive ones as

$$h_{i,j} := g_{i,j}h_{\mathrm{g},i,j}, \qquad g_{i,j} = \begin{cases} 1, & h_{\mathrm{g},i,j} > \varepsilon, \\ 0, & h_{\mathrm{g},i,j} \le \varepsilon, \end{cases} \qquad (6)$$

where $g_{i,j} = 1$ is the notation of the active (cutting) edge segment, while $g_{i,j} = 0$ denotes edge segment missed-cuts. We note that the active edge indicator $g_{i,j}$ is also time-dependent since the cutting-edge segment can lose contact at any time during machining. The static part of this parameter is originated from missed-cuts, while time-dependent part may come from flyover effect if dynamic simulation is performed. In practice, a cutting-edge segment is considered to be inactive if $h_{\mathrm{g},i,j}$ does not exceed a prescribed value $\boldsymbol{\varepsilon}$ (6).



jth tooth a)

i	$\mathbf{P}_{0,i,j}$	$\mathbf{P}_{1,i,j}$	$\mathbf{S}_{i,j}$	$\mathbf{n}_{i,j}$	$g_{i,j}$
1	$P_{0,1,j,x}$	$P_{1,1,j,x}$	$S_{1,j,x}$	$n_{1,j,x}$	$g_{1,j}$
1	$P_{0,1,j,y}$	$P_{1,1,j,y}$	$S_{1,j,y}$	$n_{1,j,y}$	
2	$P_{0,2,j,x}$	$P_{1,2,j,x}$	$S_{2,j,x}$	$n_{2,j,x}$	$g_{2,j}$
2	$P_{0,2,j,y}$	$P_{1,2,j,y}$	$S_{2,j,y}$	$n_{2,j,y}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n_m	$P_{0,n_m,j,y}$	$P_{1,n_m,j,x}$	$S_{n_m,j,x}$	$n_{n_m,j,x}$	$g_{n_m,j}$
n_m	$P_{0,n_m,j,x}$	$P_{1,n_m,j,y}$	$S_{n_m,j,y}$	$n_{n_m,j,y}$	

b)
$\mathbf{S}_{1,m}$ $h_{\mathrm{g},1,m,1}$ $h_{\mathrm{g},1,m,2}$ $\cdots$ $h_{\mathrm{g},1,m,m}$
$\mathbf{S}_{2,m}$
$\vdots$
$\mathbf{S}_{n_m,m}$
$\mathbf{S}_{1,j+1}$
$\mathbf{S}_{2,j+1}$
$\vdots$
$\mathbf{S}_{n_{j+1},j+1}$
$\mathbf{S}_{1,j}$ $h_{\mathrm{g},1,j,1}$ $h_{\mathrm{g},1,j,2}$ $\cdots$ $h_{\mathrm{g},1,j,j}$
$\mathbf{S}_{2,j}$ $h_{\mathrm{g},2,j,1}$ $h_{\mathrm{g},2,j,2}$ $\cdots$ $h_{\mathrm{g},2,j,j}$
$\vdots$ $\vdots$ $\vdots$ $\ddots$ $\vdots$
$\mathbf{S}_{n_j,j}$ $h_{\mathrm{g},n_j,j,1}$ $h_{\mathrm{g},n_j,j,2}$ $\cdots$ $h_{\mathrm{g},n_j,j,j}$

Fig. 4: Visualisation of the proposed data structure for the geometry description of broaching tools. Matrix a) is the general information matrix, and the four-dimensional matrix b) is the undeformed chip thickness history of every cutting edge segments on the tool.

Now, all the necessary parameters are known for the broaching force estimation and to perform dynamic simulation. In this manner, valuable information can be provided to the end-user separated from scaling. The manufacturer simply chooses the resolution according to their accuracy expectation, while scaling key value is kept separately from the simulation holding IPR issues essentially in-house, where simulation is performed. Consequently, all values in (1-6) are actually in the unit of pixels, the real distance unit is introduced separately later by the end-user. The summary of these parameters and the proposed data structure is visualized in Fig. 4. Notice, that in the chip thickness matrix every cutting-edge has a different number of middle points $\mathbf{S}_{i,j}$ and different depth of cutting history (3rd index).

3 IMAGE PROCESSING ALGORITHM

In order to build the proper algorithm for describing tool geometry based on the data structure presented in the previous section (Sec. 2.), first, the available information has to be collected. The exact description format and measures of a broaching tool are hardly available due to the already mentioned IPR issues. Therefore, the algorithm and the data structure must be adaptive for both reading and post-processing the geometry by separating the pixelate-able shape information from the scaling information. Therefore, the proposed algorithm is built around extracting the geometry from common image formats made in-house using an available CAD software. Once the software has recognized the geometry the common data structure describes the broaching process and it can be stored as a tool specific data.

The remainder of this section is for the introduction of the proposed image processing algorithm. Every step will be discussed thoroughly in a consecutive manner. Finally, the steps will be collected for better interpretability.

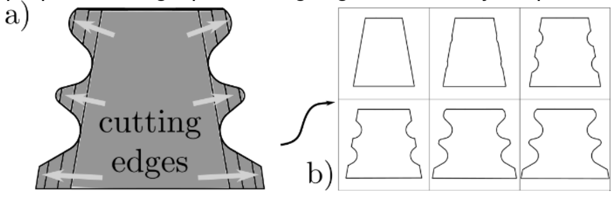


Fig. 5: Preparation of the broaching tool geometry for the image processing algorithm. a) is the front view a general fir tree broaching tool and b) is the exported individual pixelated teeth projections.

The first step is the preparation of the geometry from the available information. In this study, we used an arbitrary CAD model of the broaching tools as a starting point similarly to [Olmo 2022, Ozlu 2010]. Important to determine a common uniform frame view that could include all intricate edge geometry with the same aspect ratio (see Fig. 5).

The final resolution of the geometry description is the resolution of the exported figures. If the desired accuracy of the simulations is one micron, then one pixel should also resemble one micron at this step, however this information might be only available at the end-user. Although the pixelation creates a scarce continuation of the cutting edge, the deviation decreases with the growing resolution of the image and is always less than the resolution.

Advantageously, even high-resolution figures can be stored in a fairly small size due to common compression techniques (jpg, png), which later can be processed in a sparse variable definition. This significantly reduce the storage needed and the computational time.

To ensure easy handling of the data and cheap calculations, the simplification of the figures is done by storing the data in sparse matrices. This requires the conversion of the image into grey-scaled doubles from the exported files, which does not affect the accuracy when sufficient contrast is given (ensured when CAD models are used). By inverting the colour scheme, the fully black pixels will appear as ones in the matrix. Identification of the cutting-edge pixels then only requires finding the pixels, with values close to one:

$$\operatorname*{find}_{l,q} M_{s,l,q}, > \delta,\ l = 1,2,\dots,n,\ q = 1,2,\dots,m,\ \delta \in \mathbb{R}: [0,1], \quad (7)$$

where $M_{s,l,q}$ is the grey-scaled figure data (double) stored in sparse (s) matrix form. The indices l and q are the x and y coordinates and the colour grading is stored as values between 0 and 1. Finally, δ is a prescribed value to find all the relevant pixels that are parts of the cutting-edge. The steps from the raw pixelated data to the sparse matrix are visualised in Fig. 6.

After all the cutting-edge pixels have been found and stored as indices of the sparse matrix $M_{s,l,q}$, the next step is to define the cutting-edge in an organised manner. At this step, only the pixel coordinates are known as can be seen in Fig. 7. The connection between the edge pixels and, therefore, the cutting-edge segments is yet to be defined.

The pixelated parametrisation of the edge is beginning with the identification of the starting $\mathbf{P}_{0,i,j}$ and endpoints $\mathbf{P}_{1,i,j}$ of the cutting-edge segments. This is achieved through a pseudo arc-length-like algorithm (PA) that can trace the outer boundary of the pixelated curve, resulting in an organised point cloud. By moving through $\mathbf{P}_{0,i,j}$ and $\mathbf{P}_{1,i,j}$ a continuous curve can be traced. Notice, that $\mathbf{P}_{0,i,j} = \mathbf{P}_{1,i-1,j}$, due to the segmented description format.

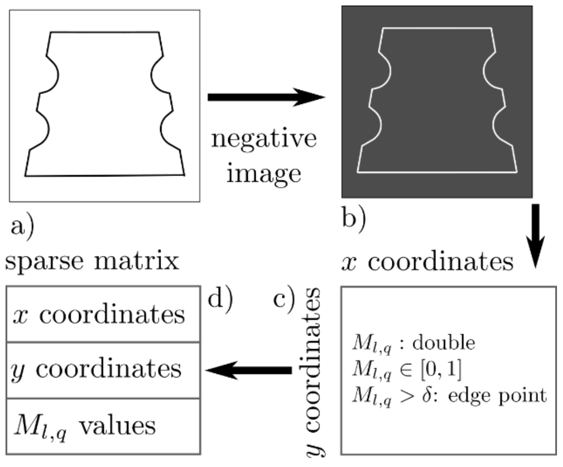


Fig. 6: Converting the image data into a sparse matrix format with double values. From a) to b) the negative of the image is taken, and then the value one is connected to the edge pixels (panel c). Panel d) is the sparse matrix representation of the image data.

The PA method works similarly to the pseudo arc-length continuation procedure [Barton 2011, Doedel 1981, Engelborghs 2001, Roose 1990]. Instead of creating a mesh and searching through grid points, then finding minimum distances, the algorithm simply scans the previous point's neighbourhood to find the next relevant point. First, an outer boundary point has to be found within the pixelated point cloud that is recognised as part of the current cutting-edge. This is simply just picking the highest or lowest coordinate index in any direction from the identified pixel points of the edge curve. This is the initialisation of the PA algorithm.

The proposed algorithm uses the previous two points $\mathbf{P}_{0,i,j}$ and $\mathbf{P}_{1,i,j}$ (or the initial one) as the origin of the scanning procedure. Then $\mathbf{P}_{1,i+1,j}$ is found by evaluating every point in $M_{s,l,q}$ around (in a circle with a 1-pixel radius) $\mathbf{P}_{0,i+1,j} = \mathbf{P}_{1,i,j}$ in a clockwise or counterclockwise manner (depending on the user defined tracing direction). The evaluation looks for large colour changes, meaning that if the previous point on the traced circle is white and the next one is black, then the next point $\mathbf{P}_{1,i+1,j}$ is the indices of the pixel with a value close to 1 (black) in $M_{l,q}$. The visual representation of the algorithm can be seen in Fig. 7.

The first iteration of the PA algorithm is initialised from only one point that is a limit pixel of the cutting edge. Therefore, 4 different starting scenarios are possible, which are related to the highest and lowest q and l values from (7). Then the first scanning point is

$$\begin{cases} M_{s,l,q_{\max}+1}, & q_{\max} := \max_n q_n, \\ M_{s,l,q_{\min}-1}, & q_{\min} := \min_n q_n \\ M_{s,l_{\max}+1,q}, & l_{\max} := \max_n l_n \\ M_{s,l_{\min}-1,q}, & l_{\min} := \min_n l_n \end{cases},\quad n = 1,\dots,m, \qquad (8)$$

where q_n and l_n are the identified indices of the edge, and m is the number of the found cutting-edge related pixels.

Notice, that the PA algorithm works on broaching tool geometries, due to their continuous and loopless topological cutting-edge structures [Choi 1999]. Consequently, if discontinuity appears due to poor image resolution or faulty image exportation, the proposed algorithm will fail by giving an error message or looping into itself. As a result of this, it is highly recommended to export the figures in high resolution, and only after that, reduce

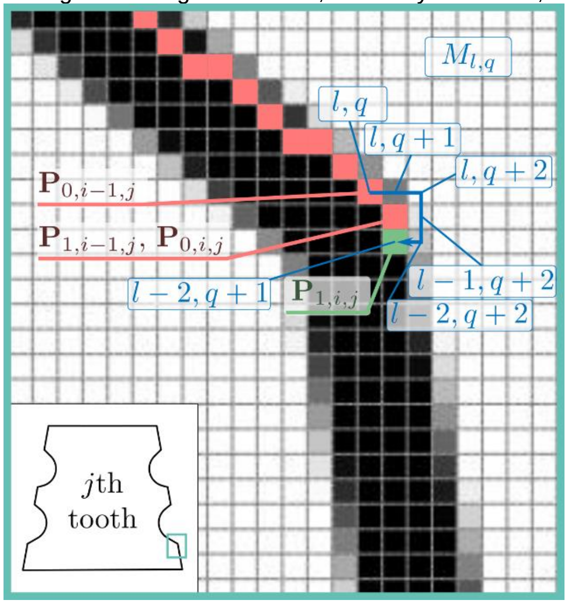


Fig. 7: Visual representation of the PA algorithm to trace the broaching tool's cutting-edge points ($\mathbf{P}_{0,i,j}$ and $\mathbf{P}_{1,i,j}$). The 1-pixel radius circular scanning and the related matrix indices in $M_{l,q}$ are marked with blue. $\mathbf{P}_{1,i,j}$ is the identified point by the algorithm and $\mathbf{P}_{0,i-1,j}$ with $\mathbf{P}_{1,i-1,j}$ are there to initialise the algorithm.

edge resolution by eliminating data points, to further reduce calculation costs and data size. This way the imported picture will be always sufficient for the algorithm.

The next step is the averaging of the identified curve. This is only necessary if the resolution of the data is not sufficiently high since it will result in horizontal and vertical line segments where otherwise smooth curves should be traced. This is due to the pixelated data structure, and the phenomenon is presented in Fig. 8. On the other hand, if the resolution is sufficiently high, then the segments between pixels will trace an acceptable smooth curve with variation within a single pixel size. Even in this case, it is advised to apply this averaging step due to calculation cost reduction. When straight vertical or horizontal lines are traced, it is unnecessary to store every data point and most of the points can be eliminated on the horizontal or vertical cutting-edges.

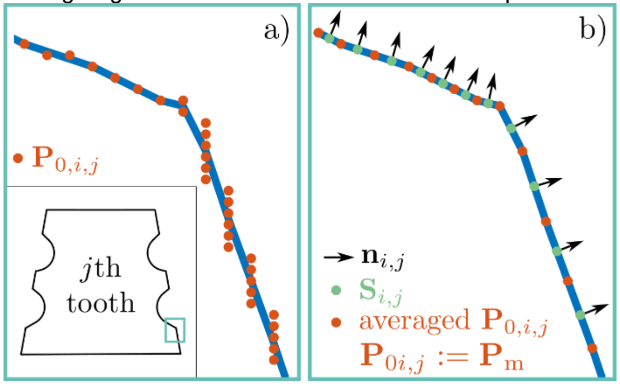
During the averaging step, the algorithm looks for vertical and horizontal lines longer than a prescribed value (e.g. 4 pixels) as Fig. 8 shows. The middle point $\mathbf{P}_\mathrm{m}$ of the vertical lines is calculated as

$$\mathbf{P}_\mathrm{m} = \frac{\sum_{i=i_\mathrm{i}}^{i_\mathrm{e}} \mathbf{P}_{0,i,j}}{i_\mathrm{e} - i_\mathrm{i}}, \tag{10}$$

where i_i and i_e are the initial or end point of the vertical or horizontal line respectively. The middle point $\mathbf{P}_\mathrm{m}$ replaces all points between $\mathbf{P}_{0,i_\mathrm{i},j}$ and $\mathbf{P}_{0,i_\mathrm{e},j}$. This way, we can save calculation costs without any loss of important information.

Once, the averaging of the pixelated cutting-edge curve is done, the middle points $\mathbf{S}_{i,j}$ can be calculated according to (1). Then the unit normal vectors $\mathbf{n}_{i,j}$ are calculated based on (2). The accurate calculation of the normal vectors is high priority since it has a direct influence on the surface through the orientation of the cutting-edge segment and chip thickness (Fig. 8).

The next step is the calculation of the geometrical chip thickness history $h_{\mathrm{g},i,j,k}$. Here, we emphasise that the length of index i varies from edge to edge, since every cutting-edge consist of a different number of points. The



a)
b)
● $\mathbf{P}_{0,i,j}$
jth tooth
$\rightarrow \mathbf{n}_{i,j}$
● $\mathbf{S}_{i,j}$
● averaged $\mathbf{P}_{0,i,j}$
$\mathbf{P}_{0i,j} := \mathbf{P}_\mathrm{m}$

Fig. 8: Averaging and further processing of the identified edge points to get the middle points of the edge segments $\mathbf{S}_{i,j}$ and their normal unit vectors $\mathbf{n}_{i,j}$. Panel a) is there to visualise the necessity of the averaging step. Panel b) is the visualised parametrisation of the cutting-edge.

length of index k is also changing as it refers to the history (chip thickness between edges j and $j-k+1$) of the current cutting-edge. The calculation of $h_{\mathrm{g},i,j,k}$ is described by (4).

The connection of consecutive edge segments is done by applying a minimum search algorithm (3) as it is visualised by the dashed lines in Fig. 9. Starting from the current edge segment, the nearest point on the previous tooth is detected, and then the same is done from the detected point repeatedly until the first cutting edge is reached. This is important if a missed-cut or flyover phenomenon occurs and the surface machined by the current edge is not the imprint of the previous edge directly. Also, $h_{\mathrm{g},i,j,k}$ is important due to the finished surface is directly related to the cumulative chip thickness and the movement of the tool together.

The last step before the creation of the uniform data structure is to eliminate all the inactive edge segments. During, the image processing algorithm, all the edge points are captured, even the ones that are not part of the cutting process or for some reason (e.g. overfly) the edge segment loses contact with the workpiece surface during operation. To eliminate the statically inactive edge segments, a small ε value is introduced in order to decide if the calculated chip thickness comes from error or noise in the pixelated data, or if the calculated chip thickness is indeed valid, and the edge segment is active (Fig. 10).

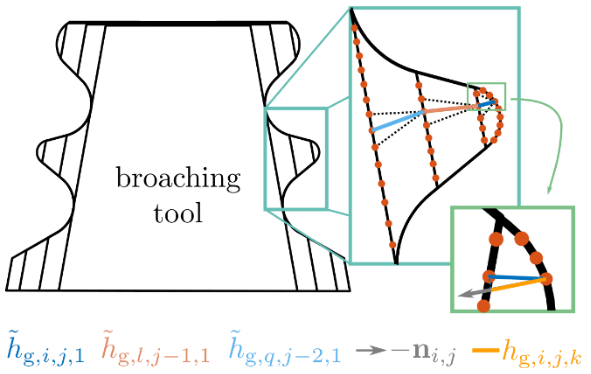


Fig. 9: Calculation of the chip thickness history. Indices i, l, and q are showing, that every cutting-edge has a different number of middle points $\mathbf{S}_{i,j}$. The $k = 1$ refers to the chip thickness with the direct neighbouring edge. The black dashed lines are there to visualise the minimum distance finding algorithm.

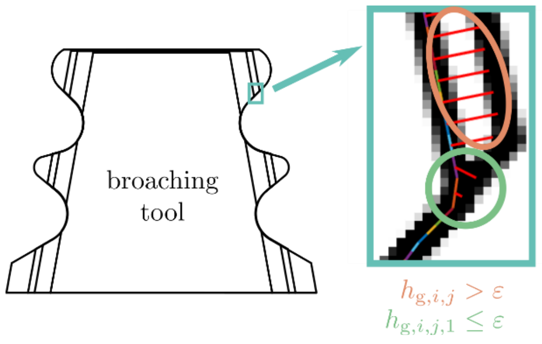


Fig. 10: Visual interpretation of the need for the definition of a small ε value that eliminates all the points that are not part of the current cutting process.

The ε value is defined prior to the calculations, based on the desired accuracy that comes from the resolution. If the required resolution is around a micron, then the definition of ε should be also around a micron. The calculation of active segments is defined in equation (6).

Finally, all the above-mentioned calculations and parameters are stored in a uniform data structure, described in Fig. 4. The individual edges form individual matrices with the pixelated description of the broaching tool geometry. Then the chip thickness history is also stored separately for every cutting-edge, where the stored data size is increasing as the number of teeth grows on a broaching tool.

The algorithm procedure can be followed in a step-by-step manner in Fig. 11 as a summary of the image processing method. After the exportation of all edges, the identification of the cutting-edge geometry is processed from edge to edge in a loop. After knowing the whole broaching tool geometry, the chip thickness calculation and the active edge segment detection can occur.

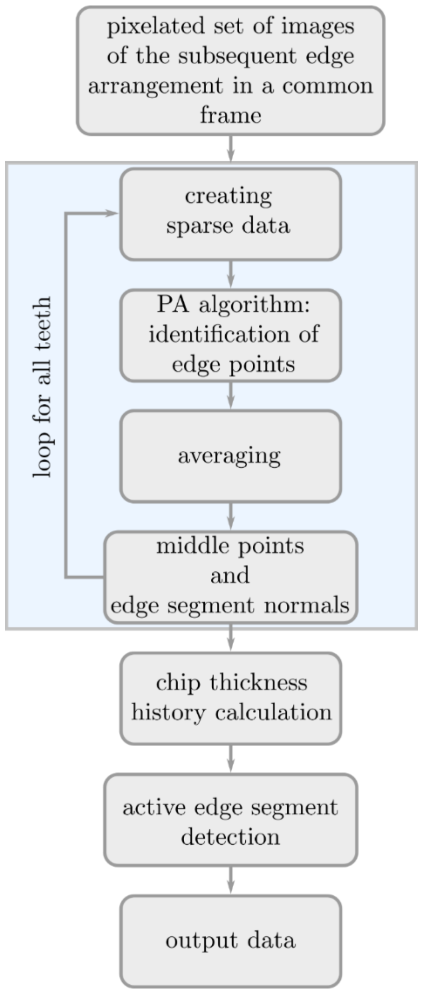
4 DISCUSSION AND CONCLUSION

The focus of this study is the creation of a uniform geometry description format and data structure with an image processing algorithm that can recreate the geometry of broaching tools based on images originated from some kind of CAD models. The proposed data structure is a point cloud (pixels) in the coordinate system (x, y, z) as it is shown in Fig. 1. It is important to bear in mind that the identified edge curves are in the plane perpendicular to the feed direction (front view).

Providing this pixelated data structure ensures the user to adaptively utilize the geometry for cutting load prediction, simulations, optimization, and Industry 4 ready manufacturing. Due to its structure, it can be easily rescaled, modified, or even parametrized.

The calculation of the chip thickness history makes it possible to generate the expected final surface of the workpiece. The chip thickness is also an indicator for detecting inactive edges, which is especially important in dynamic simulations. Here, we introduced the static indicator of the active edge segments, but it is important to notice the time dependency of this indicator during operation. We also note that submicron dislocation errors have been detected on the chip thickness value extraction, which have significant effect on the final surface generation handled by an additional smoothing synchronization cycle that also increases the calculation time.

The presented geometry processing algorithm for arbitrary


pixelated set of images of the subsequent edge arrangement in a common frame
loop for all teeth
creating sparse data
PA algorithm: identification of edge points
averaging
middle points and edge segment normals
chip thickness history calculation
active edge segment detection
output data

broaching tools is a product of a larger software package

Fig. 11: Flowchart representation of the proposed image processing algorithm for the creation of a uniform data structure for broaching tool geometry description.

developed in MATLAB environment. This paper presents only the identification and recreation of cutting-edge geometry and the introduction of a uniformly useable data structure.

5 SUMMARY

This paper introduces a uniform data structure for the geometric description of broaching tool and a sufficient image processing algorithm for the parametrization of the cutting-edges. Using the proposed data structure and image processing algorithm, both simulations and optimisation can be carried out, and relevant information can be provided for manufacturers.

6 ACKNOWLEDGMENTS

This research was partially funded from the European Union's Horizon 2020 Research and Innovation. Programme under the project InterQ (grant agreement No. 958357), and it is an initiative of the Factories-of-the-Future (FoF) Public Private Partnership.

7 REFERENCES

[Arrazola 2020] P. J. Arrazola, J. Rech, R. M'Saoubi, D. Axinte. Broaching: Cutting tools and machine tools for manufacturing high quality features in components. CIRP Annals – Manufacturing Technology, June 2020, Vol.69, No.2., pp 554-577. ISSN 0007-8506

[Barton 2011] D.A.W. Barton, B.P. Mann, S.G.Burrow. Control-based continuation for investigating nonlinear experiments. Journal of Vibration and Control, 2012, Vol.18, No.4., pp. 509-520, ISSN 1741-2986

[Brinksmeier 2012] E. Brinksmeier, R. Gläbe, L. Schönemann, Review on diamond-machining processes for the generation of functional surface structures. CIRP Journal of Manufacturing Science and Technology, 2012, Vol.5, No.1., pp.1-7. ISSN 1755-5817

[Choi 1999] B.K Choi, S.C. Park. A pair-wise offset algorithm for 2D point-sequence curve. Computer-Aided Design, October 1999, Vol.31, No.12., ISSN 0010-4485

[Doedel 1981] E.J. Doedel. AUTO: A program for the automatic bifurcation analysis of autonomous systems, Congressus Numerantium, April 1981, Vol.30, pp. 25-93

[Dombovari 2010] Z. Dombovari, Y. Altintas, G. Stepan, The effect of serration on mechanics and stability of milling cutters. International Journal of Machine Tools and Manufacture, June 2010, Vol.50, No.6, pp. 511-520, ISSN 0890-6955

[Engelborghs 2001] K. Engelborghs, V. Lemaire, J. Bélair, D. Roose. Numerical bifurcation analysis of delay differential equations arising from physiological modeling. Journal of Mathematical Biology, April 2001, Vol.42, pp. 361-385

[Hosseini 2013] A. Hosseini, A. Kishawy. Prediction of cutting forces in broaching operation. Journal of Advanced Manufacturing Systems, 2013, Vol.12, No.1., pp 1-14. ISSN 1793-6896

[Iklodi 2022] Zs. Iklodi, D.A.W. Barton, Z. Dombovari. Bi-stability induced by motion limiting constraints on boring bar tuned mass dampers. Journal of Sound and Vibration, January 2022, Vol.517, pp. 116538, ISSN 0022-460X

[Liu 2007] X. Liu, J. Yong, G. Zheng, J. Sun. An offset algorithm for polyline curves. Computers in Industry, April 2007, Vol.58, No.3., pp. 240-254, ISSN 0166-3615

[Olmo 2022] A. del Olmo, L.N. López de Lacalle, G. Martínez de Pissón, C. Pérez-Salinas, J.A. Ealo, L. Sastoque, M.H Fernandes. Tool wear monitoring of high-speed broaching process with carbide tools to reduce production errors. Mechanical Systems and Signal Processing, June 2022, Vol.172, pp. 109003, ISSN 0888-3270

[Ozlu 2010] E. Ozlu, S. Engin, C. Cook, T. El-Wardany, E. Budak. Simulation of Broaching Operations for Tool Design Optimization. 2nd International CIRP Conference on Process Machine Interactions, June 2010, pp. 10-11

[Oztemel 2020] E. Oztemel, S. Gursev, Literature review of Industry 4.0 and related technologies. Journal of Intelligent Manufacturing, 2020, Vol.31, No.1, pp. 127-182. ISSN 1572-8145

[Roose 1990] D. Roose, B.D. Dier, A. Spence. Continuation and Bifurcations: Numerical Techniques and Applications. Springer, Nato Science Series C, Vol.313, ISBN 978-94-009-0659-4

[Vogtel 2015] P. Vogtel, F. Klocke, D. Lung, S. Terzi. Automatic Broaching Tool Design by Technological and Geometrical Optimization. Procedia CIRP, 2015, Vol.33, pp. 496-501, ISSN 2212-8271

[Wang 1996] J.-J. Wang, S.Y. Liang. Chip Load Kinematics in Milling With Radial Cutter Runout. Journal of Engineering Industry, February 1996, Vol.118, No.1, pp. 111-116, ISSN 1528-8935

[Zhang 2019] C. Zhang, G. Zhou, J. He, Z. Li, W. Cheng. A data- and knowledge-driven framework for digital twin manufacturing cell. Procedia CIRP, 2019, Vol.83, pp. 345-350. ISSN 2212-8271