

Automatic trajectory generation for the milling of sculpted surfaces acquired from laser scanning systems

João L. Vilaça*, Jaime C. Fonseca+, ACM Pinho°

University of Minho, School of Health and Sciences, 4710-057 Gualtar, Braga*, Industrial Electronics Department+ and Mechanical Department°, 4800-058 Azurém, Guimarães - Portugal.

joaovilaca@ecsau.de.uminho.pt*, jaime@dei.uminho.pt+, acmpinho@dem.uminho.pt°

Abstract - The present work describes a set of algorithms for the milling of sculpted surfaces acquired from laser scanning systems. The milling operations cover three operations: roughing, semi-finishing and finishing. The developed algorithms use techniques ranging from computer vision to computer graphics following a different approach when compared with the usual milling techniques, usually feature based. The implementation of 3D path simulator was also described and testing of the different milling trajectories presented.

Keywords: sculpted surfaces, machining, spline interpolation, image processing

I. INTRODUCTION

Milling sculpted surfaces (free form surface) has a very important role in the invention of new products in the market. This kind of surface is usually used, for example, in the invention of work pieces for the automobile, furniture and aerospace industry [1-4].

However, moulding this type of surfaces with a CAD software is done differently, where moulding most of the times implies the invention of physical models (sculpted) of clay or wood, as well as its posterior digitalization. One such example entails modules for data acquisition and processing [5].

Moreover, once these surfaces show curvilinear features, milling becomes harder for most of the controllers presently existing in the industry, meaning that they're only prepared for linear and circular interpolations. For this reason, they need post processors that convert the produced trajectories into this type of surfaces (for example, spline and NURBS – “Non Uniform Rational Basis Spline”) in compatible trajectories with their interpolation systems. As a consequence there are broad and difficult programs to interpret and they are also sensitive to eventual loss of information, namely in the complex curvilinear geometry

encodes, that most of the postprocessors convert in small straight line segments.

Presently there are several CAM programs that allow the invention of trajectories for rapid and efficient milling (*MasterCam*, *Catia*, *PowerMill*, *Pró-Engineer*, and others). The function of these programs is essentially based on the identification of geometric features on the work piece to mill and on the application of the milling strategy most suitable for each type of geometry, according to the identified geometric features [6, 7]. However, this functioning rule isn't applicable for milling sculpted surface, since it's not possible to proceed to the common geometric feature identification (cubes, spheres, pyramids) as well as to link strategies for standard milling. Defining a machining trajectory with the most suitable trajectories for each sculpted surface turns out to be a real challenge.

The developed module will be introduced in this paper with the objective of creating necessary trajectories to mill the sculpted surfaces, obtained from the laser scanning systems. The created trajectories include the roughing, semifinishing and finishing operations. Yet, these created trajectories, in the different operations, only introduce the geometric information and the type of tool used. It is suggested that other parameters should be considered in future work, such as rotation and feeding rates.

A specific strategy was associated to each one of the milling operations. Each one of them was established according to the controller features of the testing machine (enables a spline interpolation) and seeks to remove the greatest amount of material in the smallest time possible, maintaining the demanded superficial finishing.

The paper is divided in the following sections: section 2 describes the developed the main function of the developed module; section 3 explains the developed algorithm for rouging operation; section 4 presents an algorithm semi-finishing and finishing operations based on spline

interpolation; section 5 introduce the techniques used to develop the milling path simulator; section 6 shows three analyses of the developed milling strategies with different kinds path characteristics; and finally in Section 6 some final considerations are set.

II. SOFTWARE OVERVIEW

The *software* module was developed in *Visual C++* and it was named “*Scanned Sculpted Surfaces CAM*” (fig. 1). It accepts a regular 2D structure input format and it has six available options:

- Invention of a project – where it is defined the name of the project and the input format is selected. Moreover, the blocks dimensions to mill are automatically set up, based on the maximum dimensions of the object related to this file;
- Tools data base – the developed program only has two kinds of tools: FEM (flat end milling) and BEM (ball end milling). The first one is used for roughing operations and the second for semi-finishing and finishing operations. In this option the user can create two types of tools, denoting its features, for example: height, diameter, radius;
- Roughing operations – roughing trajectories are defined. When defining the trajectories the tool is selected, according to the definition. The parameters for the trajectories (spacing between the trajectories and depth of cut) are automatically set up. They can be changed by the user, if he well understands to;
- Semi-finishing operations – semi-finishing trajectories are defined. When defining the trajectories the tool must be selected, so that the parameters for the trajectories can be automatically established. Most of the times, they can be modified by the user;
- Finishing operations – it is the same as the operation above, even though it must be accomplished with a smaller radius tool;
- Simulation – it’s an option that allows visualizing a three-dimensional simulation for each one of milling operations.

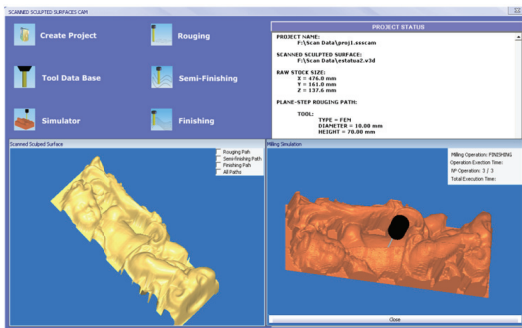


Fig. 1. *Software* interface developed for the module of automatic generation of trajectories to mill sculpted surfaces.

The developed application still has a window that shows an historical of the operations done by the user, since the beginning of the project.

III. ROUGHING

The roughing operation has the objective of removing the greatest amount of material in the smallest time possible, trying the best to achieve the results of the required surface.

The selected milling strategy for this operation is called *z-parallel*. In short, this strategy establishes parallel cutting plans. Also parallel trajectories are defined, between the limit points of the block to mill and the intersection points of the cutting plan with the required surface (surface of the digitalized object). This strategy was chosen because of being an efficient and adequate strategy for this kind of surfaces [8]. Thus, the chosen algorithm in this subpoint for the implementation of this strategy in some aspects follows the algorithm proposed by Choi e Jerard (1998) [ibidem].

The created algorithm to carry out this milling strategy develops many stages and has the following input data: the required surface; the tool (namely the FEM type); the spacing between the trajectories (ω); the depth of cut (S); and, the dimensions of the block to mill (raw-material).

The first stage is adapting the required surface to the *z-map*. This *z-map* is a map with a height distributed along a xy chart, with the dimensions m and n defined by the maximum x and y dimensions of the block to mill. At each point of the chart it is given the value of the coordinate z (height). This value $z[x_i, y_i]$ for $\{x_i, y_i: i \in [0, m], j \in [0, n]\}$ is calculated through a 3D straight line intersection, constant in x and y due to $[x_i, y_i]$, with the required surface. The values of x_i and y_i are calculated due to the chart resolution, the examples presented in this project were $2 \times 2 \text{ mm}^2$.

Knowing that the input data is distributed in a 2D chart, organized with the neighbourhood information, the calculation of the intersection is simplified, once it’s only processed in the neighbouring faces $[x_i, y_i]$. The points where there is no intersection it is given the value of zero (inferior left corner of fig. 1).

The second stage is establishing a more advanced orientation for the trajectories. This orientation is allowing the definition of trajectories with a larger length between the reference points. This way it minimizes the number of accelerations and decelerations, meaning that the trajectories main orientation is defined in the direction of the *z-map* coordinate with a larger dimension.

After the *z-map* and the trajectories main orientation is defined the cutting plan stage begins. The objective of this stage is to define a set of parallel and even plans (Zpc_k), achieved after the successive increases along the *z* axis, defined by the selected value for the depth of cut (1). Each one of these cutting plans crosses the surface, defining iso-value lines (same value as *z*).

$$Zpc_k = Zpc_0 + k \times S \quad \text{to} \quad \{Zpc_k: k \in [0, l]\} \quad (1)$$

The next stage is creating a 2D binary image with the same resolution and dimensions as the *z-map*, related to each one of the cutting plans. In this case, the value for each point of the image is $I(x_i, y_i)$ it's defined by (2):

$$I(x_i, y_i) = \begin{cases} 0, & z[x_i, y_i] < Zpc_k \\ 255, & z[x_i, y_i] \geq Zpc_k \end{cases} \quad (2)$$

In the created images it is possible to observe black and white points. The first ones represent material removal areas where the milling should be defined. The second ones represent the areas of the object where such operation should not take place. When applying the *Freeman chain code* algorithm [9, 10] to the image it is possible to obtain the contours of those areas (required surface intersection points with a cutting plan), which will be used as reference points for the next stage (fig. 2a and fig. 2c).

The next algorithm stage is denominated of offset curves, where it's defined an offset curve for each contour determined previously by the last stage [11]. The need to use this type of curves is because the trajectories are applied considering the centre of the tool. As in this case, the selected tool for this milling strategy was a FEM, it is only necessary to establish a value of offset for each one of the trajectories, with the same tool radius, to avoid that the tool moves forward to the inside of the required surface.

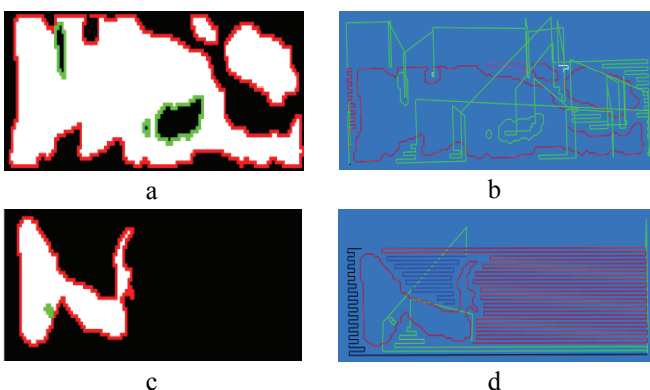


Fig. 2. 2D binary images with the representations of the contours and created trajectories for the respective cutting plan ($S=20$ mm, $FEM_{(diameter)}=15$ mm, $\omega=6$ mm): a - image with Zpc_2 ; b - trajectories with Zpc_2 ; c - image with Zpc_4 ; d - trajectories with Zpc_4 .

The offset curves can be defined inside or outside, depending on the type of contour. In this case it was used red for the outside points and green for the inside points (subcontour).

The offset curve definitions are set according to the parallel and even trajectory segments (with an increasing on the (ω) value, along the secondary axis (fig. 3)). The segments can be established the following way:

- Connects two different contour points of the roughing object;
- Connects a border point of the roughing object and a border point of the offset curve;
- Connects two different points of the outside offset curve points;
- Connects the same two points of inside points of the offset curve.

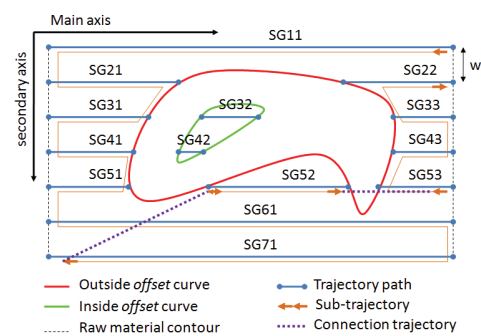


Fig. 3. Design of the created trajectory segments.

The next algorithm stage is grouping the segments in lists of subsegments. Each list is related to the segments formed along the main axis and the information of the previous and following lists. For example, the third list is related to the SG31, SG32 and SG33 segments, according to fig. 3.

To connect the segments of the lists it was chosen to use a *zig-zag* strategy. This strategy establishes connections between the segments of consecutive lines. The result is a set of sub-trajectories that have to be attached before. Yet, the segment connections have to follow some rules:

- They can only segments that can be joint are the ones that have points belonging to the same contour;
- If one segment goes from left to right then the next segment must go the opposite direction;
- The creation of a new subtrajectory must begin from the segment with the least index that hasn't yet been connected;
- Two temporary sub trajectories must be created in order to create a new sub trajectory. One of them starting from left to right and the other one starting from the opposite side. The chosen sub trajectory must be the one that has the larger number of connected segments.

After the sub trajectories are defined, the algorithm proceeds with its bond, defining trajectory connections. The sub trajectory connection begins at the ending point of the first sub trajectory, where the distance of this point is calculated for the starting and ending point of the remaining sub trajectories. The sub trajectory with smaller distance is selected. If the smaller distance corresponds to the starting point, the sub trajectory must be changed. The process repeats until all the sub trajectories are connected.

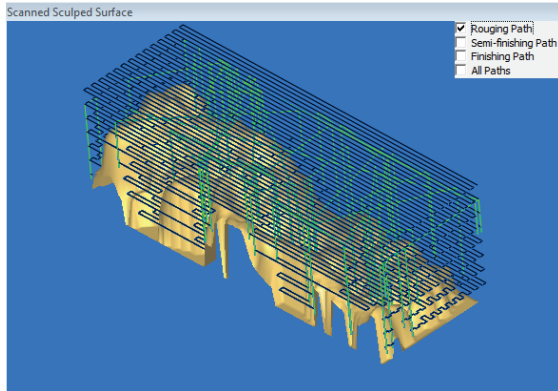


Fig. 4 – Achieved trajectories for the roughing operation with a maximum dimensions surface 170x600x150 mm³ (roughing block's dimensions), using: *z-parallel/zig-zag* strategies, $FEM_{(diameter)}=15$ mm, $\omega=6$ mm e $S=20$ mm.

All connection trajectories start from the sub trajectory's ending point, raises to z with the same value as the roughing block's height, moves to the next sub trajectory with a z constant value and lowers to z until the cutting plan's height that it started with (fig. 2b and fig. 2d).

The final stage of this algorithm is connecting the defined trajectories of each cutting plan to the next cutting plan (fig. 4). The process starts on the further outside cutting plan, following the same connection rules as the sub trajectories.

IV. SEMI-FINISHING AND FINISHING

Semi-finishing and finishing operations share the same functioning rule, using splines to define the surface contour. When the contour is defined by a spline, the trajectory is defined by a mathematical module that incorporates the points that form it, instead of being defined by a set of straight lines and/or arches. This means that a spline is treated like a unique entity, which allows smaller programs.

In this specific case, it was possible to use this mathematical module because the controller used on the testing machine allows the definition of spline interpolation to define the trajectories.

We aware that even though the referred operations share the same functioning rule their finalities are very different. For example, the semi-finishing operation has to be after the roughing operation, which normally involves tools from the BEM type and whose diameter depends on the cutting depth(s) used in the roughing operation.

Therefore, its last objective is removing the material, in order to approximate the resulting surface to the required one, leaving the same layer of material in excess.

The finishing operation must be applied after the semi-finishing operation or immediately after the roughing operation, as long as it has been realized with the small diameter tool and with inside cutting depth (S) with the radius of the used tool for the finishing operation. Similar to what happens in the semi-finishing operation, here it is also used the BEM type of tools, however, the tool's radius must be inferior. The objective of the finishing operation is to create a final soft surface the closest possible to the required surface.

A good approximation may depend on the radius of the chosen tools and on the complexity of the surface to mill. The surfaces softness may be because of the type of trajectory and spacing between the trajectories (ω), the smaller the spacing between the trajectories the better it is.

The developed algorithm for the semi-finishing and finishing trajectory definitions has the following stages: creation of an offset surface, definition of a parallel set of intersection plans with an offset surface, determination of the intersection points for each plan and treatment of the intersection points.

The offset surface (fig. 5b) is created based on the points of the required surface moved along a vector (fig. 5a), with the same direction as the normal vectors (n) and the same dimension as the cutting tools radius (R_{BEM}) selected for the operation (3).

$$\begin{cases} x_{offset}(i) = x(i) + n_x(i)R_{BEM} \\ y_{offset}(i) = y(i) + n_y(i)R_{BEM} \\ z_{offset}(i) = z(i) + n_z(i)R_{BEM} \end{cases} \text{ for: } 0 \leq i < n_points \quad (3)$$

After the offset surface is created, the algorithm defines a set of parallel plans for the plan produced between the main and z axis. Each one of these plans is defined along the secondary axis by the same increases as the ω value (fig. 5c). Each one of the plans is intersected with the offset surface. The intersection result is a set of profile points of the offset surface (fig. 5d).

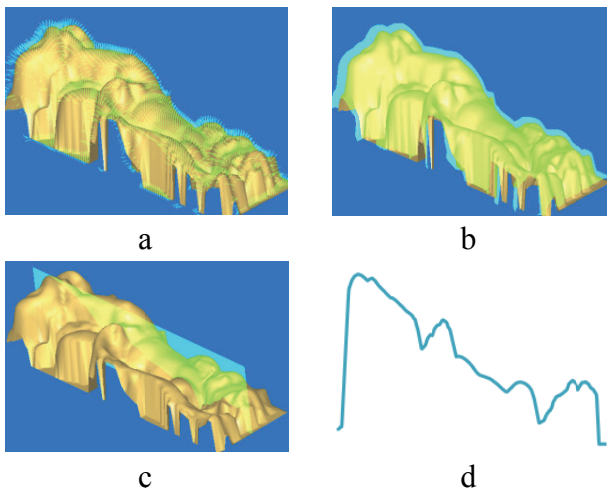


Fig. 5. a - normal vectors of the required surface points; b - offset surface; c - intersection plan; d - points that represent an offset surface profile obtained by the intersection with the intersection plan.

Nevertheless, the produced profile by the obtained points of the last stage may cause some problems. For example, it may make noise, because of some vectors used in the creation of the offset surface if they intersect – this situation may aggravate when the surface is complex or/and the cutting tool diameter is big. Besides, it may still show big changes between the consecutive points – that should be eliminated to soften the trajectory and prevent collisions. The collisions may happen, for example when the profile shows hallowed area with a smaller dimension than the tool diameter.

It was developed a correction filter to solve these problems (fig. 6) that prepares the profile points for the trajectory's mathematical module definition. It's a filter that uses the discrete derivative of the sign (in z) to search for accentuated changes. This way, the value of the partial derivative is compared to the selected tools radius, the selected point is the changing point when the value of the derivatives module is higher than the $BEM_{(radius)}$ value.

When the selection of points is finished, it is attributed the following rejection criteria of profile points: points between two consecutive changing points and points between two non-consecutive points with distances between changing points inferior than the selected cutting tool's diameter.

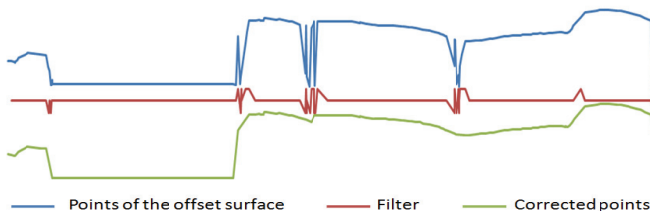


Fig. 6. Illustration of the correction filter used.

The final algorithm stage consists on defining a spline (sub-trajectory) from the resulting points of the correction

filter and establishing the global strategy for the sub trajectory's connection. In this case, it was chosen a strategy called *zig-zag*, where different splines formed are connected by linear movements. This means that the end of each spline is connected to the beginning of the following one and that each spline goes the opposite way to the previous and following spline (fig. 7).

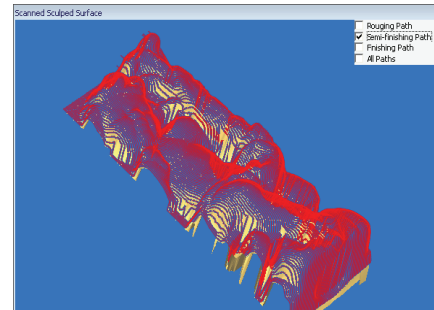


Fig. 7. The achieved trajectories for the finishing operation of a surface with maximum dimensions $170 \times 600 \times 150 \text{ mm}^3$ (roughing block dimensions), using: *zig-zag* strategies, $R_{BEM}=4 \text{ mm}$ and $\omega=1 \text{ mm}$.

V. SIMULATION

The objective for this stage is to virtually simulate, the result of the trajectory applications created for each one of the operations described in the previous points of milling a raw-material block.

Thus, a block with the same maximum dimensions is defined for the required surface. This block is represented through a *quadtree*. Each square of the *quadtree* has associated the value that represents the block height for that position.

The algorithm developed to simulate the effect of the milling block's trajectories adjusts the height value of each *quadtree* square, according to the position and the type of tool.

For the FEM tool (roughing operation), the trajectories are the linear type. To simulate the effect of this type of trajectory it is established in the first place a straight xy line between the two points that define them. Afterwards, that straight line is subdivided in points with an inferior spacing to the *quadtree* resolution. It is later defined for each one of these points the circle equation with its centre on the point and the diameter equal to the selected tools diameter. It is done a search in the *quadtree's* inside points of the circle and with a z value higher than the trajectory points. It is attributed the value of z to the trajectory points found in the *quadtree* points.

For the BEM type of tools (semi-finishing and finishing operations), the trajectories are the spline type. To simulate this type of trajectories we start off by defining a

mathematical module for each spline and calculating there z value. To make these calculations the values of the main axis are evenly spaced, even though with a smaller resolution than the *quadtree* resolution. For each one of these trajectory points of simulation it is established the sphere equation. Based on the established equation, the previous *quadtree* points are selected. At each one of the selected points it is given a new z value. This value is equal to the sphere surface for the xy position of each selected point.

VI. RESULTS AND CONCLUSIONS

Figures 8 and 9 show the simulation results of the roughing and finishing operations, respectively. The results shown for the roughing operation were accomplished using the same type of tool and spacing between trajectories, only varying the cutting depth. The given surface in fig. 8a is close to the required surface, being able to pass directly to the finishing operation, as long as in the operation it is used a tool with a radius higher than the resulting degree from the finishing operation. On the other hand, the surface represented in fig. 8b has very high steps, needing a semi-finishing operation. In these cases it is used a tool with a superior radius, in order to approach a little more the resulting surface to the required one.

The results for the semi-finishing and finishing operations (fig. 9) mainly play with the spacing between trajectories (ω) and look forward to showing the effect of these parameters variation in the softness of the final surface. The value of ω found, for a final surface with a good aspect, was $\omega \leq 0.2R_{BEM}$ (value obtained by the virtual simulation). As for choosing the tools diameter must be associated, for example, to the geometry of the surface to mill and to the detail level that is meant to reach with the finishing operation.

REFERENCES

- [1] Choi, Y. and Banerjee A., "Tool path generation and tolerance analysis for free-form surfaces", *Machine Tools & Manufacture*, Elsevier, 47, 689-696, 2007.
- [2] Omirou, S. and Nearchou, A., "A CNC machine tool interpolator for surfaces of cross-sectional design", *Robotics and Computer-Integrated Manufacturing*, 23, 257-264, 2007.
- [3] Kim, J.; Ryuh, B. and Pennock, G., "Development of a trajectory generation method for a five-axis NC machine", *Mechanism and Machine Theory*, 36, 983-996, 2001.
- [4] Feng, H. and Li, H., "Constant scallop-height tool path generation for three-axis sculptured surface machining", *Computer-Aided Design*, Elsevier, 34, 647-654, 2002.
- [5] Joao L. Vilaca, Jaime C. Fonseca, Antonio M. Pinho, "Calibration procedure for 3D measurement systems using two cameras and a laser line", *Optics & Laser Technology*, 41(2), 112-119, 2009.
- [6] Sheen, B. and You, C., "Machine feature recognition and tool-path generation for 3-axis CNC milling", *Computer-Aided Design*, 38, 553-562, 2006.

- [7] Zhang, L.; Deng, J. and Chan, S., "A next generation NC machining system based on NC feature unit and real-time tool-path generation", *Advanced Manufacturing Technology*, Springer, 16, 889-901, 2000.
- [8] Choi, B. and Jerard, R., "Sculptured Surface Machining Theory and applications", Kluwer Academic Publishers, 147-185, 1998.
- [9] Freeman, H., "Computer processing of line-drawing images", *ACM Computing Surveys (CSUR)*, 6(1), 57-97, 1974.
- [10] Wagenknecht, G., "A contour tracing and coding algorithm for generating 2D contour codes from 3D classified objects", *Pattern Recognition*, Elsevier, 40(4), 1294-1306, 2007.
- [11] Del Savio, A.; Santi, M. and Martha, L., "Traçado de Curvas Offset para Auxílio na Geração de Malhas", XXV CILAMCE – 25th Iberian Latin-American Congress on Computational Methods in Engineering, 2004.

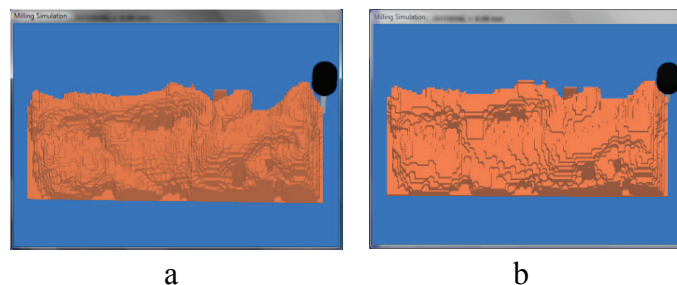


Fig. 8. Roughing operation simulation results: a - $FEM_{(diameter)=10\text{ mm}}$, $\omega=4\text{ mm}$, $S=2\text{ mm}$; b - $FEM_{(diameter)=10\text{ mm}}$, $\omega=4\text{ mm}$, $S=8\text{ mm}$.

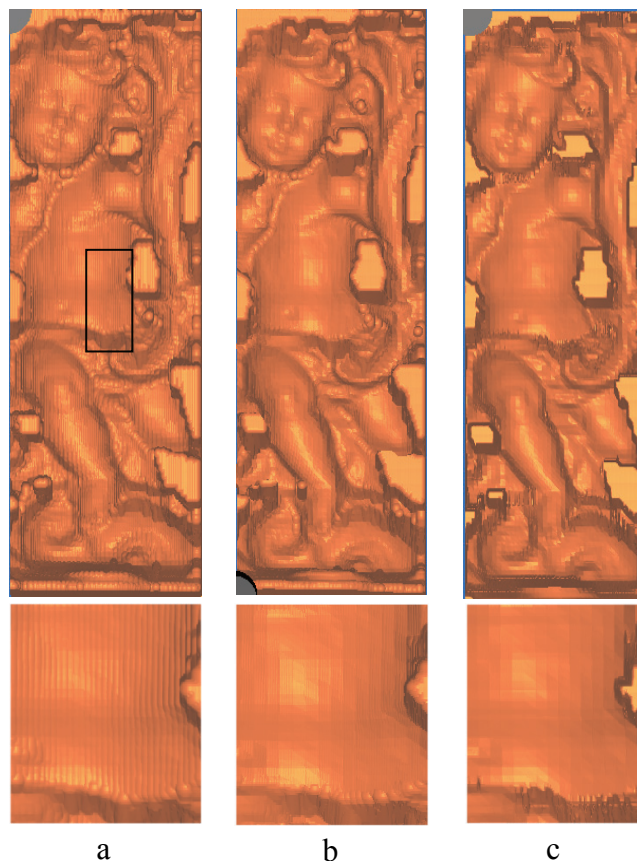


Fig. 9. Semi-finishing and finishing operation simulation results: a - semi-finishing, $R_{BEM}=4\text{ mm}$, $\omega=2\text{ mm}$; b - finishing, $R_{BEM}=4\text{ mm}$, $\omega=0.8\text{ mm}$; c - finishing, $R_{BEM}=1\text{ mm}$, $\omega=0.2\text{ mm}$.