Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/11337

To cite this version :

Anthony SURLERAUX, Samuel BIGOT, Jean-Philippe PERNOT - A comparative study between NURBS surfaces and voxels to simulate the wear phenomenon in micro-EDM - Computer-Aided Design and Applications - Vol. 13, n°6, p.792-798 - 2016

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Arts et Métiers Institute of Technology

A comparative study between NURBS surfaces and voxels to simulate the wear phenomenon in micro-EDM

Anthony Surleraux[1], Jean-Philippe Pernot[2] and Samuel Bigot[2]

[1]Cardiff University, UK; [2]Arts et Métiers ParisTech, France

ABSTRACT

The prediction of the tool wear phenomenon in the micro electro discharge machining technology would be of a great use in the optimization of tool shapes. In order to do so, the ability to rapidly and precisely simulate the process is required. The choice of a geometrical model for the electrodes is of the utmost importance as it will condition the available methods for the simulation.

Two geometrical models have been tested and implemented: NURBS and voxels embedded in an octree data structure. A limited experimental validation has been conducted and the results compared to the simulation through the use of the Hausdorff metric.

KEYWORDS

Shape modeling and optimization; continuous vs. discrete representations; surface warping; Haussdorf metric

1. Introduction

Micro Electrical Discharge Machining (micro-EDM) is a micro-manufacturing process that involves the removal of material through electrical discharges. Its main strength resides in the fact that it is able to machine any electrically conductive material independently of its hardness [9]. The process involves two electrodes (one being referred to as the tool while the other is the workpiece) that are submitted to an electrical current and immerged in a dielectric fluid. As the distance separating both electrodes decreases, there is a point (known as the machining gap) where the dielectric fluid isn't able to insulate both electrodes from each other anymore and breaks down leading to the apparition of a plasma channel between both electrodes. The resulting thermal energy leads to melting and vaporization of the material on both electrodes and the removal of material as craters. A visual description of this process is available in Fig. 1.

The main issue is that material is also removed from the tool (tool wear) and therefore influences the final shape obtained on the workpiece. The influence of tool wear is significantly greater when tackling with micro-scale features and can be observed in Fig. 2. While proven methods exist for micro- EDM milling [2,9,10] to compensate for the tool wear, die-sinking micro-EDM would require the use of multiple tools to obtain the desired geometries.

Most previous efforts have been focused on the modeling of single-spark discharges through the solving of thermal equations and specific boundary conditions. A few [1,4,8] have proposed simulation methods of micro-EDM usually using a grid of points to model the involved geometries. A recent work [5] has tackled with the geometrical simulation of a whole process through the use of Z-maps but those are inherently limited when it comes to representing overhangs.

The main idea is to develop a geometrical simulation method with the final objective of optimizing the initial shape of the tool by adding material on it to counteract the effects of tool wear. In order to do so, the simulation must be fast considering that the optimization method is likely to require at least several iterations. Actually, in a later stage, this method will take part in a shape optimization process where tens of simulations will be performed (Fig. 3).

2. The simulation tool

2.1. Introduction

Section 1 has given an overview of the micro-EDM process and the elements that are involved. What is proposed in the present paper is to represent the electrodes' geometries with:

- A representation of their boundaries with NURBS
- A volumetric representation with voxels.

The main aspects of the simulation, despite the differences in the proposed geometric representations, are

CONTACT Samuel Bigot bigots@cardiff.ac.uk

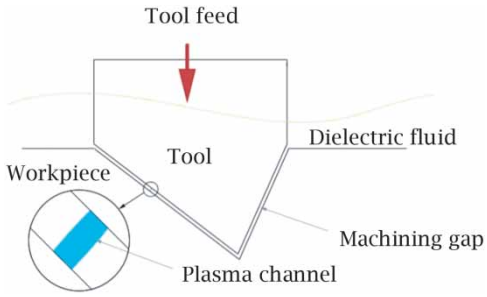

Figure 1. Basic principle of micro-EDM.

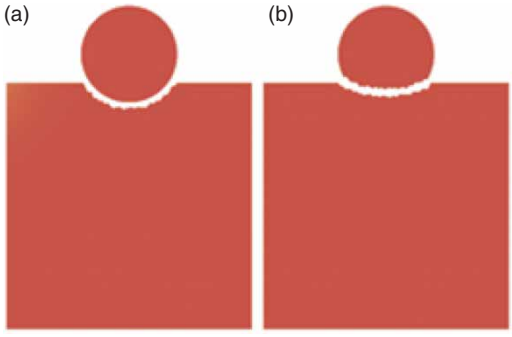

Figure 2. The influence of tool wear in micro-EDM. a) Desired result. b) Obtained result.

the same in both cases. It is an iterative process that ends when the desired tool movement is achieved. In the particular case of die-sinking micro-EDM, it is usually a simple vertical movement made by the tool until it reaches a depth name the objective depth, D_{obj}.

An iteration starts with the search for the locations of the craters. The main assumption that is being made is that the sparks takes the path of least resistance and therefore occurs at the minimum distance between both geometries. The distance search algorithm returns d, the minimum distance as well as P_t and P_w, the corresponding points.

The minimum distance value is then compared to another value known as the machining gap (defined in Fig. 1), M_g to determine if a spark can occur. Should the minimum distance be smaller than M_g, it is possible for a spark to appear and craters to be formed.

The apparition of a spark consists in volume removals for the voxel method and surface deformations for the NURBS method in order for predetermined volumes V_t and V_w to be removed on the tool and the workpiece, respectively. If no spark can take place, the tool is move down with a small increment Δ_z.

The process is repeated until a total vertical movement D_{obj} is achieved. A visual representation of the global algorithm is given in Fig. 4.

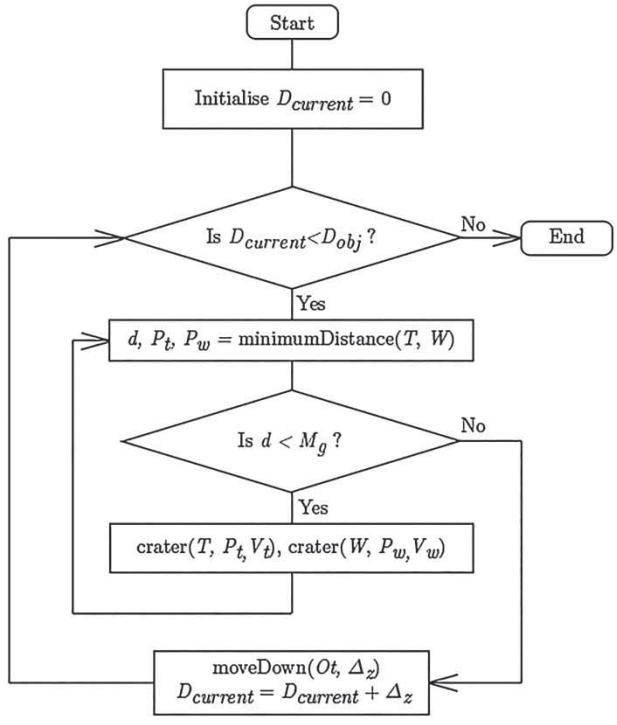

Figure 4. Global overview of the simulation process.

2.2. The NURBS method

2.2.1. Introduction

In the NURBS method, both electrodes are modeled as single NURBS patches. The patches are then refined

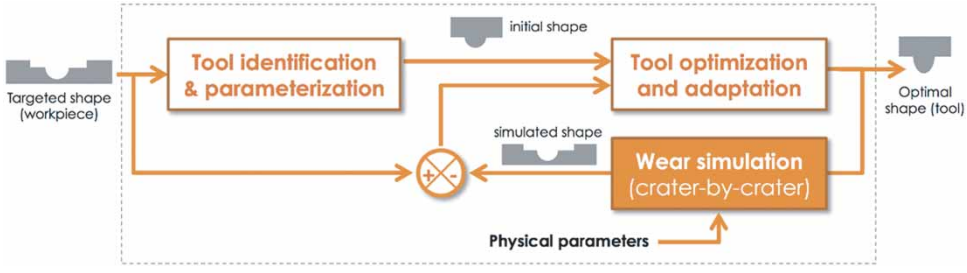

Figure 3. The crater-by-crater simulation module plugged within a shape optimization loop.

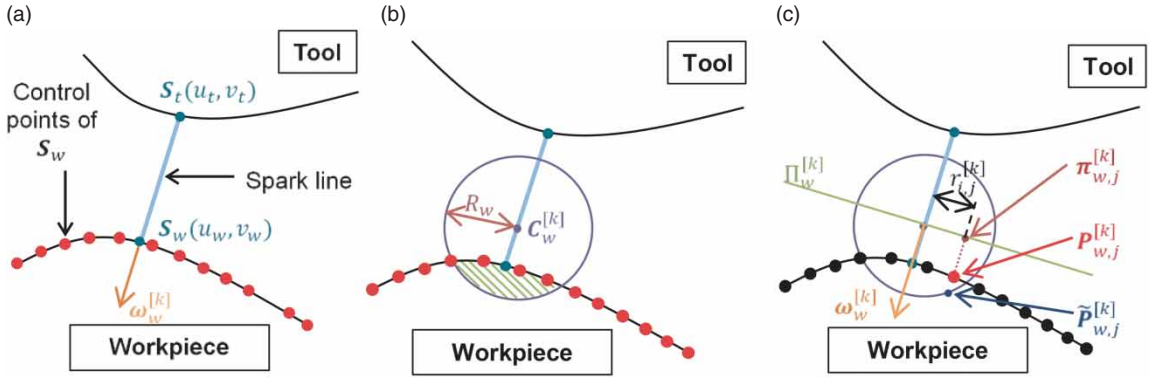

Figure 5. Definition of the various elements involved in the crater insertion process.

using the Boehm's knot insertion algorithm [7]) in order to obtain a million (1000×1000) control points for each patch and, therefore, to increase local control. The crater insertion method is illustrated by Fig. 5. and implements the surface warping technique.

For the current depth, the minimum distance search is done with the use of Particle Swarm Optimization (PSO) with four variables that are the NURBS patches parameters [6]. The output gives two points $\boldsymbol{S_t}(u_t,v_t)$ *and* $\boldsymbol{S_w}(u_w,v_w)$.

2.2.2. Crater insertion

Then, it is supposed that, during the manufacturing process, all sparks transfer the same energy therefore removing the same volumes V_t and V_w for each iteration. Those volumes are computed by modeling the craters as spherical caps and experimentally measuring the mean depth D_t^m, D_w^m and radius R_t^m, R_w^m of craters.

$$V_i = \pi \cdot \frac{R_i^m}{6}[3R_i^{m2} + D_i^{m2}], i \in \{t, w\} \tag{2.1}$$

Furthermore, this leads to the definition of what is known as the spherical caps' support spheres which have a radius R_i.

$$R_i = \frac{R_i^{m2} + D_i^{m2}}{2D_i^m} \tag{2.2}$$

If the minimum distance condition described previously is met (the distance is smaller than the machining gap), the crater insertion technique is applied. A crater will be inserted on each surface $\boldsymbol{S_i}$ $i \in \{t, w\}$).

The first step is to compute the displacement directions given by two warping unit vectors that are given in Eq. 2.3.

$$\boldsymbol{\omega}_i^{[k]} = \mathrm{sg}(i) \cdot \frac{\boldsymbol{S}_t(u_t, v_t) - \boldsymbol{S}_w(u_w, v_w)}{\|\boldsymbol{S}_t(u_t, v_t) - \boldsymbol{S}_w(u_w, v_w)\|} \tag{2.3}$$

with $\mathrm{sg}(i) = \begin{cases} 1 & \text{for } i=t \\ -1 & \text{for } i=w \end{cases}$ and k values refer to the kth crater inserted on each surface. Note that the superscript k has been omitted for the u_i values for the sake of clarity.

Fig. 5(a) shows the warping vector $\boldsymbol{\omega}_i^{[k]}$ corresponding to the case where the workpiece is to be deformed. The figure gives a two-dimensional version of the process to simplify the representation.

The second step consists in identifying the control points in the neighborhood of $\boldsymbol{S}_i$ (u_i, v_i) that need to be moved. In order to do so, the support sphere's center $\boldsymbol{C}_i^{[k]}$ has to be moved along the spark line so that the intersection volume between the sphere and the NURBS patch is equal to the volume V_i to be removed within a certain tolerance. (Fig. 5(b)). The search for this location is made through the use of an iterative dichotomy method (also known as binary search or bisection method). At each step, the intersection volume (the hashed part of Fig. 5(b)) is computed. If the volume is too small (inferior to (1-T_v)*V where T_v is a tolerance parameter), the sphere is moved towards the surface. In the opposite case (it is greater than (1+ T_v)*V, it is moved away. The process stops when a volume within 1% of V_i is found.

Once the $\boldsymbol{C}_i^{[k]}$ adequate positions are found, it is possible to determine the $N_i^{[k]}$ control points of the two $\boldsymbol{S}_i$ surfaces that need to be moved. This is done by computing for each control point the distance that separates them from the center of the sphere. If the distance is smaller than the radius of the sphere, the control point is added to the list of points to be displaced. At the end, two lists of control points are obtained.

In order to displace the control points to mimic the shape of a sphere, a reference is needed. Let $_i^{[k]}$ be the plane that includes the center of the sphere $\boldsymbol{C}_i^{[k]}$ and that has $\boldsymbol{\omega}_i^{[k]}$ as normal vector. Then, for all the control points, $\boldsymbol{P}_{i,j}^{[k]}$, $j \in \{1, \ldots, N_i^{[k]}\}$ and $i \in \{t, w\}$, the new position are computed as follows (Fig. 5(c)):

$$\tilde{\boldsymbol{P}}_{i,j}^{[k]} = \boldsymbol{P}_{i,j}^{[k]} + f_i^{[k]}(r_{i,j}^{[k]}) \cdot \boldsymbol{\omega}_i^{[k]} \tag{2.4}$$

with $f_i^{[k]}(r) = \sqrt{R_i^2 - r^2} - \boldsymbol{\omega}_i^{[k]}.(\boldsymbol{P}_{i,j}^{[k]} - \boldsymbol{\pi}_{i,j}^{[k]})$ and $r_{i,j}^{[k]} = \|\boldsymbol{\pi}_{i,j}^{[k]} - \boldsymbol{C}_i^{[k]}\|$ being $\boldsymbol{\pi}_{i,j}^{[k]}$ the projection of $\boldsymbol{P}_{i,j}^{[k]}$ on the plane $\Pi_i^{[k]}$

This process is repeated iteratively until no more craters can be inserted for the actual depth. Then, the tool is moved down along the $\boldsymbol{z}$ axis with an increment of Δ_z and the craters insertion process starts again.

2.3. The voxels method

2.3.1. Introduction

Voxel stands short for "volumetric pixel" and, as the name implies, can be considered as being the three-dimensional equivalent of pixels. As mentioned before, they are used for the representation of three-dimensional elements. The most common approach is to set them in a uniform grid aligned with the coordinate axes. Although some recent developments have increased their use in the computer graphics industry, their discrete nature have made them unsuitable for the representation of detailed scenes. Consequently, voxels have mostly been used in the visualization of volumetric data such as in the medical imagery field.

The voxels in this method are embedded in an octree structure (Fig. 6.) and used to model both electrodes in a volumetric and discrete manner. The octree provides with the ability to represent large sections of the volumes as bigger voxels and therefore reduces memory usage. Additionally, it provides with a hierarchical structure that is useful when querying a specific voxel. An octree is composed of nodes that have a parent node and up to eight children nodes. A node with no children is said to be a leaf node. The topmost node in the structure is called the root node.

2.3.2. Minimum distance search

This section deals with the search of the minimal distance between two disjoint octrees. In addition, both of those are axis-aligned. The presented method is based on [3] with the addition of some fast exit conditions.

The smallest distance d between two octree nodes can be bounded by two values such as:

$$d_{lower} \leq d \leq d_{upper} \tag{2.5}$$

In the situation where both nodes are leaves (they don't contain any children), equation 1 becomes:

$$d_{lower} = d = d_{upper} \tag{2.6}$$

Additionally, in the case where one node is a leaf but the other isn't, the calculation of the maximum distance

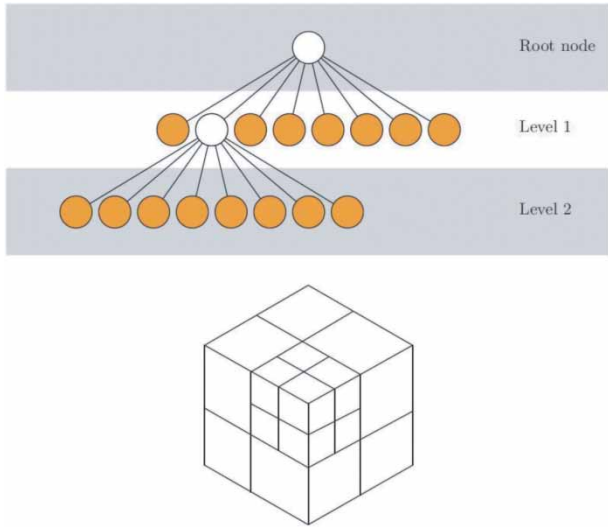

Figure 6. The tree view of an octree and its three-dimensional representation. The leaves are colored.

between them can be modified to take that fact into account and further refine the value of d_{upper}.

The search algorithm starts with the two octrees root nodes and, for each non-empty children, compares the minimum and maximum distance of each voxel couple. Those distances are easy to compute since the nodes are axis-aligned.

Node pairs that are still viable candidates for the minimum distance couple are kept and subdivided at the next iteration.

The viability condition is based on a global value, the smallest of upper bounds or supremum, $d_{smallestUpper}$. Any couple that has a lower bound distance equal or smaller to that value is kept regardless of the minimum distance. The viability condition is formally written as:

$$d_{lower} \leq d_{smallestUpper} \tag{2.7}$$

The process goes on until the voxels have been subdivided to a size s_{res}, a parameter of the algorithm. Since the algorithm relies on a bounding of the minimal distance, there is no guarantee that, after the number of iterations required to reach s_{res}, only a single couple is returned.

Additionally if all couples of candidate nodes are leaves, then, given the way the candidate nodes are culled, it stand to reason that all of those couples have the same d_{min}. At this stage, the algorithm may be terminated and a pair of nodes can be selected at random without having reached a resolution of s_{res}.

Two other early stop conditions can be added specifically for the present problem: in this case it doesn't matter whether the smallest distance is found.

Firstly, the returned result must only comply with the condition that it is smaller than the machining gap.

If at any iteration of the search, a couple with a $d_{upper} \leq M_g$ than the algorithm can be stopped and that couple returned. A couple with an actually lower d_{min} that has been ignored in that way will be used in another iteration of the simulation.

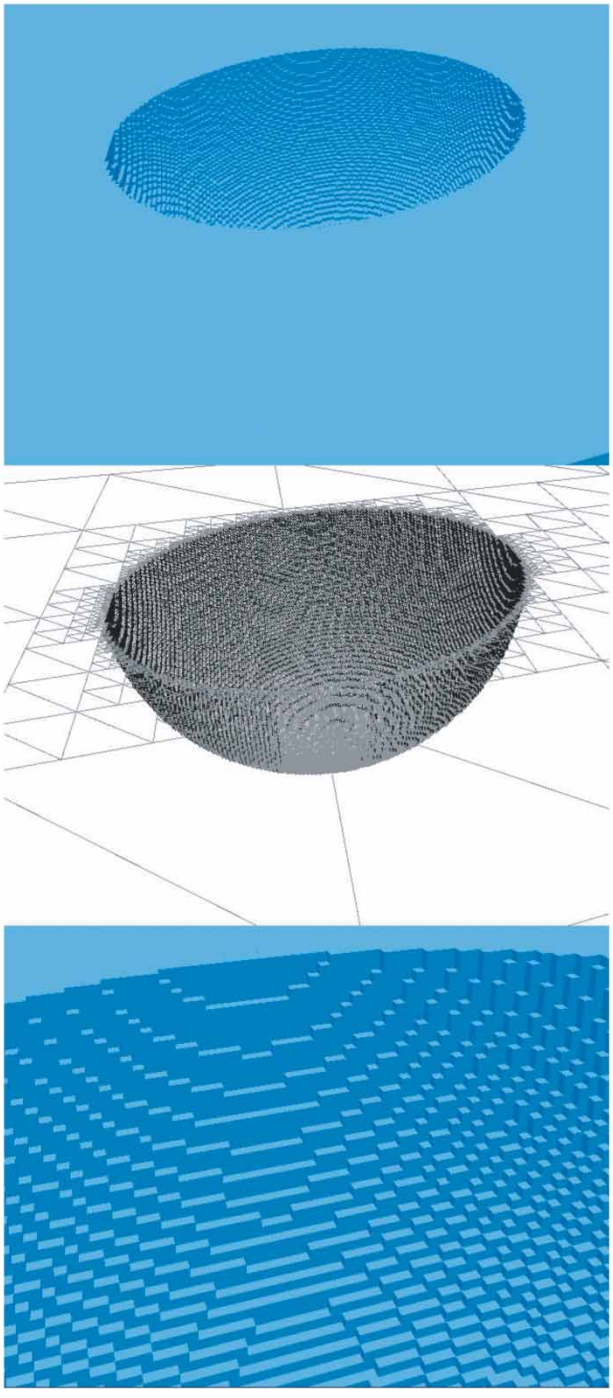

Figure 7. Single crater at a resolution of one voxel per 125 nm.

Secondly, if at any iteration of the search, the list of candidate couples doesn't contain any couple with a $d_{lower} \leq M_g$, the process can be stopped and return null.

Another matter needs to be handled. Since the algorithm might return a list of couples that aren't necessary leaves (in the case where the algorithm exits after having reached a certain number of iterations), a couple is chosen at random. The algorithm is then re-applied on the selected couple and forced to exit after having found only leaf nodes.

Using a Particle Swarm Optimization Method has been tested but has been found to give worse results.

2.3.3. Crater insertion

Once the closest nodes have been identified, the crater insertion method on an octree starts with the root node. The children are tested for intersection with a sphere defined in the same way as in section 2.2: the support sphere is displaced along the spark line until a tolerance condition on the volume to be removed is met.

Similarly to the distance algorithm, the volume removal algorithm searches the octrees in a top-down manner keeping only nodes that intersect with the sphere and deleting those that are inside. The intersection test is made by finding the smallest cubed distance d between a cube and a point and comparing it to Ri^2. The intersection test returns true if:

$$d \leq R_i^2 \tag{2.8}$$

Fig. 7. shows an example of a single crater with the octrees data structure visible underneath and a detail of the voxels at the bottom. One voxel's dimension is 125 nm.

3. Experimental validation and comparative study

3.1. The experiment

In order to assess the performances of each method, an experiment was devised in order to measure the shape differences between simulated and experimental results. The used machining parameters are displayed in Tab. 1.

Table 1. Used machining parameters.

Parameter	Value
Energy level (index)	200
Voltage (V)	90
Current (index)	20
Time on (ms)	5
Objective depth (µm)	100

The experiments were conducted using a Sarix SX-200 micro-EDM machine. The tools were made of tungsten

carbide while aluminium was used for the workpiece. A depth of 100 μm is set for the machining operation. This depth was selected in order to provide a noticeable wear on the tool without completely destroying the features on it.

The tool that was used is of a spherical shape as visible in Fig. 7(a). and it has been obtained through wire-dressing of a cylindrical electrode of nominal diameter 300 µm. The nominal diameter of the sphere is 250 µm.

This tool was measured with the help of a micro-tomographer of a resolution of 1 µm. The resulting workpiece was exported as a three-dimensional mesh. (Fig. 7(c)) The tool was also measured in the same manner after the experiment (Fig. 7(b)).

The machining parameters were:

The craters were also measured using the 3D meshes and the results are tabulated in Tab. 2.

Table 2. Experimental average craters dimensions.

Parameter	Value
Workpiece crater diameter (µm)	13.30
Workpiece crater depth (µm)	4.42
Tool crater diameter (µm)	12.40
Tool crater depth (µm)	4.39

3.2. The simulations

The NURBS elements used were a sphere of diameter 250 µm for the tool and a flat square surface with a side length of 500 µm for the workpiece as shown in Fig. 8(a).

The voxel tool was created from the micro-tomographer STL files as shown in Fig. 8(b). The workpiece is a cube of dimension 512 µm. Contrary as to what is depicted in the figure, only the tip of the spherical shape was kept in order to reduce the octree's memory footprint. Those two models do represent the same extremity of the tool using a sphere-like shape. The various tolerances used are shown in Table 3.

Table 3. Tolerances used in the simulations.

Parameter	Value
NURBS volume removal (in %)	1
Voxel volume removal (in %)	1
Voxel crater resolution (voxels per micron)	4
Voxel distance precision (voxel size)	0.25

The NURBS simulation ended after 2040 minutes while the voxels simulation took 126 minutes and is therefore more than 16 times faster.

After the simulation, the resulting models (NURBS and voxels) were compared to the experimental ones with the use of the Hausdorff metric. The Hausdorff metric results are given in Tab. 4, including the minimum, maximum, mean and RMS values of each set of sampled points' Hausdorff distance.

Table 4. Hausdorff metric results.

	NURBS		Voxels	
	Tool	Workpiece	Tool	Workpiece
d_H min (µm)	0.000000	0.000107	0.008184	0.000000
d_H max (µm)	8.629291	14.886533	2.163380	14.917241
d_H mean (µm)	1.449477	3.073571	0.858142	0.718315
d_H RMS (µm)	2.521132	3.626015	0.971600	1.682061

Fig. 9. depicts the maps of the calculated Hausdorff distances. A red color represents a small difference between the experimental and simulated elements while a blue color indicates a larger difference.

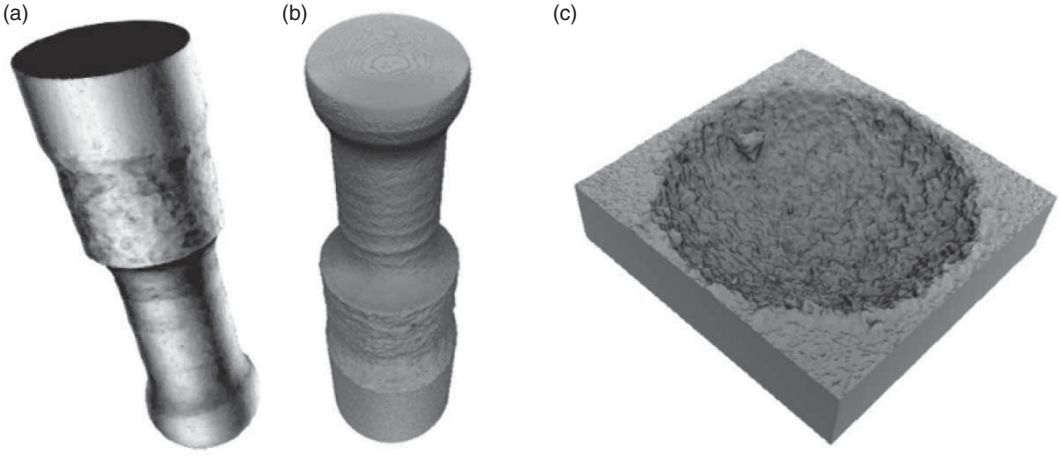

Figure 8. 3-D mesh of the experimental tool. a) before and b) after machining. c) workpiece after machining.

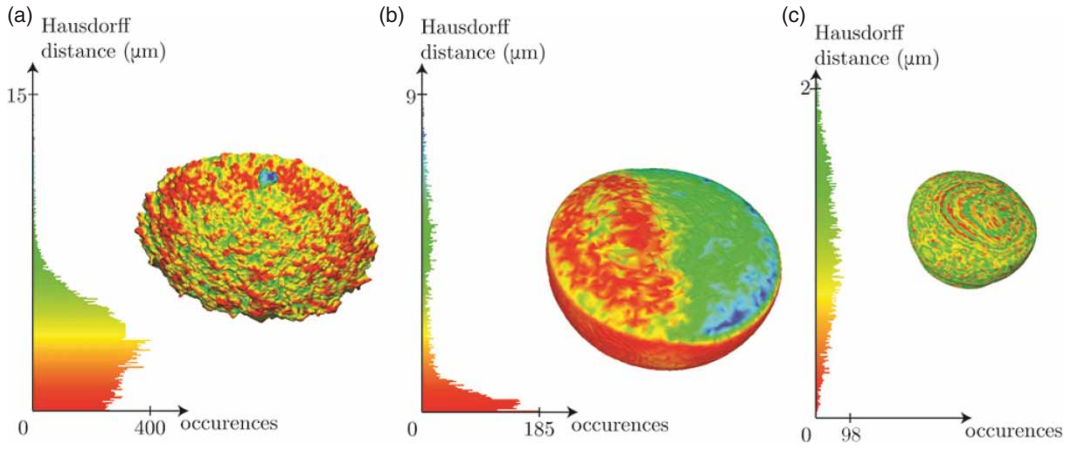

Figure 9. Hausdorff distance maps depicts the maps of the calculated Hausdorff distances. A red color represents a small difference between the experimental and simulated elements while a blue color indicates a larger difference.

4. Conclusions

Considering the Hausdorff distance RMS values, both methods offer a good prediction of the experimental results with values in the range of 2–3 µm for the NURBS method and 1–2 µm for the voxel method. Those values are to be compared with the dimensions of the models (bounding box diagonal of 403 µm for the workpiece and 366 µm for the tool).

While the NURBS method offers geometries with no sharp edges, the voxel method is significantly faster and therefore more efficient when used as part of a shape optimization loop.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [EP/F056745/1, EP/J004901/1]. Information on data that supports the results presented in this article, including how to access them, can be found in the Cardiff University research-data catalogue at http://dx.doi.org/10.17035/d.2016.0008385063.

ORCID

Anthony Surleraux http://orcid.org/0000-0002-2101-9259
Jean-Philippe Pernot http://orcid.org/0000-0002-9061-2937
Samuel Bigot http://orcid.org/0000-0002-0789-4727

References

[1] Bigot, S.; Surleraux, A.; Bissacco, G.; Valentinčič, J.: A New Modelling Framework for Die-Sinking Micro EDM, Proceedings of the 9th International Conference on Multi-Material Manufacture, 2012, 51–55. http://dx.doi.org/10.3850/978-981-07-3353-7_309

[2] Bleys, P.; Kruth, J.-P.; Lauwers, B.; Zryd, A.; Delpretti, R.; Tricarico, C.: Real-time Tool Wear Compensation in Milling EDM, *Annals of the CIRP*, 51(1), 2002, 157–160. http://dx.doi.org/10.1016/s0007-8506(07)61489-9.

[3] Borrmann, A.; Schraufstetter, S.; van Treeck, C.; Rank, E.: An Iterative, Octree-Based Algorithm for Distance Computation Between Polyhedra with Complex Surfaces, Proceedings of the International ASCE Workshop on Computing in Civil Engineering: 2007, 103–110. http://dx.doi.org/10.1061/40937(261)13.

[4] Heo, S.; Jeong, Y.H.; Min, B.-K.: Virtual EDM Simulator: Three-Dimensional Geometric Simulation of Micro-EDM Milling Processes, *International Journal of Machine Tools and Manufacture*, 49(12–13), 2009, 1029–1034. http://dx.doi.org/10.1016/j.ijmachtools.2009.07.005

[5] Heo. S.; Kim, T.-G.; Lee, C.-H.; Min, B.-K.; Lee, S.J.: Geometric Simulation Based Tool Wear Compensation in Micro Die-Sinking EDM Process, Poster presented at the 29th Annual Meeting of the American Society for Precision Engineering, 2014, Boston, MA.

[6] Kennedy, J.; fEberhart, R.: Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks IV, 1995, 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968

[7] Piegl, L.; Tiller, W.: *The NURBS Book*, Springer, 1997. http://dx.doi.org/10.1007/978-3-642-59223-2

[8] Tricarico, C.; Delpretti, R.; Dauw, D.F.: Geometrical Simulation of the Die-Sinking Process, *Annals of the CIRP*, 37(1), 1988, 191–196. http://dx.doi.org/10.1016/S0007-8506(07)61616-3

[9] Yu, Z.Y.; Masuzawa, T.; Fujino, M.: Micro-EDM for Three- Dimensional Cavities Development of Uniform Wear Method, *Annals of the CIRP*, 47(1), 1998, 169–172. http://dx.doi.org/10.1016/s0007-8506(07)62810-8

[10] Yuzawa, T.; Magara, T.; Imai, Y.; Sato, T.: Micro Electric Discharge Scanning Using a Mini-Size Cylindrical Electrode, *Kata Gijutsu*, 12(8), 1997, 104–105.