AVALIAÇÃO DA RUGOSIDADE SUPERFICIAL BASEADA NO MODELO DE REPRESENTAÇÃO DISCRETA PARA PEÇAS DE MANUFATURA ADITIVA

SURFACE ROUGHNESS ASSESSMENT BASED ON DISCRETE MODEL REPRESENTATION FOR ADDITIVE MANUFACTURED PARTS

ОЦЕНКА ШЕРОХОВАТОСТИ ПОВЕРХНОСТИ НА ОСНОВЕ ДИСКРЕТНОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДЛЯ АДДИТИВНЫХ ДЕТАЛЕЙ

OREKHOV, Alexander A.[1]; RIPETSKIY, Andrey V.[2*]; FEDOSEEV, Denis V.[3]

[1] Moscow Aviation Institute (National Research University, Department of Advanced Materials and Technologies of Aerospace Application, 4 Volokolamskoe shosse, zip code 125993, Moscow – Russian Federation (phone: +74855296100)

[2] Moscow Aviation Institute (National Research University, Department of Engineering Graphics, 4 Volokolamskoe shosse, zip code 125993, Moscow – Russian Federation (phone: +74855296100)

[3] PJSC UEC-Saturn, 163 Lenin Ave., zip code 152903, Rybinsk – Russian Federation (phone: +74991584819)

** Corresponding author*
e-mail: a.ripetskiy@mai.ru

Received 27 June 2018; received in revised form 21 November 2018; accepted 05 December 2018

RESUMO

Apesar de método de fabricação de peças, sua superfície não pode ser completamente suave, perfeita. Há sempre irregularidades mais ou menos consideráveis de várias formas e alturas - vestígios de moldes de fundição, rolos de dimensionamento, ferramentas de corte, etc. A altura, a forma e a localização dessas irregularidades dependem de vários fatores e condições relacionados às propriedades do material, tecnologia de processamento, velocidade de corte e qualidade de ferramentas. O surgimento de novos desenvolvimentos em ciência de materiais e suas capacidades utilizadas no método da produção de aditivos em fases (AM) contribui para o desenvolvimento de soluções inovadoras que aumentam a eficiência da produção de peças complexas e melhoram a qualidade. Este artigo descreve um dos métodos (hipótese) para estimar a rugosidade, que pode ser calculada durante a preparação do processo de impressão 3D. O cálculo da rugosidade da superfície é baseado num modelo 3D de um voxel criado a partir de um modelo poligonal (STL).

Palavras-chave: *rugosidade superficial, avaliação, produção de aditivos, modelo discreto, representação.*

ABSTRACT

With any method of manufacturing parts of their surface cannot be completely smooth, perfect. There are always more or less pronounced irregularities of various shapes and heights — traces of the casting mold, sizing rollers, cutting tools, etc. The height, shape and location of these irregularities depend on a number of factors and conditions related to the material properties, processing technology, cutting speed, quality of tools. The emergence of new developments in the field of materials science and their possibilities used in layer-by-layer additive manufacturing (AM) method promotes the development of innovative solutions that increase the efficiency of producing complex parts and improves the quality. This paper describes one of the method (hypothesis) for roughness assessment that can be calculated during preparation of the 3D printing process. The calculation of the roughness surface is based on the voxel 3D model generated from the polygonal model (STL).

DOI: 10.52571/PTQ.SI1.2018.531_PeriodicoES.pdf

Keywords: *CAD model, polygonal model (STL), selective laser melting (SLM), selective laser sintering (SLS), voxel 3D model.*

АННОТАЦИЯ

При любом способе изготовления деталей их поверхности не могут быть абсолютно гладкими, идеальными. Всегда имеются более или менее выраженные неровности различной формы и высоты - следы литейной формы, прокатных вальцов, режущих инструментов и т.д. Высота, форма и расположение этих неровностей зависят от ряда факторов и условий, связанных со свойствами материала, технологией обработки, скоростью резания, качеством инструментов. Появление новых разработок в области материаловедения и их возможностей, используемых в методе поэтапного производства присадок (AM), способствует разработке инновационных решений, повышающих эффективность производства сложных деталей и улучшающих качество. В этой статье описывается один из методов (гипотеза) для оценки шероховатости, который можно рассчитать во время подготовки процесса 3D-печати. Расчет поверхности шероховатости основан на 3D-модели вокселя, созданной из полигональной модели (STL).

Ключевые слова: *CAD модель, полигональная модель (STL), селективное лазерное плавление (SLM), выборочное лазерное спекание (SLC), 3D модель воксель.*

INTRODUCTION

The evaluation of surface roughness and its assessment is important at any stage of the traditional design and manufacturing process. It becomes even more difficult when we move into additive manufacturing (AM) methods. By performing layer-by-layer deposition of the metal powder material particle and further hardening process it gives the engineer not only a new opportunity of creating a very complex structure not possible to manufacture in the traditional way but also restrict AM process (Kuznetsova *et al.*, 2015; Lurie *et al.,* 2017; Davydov *et al.*, 2013). The user can manufacture complex geometrical forms such as engine block with internal cavities and channels. But there are some restriction that needs to be taken into account, such as; part location (composition) in the volume printer (Rabinskiy *et al.*, 2017a; Rabinskiy *et al.*, 2017b), layer thickness, diameter and power source positional accuracy, quality of metal powder etc. All those restrictions influence the quality of the final product with expected surface roughness.

In this paper, one of the variant (hypothesis) of product roughness assessment and calculation is analyzed. This method is intended to be used for the layer-by-layer deposition process. Product roughness assessment can be achieved through comparison of the section profile (cross-section) taken from the surface of the manufactured (printed) model with the surface from the original CAD model (Lich, 2012; Filatov *et al.*, 2009; Valetov and Filimonova, 2011).

MATERIALS AND METHODS

Current research allowing us to assess surface roughness is based on discrete model representation supported in software use in additive manufacturing in form of consistent triangulated 3D model representation. The aim of the project is to find new criteria that can help to achieve high quality prints were we calculated roughness will be the same as measured from the physical part. Presented method is suitable for lases melting and sintering processes (Anukhin, 2012; Kochetkov *et al.*, 2009; Kaladze and Shaposhnikov, 2010).

The roughness assessment should be integral part of any AM processes were the surface quality of the final product is critical (aerospace applications) and other methods of the surface improvement are not possible to apply. Reducing number of fault prints saves time and reduce the costs of entire AM processes, making them environmental friendly.

RESULTS AND DISCUSSION:

AM process requires 3-D polygonal mesh model. The mesh (shape of the printed part) is defined as a set of triangles. Surface roughness is considered typically to be the high-frequency, short-wavelength component of a measured surface represented as a set of tiny irregularities with relatively small distance from baseline. To specify baseline "L" (Figure 1) we may select two points **A** and **B** from the analyzed surface model.

The base length **L** should be taken as a line joining, **A** and **B** along the shortest way over the model surface. The additive manufacturing process can be represented as a filling a volume with small sample volumes, with the size determined to be the resolution of the printer. Those small volumes can be called as voxels. In this paper, we will consider small volumes as cubic voxels. In the filling volume process, there are no empty areas between voxels. As an illustration, we show the example of Stanford rabbit model (Stanford Bunny), on Figure 2. Figure 2 shows the voxel model generated from the polygonal model. Voxel model is re-scaled for illustrative purposes. In reality, the voxel size is significantly smaller.

In order to build the voxel model from polygonal model representation the following steps were taken; minimum volume surrounding original mesh (axis-aligned bounding box or AABB) was created, parallel to the axis, limiting any geometrical object in space to three-axis (Minimum bounding box; Gorshkov *et al.*, 2003; Rabinskiy and Zhavoronok, 2018; Ripetskiy *et al.,* 2016). Increase smaller dimensions of the box so, that there would be a cube with S side. This cube volume will fully include original mesh. Use octree (Octree) will give us voxel model of mesh surface. The volume is broken into eight sub-volumes with three mutually perpendicular surfaces, parallel to the axis of reference. In the same manner, each of sub-volumes is divided into sub-volumes (Odinokova *et al.*, 2013; Gawlik *et al.*, 2014; Paz *et al.*, 2016). The process is repeated up to the achievement of any sub-volume minimum size (Figure 3).

Let's analyze the voxel size used to generate a voxel model. As mentioned above, for simplicity we may assume that voxel is a cube with $\mathbf{S}_v$ side. By increasing volume S, we can get the size of the voxel $\mathbf{S}_v$ required for a proper surface model approximation when building octree (Equation 1). **n** – the number of voxels on the most detailed tree level (Equation 2); **S** – new volume size for octree building.

During the process of building octree, we will include in the voxel structure only those joints in the tree were volume are intersects at least with one triangle of the original mesh. Figure 4 shows the example of the flat quad tree model joins with one triangle.

The algorithm of octree building can be described with the following code:

```
For each mesh triangular
{
intersect_triangular_with_volume
 {
 Does triangular_intersect_volume?
 yes:
 {
 Is volume_equal to_voxel?
                      yes:
                     {
 add_joint_in_the tree
          }
no:
 {
 break_volume_in_8_parts
 for each part:
 intersect_triangular_with_volume
 }
 }
 no:
 {
 Transfer to next volume
 }
 }
}
```

The algorithm searching triangular intersection within the limited volume is based on (The algorithm searching triangular intersection). After octree generation is finished we will get a set of voxels, where each of them will intersect with the original mesh surface. To display the voxel tree, you need to associate each polygonal mesh cube with mesh voxel. The cube mesh will contain 12 triangles (6 faces of 2 triangles). The set of voxel meshes is a polygonal model of the voxel representation of the product. Such a model is suitable for visualization, but it is not good for carrying out calculations because it contains internal triangles that do not belong to the mesh surface.

To calculate the roughness of the surface, we are primarily interested in the outer surface of the mesh. To discard triangles that do not belong to the surface, we need to bypass all three levels (nodes that do not contain descendants - voxels) and analyze 6 neighboring nodes of the tree according to the number of faces of the voxel. If there is no node on either side of the voxel, then this voxel face belongs to the inner or outer surface of the mesh. Triangles of this face need to be included in the mesh. If there is a neighboring node on the side of the face, then the face is entirely inside the mesh and triangles of the face must be excluded from the mesh. Having skirted the tree in this way, we can make a mesh only from triangles belonging to the surface of the model. The following Figure 5 illustrates the definition of the base length.

In Figure 5 from point **A** on the surface of the mesh a line is drawn to point **B**. The line is drawn along the shortest path. To determine the shortest path we draw through the points **A** and **B** a plane. The line of intersection by the plane of the mesh surface between points **A** and **B** will be the line defining the base length. To draw a plane through two points, we need to choose a vector lying in this plane. We choose a vector equal to the half-sum of the vectors of the normal at the points, respectively (Equation 3).

N is a vector on section plane. The vector normal is defined as the normal to the triangles mesh. In the described work, the algorithm for selecting the nearest triangle of the CGAL library was used (The algorithm searching triangular intersection). Having determined the vector **N**, draw through the points **A** and **B** the plane is parallel to the vector **N**. The plane will cross the mesh, and we get a set of contours consisting of straight lines (Figure 6). Similarly, we cross the same plane with the voxel model (Figure 7). The calculation of mesh sections is realized using the CGAL library classes (Akenine-Möller, 2001; CGAL, Computational Geometry Algorithms Library).

For further analysis, it is necessary to drop those points of contours that lie outside the line **L**. We denote the set of all segments of the contours {ai, bi}. The distance between the points **A** and **B** is defined as Equation (4). We may discard all lines not satisfying at least one of the conditions (Equation 5).

Despite the fact that the points of contours are all in one plane, they are 3D points. To project them onto the plane, we introduce the coordinate system X'Y'Z '. The axis origin should be placed in point **A**. **X'** and the axis is directed to point **B**. Axis $\vec{Y}' = \vec{N} \times \vec{X}'$, and axis $\vec{Z}' = \vec{X}' \times \vec{Y}'$. The transformation matrix is written as follows Equation (6). Multiplying the contours vertices coordinates by the matrix, we obtain the coordinates of the vertices in the XY plane. The section of the mesh along the line **L** is a set of connected segments (straight lines). Each segment has a normal to the segment (Figure 8). We specify the normal vector at each end of each segment as the half-sum of the vectors of the neighbor segments (Figure 9).

To calculate the surface roughness, we define parameter $\mathbf{d_l}$. Each segment of the line **L** is split with a parameter $\mathbf{d_l}$. At each splitting point we calculate the normal vector by the formula (Equations 7-8), where: **n** – break number; $\mathbf{N_0}$ – normal vector in initial segment point; $\mathbf{N_1}$ – normal vector in end segment point; **l** – segment length (Figure 10). For each splitting point of the segment of the original model, we calculate beam intersection points in direction **N** with the section of the voxel model. The distance between the splitting point and the intersection point is defined as $\mathbf{y_i}$.

We calculate $\mathbf{R_a}$ — the arithmetic means of the absolute values of the profile deviations within the base length (Equation 9). In the process of calculating $\mathbf{y_i}$, we will store the 5 largest (maximum) and 5 lowest (minimum) values obtained from the calculation process ($\mathbf{H_{imax}}$ and $\mathbf{H_{imin}}$). We use these values to calculate another surface roughness parameter $\mathbf{R_z}$, representing a sum of altitude average absolute values of five maximum profile peaks and depths of five maximum profile valleys within a base length (Equation 10).

Possible algorithm improvements:

The considered algorithm for calculating the surface roughness can be accelerated by discarding those mesh triangles that do not fall into the vicinity of the baseline L. Dropping the triangles will greatly accelerate the calculation of the mesh cross-section for meshes with a large number of polygons, and also when calculating the surface roughness of a product made with high resolution. When calculating the mesh voxel representation of the product, you can discard those triangles of the mesh surface that are oriented inward of the part. This can be done by comparing the direction of the normal vector of the voxel face and the normal of the intersecting voxel of the original mesh triangle. For further development and improvement of the work, a possibility of parallel algorithm use will be considered for acceleration of analysis data handling and assessment of product roughness values (Anamova *et al.*, 2016; Ripetskiy *et al.*, 2016; Dmitriev et al., 2015).

CONCLUSIONS:

Validation of the mathematical model for surface roughness assessment, it's one of the challenging question and authors are now in the process of building a data base including characteristics for different AM machines and different discrete 3D models to compare and find adequate settings that can guaranty the repeatability of the process.

The article describes implementation of the surface roughness calculation in the AM processes used in Moscow aviation institute and available for different SLA/SLM machines such as EOS. From the practical point of view more than 85 % of prints have expected surface roughness as calculated by the presented method.

Presented method can be used for further investigation that can lead to design new algorithms taking even more characteristics for modern modern commercial and experimental machines from Russian or foreign hardware industrial manufacturing industry.

ACKNOWLEDGMENTS:

The work was carried out with the financial support of the state project of the Ministry of Education and Science project code 2.9219.2017 / 8.9.dc4 v j

REFERENCES:

1. Akenine-Möller, T. *Fast 3D Triangle-Box Overlap Testing*, **2001**. http://fileadmin.cs.lth.se/cs/Personal/Tomas_Akenine-Moller/code/tribox_tam.pdf, accessed August 2018.
2. Anamova, R.R., Zelenov, S.V., Kuprikov, M.U., Ripetskiy, A.V. *IOP Conference Series: Materials Science and Engineering*, **2016**, *140(1)*, 012003
3. Anukhin, V.I. *Tolerances and landing: studies. Manual*, Saint Petersburg: Peter, **2012**.
4. CGAL, Computational Geometry Algorithms Library. https://www.cgal.org, accessed August 2018.
5. Davydov, V.M., Zayev, V.V., Panochevnyi, P.N., Kozachenko, I.A., Prokhorets, O.V. *Scientists Notes PNU*, **2013**, *4(4),* 1061-1074.
6. Dmitriev, V.G., Egorova, O.V., Rabinskii, L.N. *Composites: Mechanics, Computations, Applications*, **2015**, *6(4)*, 265-277.
7. Filatov, Yu.D., Yashchuk, V.P, Filatov, A.Yu., Heisel, U., Storchak, M., Montey, G. *Superhard Materials*, **2009**, *5*, 70-81.
8. Gawlik, J., Niemczewska-Wojcik, M., Krajewska, J., Sokhan, S.V., Pascenko, E.A., Zuk, T.S. *Management and Production Engineering Review*, **2014**, *5(4),* 9-17.
9. Gorshkov, A.G., Egorova, O.V., Medvedsky, A.L., Rabinsky, L.N. *Proceedings of the Russian Academy of Sciences. Solid Mechanics*, **2003**, *3*, 148-155.
10. Kaladze, V.A., Shaposhnikov, V.N. *Polzunovsky Herald*, **2010**, *2*, 72-77.
11. Kochetkov, A.V., Chvanov, A.V., Arzhanukhina, S.P. *Bulletin of the Volgograd State University of Architecture and Civil Engineering. Series: Construction and Architecture*, **2009**, *14(33)*, 80-86.
12. Kuznetsova, E.L., Leonenko, D.V., Starovoitov, E.I. *Mechanics of Solids*, **2015**, *50(3)*, A012, 359-366.
13. Lich, R. *Engineering foundations for measuring nanometric accuracy: Educational edition*, Dolgoprudny: Publishing House "Intellect", **2012**.
14. Lurie, S.A., Solyaev, Y.O., Lizunova, D.V., Bouznik, V.M., Menshykov, O. *International Journal of Heat and Mass Transfer*, **2017**, *109*, 511-519.
15. Minimum bounding box. https://en.wikipedia.org/wiki/Minimum_bounding_box, accessed July 2018.
16. Octree. https://en.wikipedia.org/wiki/Octree, accessed July 2018.
17. Odinokova, E.V., Panfilov, Yu.V., Yurchenko, P.I. *Engineering Journal: Science and Innovations*, **2013**, *6*, 1-7.
18. Paz, V. DE F., Trevisan, L., Reguly, A. *Periódico Tchê Química*, **2016**, *13(26),* 97-105.
19. Rabinskiy, L.N., Ripetskiy, A.V., Zelenov, S.V., Kuznetsova, E.L. *International Journal of Pure and Applied Mathematics*,

2017a, *116(3),* 789-797. doi: 10.12732/ijpam.v116i3.24.

20. Rabinskiy, L.N., Ripetskiy, A.V., Zelenov, S.V., Kuznetsova, E.L. *Journal of Industrial Pollution Control*, **2017b**, *33(1)*, 1178-1183.

21. Rabinskiy, L.N., Zhavoronok, S.I. *Journal of Physics: Conference Series*, **2018**, *991(1)*, 012067.

22. Ripetskiy, A.V., Zelenov, S.V., Vucinic, D., Rabinskiy, L.N., Kuznetsova, E.L. *International Journal of Pure and Applied Mathematics*, **2016**, *111(2)*, 343-355. doi: 10.12732/ijpam.v111i2.17

23. Ripetskiy, A.V., Zelenov, S.V., Vučinić, D., Rabinskiy, L.N., Kuznetsova, E.L. *International Journal of Pure and Applied Mathematics*, **2016**, *111(2)*, 343-356.

24. Stanford Bunny. https://en.wikipedia.org/wiki/Stanford_bunny http://graphics.stanford.edu/pub/3Dscanrep/bunny.tar.gz., accessed July 2018.

25. Surface roughness schema. https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/Roughness_rus.png/1920px-Roughness_rus.png, accessed July 2018.

26. The algorithm searching triangular intersection. https://upload.wikimedia.org/wikipedia/commons/thumb/2/20/Octree2.svg/800px-Octree2.svg.png, accessed July 2018.

27. Valetov, V.A., Filimonova, E.A. *Metalworking*, **2011**, *5*, 45-46.

$$n = 2^p, p = \lceil (\ln(S/S_v)/ln(2)) \tag{1}$$

$$S = S_v n \tag{2}$$

$$\vec{N} = (\vec{N}_a + \vec{N}_b) * 0.5 \tag{3}$$

$$D = \|A - B\| \tag{4}$$

$$|a_i - A| < D, |b_i - A| < D, |a_i - B| < D, |b_i - B| < D \tag{5}$$

$$M = \begin{pmatrix} Y'_x & Y'_y & Y'_z & 0 \\ X'_x & X'_y & X'_z & 0 \\ Z'_x & Z'_y & Z'_z & 0 \\ A_x & A_y & A_z & 1 \end{pmatrix} \tag{6}$$

$$N_i = N_0(1 - l_i/l) + N_1 l_i/l \tag{7}$$

$$l_i = d_l i, i = 0..n \tag{8}$$

$$R_a = \frac{1}{n} \sum_{i=1}^{n} |y_i| \tag{9}$$

$$R_z = \left(\sum_{i=1}^{5} |H_{i\ max}| + \sum_{i=1}^{5} |H_{i\ min}| \right) /5 \tag{10}$$

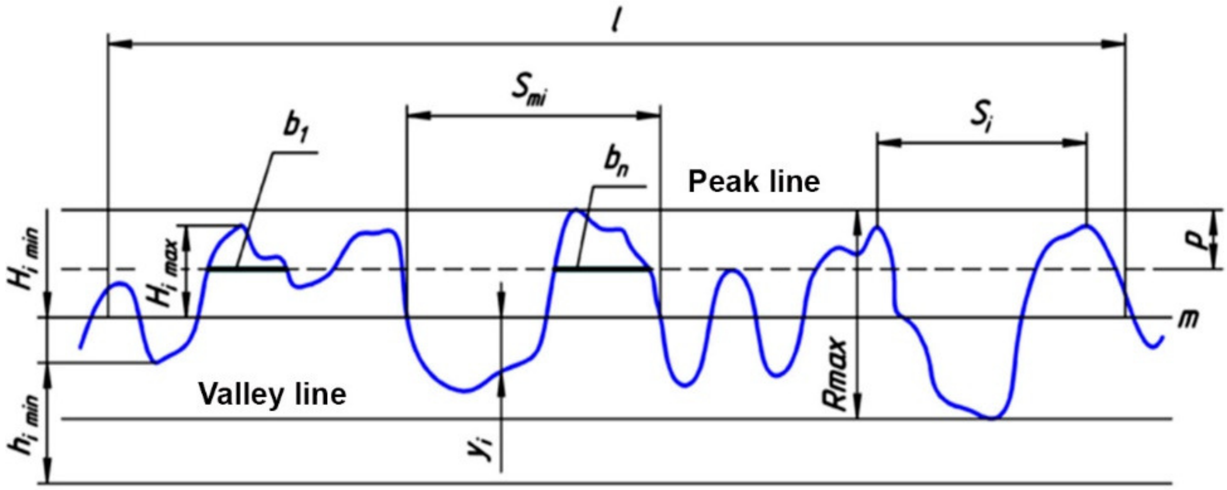

Figure 1. *Surface roughness schema (Surface…)*

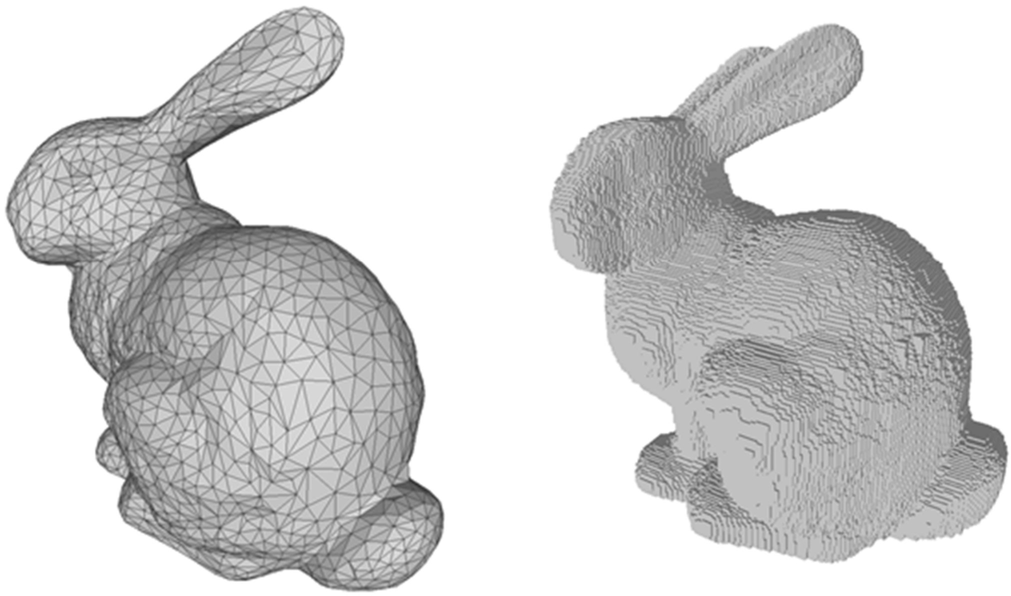

Figure 2. *Polygonal model of Stanford rabbit on the left and voxel model corresponding to it on the right side*

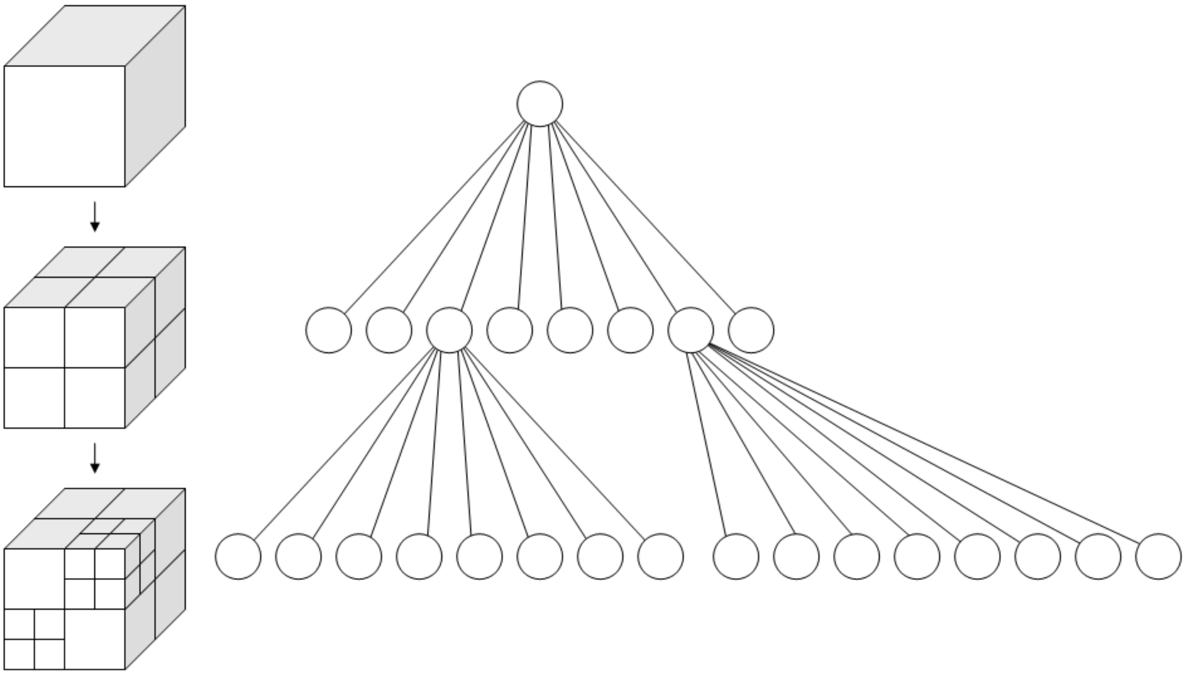

Figure 3. *Octree voxel model generation example (Minimum bounding box)*

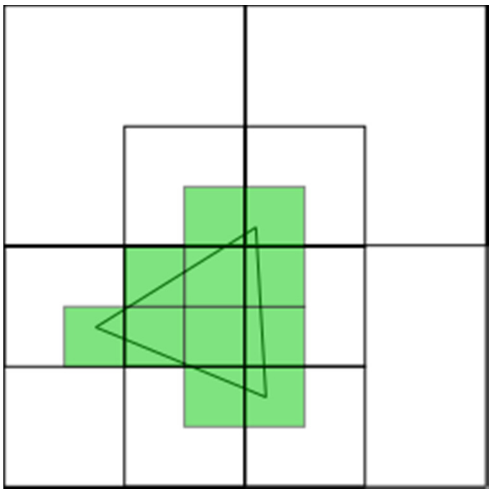

Figure 4. *Example of quadtree model join with one triangle*

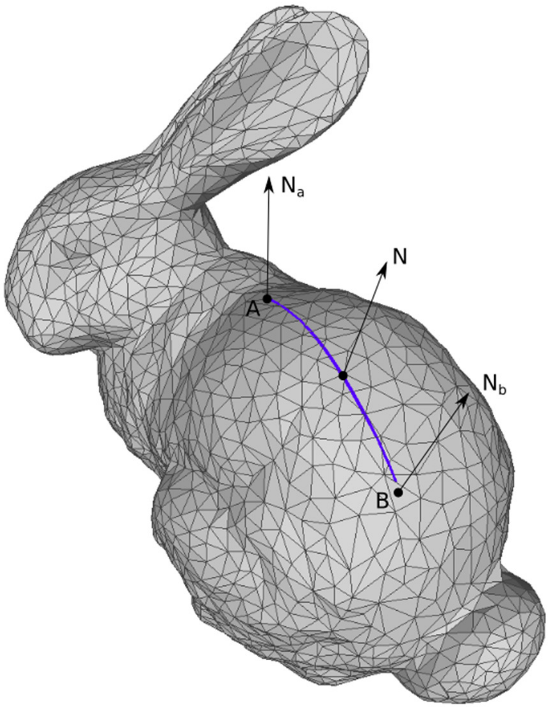

Figure 5. *Determination of the base length for the roughness surface evaluation*

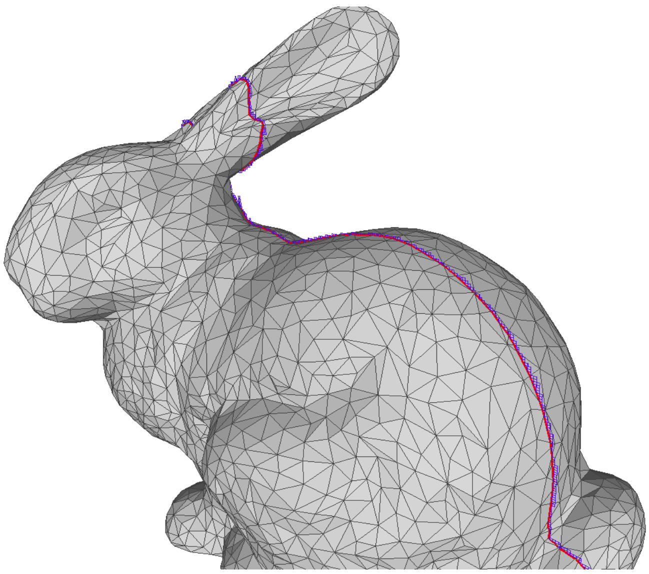

Figure 6. *Section of a mesh. The blue line shows the voxel cross-section, the red line shows the cross-section of the original mesh*

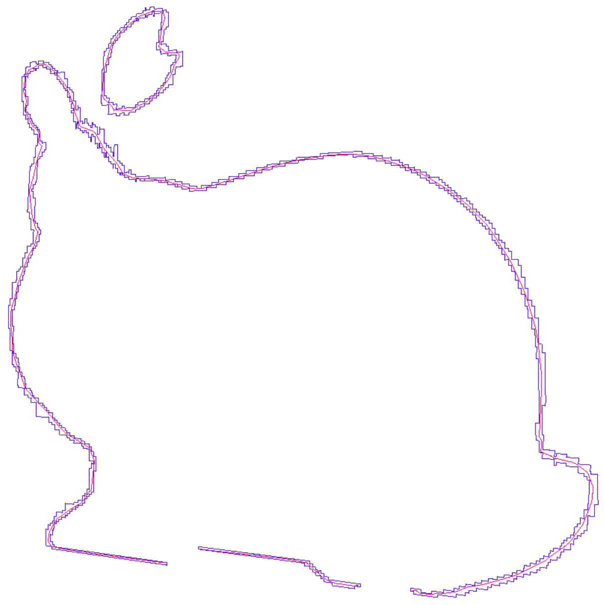

Figure 7. *Cross-section overlay. The blue line shows the voxel cross-section, the red line shows the cross-section of the original mesh*

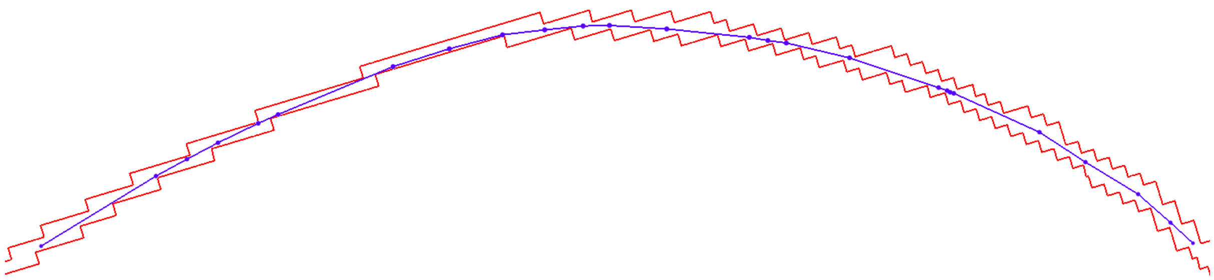

Figure 8. *Mesh section and voxel model along base length **L** is projected on the **XY** plane. Blue color shows the mesh section; red color shows the voxel model section*

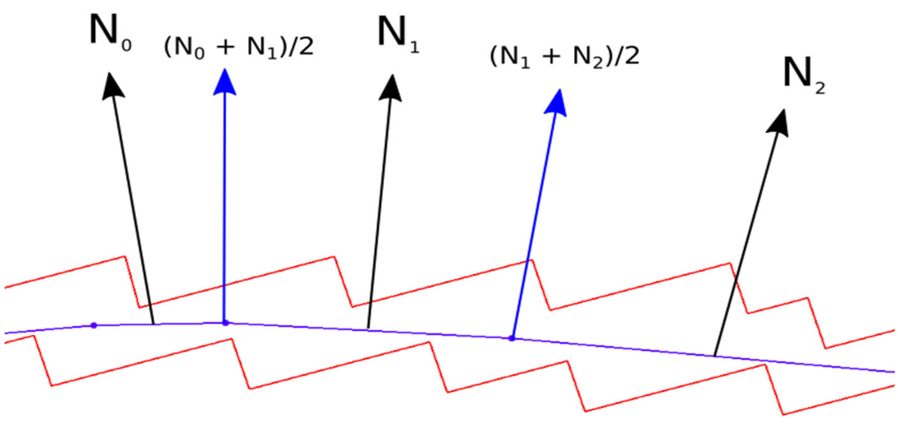

Figure 9. *Defining the normal vectors at points of the segments*

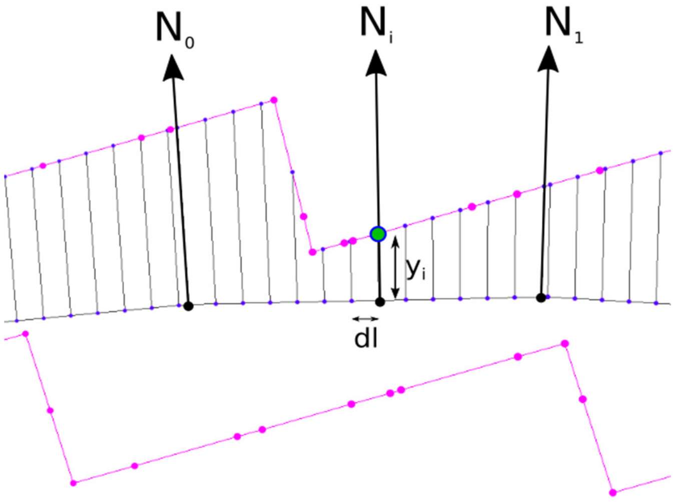

Figure 10. *Roughness surface calculation*