

PAPER • OPEN ACCESS

Real-time cutting simulation based on voxel model

To cite this article: A I Obukhov *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **709** 044062

View the [article online](#) for updates and enhancements.

Real-time cutting simulation based on voxel model

A I Obukhov^{1,2,*}, S V Evstafieva^{1,2} and L I Martinova^{1,2}

¹MSTU Stankin, department of Computer Control Systems, 127055 Vadkovsky per 3a, Moscow, Russia

² This work was supported by the Ministry of Education and Science of the Russian Federation as part of the fulfillment of the state assignment (No. 2.1237.2017 / 4.6) and was carried out using equipment provided by the collective center of Moscow State University "STANKIN"

* obukhov@ncsystems.ru

Abstract. The article discusses the approach to the implementation of cutting simulation in the CNC system. Simulation of cutting before starting the actual machining of the product allows you to avoid many errors associated with the positioning, reorientation of the tool, check the correctness of the tool's approach to the workpiece, etc. This becomes especially relevant in multi-axis machining [1-4]. The advantages of presenting a product model using a sparse voxel octree are considered.

1. Introduction

A modern CNC system cannot be considered a full-fledged product without tools for modeling the processing of products in a virtual environment [5-8]. The ability to view the process and the result of processing in three-dimensional allows you to: 1) ensure that there are no obvious errors when generating or manually writing the program, collisions and accidental undercuts on the part with a tool during reorientation and positioning; 2) select the optimal tool geometry to achieve the required surface quality; 3) change the cutting strategy or tool approaches to the workpiece according to the results of the analysis of the sequence of tool movements.

Processing simulation functions are often assigned to individual software products due to the complexity and high cost of their development, as well as the limitations of the hardware of CNC systems (relatively low performance of the computing platform, the lack of high-performance graphics subsystem) [9-13]. Nevertheless, the presence of a virtual simulation module directly in the software of the CNC system has the advantage of high integration with the core and the entire infrastructure of mathematical software. This allows you to take into account when modeling all the features of the implementation of the geometric problems of the CNC system and view the process of virtual processing in real time (both during emulation and during the actual processing of the part on the machine). Consider the approach we have implemented to develop a virtual processing simulation module.

2. Product model presentation

There are several practical methods for representing the model of a three-dimensional object as a data structure. Of compromise options, based on the use of incomplete information of solid body topology,



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

you can specify elevation maps, wireframe and surface models. These methods are of limited use. In general, it is necessary to use a full description of a volumetric body based on the structure of a constructive solid-state geometry, a boundary representation, or a voxel model (we do not consider exotic variants). Constructive solid-state geometry and boundary representation are widely used to describe models in CAD-systems, but they are poorly suited to simulate processing in real-time, since they require periodic resource-intensive restructuring of all surfaces. For this reason, for the software representation of bodies deformable in real time, the most promising is the voxel model, which is “naturally” suitable for intensive volume removal procedures characteristic of cutting modeling.

A pure voxel model (i.e., a three-dimensional array) requires large amounts of memory and limits the developer in the ability to build fast visualization algorithms and change the structure of an object. Therefore, methods have been proposed for optimized representation of a voxel array. Of interest are hierarchical structures and, in particular, sparse octree [14]. We will show how voxel octree allowed us to implement an effective algorithm for virtual processing of the product in real time.

Sparse octree is a hierarchy of levels of binary partitioning of space. The root of the tree is a cube whose side is equal to the largest dimension of the object (out of three dimensions along the axes of the Cartesian coordinates). All nodes, except for the nodes of the last crushing level (ie, the “leaves” of the tree, representing the smallest significant elements of the volume), have an array of 8 child nodes. Child nodes correspond to cubic volumes obtained by cutting the parent cube in half into each Cartesian coordinate. Each node must contain an indication of activity, i.e. show whether it is empty volume of space or containing part of the material of the object.

In our version of the tree, the structure of voxels and connections between levels is as follows (Figure 1):

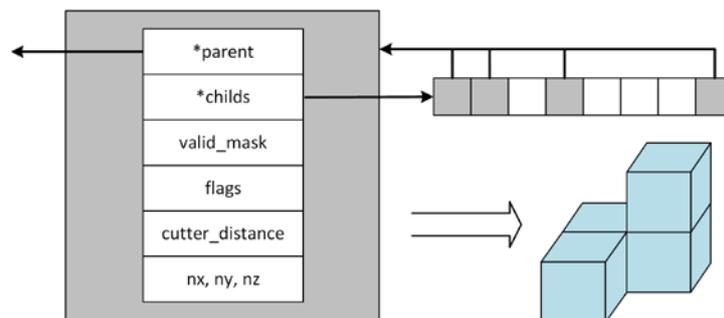


Figure 1. Voxel structure and relationship between two octree levels.

Here parent – this is a pointer of the parent block. At the root node, it is zero. The node has a pointer to an array of 8 children (childs). If the childs pointer is zero, the block is a solid area of the object or a leaf of the tree, its fragmentation is not required. Thus, memory is allocated only when part of the parent volume is removed from a solid object. Having one pointer to an array of all child nodes instead of eight separate pointers can significantly reduce the number of operations with dynamic memory when the object changes. To determine the activity of the child nodes there is a mask - `valid_mask`. 8 bits its values correspond to 8 nodes. For a solid block, the mask is 255 (i.e., all nodes are active, none are deleted). When editing an object, operations with masks of nodes are performed, and the memory allocated for each childs pointer is released only when the mask of the corresponding node is reset (this indicates that the volume of the node is completely empty).

The remaining elements of the structure determine the properties of the voxel that are not related to the tree topology: `flags` – node type indicator (raw volume of the workpiece, surface of the workpiece with a pre-calculated normal, treated surface); `cutter_distance` – current distance from voxel center to instrument surface. Determines the approximate amount of penetration of the instrument into the voxel volume; `nx, ny, nz` – components of the normal vector to calculate the lighting.

Table 1 shows a comparison of file sizes of processed and saved on disk models of different resolutions for the same part. It can be concluded that when the resolution is increased by 2 times, the

memory occupied by the model increases approximately 4 times. Such a pattern can be traced for any details. This shows the key advantage of the rarefied octree over the simple voxel representation: the memory size of the model is proportional to the area of the treated surface, not the volume (this is our empirical observation).

Table 1. Evaluation of occupied memory for models with different resolutions.

Model resolution	Size (Kb)
256 x 256 x 256	2760
512 x 512 x 512	10684
1024 x 1024 x 1024	43920
2048 x 2048 x 2048	174198

3. Visualization method

The object visualization algorithm based on the casting principle (ray casting) is taken from work [14] and adapted to execute the code on the central processor (in the original work, the code for CUDA was presented). The decision to use an optimized and streamlined algorithm with convincing published results justified itself - visualization of the processing object in real time was possible even for client computers of the average-performance CNC system (Core i3-540 processor). Tests show that frame rendering time grows rather slowly with increasing model resolution. Table 2 presents the data at the pixel size of the image of the test part 1024x1024 in the XY projection.

Table 2. Average frame rendering time for models of different resolutions.

Model resolution	Frame rendering time for Core i3-540 processor (ms)	Frame drawing time for a Core i7-3770 processor (ms)
256 x 256 x 256	11	5
512 x 512 x 512	13	6
1024 x 1024 x 1024	17	8
2048 x 2048 x 2048	22	10

Qualitative calculation of lighting in three-dimensional graphics is not conceivable without information about the orientation of the illuminated surfaces. For voxel models, approximation procedures based on the field gradient are most often used [15]. This method is also implemented in our system, but we managed to achieve a qualitatively different level of realism by implementing the idea of assigning the normal value of the nearest point of the instrument surface to the voxel (information about the closest point is an indirect result of the volume removal process during virtual processing). In Figure 2 shows a comparison of the visualization of the machined part by copying the tool surface normal (left) and the normal gradient approximation (right). To calculate the light in both cases, the classical Phong method is used [16]. The gradient is calculated by the 26 nearest neighbors [17].

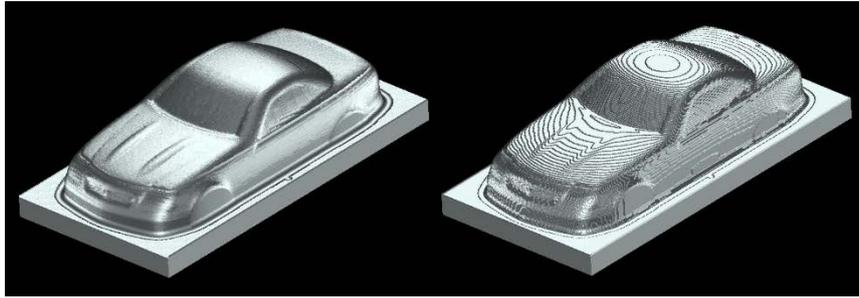


Figure 2. Two ways to visualize a virtual model of a part.

The method of copying the normal to the nearest point on the tool surface allows one to realistically convey the surface character of the processed product, eliminating the discreteness of shading levels characteristic of the gradient method and the parasitic smoothness of the sharp edges of the part. In fact, we obtained a result that is close to displaying the exact boundary representation of the real surface of the product obtained on the machine tool.

4. Virtual processing algorithm

Using the structure of a sparse octree allows not only to reduce the amount of memory occupied and to effectively visualize the model, but also to implement a fast procedure for virtual processing of the product. With a simple raster representation of voxel data, the computational complexity of the volume removal problem during tool movement is generally proportional to the cube of the volume being deleted (and resolution). The binary partitioning of space by an octree allows you to develop an algorithm whose complexity is proportional to the surface swept up by the tool. This allows you to reach a qualitatively new level of virtual processing, making it possible to implement the procedure for removing the volume dozens and even hundreds of times per second with the existing resources of modern general-purpose processors (without using mass-parallel computing on graphics cards). Such a frequency is enough to emulate real processing in a virtual environment with high accuracy.

The virtual processing procedure is recursive and advances according to the following algorithm (Figure 3):

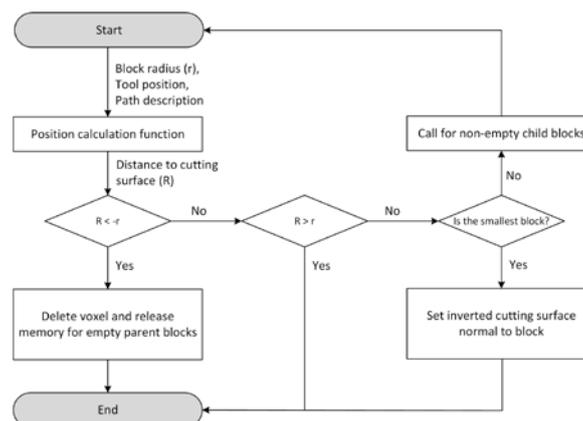


Figure 3. The procedure of virtual processing (removal of material model).

The procedure is called initially for the root block of the model, and then recursively for the child nodes.

The key element of the algorithm is the function of calculating the position, which is implemented separately for each type of tool and type of movement (with a constant or variable orientation of the tool). The return value of the function is the distance from the center of the block to the nearest tool surface. Further, this distance is used to determine whether the block is inside a remote volume, lies on its boundary or outside the boundaries. If the block is inside, it is removed from the model; if out of

bounds - the procedure ends without modifying the block. In the border case, the procedure is called for the existing child nodes of the block or is completed if the block is the smallest element of the model.

At the same time, to simplify the calculations, the blocks of the model (regardless of their level) are considered spheres described around the cubes. This approach is valid because the spheres of the child blocks always lie within the spheres of the parents (Figure 4). Thus, erroneous block skipping is impossible, and the distance from the block to the cutting surface can be considered very simply - as the distance from the center minus the sphere radius.

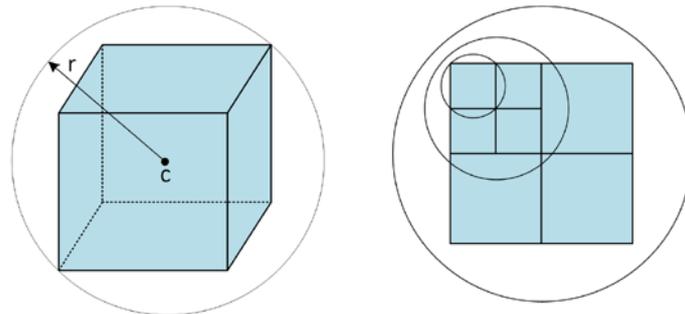


Figure 4. Block representation as a sphere in position calculations.

The function of calculating the position for the linear motion of a spherical milling cutter is implemented most easily. For tools with a flat and toroidal end, calculations are much more complicated due to the presence of surfaces with 4th order generators (except for movements in the XY plane and along the Z axis of the tool coordinate system). Analytical calculation methods for such cases have been proposed [18,19], but in practice it is simpler to use approaches based on zero-order optimization, which allows one to quickly calculate the approximate distance from a point to a curved surface.

Table 3 shows the execution time of the volume removal procedure for a tool with a radius of 10 mm, a cubic model with a nominal size of 100x100x100 mm. Depth of immersion tool in the material - 50 mm. Those an estimate of the time for removal of the volume of a vertically cut tool is given.

Table 3. Average volume removal time for models of different resolutions.

Model resolution	Frame rendering time for Core i3-540 processor (ms)	Frame drawing time for a Core i7-3770 processor (ms)
256 x 256 x 256	1	<1
512 x 512 x 512	5	3
1024 x 1024 x 1024	19	12
2048 x 2048 x 2048	75	46

These and several other experiments have shown that time grows in proportion to the area of the bounding surface of the volume being removed (in other words, to the number of smallest blocks of the model crossing this surface).

5. CNC integration

The visual component of the simulation is deployed along with the code of virtual processing algorithms, has a clearly defined interface and can be embedded in any application. In the terminal client of our CNC system “AxiOMA Control” [20-22], the component is used to implement the screen of the simulation processing mode. There are two main types: full-screen view with the opening of the

visualization panel on the entire area of the working screen and the standard view with the display of program text and technological information. The user can control the view without interrupting the virtual processing process and see the tool movement in real time (Figure 5).

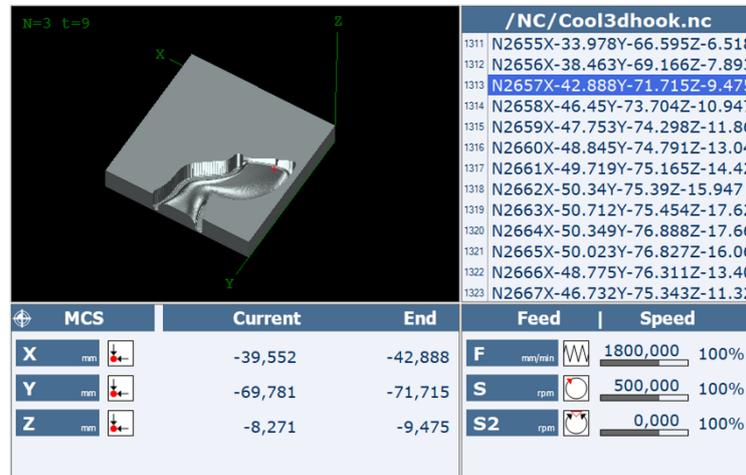


Figure 5. The main panel of the simulation mode screen.

The directive - #workpiece is provided for running simulation processing from a control program, which passes the basic parameters of the workpiece and processing mode. Example:

```
#workpiece(0, 256, 3, 120, 120, 20, -60, -120, -20)
S500 M3 M6 T1
N1350 G01 X-18.097Y-1.41Z-12.179 F2500
N1351X-18.329Y-1.594Z-15.528
N1352X-18.256Y-1.72Z-18.
N1353X-17.614Y-2.808Z-18.065
...
M30
```

Here, a blank in the form of a parallelepiped is defined and the milling mode for machining simulation is enabled. The definition of the workpiece directly in the control program greatly simplifies the process of regular testing of the CNC system on a pre-formed set of test programs.

6. Conclusion

The proposed method of presenting a product in the form of a sparse voxel model made it possible to implement an effective algorithm for virtual processing of the product in real time, in which the memory size occupied by the model and the computational complexity of the procedure are proportional to the area of the treated surface, and not to the volume.

The implemented virtual processing algorithm makes it possible to carry out the volume removal procedure hundreds of times per second with the existing resources of modern general-purpose processors. This frequency is enough to emulate the actual cutting process in a virtual environment with high accuracy.

The illumination calculation for the voxel model is implemented based on the idea of assigning the value of the normal to the nearest point of the instrument surface to the voxel, which allowed us to obtain a result close to displaying the exact boundary representation of the real surface of the product.

The software component based on the proposed solutions has been successfully integrated into the CNC system and is used both for regular tests of control programs before actual machining and in the educational process.

Acknowledgments

This work was supported by the Ministry of Education and Science of the Russian Federation as part of the fulfillment of the state assignment (No. 2.1237.2017 / 4.6) and was carried out using equipment provided by the collective center of Moscow State University "STANKIN"

References

- [1] Martinov G and Kozak N 2016 Specialized numerical control system for five-axis planing and milling center *Russian Engineering Research* **36**(3) 218-22.
- [2] Martinov G and Kozak N 2015 Numerical control of large precision machining centers by the AxiOMA control system *Russian Engineering Research* **35**(7) 534-8.
- [3] Martinov G, Kovalev I and Al Khoury A 2018 *International Russian Automation Conf (RusAutoCon)* (Sochi: IEEE)
- [4] Martinov G, Kozak N, Nezhmetdinov R, Grigoriev A, Obukhov A and Martinova L 2017 Method of decomposition and synthesis of the custom CNC systems *Automation and Remote Control* **78**(3) 525-36.
- [5] Martinov G, Obukhov A and Kozak N 2018 The Usage of Error Compensation Tools of CNC for Vertical Milling Machines *Russian Engineering Research* **38**(2) 119-22.
- [6] Martinova L, Kozak N, Nezhmetdinov R, Pushkov R and Obukhov A 2015 The Russian multi-functional CNC system AxiOMA control: Practical aspects of application *Automation and Remote Control* **76**(1) 179-86.
- [7] Martinov G, Obuhov A, Martinova L and Grigoriev A 2014 *6th CIRP International Conf. on High Performance Cutting, HPC2014* 14 pp 511-16.
- [8] Pushkov R, Salamatin E and Evstafieva S 2018 *International Conf. on Modern Trends in Manufacturing Technologies and Equipment (MATEC Web Conf)* 224 pp 1-7.
- [9] Martinova L and Martinov G 2018 *3rd Russian-Pacific Conf. on Computer Technology and Applications* (Vladivostok) pp 1 - 4.
- [10] Grigoriev S and Martinov G 2016 *7th HPC 2016 – CIRP Conference on High Performance Cutting* 46 pp 525-28.
- [11] Grigoriev S and Martinov G 2016 *48th CIRP Conference on Manufacturing systems* 41 pp 858-63.
- [12] Martinov G and Nezhmetdinov R 2015 Modular design of specialized numerical control systems for inclined machining centers *Russian Engineering Research* **35**(5) 389-93.
- [13] Martinov G, Lyubimov A, Bondarenko A, Sorokoumov A and Kovalev I 2015 An approach to building a multiprotocol CNC system *Automation and Remote Control* **76**(1) 172-8.
- [14] Laine S and Karras T 2010 Efficient Sparse Voxel Octrees - Analysis, Extensions, and Implementation *NVIDIA Technical Report NVR-2010-001*
- [15] Yagel R, Cohen D, and Kaufman A 1992 Normal Estimation In 3d Discrete Space *The Visual Computer* **8** 278-91.
- [16] Phong B T 1975 Illumination for Computer Generated Pictures. *Communications of the ACM* **18** 311-7.
- [17] Thürmer G and Wuthrich C 1997 Normal Computation for Discrete Surfaces in 3D Space *EUROGRAPHICS* **16** 15-26.
- [18] Wang W P and Wang K K 1986 Geometric modeling for swept volume of moving solids *IEEE Computer Graphics and Applications* **6**(12) 8-17.
- [19] Chung Y C, Park J W, Shin H and Choi B K 1998 Modeling the Surface Swept by a Generalized Cutter for NC Verification *Computer-Aided Design* **8**(30) 587-94.
- [20] Grigoriev S and Martinov G 2018 *3rd Russian-Pacific Conf. on Computer Technology and Applications* (Vladivostok) pp 1 - 4.
- [21] Nikishechkin P, Chervonnova N and Nikich A 2018 *International Conf. on Modern Trends in Manufacturing Technologies and Equipment (MATEC Web Conf)* 224 pp 1-9.
- [22] Martinova L and Fokin N 2018 *International Conf. on Modern Trends in Manufacturing Technologies and Equipment (MATEC Web Conf)* pp1-5.