

Article

Material Removal Optimization Strategy of 3D Block Cutting Based on Geometric Computation Method

Hui Shao ^{1,2,*} , Qimeng Liu ^{1,2} and Zhiwei Gao ³ 

¹ School of Information Science and Engineering, Huaqiao University, Xiamen 361021, China; lqimeng2021@163.com

² Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen 361021, China

³ Faculty of Engineering and Environment, University of Northumbria at Newcastle, Newcastle upon Tyne NE1 8ST, UK; zhiwei.gao@northumbria.ac.uk

* Correspondence: shaohuihu11@163.com

Abstract: During the material removal stage in stone rough processing, milling type has been widely explored, which, however, may cause time and material consumption, as well as substantial stress for the environment. To improve the material removal rate and waste reuse rate in the rough processing stage for three-dimensional stone products with a special shape, in this paper, circular saw disc cutting is explored to cut a convex polyhedron out of a blank box, which approaches a target product. Unlike milling optimization, this problem cannot be well solved by mathematical methods, which have to be solved by geometrical methods instead. An automatic block cutting strategy is proposed intuitively by considering a series of geometrical optimization approaches for the first time. To obtain a big removal block, constructing cutting planes based on convex vertices is uniquely proposed. Specifically, the removal vertices (the maximum thickness of material removal) are searched based on the octree algorithm, and the cutting plane is constructed based on this thickness to guarantee a relatively big removal block. Moreover, to minimize the cutting time, the geometrical characteristics of the intersecting convex polygon of the cutting plane with the convex polyhedron are analyzed, accompanied by the constraints of the guillotine cutting mode. The optimization algorithm determining the cutting path is presented with a feed direction accompanied by the shortest cutting stroke, which confirms the shortest cutting time. From the big removal block and shortest cutting time, the suboptimal solution of the average material removal rate (the ratio of material removal volume to cutting time) is generated. Finally, the simulation is carried out on a blank box to approach a bounding sphere both on MATLAB and the Vericut platform. In this case study, for the removal of 85% of material with 19 cuts, the proposed cutting strategy achieves five times higher the average material removal rate than that of one higher milling capacity case.

Keywords: block cutting; data reconstruction; convex polyhedron (CPH); convex polygon (CPG); path optimization; average material removal rate (AMRR)



Citation: Shao, H.; Liu, Q.; Gao, Z. Material Removal Optimization Strategy of 3D Block Cutting Based on Geometric Computation Method. *Processes* **2022**, *10*, 695. <https://doi.org/10.3390/pr10040695>

Academic Editor: Gade Pandu Rangaiah

Received: 3 March 2022

Accepted: 30 March 2022

Published: 2 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The stone processing industry has adverse effects on the environment, economy and sustainability. Stone processing causes heavy pollution from dust and CO₂ emissions, and high water and energy consumption, which brings tremendous pressure and threats to the ecological environment, especially in natural stone mining areas [1]. On the other hand, with the development of modern civilized society, the stone industry is indispensable and has become increasingly important. Moreover, the demands and varieties of stone products are increasing day by day. Therefore, the demand for the stone industry puts forward higher requirements for stone processing, especially in the rough processing stage. Building a green manufacturing system and process scheme, developing energy-saving and emission reduction optimization technologies and improving the processing efficiency of the stone industry have attracted more and more attention recently [1,2].

In the machining process of special-shaped stone products, the traditional rough processing adopts diamond wire sawing or circular sawing for simple tasks [2,3], or conventional milling for simple tasks [4], or automatic milling for complex tasks [5] with a diamond grinding wheel, etc., but such processing modes cannot realize automation completely, or are accompanied by low waste utilization, a large processing time, high consumption and serious tool wear even for straight cuts. Although the milling process has reflected automation to some extent, a large amount of dust will be generated in the machining process. The above mentioned processes can often not meet the needs of green manufacturing and high-efficiency machining in the rough machining stage.

For a material removal optimization strategy, the analysis methods of different machining tools are completely different. Owing to the distinct advantages of having a big cutting depth and high cutting linear velocity, the diamond circular saw blade is widely used in the stone cutting process, which provides a possible way to achieve efficient green processing [1,6]. It is noted that circular saws are known to likely be the cheapest and fastest motorized saw available [7,8]. With the development of the multi axis linkage technology of machining machinery, the applications of 5–6 axis NC machine tools and robots, some interesting results have been reported in recent years on the material removal method during the rough machining process using circular saw blades. In [1], an energy consumption prediction model of the stone sawing process of a circular saw was proposed. By predicting the power and energy consumption in the whole sawing process, the optimal scheme considering the variable material removal rate (MRR) could be discussed for stone processing to achieve energy saving and emission reduction. In [8], a technique to cut freeform curves with a flexible circular saw was addressed by setting the width of the cutting edge larger than the width of the saw body to ensure there was no friction between the machined surface and the saw body while cutting. Moreover, cutting any polygon down to an inner complex nonconvex shape was achieved by a sequence of straight cuts with linear-time algorithms in [9], where the cuttability of a small saw and large saw was analyzed attentively. Ref. [10] studied the algorithm for cutting polyhedral shapes with a hot wire cutter, utilizing computational geometry techniques to solve the problems of lines and segments in the cutting process. Particularly used in recent years when fabricating freeform geometries, in order to find collision-free tangential cutting directions, a conservative algorithm for line cutting with a wire cutter was presented by [11], which provided advanced techniques to remove large amounts of material. Exploring the material cutting of 2D or 3D geometric shapes, in [12], an approximation algorithm for cutting out convex polygons was presented, which can cut convex polygons from the plane at a minimum cost by designing an optimal cutting sequence. Their algorithm can achieve a constant approximation ratio of the paper diameter to the polygon diameter. On the basis of [12], in [13], an approximate algorithm for cutting out a convex polyhedron from a sphere was surveyed, in which several approximate algorithms were discussed to find the plane sequence with the minimum cutting cost. Ref. [14] proposed a method of 3D curved block cutting analysis by utilizing the advantages of topology and computational geometry in geological solid modeling, where 3D curved blocks were formed with less calculation and memory. These studies not only provide feasibility for a stone cutting plan with a diamond circular saw, but also provide a geometrical analysis method for solving the geometrical characteristics of block cutting optimization strategies to some extent.

Despite some of the new algorithms and analysis methods being investigated in [11,14], to the best of our knowledge, the theoretical research available on the optimization strategies of block sawing with a circular saw blade for special-shaped stone in the rough machining stage is currently sparse. This is due to the need to touch upon the convex polyhedron (CPH) reconstruction techniques of computational geometry and computer graphics, and the constraints of cutter head feeding along a straight line in the cutting process. The geometric challenges have also been stated for multi axis machining including the material properties, tools shapes, accessibility, collision detection, etc. [11]. Although line cutting [11] was an available new and flexible style for automatic block cutting, the cutting technique

was completely different from saw disc cutting. Consequently, these studies have motivated us to find out more about block cutting optimization. According to the characteristics of the stone rough machining stage, making full use of the advantages of the diamond circular saw and taking into account the average material removal rate (AMRR) as the cost, we propose a cutting optimization strategy for special-shaped stone blocks with a large material removal amount and a short cutting time, so as to automatically realize removing materials with high efficiency while ensuring energy saving and emission reduction.

The main contributions in this study can be summarized as follows: (i) In order to automatically obtain a set of relatively optimal cutting planes to ensure a big removal block for each cutting, the method of constructing a cutting plane (CP), in turn, is proposed according to the geometrical characteristic of the CPH in space and with the data reconstruction of the CPH, where the octree algorithm is used to search for the removal vertices of the CPH for each cutting to reduce the amount of calculation. (ii) Except for the cutting of a big block, another key point is to reduce the cutting time, whose optimization model is thus established. The convex polygon (CPG) generated by the intersection of the CP with the parent CPH is analyzed, where the optimization objective and optimization algorithm for determining the feed direction and starting point of the cutting path are addressed to produce the shortest cutting time. From (i) and (ii), the suboptimal solution of the AMRR for block cutting is obtained.

Cutting strategies with a circular saw will play a particularly important role not only in 3D stone processing but also in wood, metal and harder diamond 3D processing. This strategy may be not very satisfactory, nevertheless, which leads us to explore more feasible geometric techniques for efficient 3D cutting, whether for convex or nonconvex polyhedrons.

The rest of this paper is organized as follows: In Section 2, preliminary information regarding the block cutting mechanism is described. Section 3 addresses the reconstruction description of the CPH with vertex–face information. Section 4 investigates the scheme to design cuttable big blocks by constructing the CP. In Section 5, the cutting time optimization method is presented by considering the geometrical analysis of the cutting path. Before the conclusion in Section 7, validation studies are addressed in Section 6.

2. Preliminaries: Block Cutting Mechanism

The definitions of the symbols and units used are shown in Table 1.

Table 1. Nomenclature.

Symbols	Quantity	Unit	Symbols	Quantity	Unit
Q_0	Blank box (Blank workpiece)	/	P_{csi}	Vertex set of intersection CPG	/
Q_{i-1}	Parent CPH	/	S_i	The i th intersection CPG	/
Q_i	Child CPH	/	T	Homogeneous transformation matrix	/
Q_M	Final CPH	/	R	Radius of sawblade	mm
Q_{ri}	Removal block by the i th cutting	/	v_f	Feeding speed	mm/min
π	Cutting plane	/	V_{ri}	The i th material removal volume	mm ³
π_i	The i th CP	/	T_i	The i th cutting time	min
P_{i-1}	Vertex set of Q_{i-1}	/	W	Width of CPG	mm
P_{ri}	Removal vertex set of the i th cutting	/	L_i	Cutting path	mm/min

In the rough machining process of removing materials, due to the significant differences in shape between the original blank and the final product, rapidly removing most of the extra materials on the original blank to form a rough blank has become one of the most important processes. For any 3D special-shaped product, in the rough machining stage of removing materials, it can be representatively wrapped as a compact bounding

sphere (BS) or bounding ellipsoid (BE) to perform further cutting exploration. A BS or BE is selected depending on the 3D shape of the special-shaped product. If the product is a thin, long shape we would choose the corresponding blank box and a BE as a target to perform the rough cutting, conversely, for a short, round shape we would choose a BS. We would hope not to have any unnecessary material consumption. For instance, in Figure 1, the 3D symmetrical penguin and its BS are shown simultaneously.

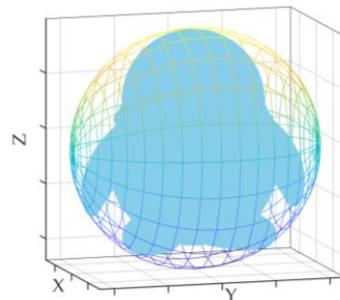


Figure 1. 3D penguin and its BS.

Hence, when confirming the machining allowance, this paper considers the compact BS to be a typical target shape for the rough cutting of a blank CPH (a blank box), Q_0 , which reflects the oriented bounding box (OBB) of the BS. This guarantees the efficient removal of the extra materials without overcutting and excessive material consumption. In the blank CPH cutting, we need to design the optimal cutting strategy to cut out the final CPH, Q_M , from Q_0 to approximate the target BS, as shown in the cutting schematic diagram in Figure 2, thus satisfying the characteristics of the circular saw processing. To solve this problem, the cutting process should be analyzed and monitored, not only combining the theory from computational geometry and computer graphics, but also taking into account the technique of the processing plan and design, which may bring complexity and challenges to the design and implementation of the cutting strategy. If π_i denotes the i th CP, the cutting optimization process reflects a series of optimization processes of π_i intersections with the CPH and the optimization process of each cutting path. The problem of the material removal following cutting can be described as follows:

- Reconstruction description of the CPH: During a continuous block cutting process, it is inevitable to be confronted with the problem of the CPH reconstruction and intersection calculation between a plane and the CPH combining with computational geometry. Therefore, it is important but challenging to update the CPH information including removal vertices, intersection vertices, intersection faces and updated faces using an appropriate, available data structure. We hope these can be efficiently implemented to carry out dynamic data storage so that any queries about the workpiece during the cutting process can be answered with the need for less data and calculation. Meanwhile, the visual image of the live cutting can be displayed to demonstrate the reconstruction of the CPH after each cutting.
- Design cuttable big blocks: For block cutting, aiming for the highest efficiency for the removal of materials, it is better when removal blocks are designed larger, and as far as possible, with the minimum number of cutting times. The difficult question is how to construct the CP so as to determine the angle of the circular saw, which is related to the vertices of the blank CPH and the BS surface.
- Optimize cutting time: The cutting time under the condition with a constant feed speed can be confirmed by investigating the cutting path. Therefore, in order to obtain the shortest cutting time, it is necessary to analyze the geometrical characteristics of the CPG generated by the intersection of the CP with the CPH, so as to calculate the cutting feed direction and cutting point of the shortest path under constraints.

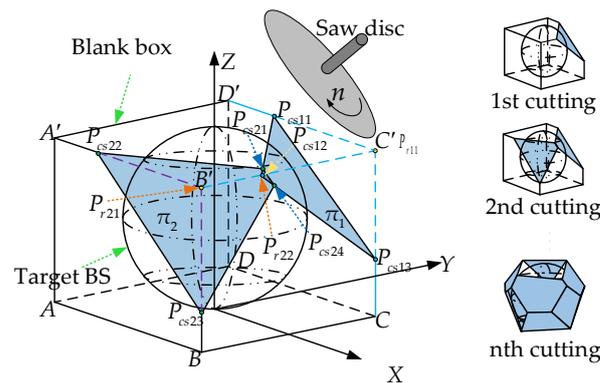


Figure 2. Schematic diagram of block cutting.

In the real machining process, it is necessary to comprehensively consider the machining efficiency and machining allowance alongside the capacity of the machine tools or robots. Therefore, the optimization strategy of 3D block cutting discussed in the paper makes the following assumptions: (1) The BS of the special-shaped product is regarded as the shape of the target object; (2) the convex polyhedral block is cuttable by a circular saw when its angle satisfies the geometrical constraints of the mechanism of the machine tools or robots and it is set up properly. (3) In the cutting process, there is no interference between the cutter and the stone. Moreover there are no obstacles around.

3. Reconstruction Description of the CPH

The essence of the removal of material when block cutting is found in the process of the intersection of the CP with the parent CPH to produce the child CPH. The vertices, edges and faces of the CPH will be updated dynamically in the cutting process, which is a complex process with tremendous and heavy computation and data storage [14,15]. In order to realize continuous cutting and the dynamic visual effect automatically, the updating of information with a relatively simple data structure is introduced to reconstruct the CPH in the updating process.

3.1. Data Structure of CPH

In the cutting process, updating the polyhedron experiences tedious and algorithmically complex updates of the data structure, which is used to describe the significant geometrical features of the convex polyhedron and bounding sphere [14]. As the cutting is completed, the cut edge causes changes in the number of faces, edges and vertices on the child polyhedron. In the literature, some data structures for a polyhedron in 3D space have been proposed, such as single level, 2 or 3 levels or half-edge data structures [15,16], which are for a face list, vertex–face list, vertex–edge–face list or doubly connected edge list. In order to realize the dynamic storage, querying and management of polyhedron data efficiently, a double level data structure for the vertex–face list is established, as shown in Figure 3, to describe the geometrical characteristics of the polyhedron, which retains the vertex–face information to guarantee that the volume of the CPH can be calculated and the cutting calculation can be implemented dynamically with a relatively small calculation and amount of storage. Here, every vertex and face of a polyhedron are indexed separately and the array data of each face lists the allocated vertices of each face so that they meet in a counter clockwise (CCW) order. Namely, the vertices are listed in a CCW order and the array of face lists is filled with the index of the vertex list.

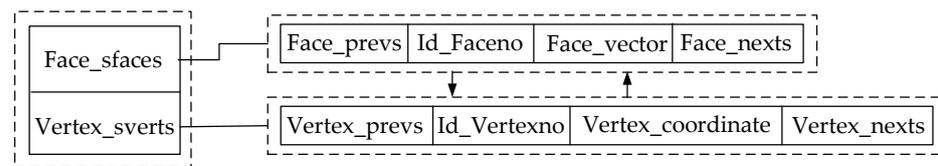


Figure 3. Vertex–face data structure.

3.2. CPH Model Reconstruction

Based on the vertex–face data structure, the reconstruction process of the CPH is as follows: Let $i = 1, 2, \dots, M$ denote the number of cuts. Given $(i - 1)$ th vertex set $P_{i-1} = \{P_{i-1,1}, P_{i-1,2}, \dots, P_{i-1,k} \dots\}$ of the parent polyhedron Q_{i-1} and the i th CP π_i in 3D space, where k denotes k -th vertex. It is then possible to compute renewed vertex and face lists of the cutting process when the removal vertices and intersection face of Q_{i-1} are confirmed; in this way, we complete the dynamic data management of the continuous cutting process and the quantitative evaluation of the cutting algorithm. The updating process follows three steps:

Step1: Judge removal vertices

In order to yield the update vertices in the cutting process, the first thing is to judge the vertices to cut off. Intuitively, from the schematic diagram of block cutting as shown in Figure 2, π_i for each cutting divides the whole space into two half spaces [15,17]. According to binary space partition algorithm (BSP), the positive and negative half space can be defined by the normal vector of π_i as the boundary. The half space pointed by the normal vector is the positive half space, which is the half space to be removed, and vice versa, the negative half space is the reserved CPH part, which is the child CPH. Aiming at vertex set $P_{i-1} = \{P_{i-1,1}, P_{i-1,2}, \dots, P_{i-1,k} \dots\}$ of the parent CPH, the vertex partition equation is established as follows

$$\begin{cases} \pi_i = K_{Ai} \cdot x + K_{Bi} \cdot y + K_{Ci} \cdot z + K_{Di} \\ \pi_{ik} > 0, P_{i-1,k} \in P_{ri} \\ \pi_{ik} \leq 0, P_{i-1,k} \notin P_{ri} \end{cases} \quad (1)$$

where $K_{Ai}, K_{Bi}, K_{Ci}, K_{Di}$ represent the equation coefficients of π_i ; π_{ik} are the solutions of the plane equation for each vertex of Q_{i-1} ; hence, the removal vertex set $P_{ri} = \{P_{ri1}, P_{ri2}, \dots, P_{rin}\}$ by the i th cutting can be obtained, and n is the number of vertices to be cut off.

Step2: Calculate intersection vertices (CPG)

If one knows the vertices that should be removed, the edges that should be cut can be deduced. This means that the CP can be calculated based on some of that information. For each cutting, one CP intersects with one CPH, then an intersection CPG is generated. Calculating CPG vertices need to judge the edges of the CP intersecting with the parent CPH and find its intersection. As can be seen from Section 3.1, the data structure stores the vertex–face list, ignoring the storage of the edge list. However, the face list is composed of vertices allocated by the right-hand rule. Therefore, we can connect two adjacent vertices in the face list to determine the edges. To judge whether the edge of the parent CPH intersects with π_i , Equation (1) can be used. If the two vertices of the edge are located on the positive and negative half space of the CP separately, the edge is intersected by the CP. Otherwise, there is no intersection. In addition, by vector parallel condition, the equation of the intersection edges of the CPH can be written as follows

$$\frac{(x - x_{iq})}{m_{iq}} = \frac{(y - y_{iq})}{n_{iq}} = \frac{(z - z_{iq})}{p_{iq}} \quad (2)$$

where (x_{iq}, y_{iq}, z_{iq}) is any vertex on the intersection edge E_{iq} of the parent CPH, $\vec{s}_{iq} = (m_{iq}, n_{iq}, p_{iq})$ is the direction vector of E_{iq} by calculating two adjacent vertices, and $q = 1, 2, \dots, h$ represents the number of the intersection edges of i th cutting.

For the intersection edges of the parent CPH and the CP, the intersection point can be solved by synthesizing Equations (1) and (2). The vertex set of the intersection CPG is expressed as $P_{csi} = \{P_{csi1}, P_{csi2}, \dots, P_{csih}\}$. It is worth noting that since each edge of the CPH is shared by two faces, they intersect with the CP to obtain the intersection point. After yielding the intersection point, we first store and query the intersection calculation in the temporary list before obtaining all intersection points in one cutting, then update the vertex list to avoid redundant and incorrect calculations.

Step3: Update faces of child CPH

Based on steps 1 and 2, the faces of the child CPH can be constructed. As can be seen from Figure 2, the vertices in the negative half space of π_i and the intersection points in the parent CPH constitute the vertex list of the child CPH. Hence, one can see that the faces in the child CPH can also be divided into two categories: one is the original face (remained face, e.g., $A'D'DA$ in Figure 2) or a part of the original face (renewed face, e.g., $A'ABP_{cs23}P_{cs22}$ in Figure 2) of the parent CPH, and the other is the new face of the intersection CPG, e.g., $P_{cs21}P_{cs22}P_{cs23}P_{cs24}$ in Figure 2. To construct the first type face, we need to delete the vertices of the parent CPH on the basis of steps 1 and 2, and reserve or add the vertices that do not belong to the data set P_{ri} to the child CPH. If the adjacent vertices lie on both sides of the CP, the indices of the intersection point $P_{csi} = \{P_{csi1}, P_{csi2}, \dots, P_{csih}\}$ are added to the face list of the child CPH replacing data information of all P_{ri} and replenishing renewed vertex indices in the corresponding face list.

To construct the second type face, since a disordered vertices set of the intersection CPG has been obtained in step 2, and the data structure follows CCW order rule, it is necessary to reorder the CPG vertices obtained in step 2. It is difficult to sort a random point with a feature in 3D space even if they are in one plane. Therefore, utilizing a uniform linear coordinate transformation method, we hope that the 3D data description of the CPG vertices can be converted into a 2D data description in a plane to analyze where the centroid (x_c, y_c, z_c) of the transformed CPG coincides with the coordinate origin. If the normal vector N_i of the CPG, as shown in Figure 4, which is calculated from the vertices of CPG, is not perpendicular to any plane of XOY , YOZ and XOZ in frame O of CPH, it is possible to express the coordinate transformation between the frames O and O' . Let the centroid O' be the origin of the frame of the CPG, and the normal vector N_i be a coordinate axis. Referring to Figure 4, frame O' is obtained from the frame O by translating it along X , Y and Z axes by x_c, y_c and z_c , respectively, followed by two times rotation of β about Z and γ about X with respect to the current frames. Therefore, the 4×4 homogeneous transformation matrix can be written as $T = T_{P_c} \cdot R_\beta \cdot R_\gamma$ where

$$R_\beta = \begin{bmatrix} \cos \beta & \sin \beta & 0 & 0 \\ -\sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_\gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma & 0 \\ 0 & -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{P_c} = \begin{bmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

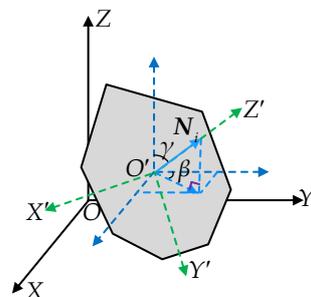


Figure 4. Coordinate description.

R_β , R_γ and T_{P_c} denote the rotation and translation transformation matrices separately. Completing the transformation to reach frame O' from frame O , the vertices of CPG lie in

plane $X'O'Y'$, $Y'O'Z'$ or $X'O'Z'$ of the coordinate system with the centroid origin, so that its vertices can be sorted simply according to the angle between the vector of the origin to the corresponding vertex and the Z' , X' or Y' axis. Otherwise, the vertices of the CPG can be sorted directly based on the axes of the CPH without coordinate transformation. At this point, the vertex list of the second type face has been updated completely.

The dynamic updating algorithm of the CPH, i.e., the CPH list updating, can be summarized as follows (Algorithm 1):

Algorithm 1 CPH List Updating.

Input: Vertex-Face List of Q_{i-1} , π_i

Output: Vertex-Face List of Q_i

```

1: function ConstructCPHList(Vertex-Face List of  $Q_{i-1}$ ,  $\pi_i$ )
2:   // Construct the Vertex-Face List of the first type face of  $Q_i$ 
3:   for all  $F$  in Face List of  $Q_{i-1}$  do
4:     for all  $vertex_{i-1,j}$  in  $F_m$  do
5:        $vertex_{i-1,j} \leftarrow$  Vertex of  $Q_{i-1}[F_m[j]]$  //Index the corresponding vertex
6:        $vertex_{i-1,j+1} \leftarrow$  Vertex of  $Q_{i-1}[F_m[j+1]]$ 
7:       if  $\pi_i(vertex_{i-1,j}) \leq 0 \wedge \pi_i(vertex_{i-1,j+1}) \leq 0$  then //Two vertices are in negative half-space
8:         Vertex List of  $Q_i \leftarrow vertex_{i-1,j}, vertex_{i-1,j+1}$ 
9:         Face List of  $Q_i \leftarrow$  indexes of  $vertex_{i-1,j}$  and  $vertex_{i-1,j+1}$ 
10:      else if  $\pi_i(vertex_{i-1,j}) \leq 0 \wedge \pi_i(vertex_{i-1,j+1}) > 0$  then //Two vertices lie on either side of  $\pi_i$ 
11:         $E_{i-1,j} \leftarrow$  line( $vertex_{i-1,j}, vertex_{i-1,j+1}$ )
12:         $P_{csij} \leftarrow E_{i-1,j} \cap \pi_i$  //Find the intersection of  $E_{i-1,j}$  and  $\pi_i$ 
13:        if  $P_{csij} \notin T empList$  then
14:           $T empList \leftarrow P_{csij}$  //  $T empList$  is used to store the  $P_{csij}$ 
15:          Vertex List of  $Q_i \leftarrow P_{csij}$ 
16:        end if
17:        Face List of  $Q_i \leftarrow$  index of  $P_{csij}$ 
18:      end if
19:    end for
20:  end for
21:  // Construct the Vertex-Face List of the second type face of  $Q_i$ 
22:   $[N_i] \leftarrow$  ComputeNormalVector( $P_{csi}$ )
23:   $[P_{csi}] \leftarrow$  3DCoordinateTransformation( $N_i, P_{csi}$ ) //3D Coordinate transformation for vertex sorting
24:  Face List of  $Q_i \leftarrow$  index of  $P_{csi}$ 
25:  return Vertex-Face List of  $Q_i$ 
26: end function

```

4. Design Cuttable Big Blocks

In order to realize a high efficiency of cutting, we propose a strategy including a set of reasonable cutting schemes to ensure a large amount of cutting materials with a shorter cutting time. The cost of one cut is the MRR (i.e., the ratio of removal block volume to cutting time) originated by the saw disc intersecting with Q_{i-1} . Our objective is to find a series of cuts whose total cost-AMRR is relatively large. First, in order to obtain a large material removal amount, according to the geometrical characteristics of the symmetric convex bounding box and the BS, the removal vertices of the CPH are searched whose distance from the surface of the BS is at maximum. The direction is regarded as the normal vector of the CP for each cut. After that, the CP is constructed based on the extracted normal vector at the corresponding tangent point on the surface of the BS. It is noted that, in this process, to ensure a small machining allowance and no overcut at the same time, the data amount of the BS saved in the triangular mesh format with a certain accuracy is not optimistic. In particular, due to the increase in the convex vertices after being cut again and again, the search process causes a problem by requiring a large amount of calculation, thus increasing the burden of searching. The octree decomposition of the 3D model to reduce the computational complexity has been considered in many applications [18–20], such as image processing, collision detection algorithms, mesh generation procedures, and so on,

which allows the search time to be easily reduced. For this reason, for each cutting, the algorithm of the octree partition is introduced to divide the OBB and BS into subdivisions at the same time, in order to search for the vertices of the CPH that are to be removed, which will reduce the load of the calculation and increase the speed and the search efficiency.

4.1. Octree Space Partition

To build an octree for the given set of 3D mesh points in the geometric space, firstly, it is necessary to decide the root node, which can be defined as an OBB. Then, we can subdivide it into multi-level equally sized cubes, called octants as shown in Figure 5, where each cube region in the space is a node of the octree [18,21]. Different from the binary tree, in which each node has two branches, each node of the octree has eight branches. The octree representation of 3D objects recursively subdivides the root cubic data into eight sub-cube arrays.

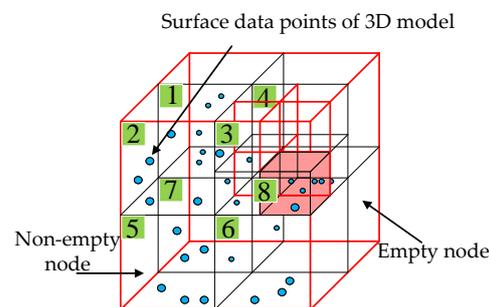


Figure 5. Schematic diagram of space partition.

Based on the above description, we need to build a compact OBB wrapping the BS with a reasonable machining allowance. Generally, the size of the blank workpiece is selected as the size of the OBB. From this point, a compact CPH Q_M , bounding the target BS, can be cut out from the blank workpiece (box). Therefore, we obtain the root node of the octree, which contains a data set P_Σ consisting of all points of the blank workpiece and spherical shell of the BS. Then, we calculate the geometrical centroid (x_{bc}, y_{bc}, z_{bc}) of the blank workpiece, which overlaps the center of the BS. Based on this, P_Σ can be generally decomposed. The subdivision nodes of the octree are recursively divided along the three coordinate axes according to the octree depth, which are determined according to the labeled sub-cube, while any point $P(x_s, y_s, z_s)$ of the data set are subdivided into a different sub-cube according to the coordinate location. When the number of the octree level increases, the cost of the octree storage occupation and node query time consumption will also increase. Comprehensively considering the factors of the search efficiency and node query time in this paper, an octree with a depth of two levels and eight subspaces is established, which subdivides all the data in the data set into eight groups, assigning the node cubes $G_o (o = 1, 2, \dots, 8)$ of the subdivision depending on the coordinate range, respectively, as in Equation (3). Since each octree node has eight branches, it is convenient to number a child node using an appropriate index ranging from one to eight. This index denotes the subregion covered by each child. An example of the space partition of a BB and BS is shown in Figure 6, which is used to reduce the calculation amount for searching in the subdivision region. The data points in each sub-cube are shown in a different color. For complex cases of octrees with many levels, we can divide subdivision by discussing solid angles distributed in a corresponding space region.

$$\begin{aligned}
 G_1 &: \{P \in P_\Sigma : x_s < x_{bc} \wedge y_s < y_{bc} \wedge z_s > z_{bc}\} \\
 G_2 &: \{P \in P_\Sigma : x_s < x_{bc} \wedge y_s > y_{bc} \wedge z_s > z_{bc}\} \\
 G_3 &: \{P \in P_\Sigma : x_s > x_{bc} \wedge y_s > y_{bc} \wedge z_s > z_{bc}\} \\
 G_4 &: \{P \in P_\Sigma : x_s > x_{bc} \wedge y_s < y_{bc} \wedge z_s > z_{bc}\} \\
 G_5 &: \{P \in P_\Sigma : x_s < x_{bc} \wedge y_s > y_{bc} \wedge z_s < z_{bc}\} \\
 G_6 &: \{P \in P_\Sigma : x_s > x_{bc} \wedge y_s > y_{bc} \wedge z_s < z_{bc}\} \\
 G_7 &: \{P \in P_\Sigma : x_s < x_{bc} \wedge y_s < y_{bc} \wedge z_s < z_{bc}\} \\
 G_8 &: \{P \in P_\Sigma : x_s > x_{bc} \wedge y_s < y_{bc} \wedge z_s < z_{bc}\}
 \end{aligned} \tag{3}$$

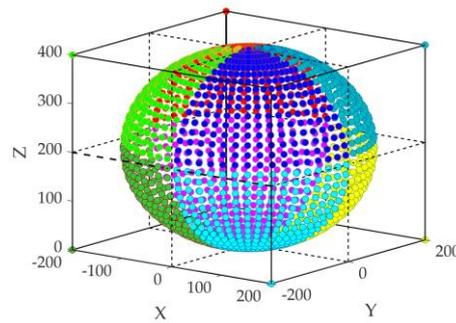


Figure 6. Space partition of the BB and BS.

4.2. Construct Cutting Plane

To achieve a short total cutting time, under the principle of “cutting big blocks as few times as possible” in the cutting process, we construct a series of CPs to ensure that the material amount is removed as much as possible. The idea is to intuitively remove the most convex vertex of the convex blocks from a practical point of view. Firstly, the maximum distance d_{imax} from the vertex of the parent CPH Q_{i-1} to the surface of the BS can be found. On the basis of Section 4.1, we can subdivide the vertices of Q_{i-1} into the sub-cube region G_o by the octree algorithm. Then, each d_o in the sub-cube G_o can be further calculated, where d_o is defined as the distance of a normal vector on the BS to the vertex of Q_{i-1} in the G_o . As shown in Figure 7, from the schematic diagram of cutting, we can see a data point on the surface of the BS is represented as P_{ti} , and a corresponding vertex of Q_{i-1} is represented as P_{vi} , where the normal vector of the tangent plane at P_{ti} is $\overrightarrow{P_{ti}P_{vi}}$ and the distance $|P_{ti}P_{vi}|$ is d_o in G_o . Choosing the maximum value of $\{d_o\}$ as the cutting thickness d_{imax} of the removal block, in that way the tangent plane of the BS at the corresponding P_{ti} is selected as the CP π_i . $\overrightarrow{P_{ti}P_{vi}}$ indicates $N_i(A_i, B_i, C_i)$ also.

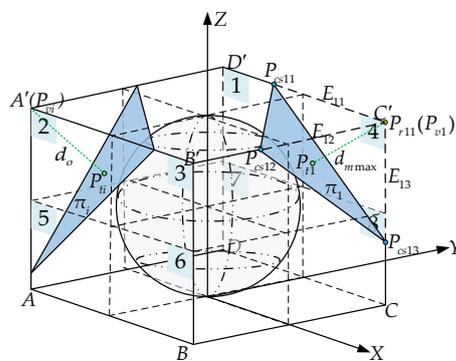


Figure 7. Diagram of constructing CP.

The coefficient of π_i in Equation (1) can be calculated based on the point P_{vi} or P_{ti} and normal vector $N_i(A_i, B_i, C_i)$. For instance, a CP π_i is shown in Figure 7 constructed in the sub-cube region G_4 . The cutting along π_i will produce a big block removal effect by

adjusting the orientation of the disc saw. Accordingly, using the CPH-updating algorithm of Section 3, the child CPH Q_i can be obtained.

4.3. Calculate Material Removal Volume (MRV)

Some methods have been proposed to calculate the volume of the CPH [16,22,23]. Combining these, we introduce a volume calculation method using vertex coordinates and face information. As above mentioned, the vertices of the removal block are $P_{ri} = \{P_{ri1}, P_{ri2}, \dots, P_{rin}\}$ and $P_{csi} = \{P_{csi1}, P_{csi2}, \dots, P_{csih}\}$; nevertheless, for the convenience of describing the calculation of the volume of the block to be removed, Q_{ri} , we assume it has n_l polygonal faces $S_{r1}, S_{r2}, \dots, S_{rl}, \dots$ each with a different amount of n_q vertices $P_{r1}, P_{r2}, \dots, P_{rq}, \dots$. The simplest contour of Q_{ri} is a pyramid feature. If Q_{ri} is not a pyramid, we can decompose the polyhedron Q_{ri} into multiple pyramids Q_{ril} with a common tip O_p in a geometrical centroid of Q_{ri} as shown in Figure 8. After that, to obtain the volume of the pyramid, if S_{rl} is not a triangle, we can take the polygon S_{rl} and decompose it into triangles from any vertex as a common vertex P_{r1} . The vertices $P_{r1}, P_{r,q-1}, P_{r,q}$ of each triangle are kept in a CCW order to the triangles with respect to their outward direction. Moreover, the triangle arrays are stored in the vertex order of CPG. Eventually, a pyramid Q_{ril} is decomposed into multiple tetrahedrons $O_p - P_{r1}P_{r,q-1}P_{rq}$ with a common pyramid tip. When O_p is assigned at the origin, one-sixth of the mixed product of $\overrightarrow{O_pP_{r1}}, \overrightarrow{O_pP_{r,q-1}}, \overrightarrow{O_pP_{rq}}$, namely, one-sixth of the dot product between the normal vector of each triangle $\Delta(P_{r1}, P_{r,q-1}, P_{r,q})$ and any vector of $\overrightarrow{O_pP_{r1}}, \overrightarrow{O_pP_{r,q-1}}, \overrightarrow{O_pP_{rq}}$, can be denoted as the determinant

$$\frac{1}{6}D(P_{r1}P_{r,q-1}P_{rq}) = \frac{1}{6} \begin{vmatrix} x_{r1} & x_{r,q-1} & x_{r,q} \\ y_{r1} & y_{r,q-1} & y_{r,q} \\ z_{r1} & z_{r,q-1} & z_{r,q} \end{vmatrix}$$

which is the volume of $O_p - P_{r1}P_{r,q-1}P_{rq}$. Thus, by the vector method the following volume equation of the pyramid Q_{ril} for the face S_{rl} can be derived

$$V_{S_{rl}} = \sum_{q=2}^{n_q} V(O_pP_{r1}P_{r,q-1}P_{rq}) = \sum_{q=2}^{n_q} \frac{1}{6}D(P_{r1}P_{r,q-1}P_{rq}) \tag{4}$$

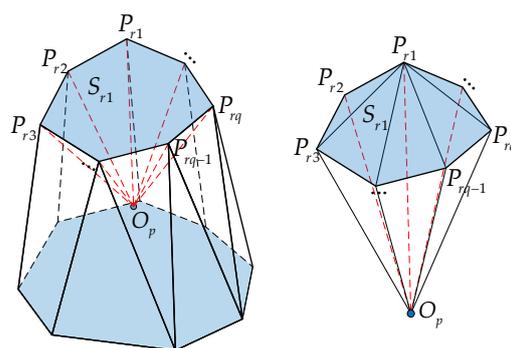


Figure 8. CPH decomposition and face decomposition.

Obviously, the volume equation of the polyhedron Q_{ri} is

$$V_{ci} = \sum_{l=1}^{n_l} V_{S_{rl}} = \sum_{l=1}^{n_l} \sum_{q=2}^{n_q} \frac{1}{6}D(P_{r1}P_{r,q-1}P_{rq}) \tag{5}$$

From the above, we can see that it is easier to calculate the CPH volume from the face index and all vertex information on the faces with less calculation. Of course, one can also

finally calculate the total removal volume directly from the remaining block. Here, in order to observe each cutting, we choose to analyze the removal block each time.

5. Optimize Cutting Time

When the material removal volume is constant, the shorter the cutting time is and the greater the MMR is. In order to minimize the cutting time, we summarize the cutting process into the second problem finding the shortest path. By analyzing the geometrical characteristics of the CPG, a path optimization algorithm to determine the feed direction and cutting point is proposed.

5.1. Time Optimization Model

By cutting with a guillotine style accompanying the aforementioned strategy, the CPH is completely separated into two convex blocks: one is the removal block Q_{ri} and the other is the child CPH Q_{i-1} . To sum each V_{ci} of Q_{ri} , we can obtain the total MRV. Due to the cutting block being relatively big, we establish the objective function with minimum cutting time as follows

$$\begin{aligned} \min \sum_{i=1}^M (T_i), \quad T_i &= \frac{L_i}{v_f} \\ \text{s.t. } \pi_i &= K_{Ai} \cdot x_i + K_{Bi} \cdot y_i + K_{Ci} \cdot z_i + K_{Di} \\ Q_0 &= \bigcap_{m=1}^z F_{0m} \\ \{S_i, Q_i\} &= Q_{i-1} \cap \pi_i \\ L_i &= W(S_i) \\ L_i &< R \end{aligned} \quad (6)$$

where M is the aforementioned number of cuts, generated automatically according to the required material removal percentage (MRP). T_i is the time consumed at i th cutting. L_i is the feed stroke along each cutting path. v_f is the feed speed, which is normally set as a constant. R is the radius of the circular saw blade. F_{0m} represents a face of the Q_0 , and m is the face number of the initial blank box, selected as six here. S_i is the intersection CPG generated by the i th cut. The width W of S_i needs to be calculated by the optimization algorithm.

Assuming the time consumption of the motion in space is ignored, the cutting time is only related to the feed stroke along the intersection with the CPG, which is decided by the geometrical characteristics of the CPG satisfying the shape constraint of the circular saw at the same time.

5.2. Intersection CPG Analysis

It can be seen from Section 5.1 that in order to obtain the shortest cutting time, it is necessary to seek out the shortest cutting path on the CPG. Through analysis, it can be found that here the path is generally characterized by the span of the CPG along a certain direction. Therefore, confirming the shortest cutting path can be worked out using the minimum span (i.e., width) of the CPG [24]. In recent years, the width calculation of the CPG has been widely used in collision detection and other calculations [19], but few researches and applications have explored its use in the field of machining. For the intersection CPG S_i , its width is defined as the minimum distance between the supporting parallel lines of S_i , decided by the vertex–edge (V–E) pairs here, in which the relative edge is formed by the intersection of two faces. The vertices set of the CPG is $P_{csi} = \{P_{csij}\}, j = 1 \cdots h$. The width calculation is as follows. Utilizing the supporting parallel lines, the V–E pairs can be scanned to obtain the maximum distance $D_j = \max\{D_{j,j+1}\}$ between any vertices of P_{csi} and one edge or extended edge $E_{j,j+1}$ in $O(h)$ time for each edge, which is the span of one edge. Once the scan is completed, we can compare each of these pairs $\{D_j\}$ and note the smallest span distances, $\min\{D_j\}$. As a result, that distance is the width $W(S_i)$, whose direction decides an initial feed direction. Here, the corresponding vertex is represented by P_{dj} , and the corresponded edge is E_w . For instance, in Figure 9, for the intersection CPG $P_{cs1}P_{cs2}P_{cs3}P_{cs4}$, the width is $D_{4,3}(W)$.

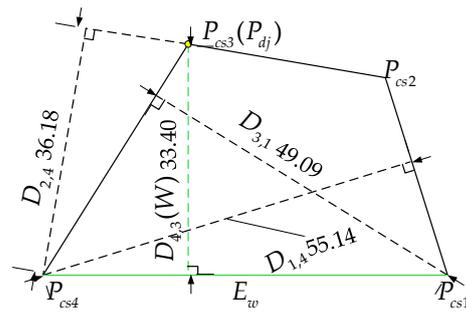


Figure 9. The width of intersection CPG.

5.3. Cutting Path Optimization

According to the analysis of the CPG in Section 5.2, if the width direction of the CPG is used as the cutting feed direction, and the relative vertex and the perpendicular point on the edge are chosen as the starting point and ending point of the cutting, respectively, then sometimes, the cutting requirements are unable to be met. In other words, the cutting range of the circular saw is not able to cover the whole CPG area without a guillotine cut; an example as shown in Figure 10a. Therefore, a bounding rectangle method of the CPG is proposed to determine the feed direction and starting point for the cut, as shown in Figure 10b, where the black dash line is the bounding rectangle of the CPG, and the red dash line demonstrates the feed direction. From the length E_w and width W of the bounding rectangle, we can calculate its centroid O_r . Along O_r , the vertical line to E_w can be drawn, intersecting E_w with point P_s , and intersecting the opposite edge with point P_e . P_s and P_e are chosen as the starting point and the ending point of the cut respectively. The vector direction from P_s to P_e is the optimal feeding direction here. If the cutting with P_e fails to cut through the block, a certain cut depth compensation should be considered to recalculate P_e , which can be found by calculating the intersection chord length between the edge $P_e P_{dj}$ and the circular saw, achieved by using the bottom vertices of the polygon.

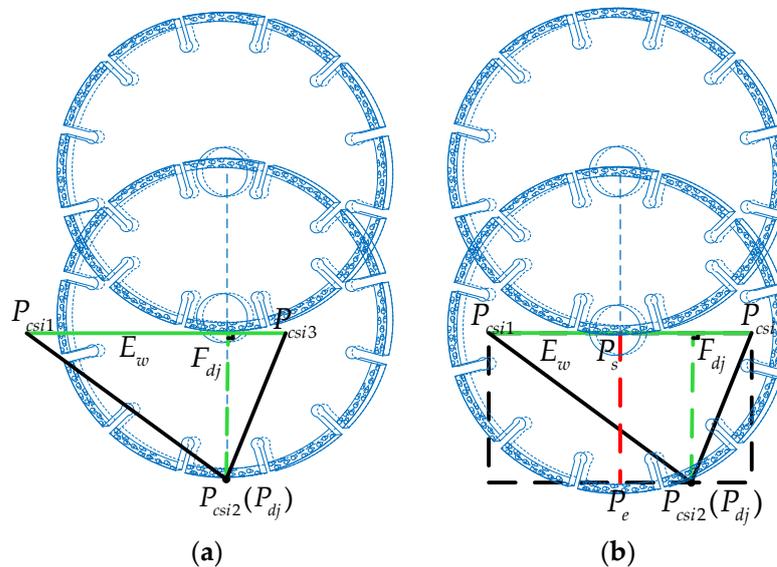


Figure 10. Feed direction of cutting. (a) Feed direction along W ; (b) feed direction along $\vec{P_s P_e}$.

6. Simulation Results and Analysis

6.1. Simulation Verification

A minimum BS of a three-dimensional penguin is used as the target for the block cutting simulation. The triangular mesh model of the BS with a radius 200 mm is generated by using CAD/CAM software, which is stored in STL format. All of the data is loaded on to

the MATLAB platform, which is used to verify the proposed strategy, thus benefiting from the powerful computing and drawing ability of MATLAB. In order to avoid any impact on the finished machined product due to the brittle fracturing of materials in the machining process, and ensuring a certain machining allowance, we select a $400 \times 400 \times 400 \text{ mm}^3$ blank box, considering that polyhedron is able to be cut by a large enough circular saw. To verify the proposed strategy for the general cutting condition, the sawing parameters are given as follows: a rotation speed of the saw blade is 1400 r/min; the radius of the circular saw blade is 400 mm; the feed speed is set to 180 mm/min. When the MRP reaches 85% of the total materials that should be removed, the cutting search stops. Importing the data set of the BS, the data of the spatial region is subdivided according to the octree algorithm in Section 4.1. For contrast, the simulation experiments are carried out with and without the octree partition. The search time after data partition is reduced by 31.64% compared with that without data partition. Some effect graphs of the workpiece's dynamic reconstruction during cutting are shown in Figure 11, resulting in a total of 19 cuts. It can be seen from Figure 11c that the contour of the CPH after cutting is closer to the target BS. Through the visualization analysis, it can be seen that there is no overcutting phenomenon in the sawing process, and the proposed methods are feasible. Figure 12 shows the cutting time after optimization, which is less than or equal to that before optimization. Moreover, we can find that the total cutting time after path optimization is about 21.3 min, which is about 11.33% less than that without optimization. In Figure 13, the MRV based on the maximum removal thickness and MRR are shown, where, in order to show these clearly, the blue solid line and red dash line have been employed to illustrate them with different labels on the left and right longitudinal axes, respectively. One can see that after nine cuts, more than 70% of MRP has been reached. After this the increase in MRP becomes slower with the increase in cutting time, i.e., the removal volume of each cut becomes smaller. This is an inevitable result of block cutting, in which each cut leads to the rough blank moving closer and closer to the target BS. By balancing the cutting times and the removal volume, in this study, we targeted 85% of the total MRP according to some engineering experience in the milling process [5] and the characteristics of the saw disc. After 19 cuts, the MRV reaches $2.6 \times 10^7 \text{ mm}^3$, accounting for about 85.34% of the total blank materials that should be removed. Accordingly, there is no need to set a higher desired MRP for rough cutting. Through the further analysis of Figure 13 (the 12th–14th cuts bring an increase in MRV), it can be seen that the cutting algorithm based on the maximum thickness of the removal block cannot completely ensure the complete removal of the material for each cutting, but it is nearly at maximum, which is why we also call this process the relatively big block cutting method. The reason that this phenomenon arises is because the contour of the removal block becomes more and more irregular with the increase in the number of cuts. Therefore, the maximum thickness of the removed block materials only reflects a relatively big block not the maximum removal amount.

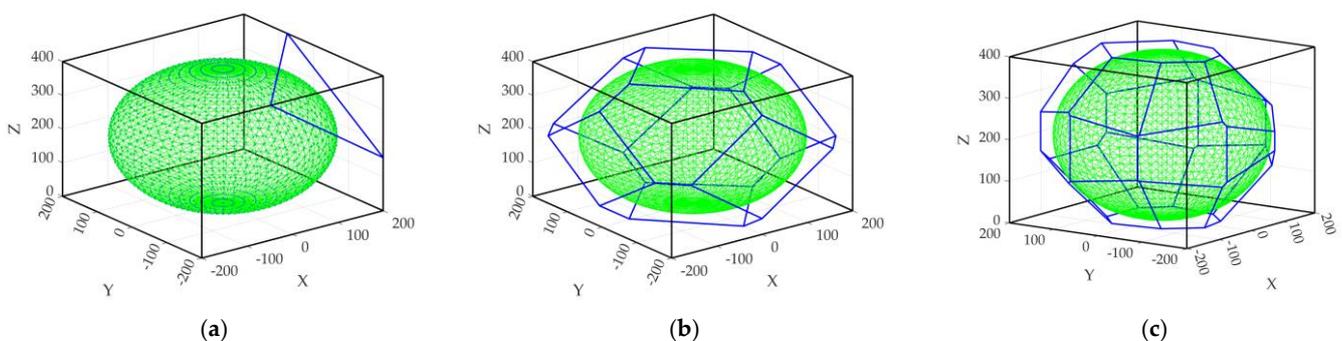


Figure 11. Dynamic reconstruction effect of CPH. (a) 1st cutting; (b) 8th cutting; (c) 19th cutting.

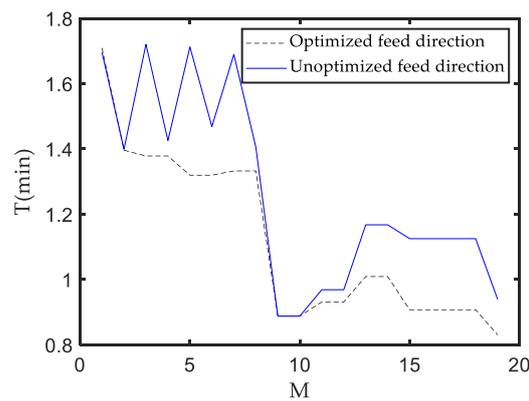


Figure 12. Cutting time.

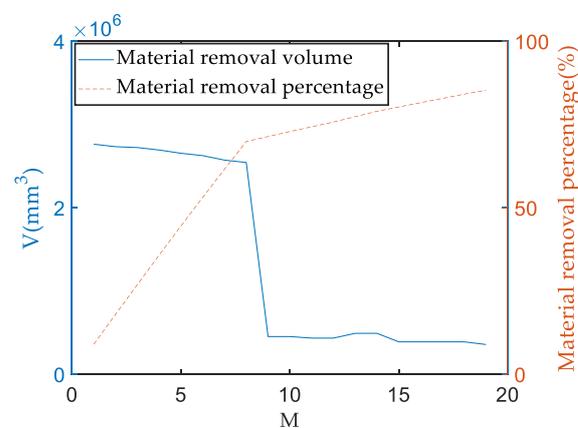


Figure 13. MRV and MRP.

Additionally, an accurate simulation model of the NC (numerical control) machine tool is established with a 5-axis cutting characteristic and circular saw blade as a simulation platform in a Vericut environment. Given the same cutting parameters as above mentioned, the cutting simulations are performed by a generated G code on the NC cutting machine model to verify the proposed cutting strategy. Some cutting results are shown in Figure 14. The cutting time displayed in the Vericut environment is about 23.4 min. Ignoring the travel time in space, it is almost the same as the cutting time in the MATLAB platform. Moreover, compared with the reconstructed CPH of each cut in the MATLAB platform, the features of each CPH processed by the 5-axis NC machine tool in the Vericut platform are roughly the same in shape and size, which further verifies the effectiveness and feasibility of the cutting strategy for removing blocks proposed in the paper.

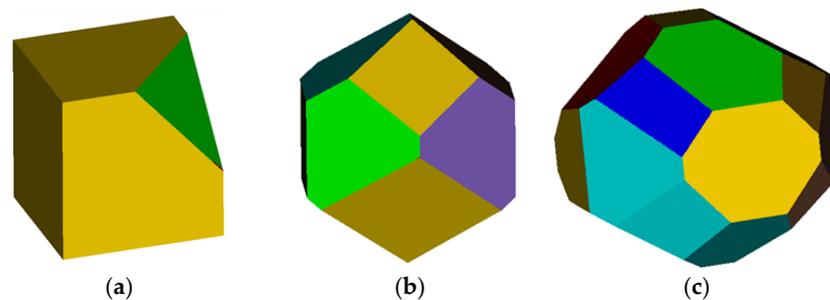


Figure 14. Cutting effect in Vericut platform. (a) 1st cutting; (b) 8th cutting; (c) 19th cutting.

6.2. Comparison of Sawing and Milling

For the removal of materials during stone rough machining, a milling mode is often employed in the newest automatic level. In order to verify the effectiveness of the block cutting strategy proposed in the paper, the milling mode with a high capacity during stone rough machining is chosen as a comparison. The milling parameters are given as follows [5]: a rotation speed of 6000 r/min; a milling depth of $a_p = 2$ mm; a milling width of $a_w = 30$ mm, which is the tool diameter of a diamond grinding wheel; a feed speed of $v_f = 4000$ mm/min. For the same MRV 2.6×10^7 mm³, the milling time is 108.4 min and the AMRR is 2.4×10^5 mm³/min. In this case, compared with the cutting results, it can be seen that the AMRR of the cutting strategy is more than five times that of the milling. The main reason of the difference is that milling is limited by its processing mode and milling depth. For the case of removing the materials of a big block, layered milling must be adopted, which consumes a lot of time and produces dust. However, the block sawing method with a saw blade can directly carry out the cutting operation with a large feed stroke and a large block thickness; thereby, the efficiency can be significantly improved. For special shapes, if a higher cutting performance saw blade [25] is employed, AMRR will be improved further. More tool cases with different machining capacities in the simulation will be performed in the future.

7. Conclusions and Future Scope

To address block cutting with a saw disc in the 3D space usually needs with a lot of hard labor and time. This article has addressed a series of works concerned with analyzing the geometrical characteristics of convex polyhedrons and convex polygons in order to complete block cutting automatically and rapidly. This has made efficient and continuous block cutting available. The optimization strategy for cutting a convex polyhedron out of a blank box has been presented by combining computational geometric theory and computer graphics knowledge. Dynamic cutting data management has been implemented with the vertex–face polyhedron data structure, which means the convex polyhedron model reconstruction is completed in the updating process. A range of cutting planes with the maximum thickness of the removal block have been constructed, and the space partition with the octree algorithm has been used in the process of searching for the vertices to be removed, which can reduce the search time. The geometrical characteristics of the convex polygon cutting plane generated by the cutting intersection have also been analyzed, and the method for an optimized cutting time has been presented. Finally, the suboptimal solution of the average material removal rate at the rough machining stage has been quantitatively analyzed. Simulation and comparison results in MATLAB and the Vericut platform have been provided to demonstrate the effectiveness of the proposed strategy. In particular, the Vericut platform is able to reflect the real processing environment. We have investigated a block cutting strategy with a practicable automatic strategy for the first time. Realistically, it is necessary to utilize multi axis machine tools or robots with good capacity.

In this work, our discussion concentrated on the problem of cutting a symmetric convex polyhedron with a bounding sphere target. However, there are some directions that can be extended further: (i) While we constructed cutting planes, we selected the suboptimal solutions with relatively big blocks and high efficiency. This leads us to think deeply about the optimal methods for constructing a cuttable maximum block for each cutting; (ii) as well as typical bounding sphere targets, we can explore the changing factors that affect the AMRR when cutting the symmetric convex polyhedron with bounding ellipsoid targets; (iii) as only one high capacity milling comparison has been completed, some different saw disc parameters and milling modes can be analyzed to obtain a more general conclusion of the higher MRR with block cutting under a similar mechanical level; (iv) instead of the bounding sphere and the bounding ellipsoid, we can explore cutting strategies according to the target polyhedrons of the compact bounding convex or nonconvex contours of special-shaped products.

Finally, we hope this work can stimulate research and applications in this field.

Author Contributions: Conceptualization, H.S. and Q.L.; writing—original draft preparation, H.S., Q.L. and Z.G.; writing—revision, H.S., Q.L. and Z.G.; supervision, H.S.; project administration, H.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Fujian Province, China, grant number 2021J01291; and the Science and Technology Planning Project of Quanzhou City, Fujian Province, China, grant number 2017T001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the research support from the Natural Science Foundation of Fujian Province, China, grant number 2021J01291; the Science and Technology Planning Project of Quanzhou City, Fujian Province, China, grant number 2017T001; the research support from Northumbria University, UK.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bai, S.W.; Qiu, H.; Rabe, L.; Jia, S.; Elwert, T.; Wang, Q.Y.; Wang, Q.; Sun, Y. A study on energy-saving optimization strategy for the stone processing industry—An improved method for modeling cutting power and energy consumption: A case study of block sawing process. *J. Clean. Prod.* **2021**, *300*, 126922. [CrossRef]
- Şirin, E.; Bonduà, S.; Elkarmoty, M. Environmental and economic optimization for block cutting of dimension stones in a limestone quarry. *Resour. Policy* **2021**, *74*, 102396. [CrossRef]
- Kim, D.; Kim, H.; Lee, S. Characterization of diamond wire-cutting performance for lifetime estimation and process optimization. *J. Mech. Sci. Technol.* **2016**, *30*, 847–852. [CrossRef]
- Kumar, V.; Saraswat, P.; Kumar, A. Experimental investigation of rotary ultrasonic face milling on red granite: A comparison with conventional grinding. *Mater. Today Proc.* **2022**. [CrossRef]
- Yin, F.C.; Ji, Q.Z.; Wang, C.Z. Research on machining error prediction and compensation technology for a stone-carving robotic manipulator. *Int. J. Adv. Manuf. Technol.* **2021**, *115*, 1683–1700. [CrossRef]
- Buyuksagis, I.S.; Rostami, J.; Yagiz, S. Development of models for estimating specific energy and specific wear rate of circular diamond saw blades based on properties of carbonate rocks. *Int. J. Rock Mech. Min. Sci.* **2020**, *135*, 104497. [CrossRef]
- Gao, Z.; Chen, M.Z.Q.; Zhang, D. Special Issue on “Advances in Condition Monitoring, Optimization and Control for Complex Industrial Processes”. *Processes* **2021**, *9*, 664. [CrossRef]
- Yamada, Y.; Sasahara, H. Free-form curves cutting using flexible circular saw. *Precis. Eng.* **2014**, *38*, 611–616. [CrossRef]
- Demaine, E.D.; Demaine, M.L.; Kaplan, C.S. Polygons cuttable by a circular saw. *Comput. Geom.* **2001**, *20*, 69–84. [CrossRef]
- Jaromczyk, J.W.; Kowaluk, M. Sets of lines and cutting out polyhedral objects. *Comput. Geom.* **2003**, *25*, 67–95. [CrossRef]
- Van Sosin, B.; Bartoň, M.; Elber, G. Accessibility for line-cutting in freeform surfaces. *Comput.-Aided Des.* **2019**, *114*, 202–214. Available online: <https://www.researchgate.net/publication/332670481> (accessed on 13 January 2022). [CrossRef]
- Dumitrescu, A. An approximation algorithm for cutting out convex polygons. *Comput. Geom.* **2004**, *29*, 223–231. [CrossRef]
- Tan, X.; Wu, G. Approximation algorithms for cutting a convex polyhedron out of a sphere. *Theor. Comput. Sci.* **2013**, *508*, 66–73. [CrossRef]
- Zhang, Q.H.; Su, H.D.; Lin, S.Z.; Shi, G.H. Algorithm for three-dimensional curved block cutting analysis in solid modeling. *Comput. Methods Appl. Mech. Eng.* **2020**, *360*, 112721. Available online: <https://www.researchgate.net/publication/337304188> (accessed on 13 January 2022). [CrossRef]
- Boon, C.W.; Houlsby, G.T.; Utili, S. A new rock slicing method based on linear programming. *Comput. Geotech.* **2015**, *65*, 12–29. [CrossRef]
- Chiodi, R.; Desjardins, O. General, robust, and efficient polyhedron intersection in the Interface Reconstruction Library. *J. Comput. Phys.* **2022**, *449*, 110787. [CrossRef]
- Chen, Z.; Tagliasacchi, A.; Zhang, H. Learning mesh representations via binary space partitioning tree networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *1*, 1. [CrossRef]
- Kim, Y.H.; Ko, S.L. Improvement of cutting simulation using the octree method. *Int. J. Adv. Manuf. Technol.* **2006**, *28*, 1152–1160. [CrossRef]
- Zhou, K.; Gong, M.; Huang, X. Data-parallel octrees for surface reconstruction. *IEEE Trans. Vis. Comput. Graph.* **2010**, *17*, 669–681. [CrossRef]
- Xu, J.; Liu, Z.; Yang, C. A pseudo-distance algorithm for collision detection of manipulators using convex-plane-polygons-based representation. *Robot. Comput. -Integr. Manuf.* **2020**, *66*, 101993. [CrossRef]
- Trinidad, M.C.; Vinacua, A.; Carruesco, A. Sweep Encoding: Serializing Space Subdivision Schemes for Optimal Slicing. *Comput.-Aided Des.* **2022**, *146*, 103189. [CrossRef]

22. Goldman, R. IV.1 Area of Planar Polygons and Volume of Polyhedra. In *Graphics GEMS II*, 2nd ed.; Arvo, J., Ed.; Academic Press/Morgan Kaufmann: San Francisco, CA, USA, 1991; pp. 169–171. [[CrossRef](#)]
23. Berg, M.; Cheong, O. *Computational Geometry: Algorithms and Applications*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–17.
24. Lassak, M. Spherical Geometry—A Survey on Width and Thickness of Convex Bodies. In *Surveys in Geometry*; Papadopoulos, A., Ed.; Springer: Cham, Germany, 2022; pp. 7–47. [[CrossRef](#)]
25. Sánchez Egea, A.; Martynenko, V.; Martínez Krahmer, D.; López de Lacalle, L.; Benítez, A.; Genovese, G. On the Cutting Performance of Segmented Diamond Blades When Dry-Cutting Concrete. *Materials* **2018**, *11*, 264. [[CrossRef](#)] [[PubMed](#)]