공학박사학위논문

Optimal generation of inner support structure for FDM type 3D printer

FDM 방식 3D 프린터를 위한 내부 지지 구조의 최적 생성

2016년 11월

서울대학교 대학원

기계항공공학부

이 주 성

ABSTRACT

Optimal generation of inner support structure for FDM type 3D printer

Jusung Lee
School of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

Among 3D printers, fused deposition modeling (FDM) type printers are widely used because of their low cost and simplicity, but their long manufacturing time remains a major concern. The widely used 2D pattern inner support structure that is uniformly produced without considering the features of an object's shape is a major cause of long manufacturing time. Therefore, in this study, the aim was to generate a new type of inner support structure to reduce the manufacturing time and material usage of FDM type printer. The proposed inner support structure has the following characteristics: efficiency, printability, self-supporting, simple path, and isotropy.

The proposed algorithm uses 3D block partitioning based on an extension of the conventional 2D pattern method. After an object is divided into unit blocks, they are merged according to the purpose of the inner support structure. The merged blocks are divided by supporting planes to ensure self-supporting and simple path properties.

In this study, the M1 method for maximizing efficiency, the M2 method for reinforcing overall strength, and the M3 method for reinforcing strength selectively were categorized according to the merging method of the unit block, and the performance of each method was confirmed experimentally. In addition, the dimensions of the unit block were extended and the extensibility and features of the extended algorithm were confirmed.

In conclusion, the inner support structures generated using the proposed methods are considerably more efficient in terms of manufacturing time and material usage, can be manufactured stably at all times, and have isotropic characteristics with respect to compressive load

Keywords: 3D printer, Inner support structure, Efficiency, Printability, Strength reinforcement, Isotropy, Fused deposition modeling

Student Number: 2010-20708

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1.

Introduction

Additive manufacturing (AM) technology is a new method of making products by building materials layer by layer, and with its advent, there has been a big change in consumer culture and manufacturing. Traditionally, consumers have been forced to buy standardized and mass-produce products, but with recent emphasis being more on the individual, businesses are focusing on the various tastes of consumers. In a market structure that requires new products to satisfy consumers' needs every day, AM technology is helping to accelerate prototyping. Unlike conventional manufacturing techniques that require diverse equipment to manufacture products, AM technology can produce diverse products with 3D printers and materials. For this reason, 3D printing has emerged as one of the next-generation manufacturing technologies and has already gone beyond the role of prototyping.

Recently, the number of companies producing 3D printers has grown as major patents related to 3D printing have expired. These companies are developing new technologies to complement existing production techniques which offers the following possibilities.

Simplification of production processes

Traditional production processes utilize injection molding, compression molding, and cutting machines to produce individual parts separately, and then assemble or weld them together. On the other hand, AM technology does not require a variety of

mechanical facilities to produce parts, but it can shorten production processes such as assembly and welding.

Personal-customized products

Since the Industrial Revolution, a market structure that produces standardized products has been in existence, and manufacturing mass-produced products in bulk is advantageous in terms of cost. Nevertheless, consumers want to buy products that fit their bodies, tastes, and styles, but are coerced into buying standardized products instead. However, it is expected that AM technology will be rapidly introduced into areas requiring customized products such as hearing aids, prosthetic legs, implants, and shoes.

Complex structural design

AM technology can produce more complex structures than traditional methods like injection molding, compression molding, and computer numerical control (CNC). For example, it is possible to produce lightweight, durable products by emulating the internal structure of the human skeleton. Furthermore, the design of clothes that cannot be produced using traditional materials could be produced using AM technology.

On-demand manufacturing

With AM technology, the manufacturing process will be on-demand so that users can

produce anything they want whenever they need it. In particular, with the storage and time constraints of space exploration, AM technology could produce alternative parts without transporting all of the necessary materials into space.

Due to the possibilities mentioned above, the 3D printer market is growing, and various kinds of printers are being developed. Moreover, the use of 3D printers in homes is also increasing. Depending on the material or manufacturing method used by a printer, they can be classified as being of either the fused deposition modeling (FDM), stereo lithography (SLA) or selective laser sintering (SLS)-type. In particular, FDM-type printers are widely used because they are simple to operate and the cost of materials is low.

However, despite the advantages of FDM-type printers, the length of time needed to produce items causes inconvenience to users; it melts filaments one line at a time and takes a long time to produce items compared to SLS and SLA which use a laser to generate one layer at a time. Therefore, research for improving the production speed of FDM-type printers is needed. The operational principles and characteristics of each method are described in the following section.

The product to be manufactured using FDM can be essentially divided into three parts: the skin, the outer support structure, and the inner support structure. The skin is a necessary part to express the shape of the object, the outer support structure is built to support overhang regions that cannot be stacked and is treated as a by-product since it is removed after manufacturing, and the inner support structure plays two

roles in ensuring the printability and the strength reinforcement of the object. Unlike the outer support structure, the inner support structure remains inside the object after the production has completed. Therefore, research to reduce manufacturing time by minimizing the inner and outer support structures is proceeding. On the other hand, research to reduce the weight of an object is progressing by minimizing only the inner support structure.

The 2D pattern method, which is mainly used for the inner support structure, is advantageous in that it is easy to generate and the fact that it is self-supporting is always guaranteed. However, when it is produced at a low density, the roof region cannot be manufactured while being stably supported. Conversely, when produced at a high density, the manufacturing time and material usage increase. In addition, the process is inefficient because the inner support structure is produced at a uniform density without considering the shape. For example, if the strength of a particular part is weak, its density is increased to reinforce the structure, which results in excessive production of the inner support structure in areas where strength reinforcement may not necessarily be needed. In conclusion, the 2D pattern method is the most simple and stable, but studies are underway to improve it because it uses materials excessively for strength reinforcement without consideration of shape.

Related studies are mostly focused on increasing the reinforcement effect of the inner support structure against the material usage, namely the strength-to-weight method. Studies on this method have focused on the characteristics of adaptive density such as the method of filling the inside structure with high density material where the

stress is high and with low density material where the stress is low through stress analysis. However, these studies do not take into account the role of the inner support structure in ensuring the possibility of manufacturing, namely printability. Even the self-supporting property of the inner support structure may not be guaranteed for the strength-to-weight method.

Therefore, in this thesis, a new inner support structure generation method to improve the manufacturing speed and reduce material usage is proposed. The proposed method satisfies the object's printability and self-supporting characteristics, and minimizes manufacturing time and material usage by generating minimal inner support structure (efficiency) while improving its isotropic properties.

1.1 Operational principles and characteristics of 3D printers

3D printers can be roughly classified into three types according to their operating principles or materials used: SLA, SLS, and FDM. Before describing the inner support structure generation method for FDM, an explanation of how it works and its advantages and disadvantages is presented. This explains why we want to apply it to FDM-type printers and what we have to consider.

1.1.1 Stereo Lithography Apparatus (SLA)

SLA is the first commercial 3D printer based on the stereo lithography technique invented by Charles Hull in 1986. In this method, an ultraviolet (UV) laser is projected onto a liquid plastic resin (photopolymer) to cure it, as shown in Fig. 1.1. In order to complete the production, a larger amount of material must be secured than the material used for the object. This is because the liquid resin fills the tank and the UV laser is projected while the bed is lowered.

SLA has the advantages of higher accuracy and a faster production time compared to FDM, but it has disadvantages such as limited materials, expensive material costs, heat distortion, and cleaning.

1.1.2 Selective Laser Sintering (SLS)

SLS was developed and patented by Dr. Carl Deckard and his academic adviser, Dr. Joe Beaman at the University of Texas in the mid-1980s. SLS sinters the powdered manufacturing material by projecting a high energy pulse laser directly onto it.

Similar to SLA, SLS uses the method of covering the material layer by layer in the form of a powder, as shown in Fig. 1.2. Unlike other types of 3D printer, support structures are not required to support the overhang or roof region because the un-sintered material supports the object. However, post-processing steps to remove un-sintered material in the object and to modify the design by generating holes in the object are required. SLS has the advantages of high precision, a variety of materials, a faster manufacturing time compared to FDM, and no supporting structure, but it is expensive and requires post-processing.

1.1.3 Fused Deposition Modeling (FDM)

FDM was developed by Mr. Scott Crump in the late 1980s and was commercialized in 1990 by Stratasys. FDM (or fused filament fabrication) is a method whereby filament-type plastic material is melted through a high-temperature nozzle and then stacked up to be bonded to the lower layer, as shown in Fig. 1.3. This method works on the opposite principle to the CNC machine, which cuts the material and shapes it. In the case of a simple shape, FDM uses only the amount of material necessary to make an object, but manufacture with the CNC machine results in considerable material loss. FDM has the advantages of low cost and simple operation compared to other 3D printing methods, but there are drawbacks such as long manufacturing time and low precision.

1.1.4 FDM’s unique features compared with SLA and SLS

FDM uses motors for moving the nozzle unlike SLA and SLS, which use lasers for rapidly curing or sintering of materials, respectively. Therefore, the more material discharged through the nozzle, the longer the manufacturing time. In addition, if the movement path of the nozzle is complicated, the manufacturing time is further delayed due to acceleration/deceleration of the motor. The manufacturing time of SLA and SLS is influenced by the number of layers governed by the height of the object, but FDM is influenced by material usage and nozzle movement. To reduce the manufacturing time, SLA and SLS simply change the height of the object by rotating it, but FDM has a more difficult problem in that it requires minimizing the inner support structure and improving the path of the nozzle. These are the reasons why the focus is on FDM in this thesis.

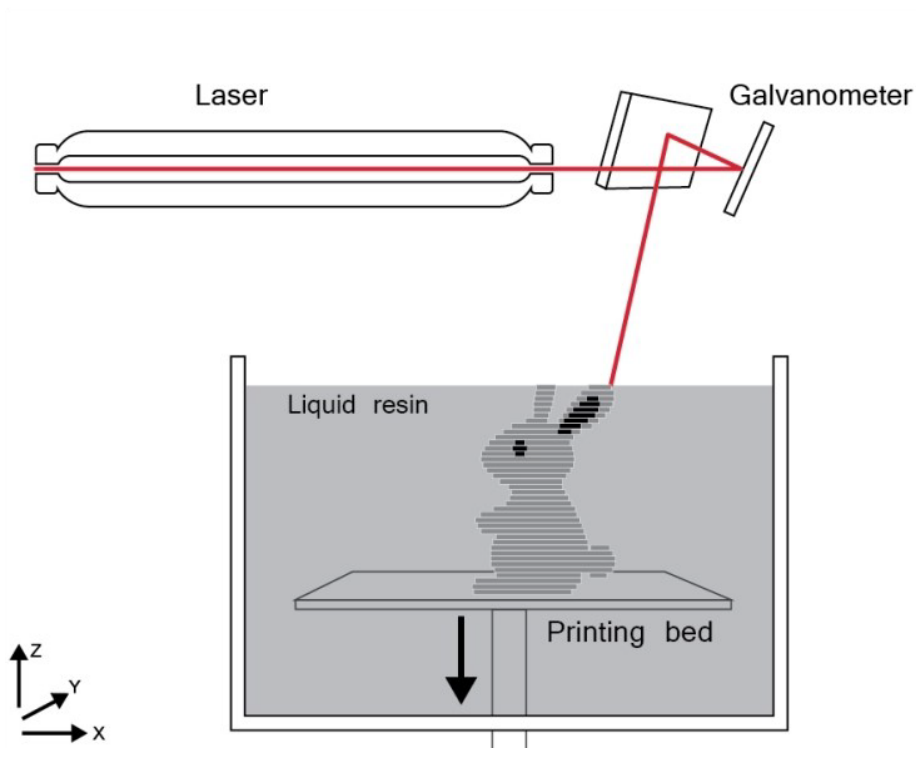


Fig. 1.1 Mechanism of the SLA-type printer

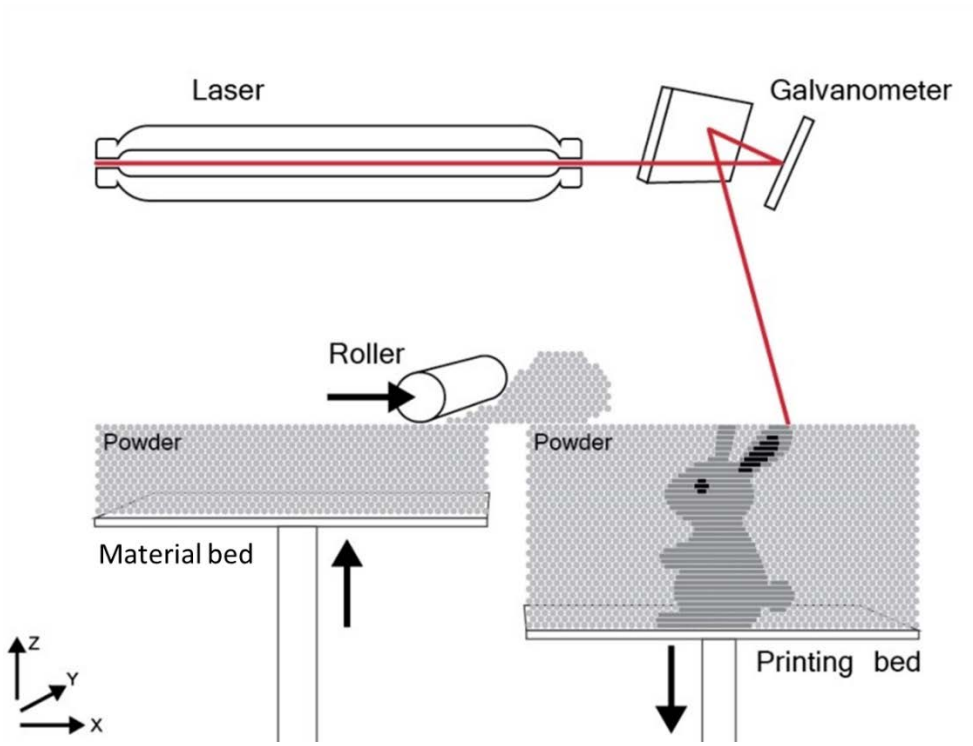


Fig. 1.2 Mechanism of the SLS-type printer

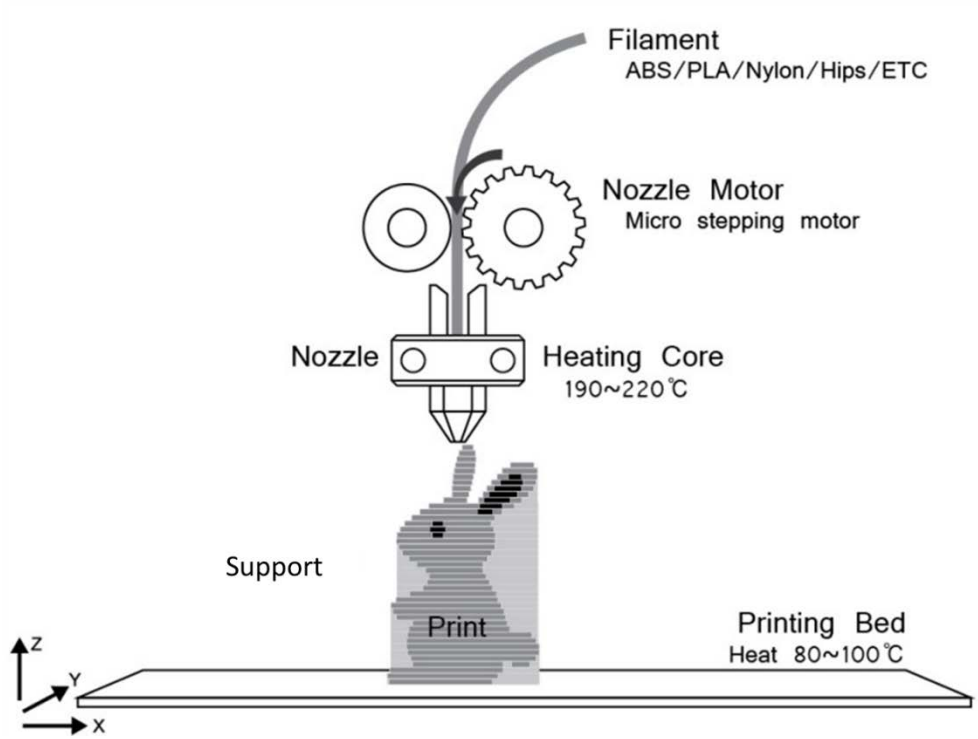


Fig. 1.3 Mechanism of the FDM-type printer

1.2 Terms used in FDM

To help understand this thesis, the main terms used for FDM-type printers are explained here.

Layered angle: when the layer is stacked, it means the angle in relation to the previous layer, and it is affected by the shape to be manufactured. As shown in Fig. 1.5, it is calculated as the absolute value of the angle between the face normal of the shape and the stacking direction vector. If the layered angle is lower than 45 degrees, there is no supporting layer under the current layer, so it cannot be manufactured stably. To prevent this, a support structure called an **infill structure** is formed inside the object (inner support structure), and another is formed outside the object (outer support structure). In general, the term layered angle is used to describe the shape of an object, but it is also used in this thesis to describe the self-supporting property of the inner support structure.

Overhang region: this is an area that cannot be stacked because there is no supporting layer below and is divided into an overhang face, overhang edge, and overhang point (see Fig. 1.4). The overhang triangle is determined based on the overhang angle.

Support structure (outer support structure): this is a structure for supporting overhang regions and is considered a by-product since it must be removed after fabrication is complete. Generally, it is the shape created by extruding the overhang region over to the next shape.

Roof region: similar to an overhang region, this is an area that cannot be stacked, is

divided into a roof face, roof edge, and roof point, and should be supported within the object. The criterion for determining the roof face is the roof angle, which is the angle between the face normal and the vector in the stacking direction (see Fig. 1.4). Generally speaking, the roof angle of the roof face is smaller than 45 degrees.

Infill structure (inner support structure): this is used to fill the inside of the object and plays the roles of reinforcing the strength of the object and supporting the roof region. Because it cannot fill the entire interior, it creates a space inside, usually using a 2D grid pattern, and is generated by controlling the spacing of the grid determined by the infill density.

Infill density: in the equations below, the infill density (d) represents the percentage of infill structure occupied by the object. Using the width (T) of the nozzle, the spacing (s) of the grid pattern is determined as follows:

$$\mathrm{d} = \frac{\mathrm{s}^2 - \frac{4sT}{2}}{\mathrm{s}^2} * 100,$$

$$\mathrm{d} = \frac{\mathrm{s} - 2T}{\mathrm{s}} * 100,$$

$$\mathrm{s} = \frac{2\mathrm{T}}{100 - \mathrm{d}} * 100.$$

The red region in Fig. 1.6 represents the region divided by the path of the nozzle, and the blue region represents the region where the material is not stacked.

Tool path: The tool path of the nozzle created by the slicing engine is stored in the form of G-code and consists of an inset, skin, infill, and support. An inset is obtained by offsetting the outline polygon obtained by slicing the model, and two insets are generally used. On the G-code viewer, the outermost inset is displayed as a red line

and the other as a yellow line; they are related to the thickness of the X and Y axes of the model. The skin is related to the thickness of the Z axis of the model and is obtained using the difference operator of the outline polygons. It is displayed in purple on the G-code viewer. In addition, the tool path of the infill in the current layer is shown as a green line, and the infill of previous layers is shown as light green.

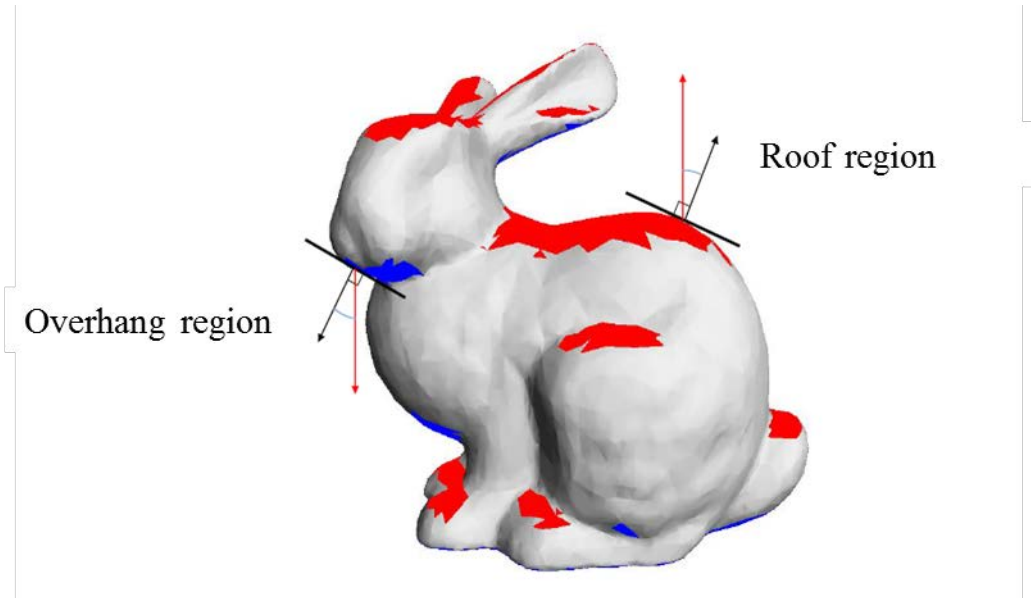


Fig. 1.4 Definition of the roof region (red) and the overhang region (blue)

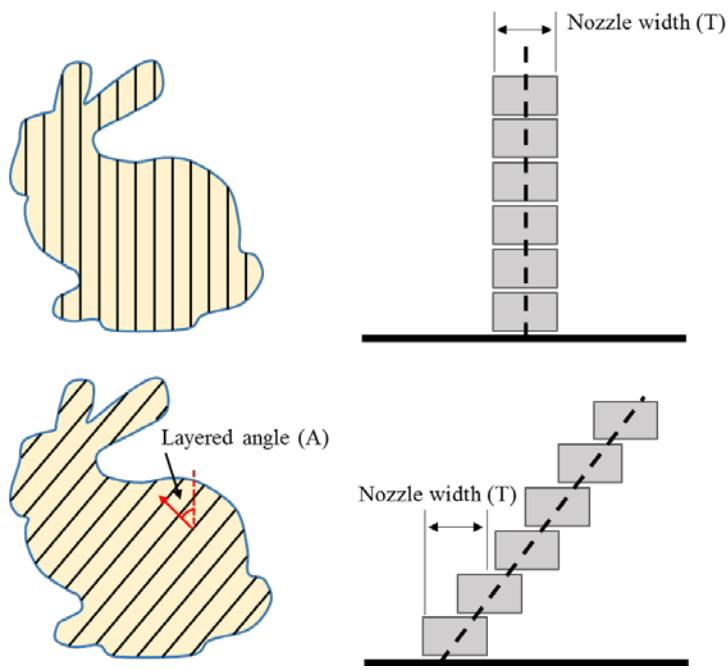


Fig. 1.5 Layered angle of the infill structure

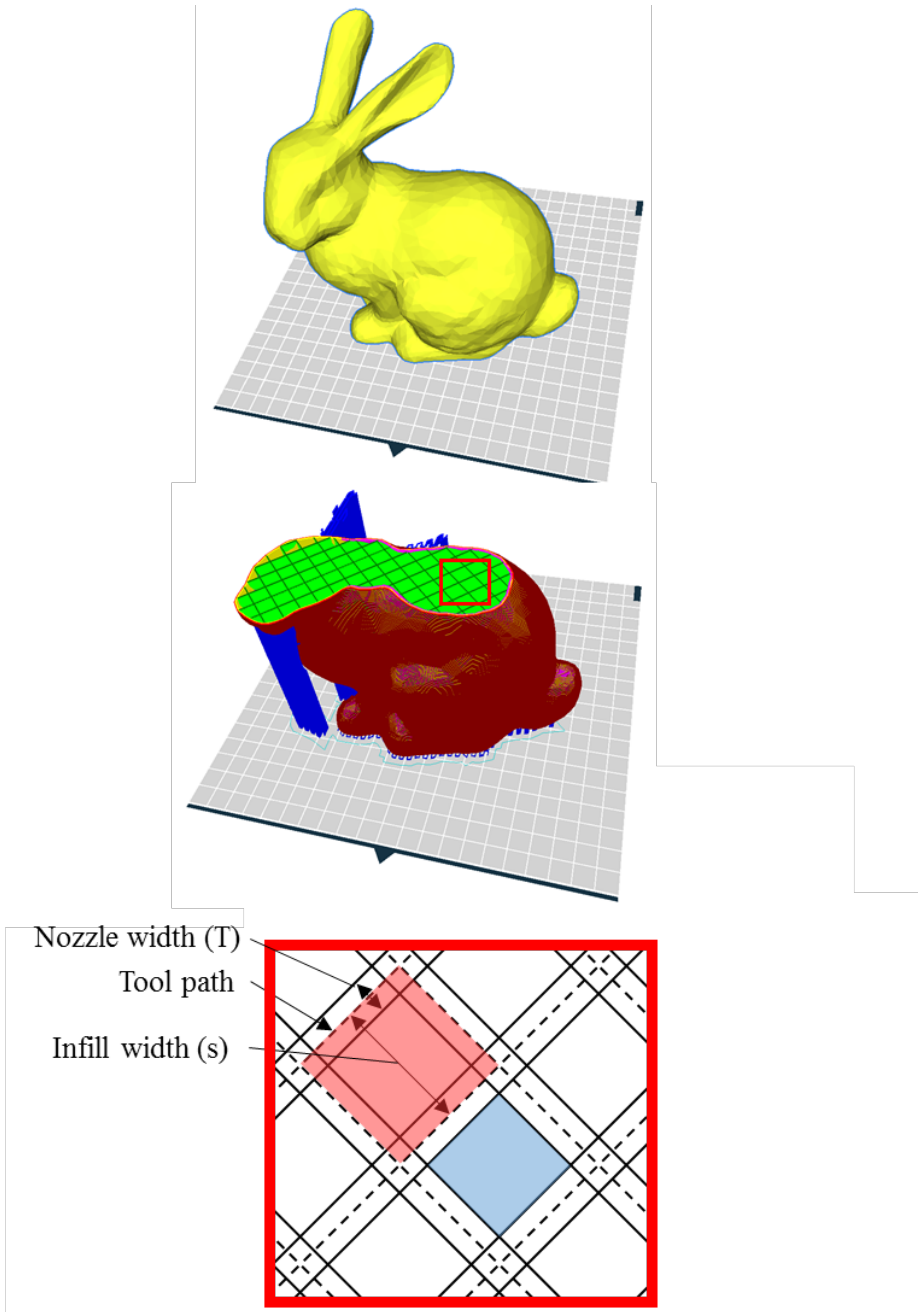


Fig. 1.6 Method to determine the width of the infill structure using the nozzle width and the infill density

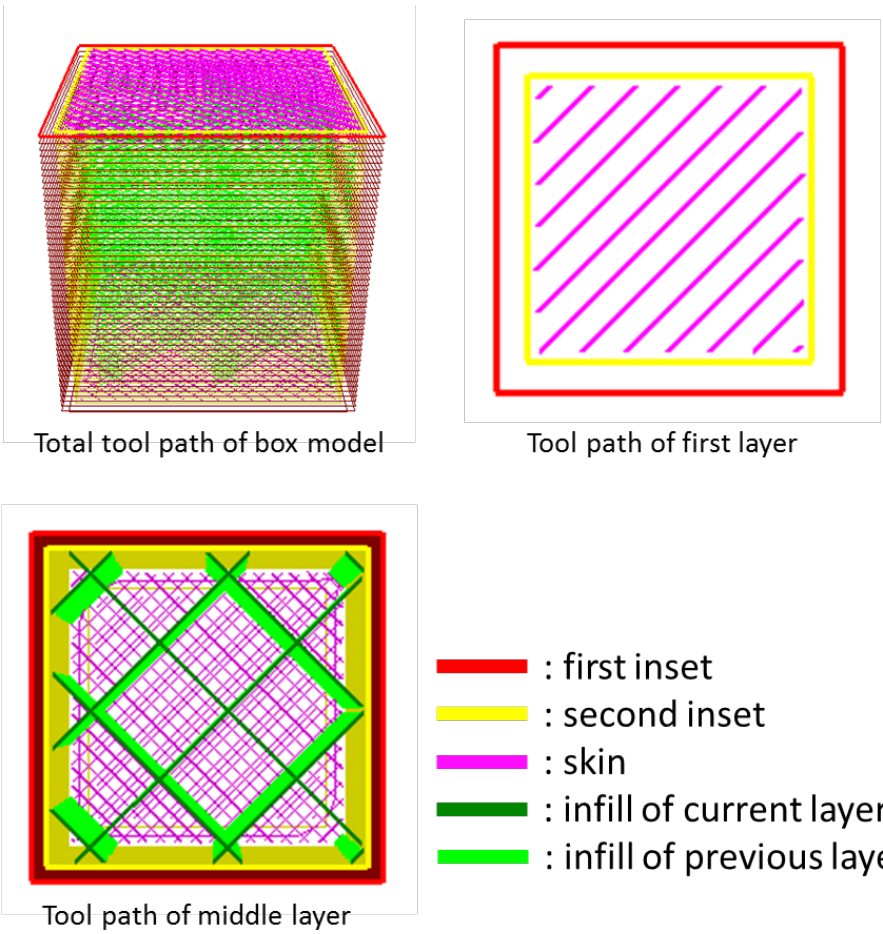


Fig. 1.7 Components of the tool path

CHAPTER 2.

Related work and research objectives

2.1 Related work

Recently, as interest in AM technology has increased, various studies have been conducted related to AM technology on various topics, including inner support structures, outer support structures, object segmentation, stacking direction determination, balancing of objects, and deformation properties using microstructures. In this chapter, the studies on each sub-field are explored.

2.1.1 Inner support structure (infill structure)

Most studies about infill structure focus on the reinforcement effect, and there are a number of researchers that have tried to produce infill structure that is strong against external loads while using a minimal amount of material. For this purpose, the studies about inner support structure mainly use the method of changing the density of the infill for the stressed area by performing the finite element method among other analytical methods [1-3].

The most straightforward method is to make the object hollow [4], but this is difficult for home users to apply because the size and density of the lattice used for the inner support structure are selected based on user experience. Lu et al. [5] described generating honeycomb-like cells in the object to minimize the inner support structure, but these cells have overhang regions, and so additional inner support structures are

needed for manufacturing. Another method is to fill the inside of the object with a 2D pattern, for example, lines and grids, and because of its simplicity, this is the most widely used method. Kindinger [6] offered various 2D patterns, such as honeycombs, balsa, and foam to aid the selection of a suitable lightweight 2D pattern structure, and Chen [7] expanded 2D pattern structures into 3D ones, which can both be referred to as cellular structures. These can be in the form of complex structures such as lattices or trusses created from a combination of simple cells. Chen [7] defined these using a variety of truss structures, and a combination of these are used as inner support structures. Wang et al. [8] defined an inner support structure called a skin-frame that is generated by placing truss structures only on the surface of an object. Inspired by this work, Zhang et al. [9] generated a medial structure based on the medial axis of the object to ensure strength, and they added a bar connecting this medial structure to the skin-frame structure. In addition, Wu et al. [10] used topology optimization. Similar to Lu et al. [5], Wu et al. and Li et al. [11, 12] created infill structures that are bony, porous, and lightweight.

There have been some studies on properties other than strength reinforcement. Wu et al. [13] considered the manufacturing stability in addition to the strength of an object. In order to achieve this, a rhombus structure that can always be stacked is used, but it has a drawback in that it remains a 2D example. Wu et al. [14] were interested in the mass distribution of the object, and to optimize this, a data structure called ray-rep, which is a representation of the model at the intersection of the ray and the boundary, was used.

2.1.2 Outer support structure (support structure)

The outer support structure supports the overhang regions so that the sections of the manufactured item can be stacked. However, it is treated as a by-product since it is is removed after fabrication is completed, and the more outer support structures generated, the longer the production time, and so there have been many studies on reducing it.

Common support generation methods have been well documented [15]. The most widely known method is to generate support for overhang areas using a tree structure such as the one proposed by Schmidt et al. [16] where the point of contact with the model is sharpened to facilitate removal. To improve the performance of this, Vanek et al. [17] reported a method for generating a tree structure using a minimum amount of material. On the other hand, Dumas et al. [18] suggested a structure using bridging, pointing out that the tree structure is not suitable for outer support in FDM-type 3D printers; the reasons for this are that it is poor in manufacturing stability, is ineffective above a certain height, and weak in load.

Similar to the tree structure, Qiu et al. [19] used a method of attaching support to the nearest model through a search using a cone shape, but since there are many structures connected to the model, it is difficult to obtain a neat model after production has completed. Huang et al. [20] used the sloping wall structure which can reduce the volume occupied by the support structure, but it is difficult to reduce the amount of material used to fabricate it. Strano et al. [21] used an implicit function

approach to generate geometric supports. This structure is suitable for SLM using heavy metal powder because it has a strong load bearing capacity, but is not suitable for FDM because of its complicated shape.

Unlike the studies on the geometry of the support structure, there are also those based on reducing the amount of support structure by modifying the model. Hu et al. [22] tried to minimize the amount of support structure generated by modifying the shape of the model and found that the amount of support can be considerably reduced with only a slight deformation. The model was modified while maintaining the detailed features of the shape.

2.1.3 Object segmentation

The size of the model that the user wants to produce varies, but commercial 3D printers are standardized. People with knowledge of 3D printers could create one that is larger than the size of the item to be produced, but there is obviously a size limitation. Hence, there have been studies to divide the model to solve this problem. Luo et al. [23] investigated how to divide a large object into multiple parts that are 3D printable shapes and can be reassembled. Yao et al. [24] explored partitioning to save time and space by considering stress load, surface details, interfacial area, packed size, printability, and assembly. Vanek et al. [25] considered how to overlap divided parts since they also have the effect of reducing the amount of support needed. Hu et al. [26] assessed how to divide an object into pyramidal shapes. When divided into pyramids, the parts have a material and time saving effect because no

support structure is required. Similar to this approach, Wei et al. [27] proposed a method to divide an object into parts that do not need an infill structure.

2.1.4 Stacking direction determination

The stacking direction has a great influence on the surface quality, the strength of the object, the amount of support needed, and the production time. Therefore, there have been many studies to determine the stacking direction by considering various factors. Ezair et al. [28] reported how to determine the direction of lamination to minimize the amount of support structure needed. For this purpose, they proposed a method to approximate the amount of support structure generated, and then to suggest the direction of optimization through simulation in various directions. By improving this, Hu et al. [29] proposed a method of finding the stacking direction that produces the minimum support structure while modifying the model slightly.

Because materials are stacked, an object manufactured using 3D printers has an anisotropic property which causes a specific region of the object to have different stiffness along the stacking direction. Umetani et al. [30] detected the weak areas in an object and proposed the stacking direction to make these strong. They used cross-sectional structural analysis to search for weakened regions due to anisotropic features.

The surface quality varies depending on the stacking direction. There are areas where the detailed shape is important in manufacturing an object and those where it is not. Zhang et al. [31] determined the optimal lamination direction by inputting the

importance of the surface quality from the user.

2.1.5 Balancing of objects

After the object is manufactured, it is very important that it is balanced. In order to achieve this, when the center of gravity of the object is projected on the floor, the projection point must be in the convex hull region of the object floor. Various studies have been carried out on this basic concept.

Christiansen and Prevost [32, 33] studied how objects can be balanced by hollowing certain areas inside an object. Because it is not possible to balance by just hollowing, deforming a part of an object is also used. Bacher et al. [34] optimized the moment of inertia so that the object can rotate with the desired axis of rotation as well as static balance. Wang et al. [35] extended the balance idea to buoyancy in water by studying the floating poses of objects using user-defined waterline height and orientation.

2.1.6 Deformation properties using microstructures

Using 3D printing, it has become possible to produce complex structures that could not have been produced before, and one of these is product design using microstructures. It is possible to construct a structure having new properties by continuously connecting very small unit structures (microstructures), and it is also possible to modify the elasticity of an object by modifying the shape of the microstructure [36, 37]. Bickel et al. [38] experimentally measured microstructure

deformation and applied it to a shoe insole design, and Brennan-Craddock et al. [39] used microstructure to create a shock absorbing structure.

2.1.7 Other topics

Research has been conducted on various subjects not covered by the topics mentioned above. There are studies that simplify problems by expressing objects in voxels [40-43], and studies that model and produce bone joints that do not require assembly using 3D printers [44, 45]. There is also research on new 3D printing methods for fast object creation, and research is also under way on how to make a thin shape such as a hair that is difficult to produce in a laminated manner [46, 47].

2.2 Problems with conventional infill structures

A widely used infill structure method is to stack the paths generated uniformly in the stacking direction as a 2D pattern so that the layered angle has a value of 90 degrees. Furthermore, the self-supporting property of the infill structure can be guaranteed by stacking uniformly in the stacking direction, and it is easy to generate because uniformly generated patterns are removed along the cross section of the object. Various patterns such as line, grid, and honeycomb have been used.

However, since the infill structure is uniformly generated, there is the disadvantage that the shape of the object cannot be considered. For example, when the strength of a part of an object is weak, the infill density increases to reinforce the strength, which increases the amount of infill structure throughout the object. If it is produced with

sparse density to save on material usage, the part to be supported (roof region) may not be.

In addition, the infill structure has two roles: strength reinforcement and roof region support which are linked to each other so that only the effect of one role at a time cannot be controlled. For example, some people may want fast, reliable output, regardless of the strength of the object. However, a certain value of infill density is required to support the roof region which induces a forced strength reinforcement effect so as to be able to manufacture stably. Therefore, in the past, if the strength reinforcement and roof region support, referred to as printability, are controlled by one parameter called infill density, it should be able to be controlled by two parameters.

The problems of the conventional 2D pattern method are summarized as follows.

- Excessive material is used without considering the geometry of the shape.
- Strength reinforcement and printability are dependent on each other.

2.3 Requirements of infill structures

The new infill structure proposed in this thesis should satisfy the following conditions:

1. Only the required amount considering the shape should be generated (efficiency).
2. It must support the roof region above a certain density (printability).
3. It should be stackable by itself (self-supporting).
4. It should only be a simple linear path (simple path).
5. It should have the isotropic property with respect to load (isotropy).

The first condition is related to the disadvantages of conventional infill structures mentioned in Chapter 2.2. Infill structures are required in the roof region and weaker regions, and outside of these, only a minimum number of them should be generated. For stable fabrication, infill structures should be created to support the bottom of the roof region with a layered angle of 45 degrees or smaller. For strength reinforcement, infill structures should be created around the region where the lowest strength is expected from the results of the stress analysis.

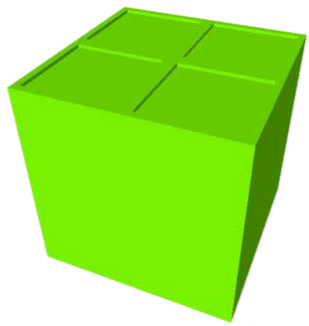
The second condition is that the infill density should be maintained above a certain value in the region close to the roof region in order to guarantee manufacturing stability. When seeking to reduce material usage, infill structures near the roof region should not be removed. For example, if the density of the 2D grid pattern is adjusted

as shown in Fig. 2.1, it is not possible to support the bottom of the roof region stably. The third condition is to guarantee the manufacturing possibility of the infill structure, which is an advantage of the conventional 2D pattern method. The geometric constraint that the layered angle of the infill structure is larger than 45 degrees should always be satisfied. If the layered angle is low, as shown in Fig. 2.2, manufacturing is not stable. In most previous related studies, this self-supporting requirement has not been considered. These studies claim to have proposed methods that reduce the amount of material used in infill structures. However, if a self-supporting structure is not guaranteed, then the reduction of material usage and manufacturing time cannot be guaranteed similarly because the infill structure may require a supplementary support structure that is not included in the final product specifications. Although the self-supporting condition in methods such as SLS is always satisfied, the concern in this study is with FDM-type printers where the self-supporting condition needs to be considered, and guaranteeing it is one of the main contributions of this thesis.

The fourth condition is a consideration of the operational principles of the FDM-type printer. The manufacturing time of the FDM-type printer is proportional to the amount of material used, and is also influenced by the nozzle path. In related studies, a truss structure is used to reduce material usage efficiently [8, 9], but the circular and tiny cross sections of a truss make it unsuitable for FDM-type printers; its circular geometry affects the manufacturing time because continuous acceleration and deceleration of the motor is required as the nozzle follows the curved tool path.

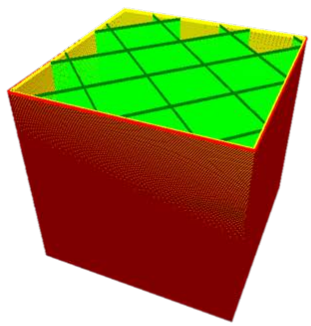
As truss structures are generally circular or rectangular, the actual manufacturing time might even increase, although the total material usage might decrease. This tendency is demonstrated in Fig. 2.3 where, in the upper and lower right diagrams, the tool paths are circular and straight, respectively. The circular tool path takes 32 hours and 37 minutes to complete, whereas the straight tool path takes only 6 hours and 57 minutes. Both use filaments of 28.12 m in length, but although a similar amount of material is used, more manufacturing time is required for the printer to replicate circular and tiny tool paths.

Finally, conventional 2D pattern infill has anisotropic characteristics with respect to the load. As shown in Fig. 2.4, it is strong in the stacking direction, but has little strength in the side direction; this is because it is not a 3D infill structure. Of course, the material produced by lamination has anisotropic characteristics [48]. In order to improve this disadvantage, the proposed infill structure should be structurally isotropic.

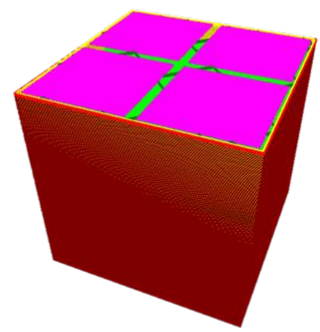


: first inset
: second inset
: skin
: infill of current layer
: infill of previous layers

Object including roof region

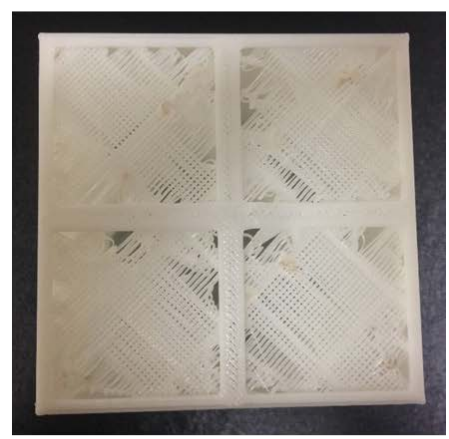


Tool path for 490th layer



Tool path for 491th layer

(Layered result)



(Layered result : top view)

Fig. 2.1 An example where infill structures do not sufficiently support the roof region

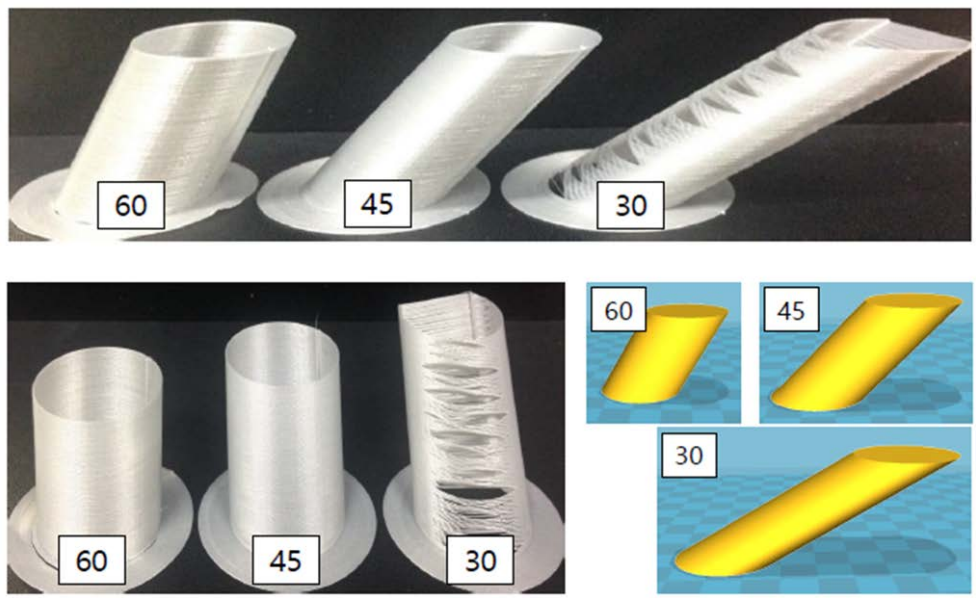
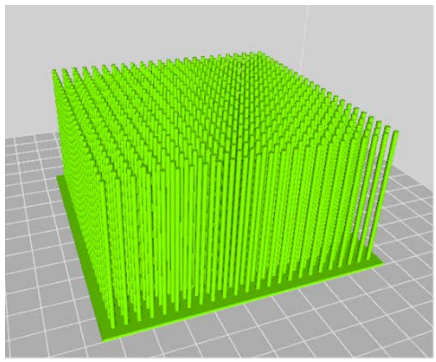
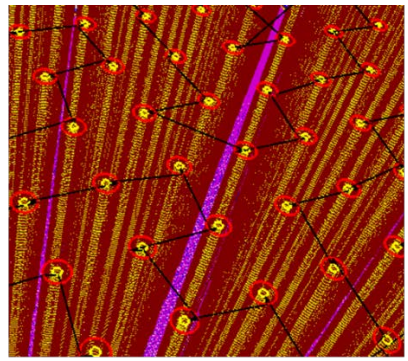


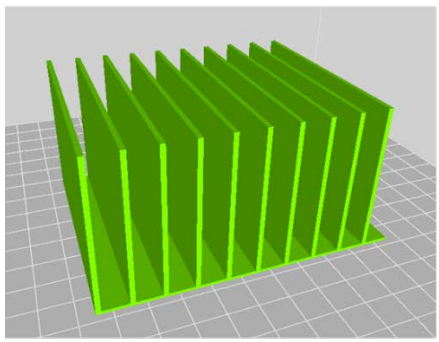
Fig. 2.2 Differences in quality depending on the layered angle in experimental tests



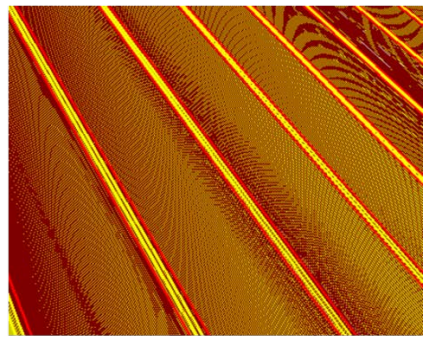
Object including circular tool path



Circular tool path



Object including straight tool path



Straight tool path

Fig. 2.3 Difference in manufacturing time according to path complexity. The black line indicates the tool path

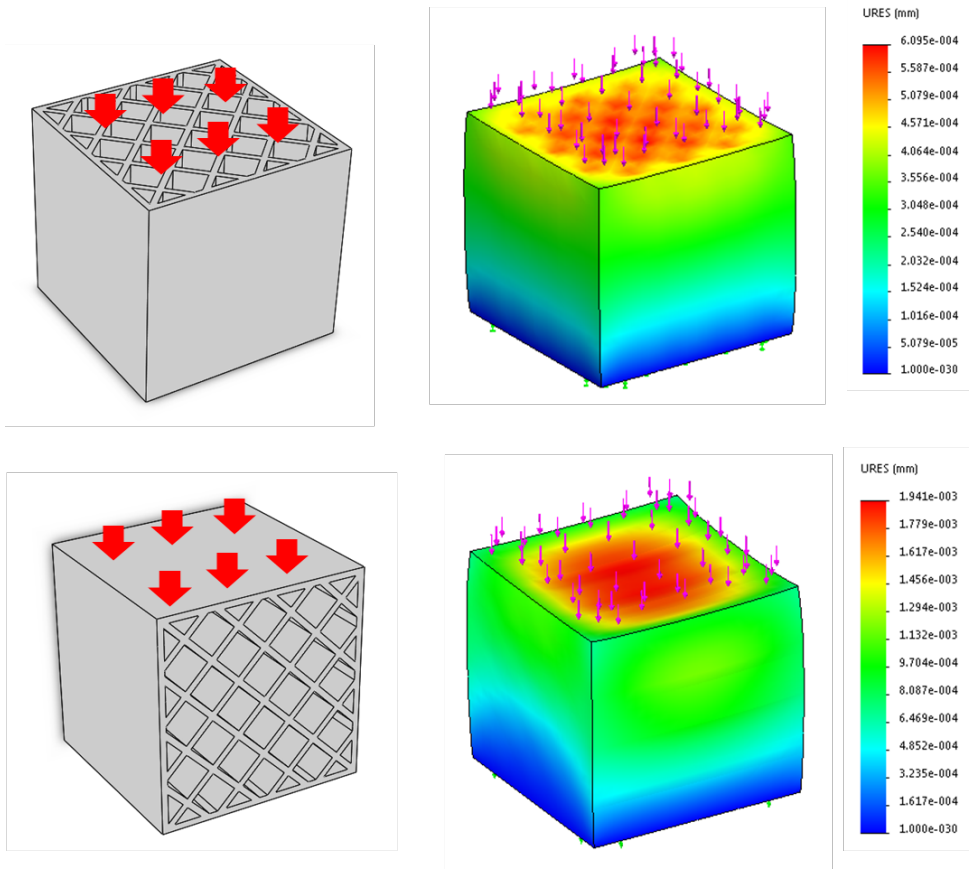


Fig. 2.4 Compression test simulation. Deformations according to compression direction: parallel to the manufacturing direction (top) and parallel to the manufacturing direction (bottom)

CHAPTER 3.

Methods

The proposed infill structure generation method is divided into 5 steps as shown in Fig. 3.1: the first step is unit block generation, the second step is to merge the inner unit blocks, the third step is a search to find the roof point in the merged block, the fourth step is to split the merged blocks to remove the roof point, and the final step is to create the tool path for the infill structure using the block data structure.

In this study, 2D and 3D examples of simple and complex shapes were used in parallel to enable easy understanding. A cube or square was used as a simple shape and the Stanford Bunny model [49] for a complex shape. The differences between the 2D and 3D implementations are discussed later.

In order to understand the proposed method, it is necessary to understand the concept of a block, so this is explained first.

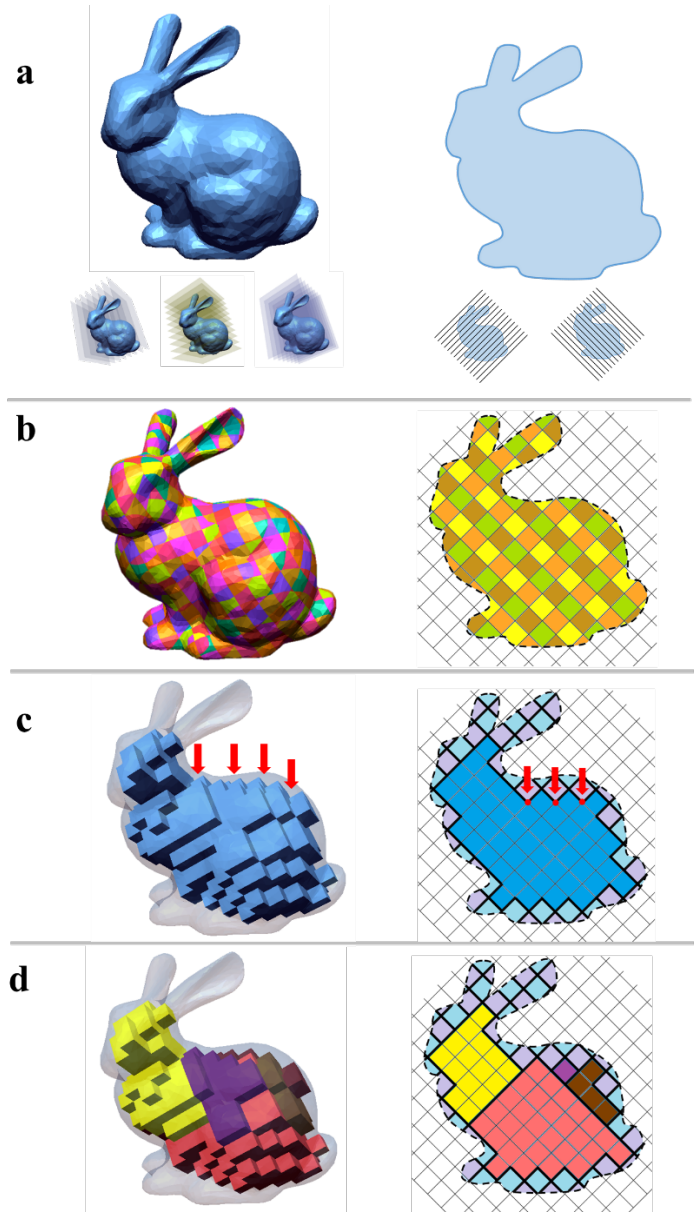


Fig. 3.1 Algorithm overview. Step 1 (a to b): make slices in three planar directions to create unit blocks. Step 2, 3 (b to c): merge the inner unit blocks into a combined block and find the roof points for splitting. Step 4 (c to d): split the combined block into sub-blocks based on the roof points. The first column is representative of 3D, and the second column is representative of 2D

3.1 The concept of blocks as a basis for the infill structure

From simple geometry to complex mechanical parts, there are a wide variety of infill structure shapes users can employ. Nevertheless, it is very difficult to create an infill structure while considering the five above-mentioned conditions (efficiency, printability, self-supporting, simple path, and isotropy). Therefore, in this thesis, the concept of the block is introduced to simplify the problem geometrically, and infill is reinterpreted as blocks.

In the conventional 2D pattern method, the infill structure divides the inside of the object into several spaces. For example, with a 2D grid pattern infill structure, the interior of the object is either divided into square pillar spaces or as a honey-comb pattern into hexagonal pillar shapes, and the wall or contact surface between spaces is called the infill structure. If the rule for dividing the internal space of an object is newly defined, then the contact surface between these spaces can be defined as a new infill structure.

A unit block is a unit of volume obtained by dividing an object at equal intervals using N planes that different from each other. Fig. 3.2 shows an example in which a 3D object (Stanford Bunny model) is divided at equal intervals using one plane.

The blocks can be used to redefine the infill structure (the infill structure is defined as the contact surface between blocks), and the contact surface between the blocks is the same as a 2D line pattern infill structure. In this way, if a block is defined as an inner space partition and the contact surface of the inner space is defined as the infill structure, another conventional 2D pattern infill structure-like grid can be

redefined. Furthermore, the space can be split into three planes to improve the disadvantages of a 2D pattern infill structure.

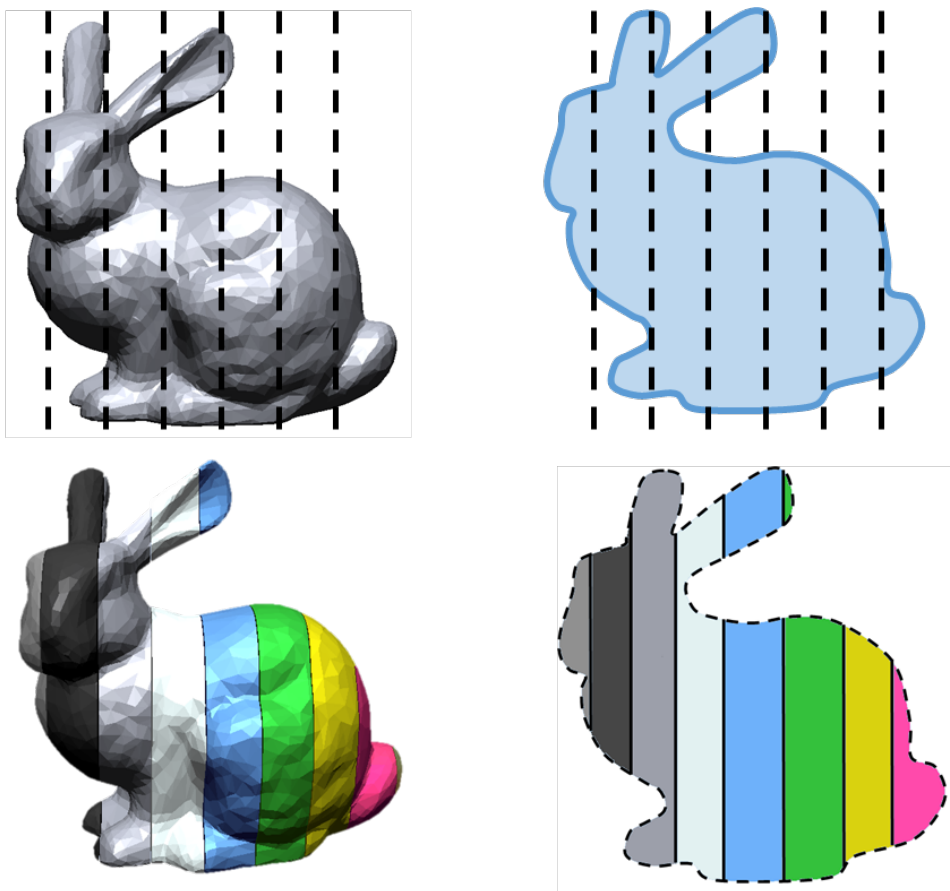


Fig. 3.2 An example of blocks and infill

3.2 Stackable space partitioning

Cube unit blocks are created by dividing the object into three planes perpendicular to each other at uniform intervals. However, the top and bottom of the cube do not satisfy the self-supporting condition for an infill structure because these make up the roof region, as shown by the red lines in Fig. 3.3b. Specifically, the angle between the normal direction of all of the planes and the stacking direction should be 45 degrees or more, and to achieve this, the cube must be rotated. The rotation matrix that makes the vectors of the two farthest points of the cube parallel to the stacking direction can be written as

$$\mathrm{R} = \begin{bmatrix} \frac{\sqrt{6}}{3} & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{6} \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{bmatrix}.$$

After rotation, the layered angle of each side is about 54 degrees and all sides meet the self-supporting condition.

The infill density represents the volume occupied by the infill inside the object, and can be expressed by the following equation where **T** is the thickness of the nozzle, **A** is the layered angle, **d** is the infill density, and **s** is the unit length of the cube:

$$\mathrm{d} = \frac{\mathrm{s}^3 - \frac{6\mathrm{s}^2 \cos(\mathrm{A}) * T}{2}}{\mathrm{s}^3} * 100,$$

$$\mathrm{d} = \frac{\mathrm{s} - 3 * \cos(\mathrm{A}) * T}{\mathrm{s}} * 100,$$

$$\mathrm{s} = \frac{3 * \cos(\mathrm{A}) * \mathrm{T}}{100 - \mathrm{d}} * 100.$$

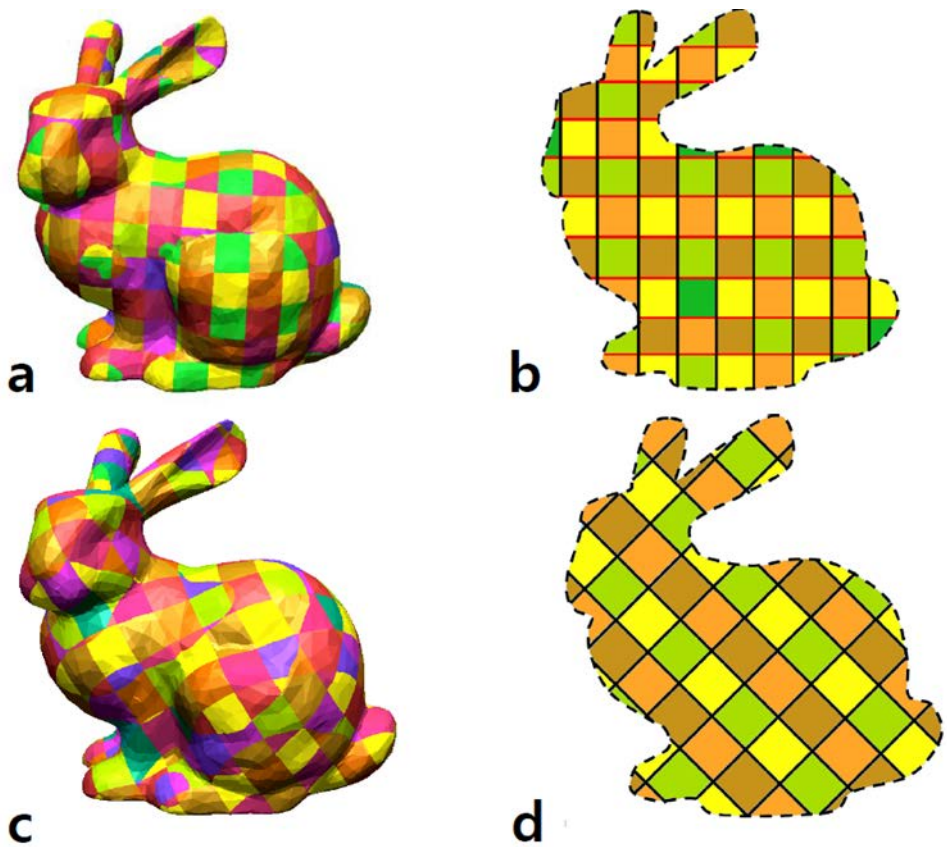


Fig. 3.3 3D pattern infill: (a) is the object divided using three perpendicular planes, (b) is a 2D representation of (a), (c) is the object divided using three rotated planes, and (d) is a 2D representation of (c). The red/black solid lines indicate roofs/non-roofs, respectively, in (b) and (d)

3.3 Merging unit blocks

The unit blocks can be divided into boundary unit blocks (BUBs) and inner unit blocks (IUBs) depending on whether the surface of the object is included or not, and the BUBs can be divided into roof unit blocks (RUBs) and non-roof unit blocks (NRUBs) depending on whether the roof region is included or not.

In the previous section, the rotated unit blocks satisfy the self-supporting condition. In this section, the self-supporting condition of the rotated unit blocks is shown to be satisfied even after merging with neighboring blocks. Unit block merging means the disappearance of the contact surface, which means a reduction in the infill structure made possible by merging the unit blocks.

The one of the purposes of creating infill structure is to generate a certain density or greater near the roof region to support it (printability condition). If the unit blocks near the roof region are maintained, merging unit blocks far away from the roof region does not affect the printability condition of the object. In particular, the amount of infill structure can be efficiently reduced through the merging of unit blocks excluding the RUBs.

A unit block can be merged with a neighboring unit block as shown in Fig. 3.4, but merged blocks do not always maintain the self-supporting condition. In the case of merged block 5 shown in Fig. 3.4, there is no supporting structure at the bottom, and so a roof point that cannot be stacked occurs.

In summary, the merging of unit blocks means that the contact surface between the blocks disappears, thus reducing infill. Therefore, it is effective in reducing the

number of unit blocks by as many as possible, except for the RUBs. However, merging is not always possible because of the self-supporting condition of the infill structure. Considering these conditions, merging is a very complicated problem because there are a large number of cases, and it is therefore necessary to simplify the problem. In the merging process, the following conditions must be satisfied.

1. The blocks should be a simple shape.
2. Roof points should not occur.

The first condition implies that the boundary shape of the merged blocks should be as flat as possible. If the boundary shape is a zigzag instead of a flat plane, the movement of the nozzle of the FDM becomes complicated and the manufacturing time is increased due to the deceleration and acceleration time of the motor.

In the second condition, a roof point is on a block that cannot be stacked because there is no supporting structure below the point. The presence of a roof point means that the infill structure's self-supporting condition is insufficient (the red dots in Fig. 3.1c indicate the roof points). A discussion of how to find the roof points can be found in Section 3.5

In conclusion, the problem of merging unit blocks can be thought of as the problem of segmenting or grouping neighboring blocks, and the problem to be solved can be defined as the problem of minimizing the area of the contact surface while satisfying the two conditions mentioned above.

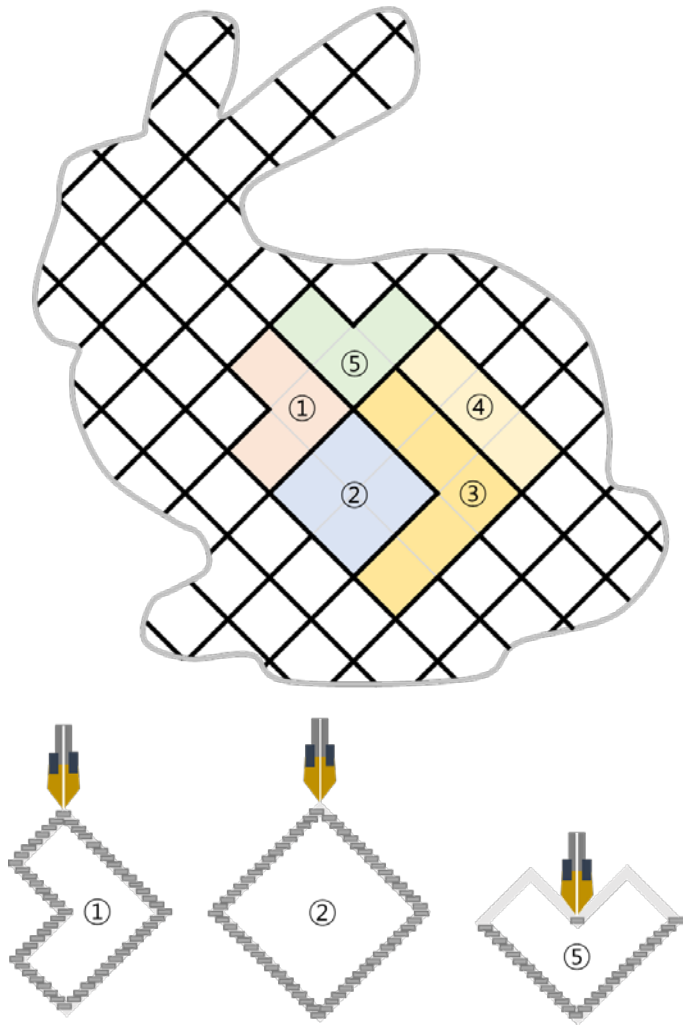


Fig. 3.4 An example showing unit blocks that can be merged. Blocks 1,2,3, and 4 can be merged, but not 5 because it includes a roof point

3.4 Eliminating roof points

The amount of infill structure can be reduced through the merging of unit blocks, but it is very difficult to minimize the amount of infill structure while satisfying the conditions of blocks having a simple shape and not containing a roof point. Therefore, an attempt to solve this problem is made by changing the order of the steps:

1. Merge all unit blocks except RUBs.
2. Find the roof points that exist in the merged blocks.
3. To remove a roof point, split the merged block that contains the roof point using one of the candidate planes passing through the roof point.

In order to remove a roof point, the merged block including the roof may be divided by using one of the candidate planes passing through the roof; there are two candidate planes in the 2D example and three in the 3D plane example. The division of the merged blocks leads to the creation of the contact surface, and the roof point is removed because the contact surface is a structure supporting the roof point.

In the first step, all units are merged so that the amount of infill can be minimized. In order to remove the roof point in the third step, the plane is used to split the merged blocks so that the contact plane becomes the shape of the plane and the simple shape condition can be maintained.

In the process of removing roof points, the merged blocks are divided and infill structure is generated. When there are N roof points in the merged block, the amount of infill structure generated by the removal order of roof points and the supporting

direction of the roof point is determined. Thus, the problem of minimizing the amount of infill structure turns into the simple problem of determining the removal order and supporting the direction of the roof points.

The original problem is the problem of dividing M unit blocks into arbitrary N groups while satisfying the given conditions, but determining the order of supporting planes automatically satisfies these conditions. The problem of determining the order is relatively easy because it is a 1D problem.

By changing the approach to the problem, both the simple pass condition and the roof condition are satisfied, and now only the order of removing the roof points needs to be discussed. Before that, a discussion on how to find the roof points is presented in the next section.

3.5 Finding roof points

A roof point occurs on a block that cannot be stacked because there is no supporting structure below the point. It refers to a point on a shape that cannot be supported similar to the roof region, but the roof region is in the shape of the object while the roof point is in the infill structure. Each corner point of a unit block can be a roof point candidate, and so all corner points are searched when looking for them. With the exception of a unit block in a location higher than the corner point, if the rest of the unit blocks adjacent to that corner point (3-unit blocks in the 2D case, 7-unit blocks in the 3D case) belong to the same combined block, then that corner point is a roof point. Fig. 3.5 shows examples of a roof point and non-roof points in a 2D case.

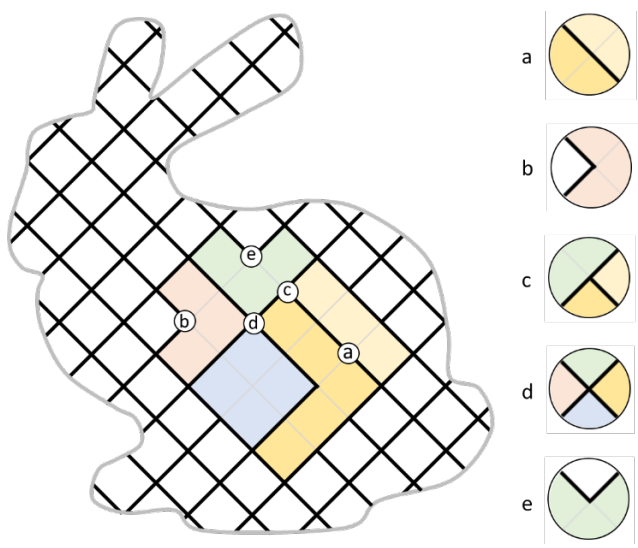


Fig. 3.5 Non-roof points (a, b, c, and d) and roof point (e). The black solid lines represent the inner support structure

Roof points are no longer generated during the splitting step, and so all of the corners are searched only after merging all of the unit blocks, then the existing roof points are repeatedly searched again. In the 3D case, when one roof point is removed, another roof point may be removed, so it is necessary to check again for the roof points each time the division is made.

3.6 Splitting the combined block

In order to minimize the amount of infill, the supporting-plane area needs to be minimized, which removes all of the roof points. The amount of infill produced to remove the roof point can be calculated as the area of the intersection of the combined block and the plane passing through the roof point. To minimize the amount of infill, heuristic methods such as a greedy algorithm [50] that find a partial solution for the minimum value only with information obtained from the current status are used.

Various approaches to determine the order of splitting the combined block (or removing roof points) and the final splitting methods are discussed. To understand the splitting algorithms described below, it is necessary to understand the notation and functions below.

M: the problem model or state which contains the coupling information of the unit blocks. In other words, whether a unit is stored as a rectangle or voxel in the inner support structure or not. In Fig. 3.6, the bold black lines denote the problem model.

P: the set of roof points. In Fig. 3.7, the red circular points are roof points whereas in Fig. 3.8, the black circular points indicate that the roof points have disappeared because of the splitting process.

i: the roof point index.

j: the supporting direction index shown as direction 1 and direction 2 in Fig. 3.7. Generally, there are two supporting directions in 2D and three supporting directions in 3D.

$\mathbf{P_i}$: the i^{th} roof point.

t: the splitting phase or number.

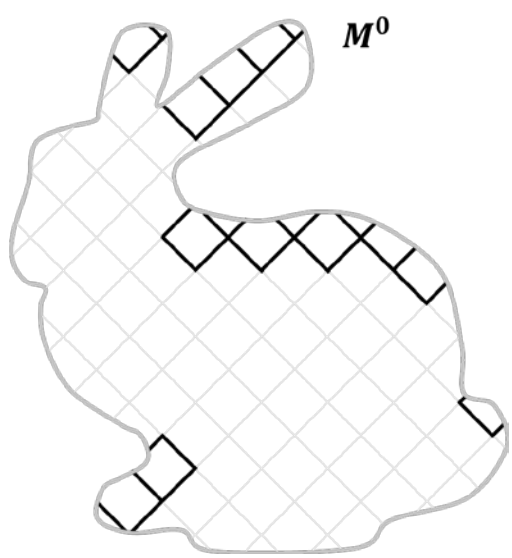


Fig. 3.6 Example of problem model M^0

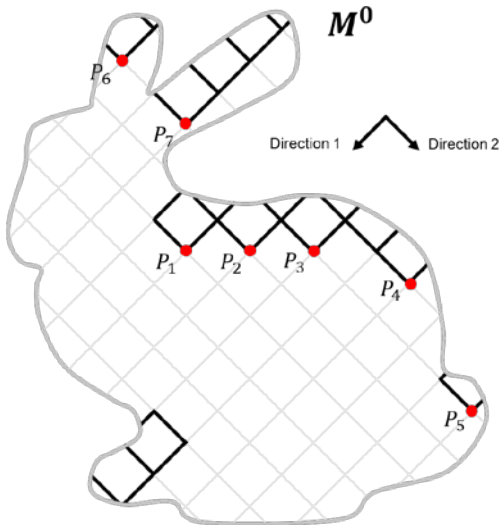


Fig. 3.7 Example of roof points P_i

$\mathbf{M^t}$: the problem model divided by the t supporting planes. In Fig. 3.8, problem model M^1 is divided by one supporting plane. Problem model M^0 means a non-split state.

$\mathbf{O}$: the array of the supporting planes; those that divide the boxes are stored sequentially.

$\mathbf{O_t}$: The t^{th} supporting plane stored at O.

$\mathbf{c_{ij}^t}$: the supporting plane which passes through the i^{th} roof point P_i with the j^{th} supporting direction in the M^t state. A bold black dotted line in all of the figures indicates that the problem model M is unaffected until the split function is used. For example, in Fig. 3.8, the areas of the supporting plane c_{31}^0 above and the supporting plane c_{31}^1 below are different. Because the problem model M^0 has not yet been divided, c_{31}^0 is not affected by c_{12}^0. On the other hand, c_{31}^1 is influenced by c_{12}^0

because it is calculated in the M^1 state. Thus, supporting planes may be affected by previously used ones (0). As a result, being affected by other supporting plane means decreasing the area of own. This is very important.

$\mathbf{c_i^t}$: the minimum supporting plane, which is the smallest supporting plane among c_{ij}^t; $j = 1,2,3$ passing through the i^{th} roof point P_i.

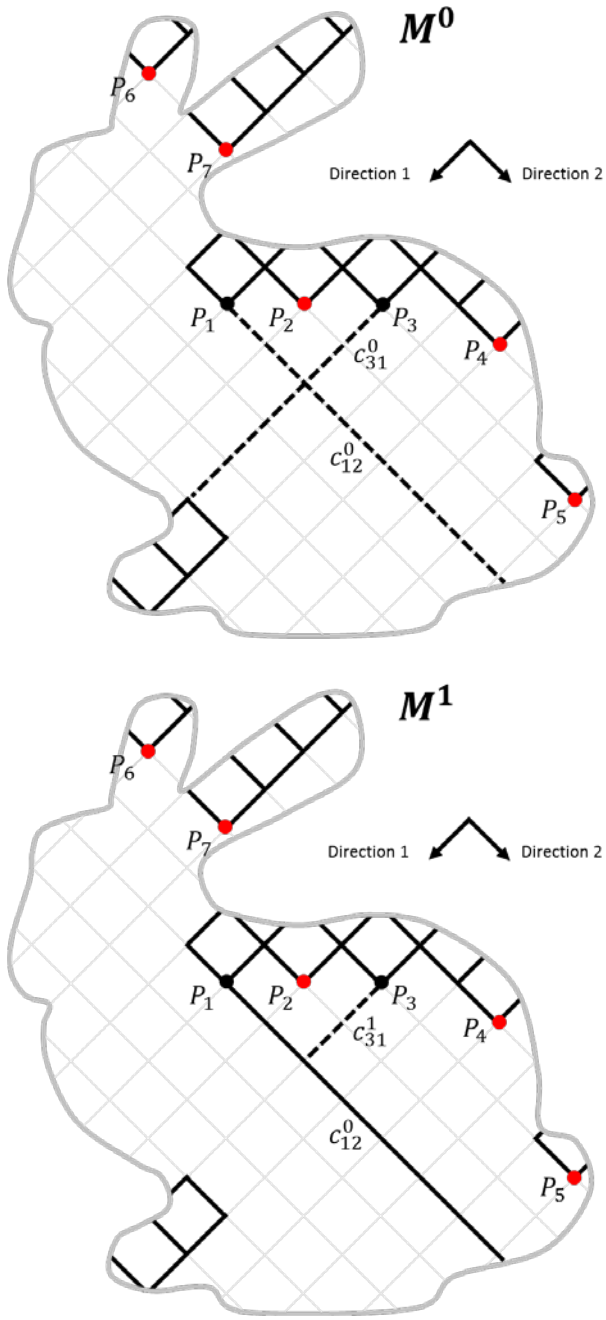


Fig. 3.8 Example to explain the phase t and supporting plane c_{ij}^t

Split function: Fig. 3.9 gives an example of a split function that returns the next problem model M^{t+1} by adding supporting plane $\mathrm{c}_{\mathrm{ij}}^{t}$ to current problem model M^{t}. As mentioned above, none of the supporting planes $\mathrm{c}_{\mathrm{ij}}^{t}$ affect problem model M until the split function is used:

$$\mathrm{M}^{t+1} = \mathrm{split}(\mathrm{M}^{t}, \mathrm{c}_{\mathrm{ij}}^{t}).$$

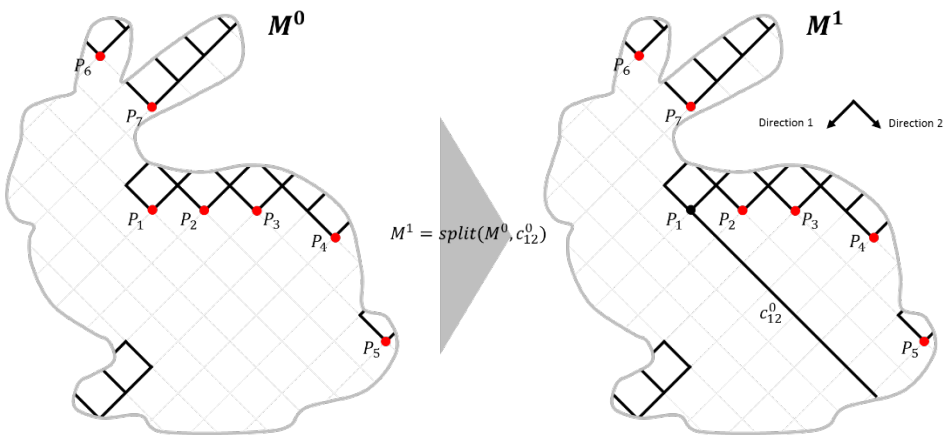


Fig. 3.9 Example of a split function

Area function: Fig. 3.10 provides an example of an area function that returns the area of supporting plane $\mathrm{c}_{\mathrm{ij}}^{t}$. If the array of supporting planes are inputted, it returns the sum of the areas of all of the supporting planes.

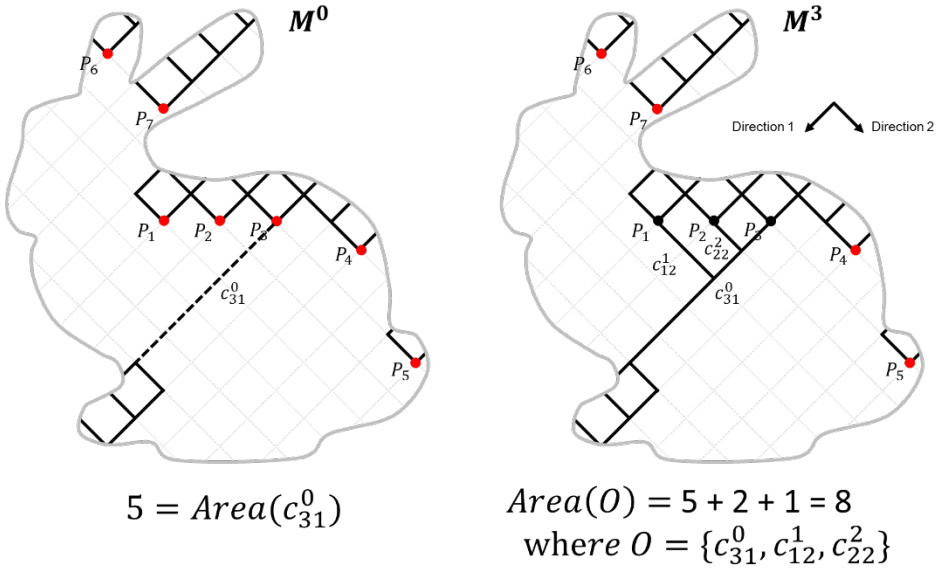


Fig. 3.10 Example of an area function

3.6.1 Splitting algorithm A

Initially, the most intuitive approach for determining the minimum sum of areas was tried. There are two issues that must be considered when removing all of the roof points, the first being to decide which direction to use to remove a roof point. For each roof point, the supporting plane having the smallest area when the combined block is divided should be chosen. This is because as the supporting-plane area decreases, the inner support structure also becomes smaller. Of all the supporting planes passing through the roof point, the plane having smallest area is called the minimum supporting plane. The second issue is to determine in which order to divide the roof points. Even when multiple root points exist, the supporting-plane area which is the smallest should be chosen. From the perspective of greedy algorithms, each choice reduces a given problem into a smaller one and the sum of small things can be the smallest.

The first step is to find the minimum supporting plane in the direction of the smallest area to remove each roof point. The second step is to find a supporting plane that has the smallest area among all of the minimum supporting planes. The last step is to split the problem model using the selected supporting plane and update the other roof points.

The above-mentioned algorithm is represented by the following pseudocode:

Splitting algorithm A

Data: M, P
Result: O
initialization
t = 0;
while P *is not empty* **do**
  for $\forall P_i \in P$ **do**
    $c_i^t = \arg\min_{c_{ix}^t} Area(c_{ix}^t)$
  end
  $O_{t+1} = \arg\min_{c_x^t} Area(c_x^t)$
  M^{t+1}=$split(M^t, O_{t+1})$
  t = t+1;
  update P
end
return O

Algorithm 3.1 Pseudocode for splitting algorithm A

To easily understand the operating principles of the algorithm and verify its performance, a simulation with a simple 2D rectangle was carried out, as shown in Fig. 3.11. The splitting processing and final problem model M^6 applied with splitting algorithm A are shown in Fig. 3.12.

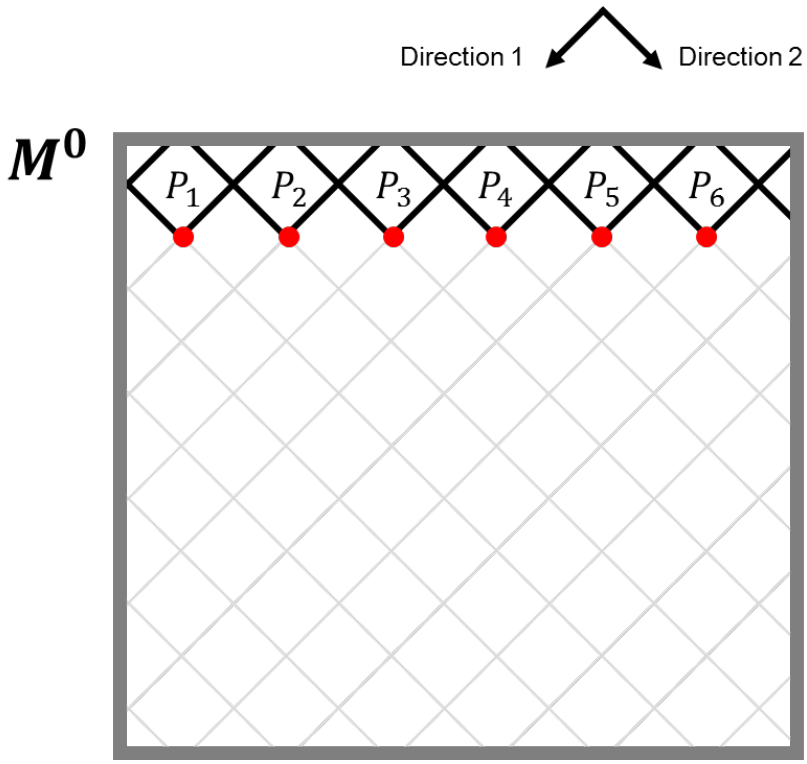
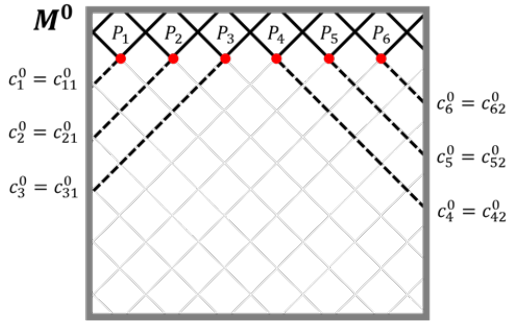
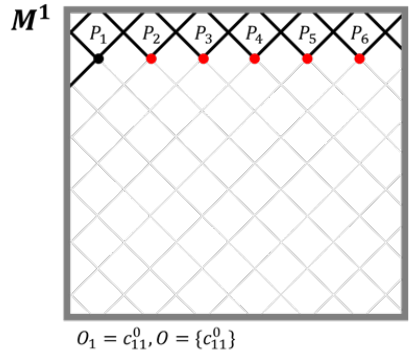


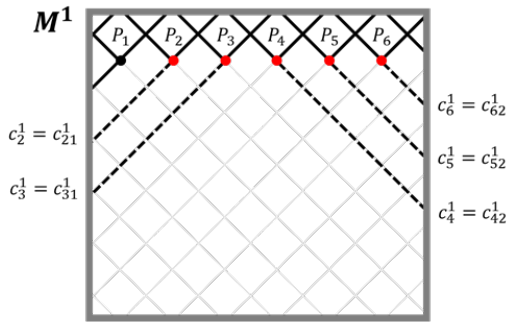
Fig. 3.11 Simple 2D rectangle case for initial problem model M^0



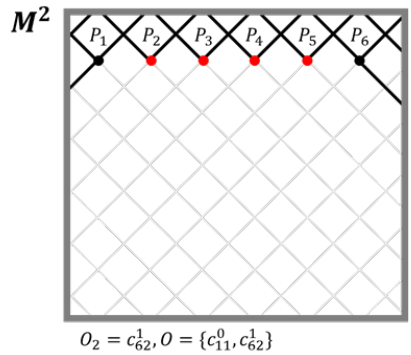
1. Find the minimal cut-planes



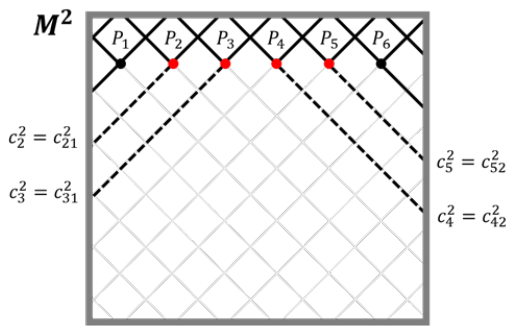
2. Split and update roof points



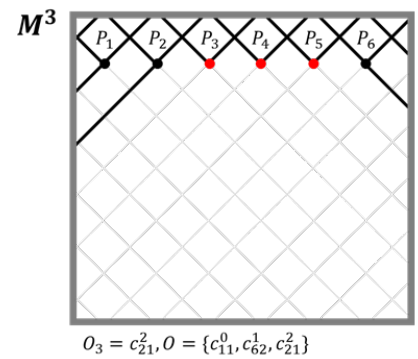
3. Find the minimal cut-planes



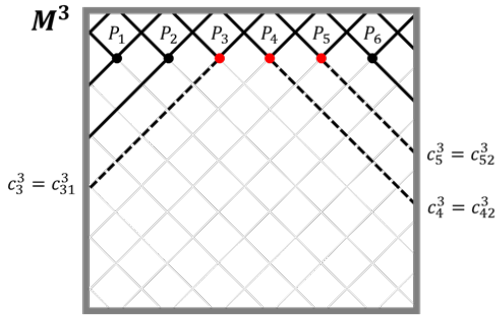
4. Split and update roof points



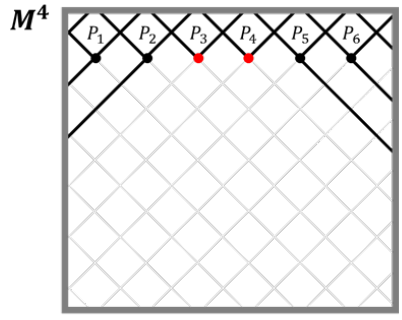
5. Find the minimal cut-planes



6. Split and update roof points

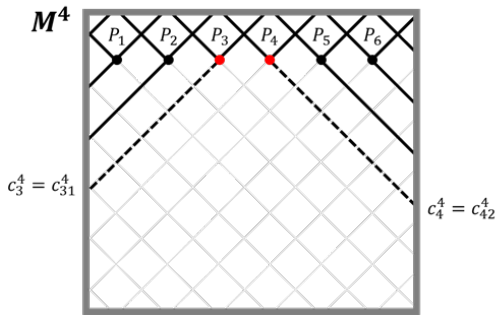


7. Find the minimal cut-planes

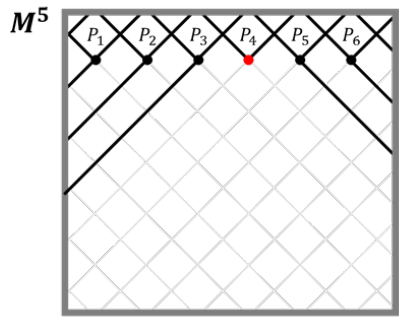


$O_4 = c_{52}^3, O = \{c_{11}^0, c_{62}^1, c_{21}^2, c_{52}^3\}$

8. Split and update roof points

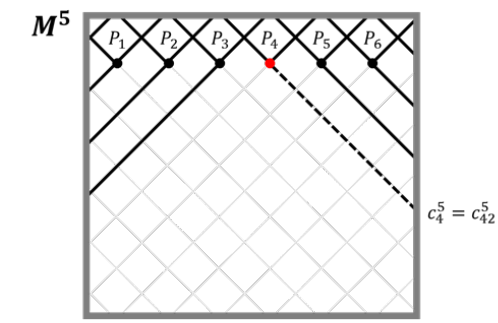


9. Find the minimal cut-planes

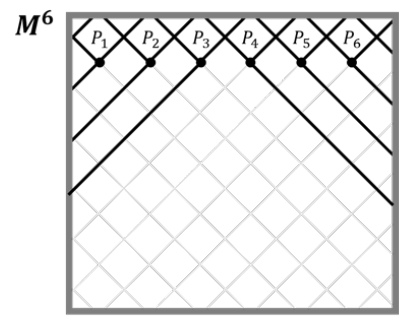


$O_5 = c_{31}^4, O = \{c_{11}^0, c_{62}^1, c_{21}^2, c_{52}^3, c_{31}^4\}$

10. Split and update roof points



11. Find the minimal cut-planes



$O_6 = c_{42}^5, O = \{c_{11}^0, c_{62}^1, c_{21}^2, c_{52}^3, c_{31}^4, c_{42}^5\}$

$Area(O) = 1 + 2 + 3 + 4 + 5 + 6 = 21$

12. Split and update roof points

Fig. 3.12 The splitting process and final problem model M^6 with splitting algorithm A

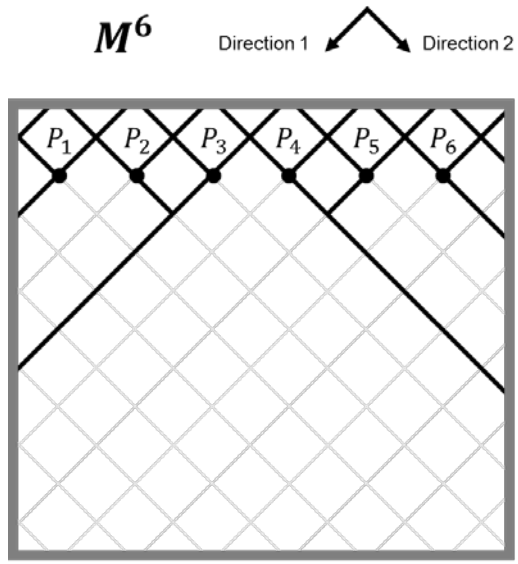


Fig. 3.13 Simple 2D rectangle case: final problem model M^6 with manual splitting

When the area of the unit square of the unit blocks is 1, the area of the newly generated infill is 21 with the six splits, as shown Fig. 3.12. However, this splitting order is thought to be inefficient. As shown in Fig. 3.13, by manually determining the order of planes, the area is less than 16, which is less than 21. Therefore, selecting the supporting plane with the smallest area among the minimum supporting planes is not the best method.

3.6.2 Splitting algorithm B

As mentioned above, splitting algorithm A is not the optimum method, and so it is necessary to find another way to solve this problem. For each roof point, the fact that the minimum supporting plane should be chosen seems to be right. Therefore, it is necessary to reconsider the order of the roof points for splitting.

Contrary to splitting algorithm A, the supporting plane having the largest area should be used primarily among all minimum supporting planes passing through the roof points (a hint of this can be found in Fig. 3.8). The area of the minimum supporting plane needed to remove P_3 is reduced from $\text{area}(c_{32}^0) = 4$ to $\text{area}(c_{31}^1) = 2$ by supporting plane c_{12}^0 of roof point P_1. Like this, the supporting planes used in the splitting can affect a change in the area of the other minimum supporting plane. In particular, the larger the supporting plane, the more likely it is to influence the other supporting plane. Conversely, a small supporting plane cannot affect a change in the area of the other minimum supporting plane. For example, supporting planes such as $\text{area}(c_{42}^0) = 1$ and $\text{area}(c_{52}^0) = 1$ do not affect any other supporting planes at all. Thus, splitting algorithm B can be created by changing splitting algorithm A as follows: the first step is to find minimum supporting planes about each roof point, the second step is to find a supporting plane that has the largest area among all of the minimum supporting planes, and the last step is to split the problem model using the selected supporting plane and update the other roof points.

The above-mentioned algorithm is represented by the following pseudocode:

Splitting algorithm B

Data: M, P
Result: O
initialization
t = 0;
while P *is not empty* **do**
  for $\forall\ P_i \in P$ **do**
    $c_i^t = \arg\min_{c_{ix}^t} Area(c_{ix}^t)$
  end
  $O_{t+1} = \arg\max_{c_x^t} Area(c_x^t)$
  M^{t+1}=$split(M^t, O_{t+1})$
  t = t+1;
  update P
end
return O

Algorithm 3.2 Pseudocode for splitting algorithm B

To verify the performance of splitting algorithm B compared with splitting algorithm A, identical simulations with simple 2D rectangles were carried out. Fig. 3.14 shows the final problem model M^6 using splitting algorithm B. The detailed operating principles of splitting algorithm B are similar to those of splitting algorithm A shown in Fig. 3.12.

The area of the newly generated infill is 16, which is the optimal solution because there are no better results among all of the cases with the six supporting planes. In order to verify its efficiency with a more complex problem, splitting algorithm B was applied to the problem model with many roof points shown Fig. 3.15.

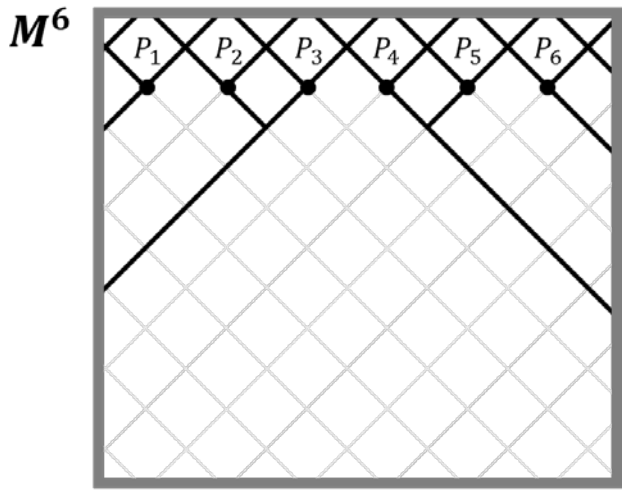


Fig. 3.14 Simple 2D rectangle case: final problem model M^6 with splitting algorithm B

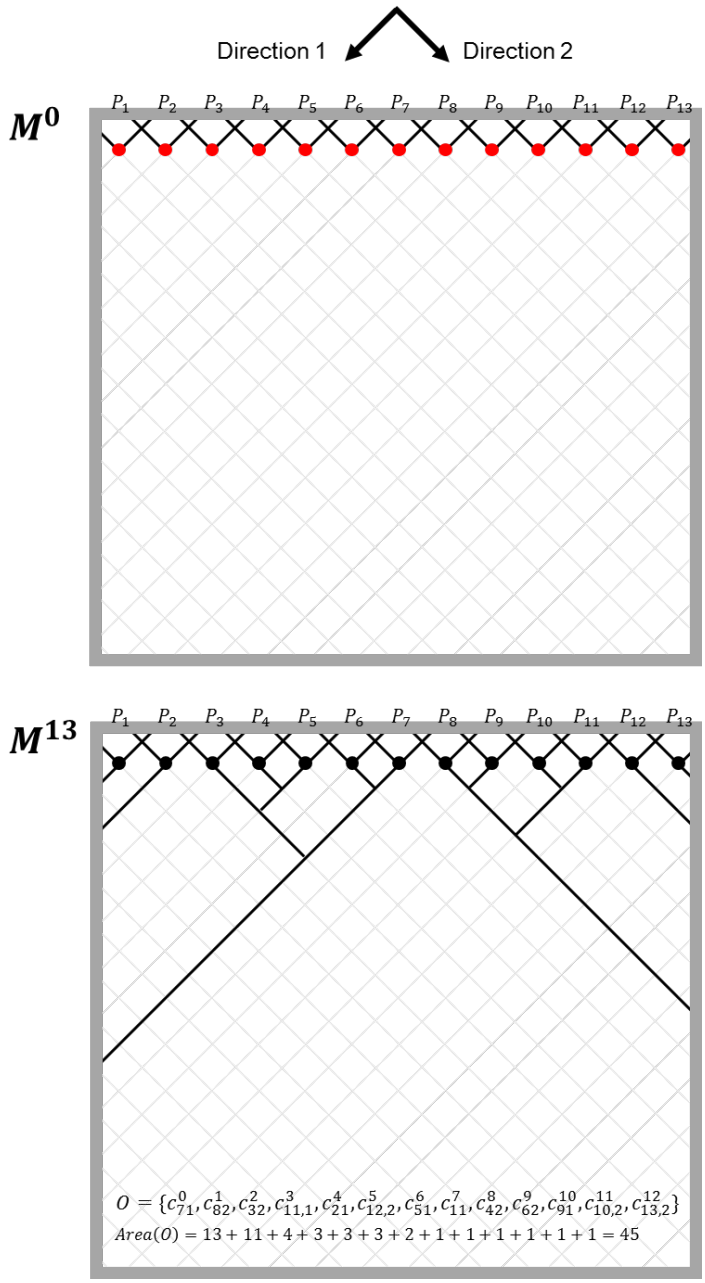
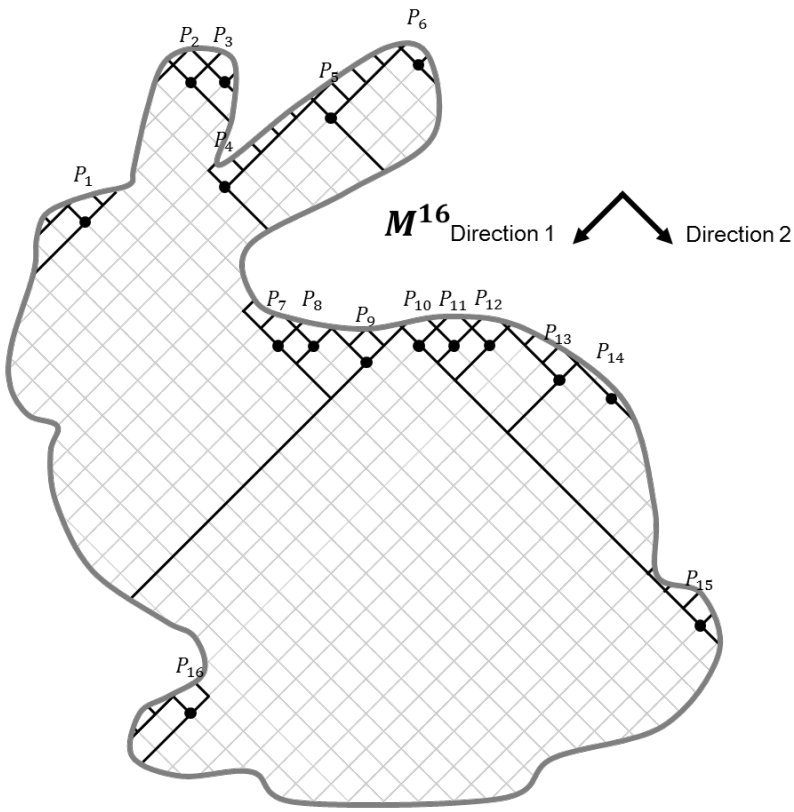


Fig. 3.15 Simple rectangle case with many roof points: initial problem model M^0 (top) and final problem model M^{13} with splitting algorithm B (bottom)

As shown in Fig. 3.15, the infill generated by splitting algorithm B is produced in a tree-like structure similar to that reported previously [17], and an effect similar to supporting the overhang area by a minimum amount can be expected. Furthermore, it works well even with complex shapes like the Stanford Bunny model (see Fig. 3.16).



$$O = \{c_{91}^{0}, c_{10,2}^{1}, c_{11}^{2}, c_{22}^{3}, c_{32}^{4}, c_{52}^{5}, c_{72}^{6}, c_{13,1}^{7}, c_{16.1}^{8}, c_{12,1}^{9}, c_{3,2}^{10}, c_{62}^{11}, c_{81}^{12}, c_{11,1}^{13}, c_{14,2}^{14}, c_{15,2}^{15}\}$$

$$Area(O) = 14 + 13 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 2 + 1 + 1 + 1 + 1 + 1 + 1 = 56$$

Fig. 3.16 Stanford Bunny case: final problem model M^{16} with splitting algorithm B

3.6.3 Splitting algorithm C

This algorithm generates an enormous number of cases for making supporting planes by choosing order and direction when the merged block is divided at the roof points. The possible number of directions for splitting the merged block is three, and if N roof points exist on the merged block, $3^N * N!$ splitting results are expected for dividing the merged block. Among the whole cases of splitting, finding the global minimum is extremely difficult, especially with this kind of minimum spanning tree problem, which is a non-deterministic polynomial-time hard (NP-hard) problem that cannot be solved unless all possible cases are examined.

In splitting algorithm B, the criterion proposed to solve this NP-hard problem, is the largest supporting plane among the minimum supporting planes that passes through each roof point. The reason for choosing this criterion is that the minimum supporting plane with a large area can have a large effect on the other supporting planes. In fact, a supporting plane affects the supporting-plane area of other roof points, and the effect here is that it intercepts with other supporting planes. As shown in Fig. 3.8, a supporting plane with a large influence intercepts another supporting plane and prevents further division of the merged block by another supporting plane. The adaptive characteristic of the infill structure shown in Figs. 3.15 and 3.16 is a result caused by this interception phenomenon. The total area of the supporting planes can be reduced due to the interception phenomenon between the split planes. Therefore, the idea of blocking the other supporting planes generated by splitting

algorithm B by as much as possible is the right approach to reduce the total infill structure.

However, there are possibilities that splitting algorithm B does not always have the effect of reducing the total area of the supporting planes. First, it is not always certain that the largest supporting plane selected among the minimum supporting planes passing through each the roof point can always affect the other supporting planes. The largest supporting plane is likely to intercept another supporting plane but may not intercept one along the direction of the split plane. When using algorithm B on a wide shape as shown in Fig. 3.17, the supporting plane is generated in parallel, and in this case, the total amount of infill structure cannot be reduced. This phenomenon occurs when the area of the minimum supporting plane is the same, which is often seen in a wide shape.

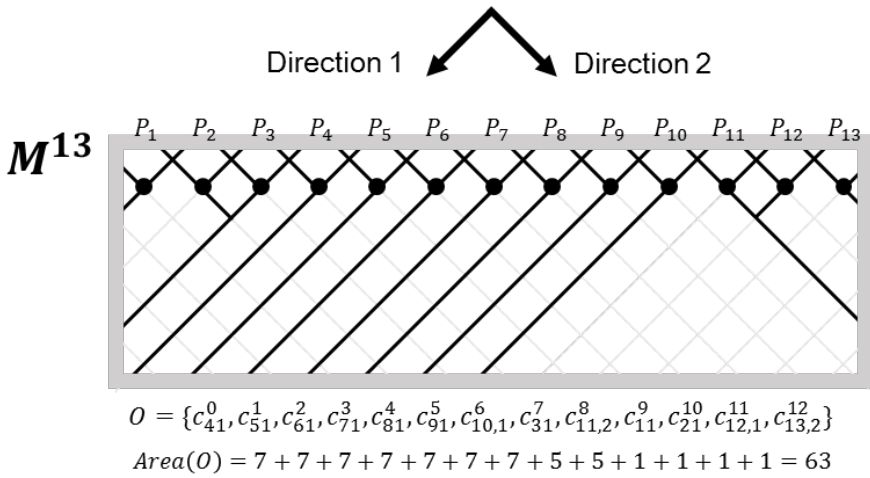


Fig. 3.17 Ineffectual case of splitting algorithm B

Second, selecting a supporting plane among the minimum supporting planes is the wrong approach. Although there is a loss due to the large supporting-plane area in the current phase, there is the possibility of a gain by reducing the total area of the other minimum supporting planes in the next phase, and so the best choice in the present state is not necessarily the best choice when the complete process is viewed as a whole, which is a known disadvantage of greedy algorithms.

Therefore, splitting algorithm B has the possibility of decreasing the infill structure but is not a sure way to guarantee it. To solve this uncertainty, a new method has been devised to quantify the idea of reducing the total infill structure by blocking other supporting planes. To calculate this, the following new function needs to be defined.

Virtual split function: Fig. 3.18 shows an example using the virtual split function. It returns the next temporary problem model M_{temp}^{t+1} by adding the supporting plane $\mathrm{c}_{\mathrm{ij}}^{t}$, which is a cup plane measuring the influence, to current problem M^{t}:

$$\mathrm{M}_{temp}^{t+1} = \mathrm{virtualSplit}(\mathrm{M}^{t}, \mathrm{c}_{\mathrm{ij}}^{t}).$$

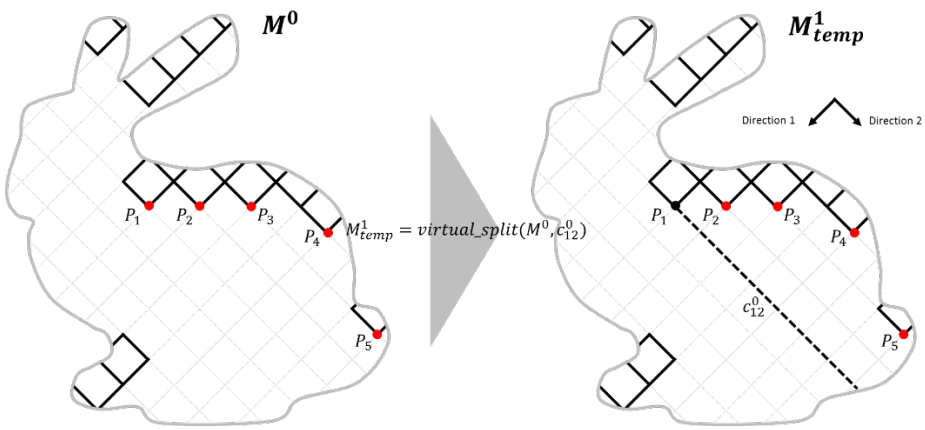


Fig. 3.18 Example using the virtual split function

The amount of infill structure reduced due to interference between the supporting planes is called the **influence**. The greater the influence of the supporting plane, the less the total infill structure. Therefore, planes with the greatest influence should be selected.

Estimating influence algorithm

Data: M, P, c_{ij}^{t}
Result: I
$O_{current}$ = splittingAlgorithmB(M, P);
M_{temp}=virtualSplit(M, c_{ij}^{t});
update P
O_{next} = splittingAlgorithmB(M_{temp}, P);
I = Area($O_{current}$) - Area(O_{next}) - Area(c_{ij}^{t});
return I

Algorithm 3.3 Pseudocode for the estimating influence algorithm

As shown in Algorithm 3.3, the influence is defined by the current and future infill usage. To calculate the influence I_{ij}^{t} of each supporting plane, the sums of the total area needed to remove all of the roof points before and after splitting using the supporting plane are compared. The larger the influence, the more the supporting plane reduces the area of the minimum supporting plane of the other roof point.

To sum up, the virtual split function is a key step in splitting algorithm C that allows the quantitative calculation of the infill structure reduction by looking ahead one step. The influence is calculated as the difference between the area of the minimum infill structure needed in the current phase and the predicted area in the virtual next phase. Here, the area of the minimum infill structure does not mean the true value in each phase. If the area of the minimum infill structure generated can be accurately calculated, it is a contradiction because it means that the solution to the NP-hard problem is already known. Therefore, the area of the minimum infill structure required in each step is approximated, and splitting algorithm B is used to achieve

this.

If the virtual split function is applied to all supporting planes, information about the most influential supporting plane that reduces the infill structure can be obtained. The algorithm is expressed in Algorithm 3.4.

Splitting algorithm C

Data: M, P
Result: O
initialization
t = 0;
while *P is not empty* **do**
  for $\forall\, c_{ij}^{t}$ **do**
    I_{ij}^{t} = estimatingInfluenceAlgorithm(M^{t}, P, c_{ij}^{t});
  end
  $O_{t+1} = \arg\max_{c_{xy}^{t}} I_{xy}^{t}$
  M^{t+1}=$split(M^{t}, O_{t+1})$
  t = t+1;
  update P
end
return O

Algorithm 3.4 Pseudocode for splitting algorithm C

The following Figs. 3.19 and 3.20 are examples to help understand how splitting algorithm C works. $I_{52}^{1} = 27$ has a higher influence than $I_{51}^{1} = 0$, and when c_{52}^{1} is selected, the total infill is reduced to 36, whereas when c_{51}^{1} is selected, the total infill becomes 63. However, the area of the two supporting planes is the same, and so it is optimal to determine the order of the supporting planes using their influence.

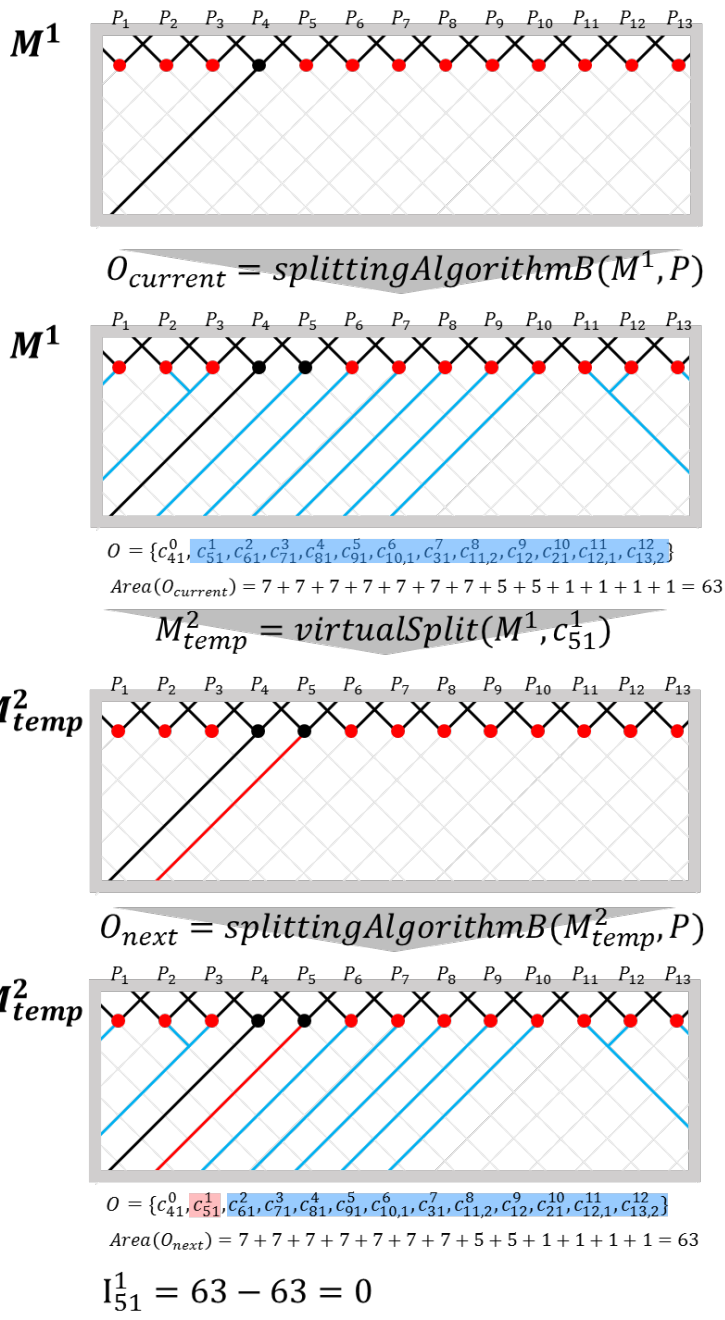


Fig. 3.19 Example of influence: estimation of I_{51}^2

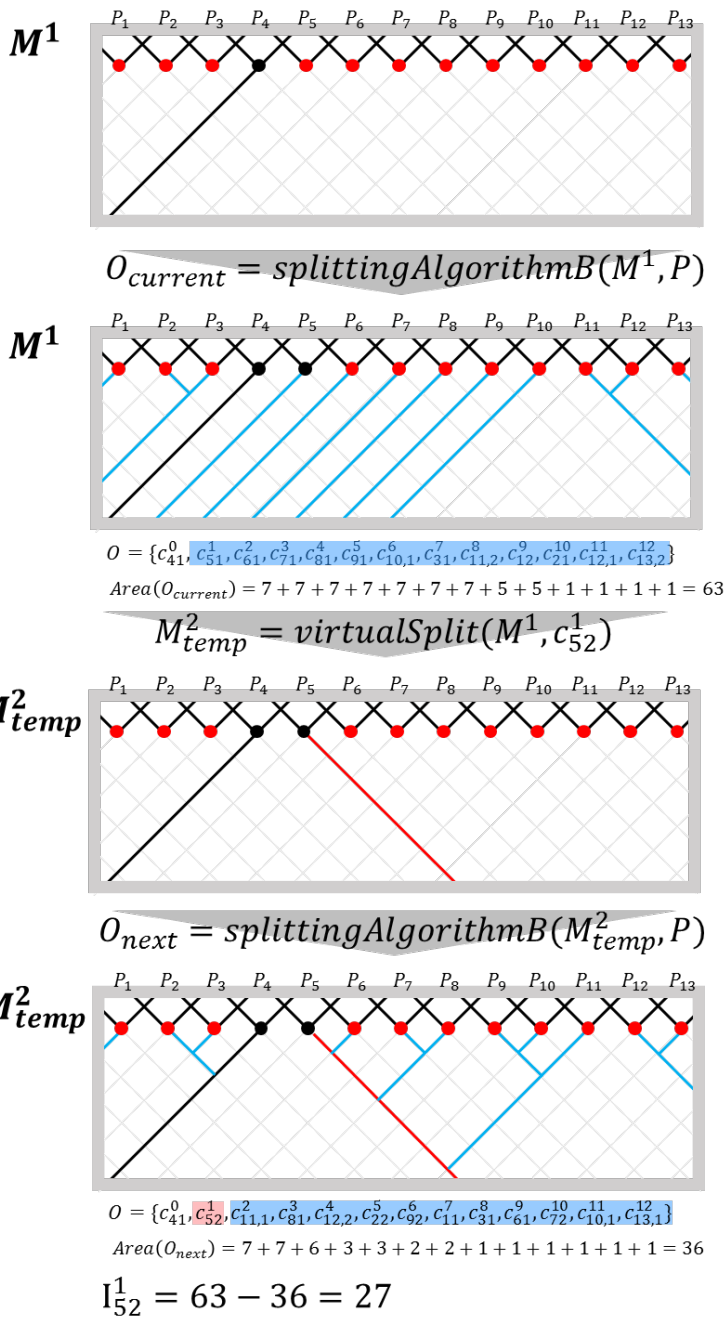


Fig. 3.20 Example of influence: estimation of I_{52}^2

3.7 Difficulties and solutions of 3D implementation

In two dimensions, the unit block is a rotated square, and in three dimensions the unit block is a rotated cube. Therefore, the candidate plane is two in two dimensions and three in three dimensions.

In the 2D case, the supporting plane includes only one roof point, but in the 3D case, a plurality of roof points share one supporting plane. For **N** roof points, there are **M (< 3N)** candidate planes, not **3N**. Therefore, it is efficient in terms of computational cost to remove the overlapping planes among the **3N** planes and then split the merged block only for **M** supporting planes.

If there are **N** roof points currently in the 2D case, there are **N-1** roof points in the next step. On the other hand, if there are **N** roof points in the 3D case, there are **N-s** roof points in the next step. Here, **s** is the number of roof points sharing the plane to be divided in the current phase. This is a more complicated problem because the number of roof points in the next step cannot be accurately predicted in the current phase.

3.8 Generating the tool path

In order to convert the infill structure defined as the contact surface between blocks into the actual nozzle path, post-processing is required. Since the shape information of an object is used only to create unit blocks or to classify a unit block as a BUB or an IUB, there is no detailed information on the shape in the data structure of the block. Therefore, it is necessary to modify the infill information obtained from the blocks according to the shape.

In order to change to the path of the nozzle, the cross-section of the contact surface between the blocks at an arbitrary height value in the stacking direction (Z axis) is required. If the intersection line of each unit rectangle valid for infill is obtained, the geometry information extraction for tool path generation is completed.

It is necessary to merge the obtained lines. It looks like one line segment, but it is expressed in multiple lines. When a tool path is created by dividing one long segment into a plurality of segments, unnecessary motor acceleration and deceleration occurs, resulting in a delay in manufacturing.

Since the infill structure for the BUBs is generated as well, the line segments are slightly deviated from the shape of the object. A point intersecting the polygon corresponding to the shape of the object is extracted, and line segments are divided. Among the divided line segments, those on the outside of the object are removed.

3.9 Reinforcing methods

The algorithm described in Section 3.6.3 focuses on ensuring the printability of an object by supporting the roof region with minimum infill. However, this may weaken the overall structure of the object. To compensate for this, ways to increase the strength of an object, in whole or in part, are discussed in this section.

The strength of an object can be controlled in the unit block merging step where a truss structure is created on the surface of the object to reinforce the strength. Likewise, if the BUBs (the RUBs and NRUBs), including the surface of the object, are excluded from the merging step, the strength reinforcing effect can be seen, as shown Fig. 3.21.

Depending on the shape of the object, the strength of a particular area may be important. In the conventional 2D pattern method, only the strength of a particular area cannot be reinforced since the infill density determines the strength of the object. Therefore, it is difficult to reinforce the strength of certain parts of an object because it is uniformly generated in all regions depending on the infill density. The method proposed in this thesis can increase the infill structure of a particular area by maintaining unit blocks existing in the area to be strengthened without merging. It is possible to generate the infill structure at the maximum density in the weak region and the roof region, and to generate the infill structure only with the minimum amount necessary for stable manufacturing in the other regions.

Hence, the methods proposed in this thesis can generate infill structure in three ways according to the requirements of fast manufacturing, overall strength reinforcement,

and partial strength reinforcement. The effectiveness of each method is confirmed in the Chapter 5. Hereinafter, the method for quick fabrication is referred to as M1, the method for overall strength reinforcement is referred to as M2, and the method for reinforcing the strength of a specific region is referred to as M3.

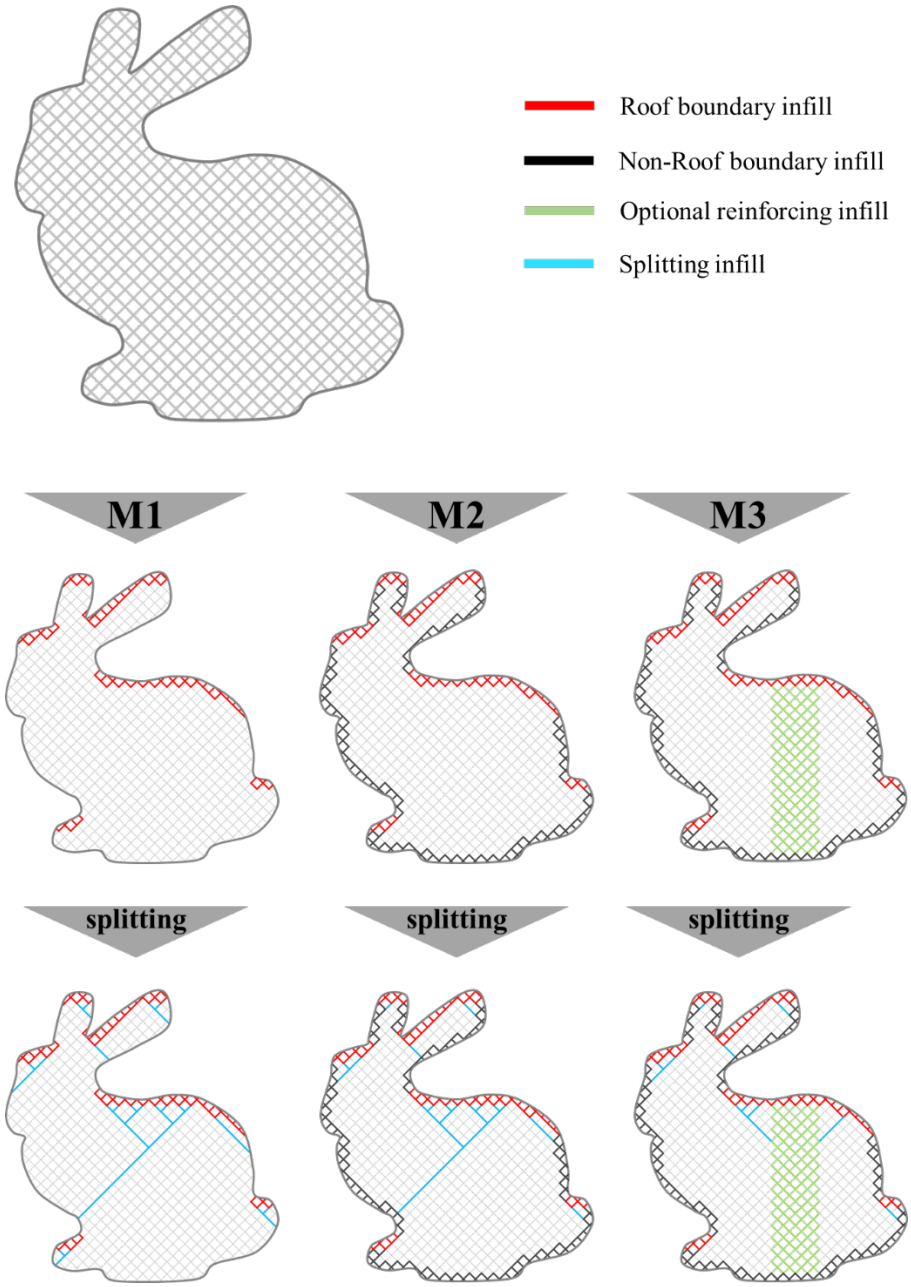


Fig. 3.21 Differences between M1, M2, and M3

CHAPTER 4.
Extension of the method

4.1 Extension of the dimensions

The contact surface between the blocks acting as an infill structure is formed along the plane of the unit blocks, and so the shape of the unit blocks directly affects the infill structure generation results. In the previous chapter, cubic unit blocks are created using partitioning planes in three directions. The three-way partitioning planes have only three directions in the supporting plane, and so the infill structure, which can be viewed as a branch of the tree structure, also extends along three directions, which causes the form of the infill structure generated to be too simple. In other words, there are three limited directions that lack the flexibility to solve problems. In this chapter, four partitioning planes are applied to expand the dimensions of the unit blocks and supporting planes. If the generation of an extended infill structure and the improved effect are confirmed, it is possible to prove the extensibility of the algorithm.

4.1.1 Defining the extension of the dimensions

Four partitioning planes produce unit blocks with a four-directional surface, and the number of spread directions for infill increases to four as well. In this case, a greater probability of improvement of the infill structure exists compared to using cubic blocks because there is an increase in available patterns for infill structure generation with the extra direction. Specifically, it can be said that the dimensions of the blocks are extended when the number of partitioning planes increases. The term dimension is used because it has the same meaning as the direction for infill spreading, which is the split direction in the splitting step. Hence, extended unit blocks with an extra dimension can be introduced so that they can generate a more efficient infill structure with less material by allowing more directions for splitting.

4.1.2 Requirements of the unit block

There are several types of unit block that can be created with four partitioning planes. Although the extension of a unit block is effective in terms of space utilization, it is not possible to simply select a unit block as a result of dividing the available space in several directions. In order for a unit block to be used for infill generation, several basic conditions must be met.

1. **Stability**: the unit block's surface should be stable with a given threshold angle, which is usually 45 degrees, and should be printable with no roof region.
2. **Minimum number of types of unit block**: a model should be assembled with a

limited and minimum number of unit blocks, but too many types of unit block makes the solution difficult.

3. **Symmetricity**: symmetric unit blocks prevent efficiency fluctuation which can arise in the partitioning direction during space partitioning. Therefore, unit blocks must have the isometric property so that rotation or mirroring of a block does not affect the partitioning efficiency.

A unit block solution that can be created by four partitioning directions has several candidates, as shown in Fig. 4.1. The first (uppermost) unit blocks are inappropriate because they contain roof planes. The second unit blocks are unsuitable because they consist of three types, which is too many. The third unit blocks are also unsuitable because they have an asymmetric shape; since they have no isotropic property, a change in the efficiency of spatial division along the direction may appear. Therefore, it is the final case with a combination of octahedrons and tetrahedrons that satisfies all of the conditions mentioned above; this multi-block solution consists of only two types, each with a layered angle of 54.74 degrees and a symmetrical structure. As a result of selecting unit blocks satisfying all of the above conditions, the multi-block solution of octahedrons and tetrahedrons was used in this study.

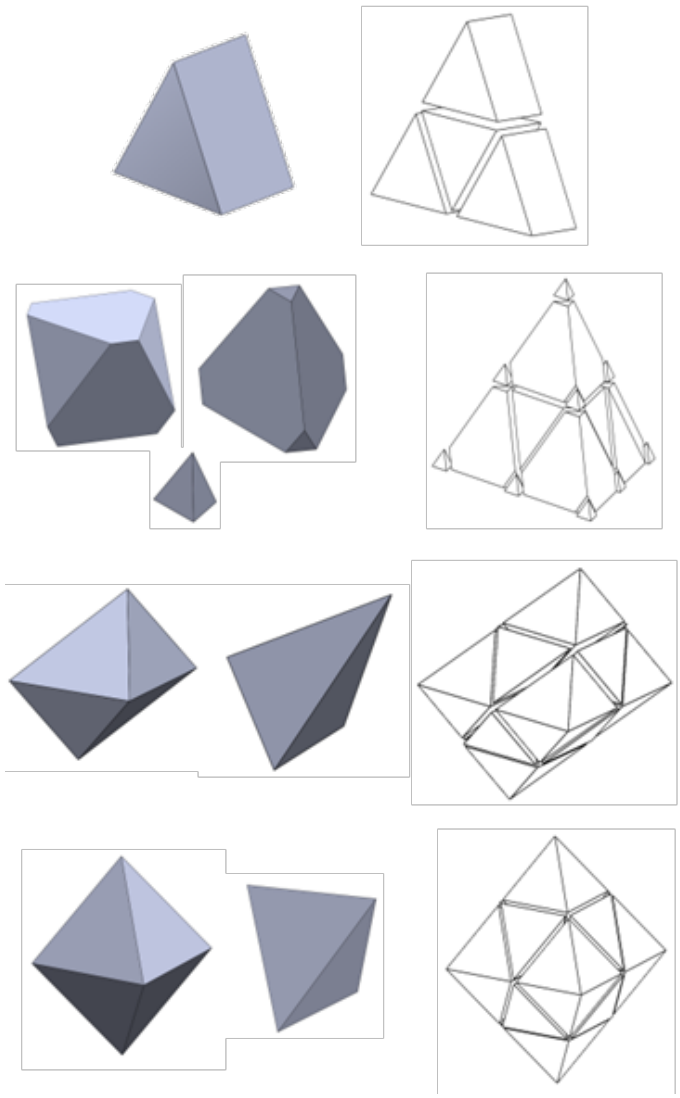


Fig. 4.1 Four directionally partitioned blocks as unit-block candidates

4.1.3 Considerations for the implementation

Several modifications are needed to implement an extended block with four supporting directions. In the merging step, when the different block types are merged, they must be merged together because they share faces. When examining the BUBs and RUBs in the merging step, the two different types of unit blocks should be examined. In particular, when finding a roof point in the merged blocks, it is important to consider that it can be defined differently for each block type. For example, in the extended unit blocks used in this study, an octahedron can contain a roof point but a tetrahedron can be part of a roof edge. Finally, since the dimensions of the unit block has increased to 4, the split direction also increases to 4, and so the splitting step should be executed in all 4 directions.

4.2 Parallelepipeds as unit blocks

If the layered angle condition is satisfied, the unit block need not be a regular hexahedron. It is expected that the larger the stacking angle is, the more stable the fabrication and the stronger the load in the stacking direction are expected to be. Therefore, infill can be generated using a parallelepiped instead of the previously used regular hexahedron. In order to implement this, it is necessary to eliminate the condition that they are orthogonal to each other. The following figure is a 2D representation of the results obtained using a parallel hexahedron. The effect of this is discussed in Chapter 5.

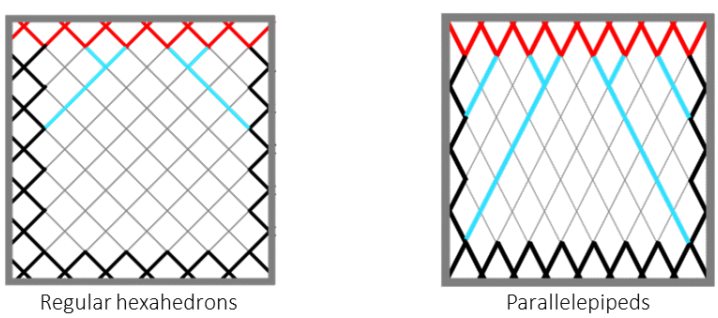


Fig. 4.2 Regular hexahedrons and parallelepipeds as unit blocks

4.3 Extension to SLA and SLS

The focus of this study is on FDM-type 3D printers. However, in this section, a discussion on how to generate optimal infill for 3D printers such as SLA and SLS is presented. Since the FDM-type printer uses motors to move the nozzle, the tool path must be simple, so it must be as straight as possible and has the feature of dividing the inner space of the object so that an independent closed structure is created inside the object.

3D printers such as SLA and SLS require a process to remove unused material. If a hole in an object is created to remove unused material and there are independent closed spaces, a number of holes in the object need to be created. In SLA and SLS, the complexity of the tool path does not affect the manufacturing time, so a straight tool path is not necessary.

In the FDM-type printer, the activated face of the unit block is infill structure. In the case of SLA- and SLS-type printers, if each edge of the face is composed of infill structure, it is easier to remove unused material from an open structure, as shown in Fig. 4.3.

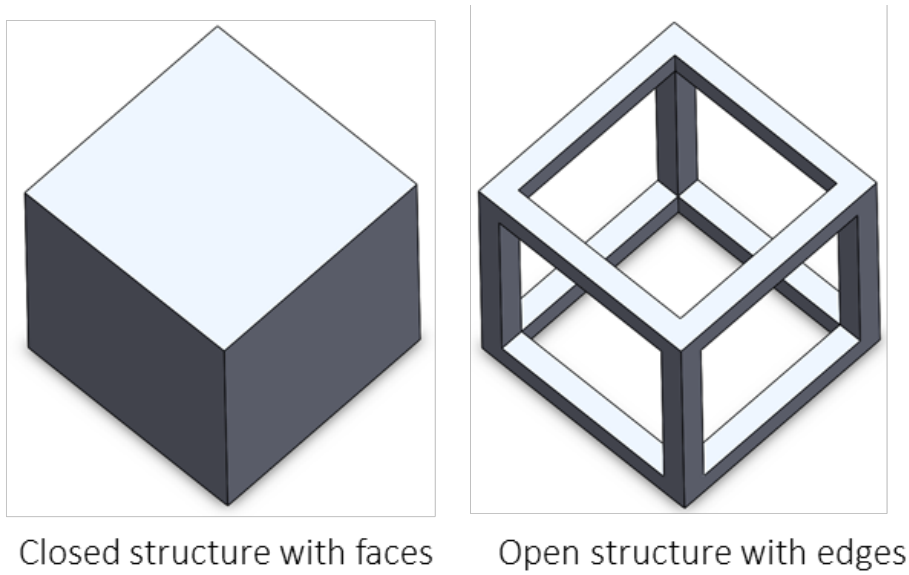


Fig. 4.3 Closed and open structures

CHAPTER 5.

Results and Discussion

The proposed algorithm was applied using CuraEngine [51], which is an open-source application for processing 3D models into 3D printing instructions. An Ultimaker 2 3D printer [52] and DP200 [53] were used for manufacturing with the layer thickness of all models fixed at 0.2 mm and polylactic acid (PLA) used as the material. In addition, because this study is related to infill structure, material usage and manufacturing time to generate the support structure were not recorded.

To verify the proposed algorithm, the model was tested under various conditions, and its performance was classified by two categories: efficiency and isotropy. In order to compare the efficiencies, material usage and manufacturing time were compared when producing at the same density, which means having the same printability for the roof region. For verifying the isotropic property of the proposed method, the rigidity of the object with respect to the load was tested in various directions, and splitting algorithm C was used for M1, M2, and M3. The parameter settings are shown in Table 5-13.

5.1 Efficiency

First, the performance of each method is compared using the Stanford Bunny model whose characteristics are detailed in Table 5-1.

Table 5-1 Skin information and geometry of the Stanford Bunny model.

Model	Geometry					Skin	
	Width (mm)	Length (mm)	Height (mm)	Volume (mm^3)	Area (mm^2)	Manufacturing Time (minutes)	Material Usage (grams)
Stanford Bunny	154	118	153	742490	56097	367	65.5

Table 5-2 Manufacturing time and material usage for the Stanford Bunny model.

Model	Manufacturing Time: minutes (percentage)			Material Usage: grams (percentage)		
	Grid	M1	M2	Grid	M1	M2
Stanford Bunny	813	170 (20)	326 (40)	174.5	32.3 (18)	60.2 (34)

From Table 5-2, M1 reduces material usage to approximately 18% of that for the conventional grid pattern method and likewise reduces the manufacturing time to approximately 20%, and M2 reduces material usage to approximately 34% of that for the conventional grid pattern method and reduces manufacturing time to approximately 40%. Hence, M1 is more effective than M2 in terms of the amount of material used and manufacturing time. The extent of the time saved is less than that for material saved because although the amount of material used is reduced, the tool path must still cover the entirety of the original geometry. Nevertheless, the proposed methods M1 and M2 saved time compared with the conventional grid pattern method.

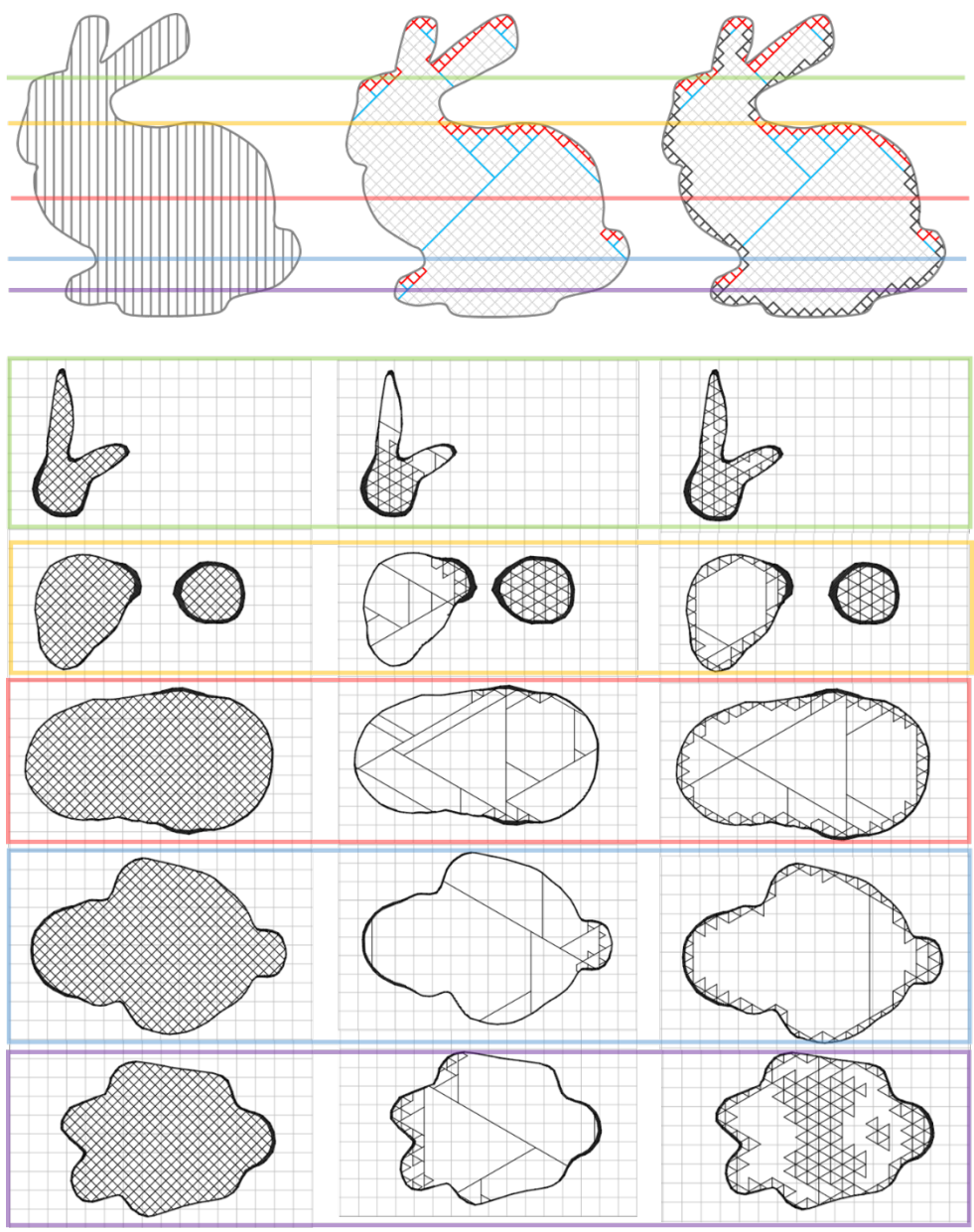


Fig. 5.1 Tool paths for grid, M1, and M2 for the Stanford Bunny model.

In Fig. 5.1, the tool paths for each layer of the Stanford Bunny model generated differently by each method are shown. The first column shows the tool path for the 2D grid pattern infill structure, and it can be seen that a uniform infill structure is created in all layers. The second column shows the tool path for the M1 method that supports only the roof region, such as the head and the back of the model. Except for the roof region, it was confirmed that the infill structure is very sparse. The third column shows the tool path for the M2 method which supports the region corresponding to the boundary of the model. It can be seen that a truss-like structure was formed at the boundary of the model in all layers.

Next, in the second experiment, the 2D grid pattern method and the two proposed methods M1 and M2 were compared in terms of material usage and manufacturing time for several models obtained from Pinshape [54]. As shown in Figs. 5.2 and 5.3, and Tables 5-3 and Table 5-4, although the efficiencies of material usage and manufacturing time are different depending on the shape (similar to the first experiment), the M1 method is more effective than M2 in terms of the amount of material used and manufacturing time, and the extent of the time saved is less than that for material saved.

Table 5-3 Skin information and geometry for the test models.

Model	Geometry					Skin	
	Width (mm)	Length (mm)	Height (mm)	Volume (mm^3)	Area (mm^2)	Manufacturing Time (minutes)	Material Usage (grams)
Cat	200	85	73	261146	31759	215	34.5
Fawn	150	94	124	397679	41964	282	46
Fox	100	122	167	516021	44877	279	48.5
Giraffe	150	108	167	355511	41015	270	44.2
Moai	150	110	187	836458	65373	399	71.2
Skull	100	149	127	1064351	57825	353	62.5
Yoda	150	109	107	390533	40168	276	45.3

Table 5-4 Manufacturing time and material usage for the test models.

Model	Manufacturing Time: minutes (percentage)			Material Usage: grams (percentage)		
	Grid	M1	M2	Grid	M1	M2
Cat	276	88 (31)	165 (59)	59.3	16.3 (27)	31.2 (52)
Fawn	417	122 (29)	237 (56)	88.7	23.4 (26)	45.3 (51)
Fox	564	88 (15)	244 (43)	120.2	15.4 (12)	43.8 (36)
Giraffe	383	106 (27)	216 (56)	81	19.7 (24)	41.6 (51)
Moai	869	119 (13)	372 (43)	196	22.5 (11)	65.6 (33)
Skull	1161	184 (15)	372 (32)	253.2	34.4 (13)	68.1 (26)
Yoda	428	110 (25)	200 (46)	90.1	19 (21)	36.5 (40)

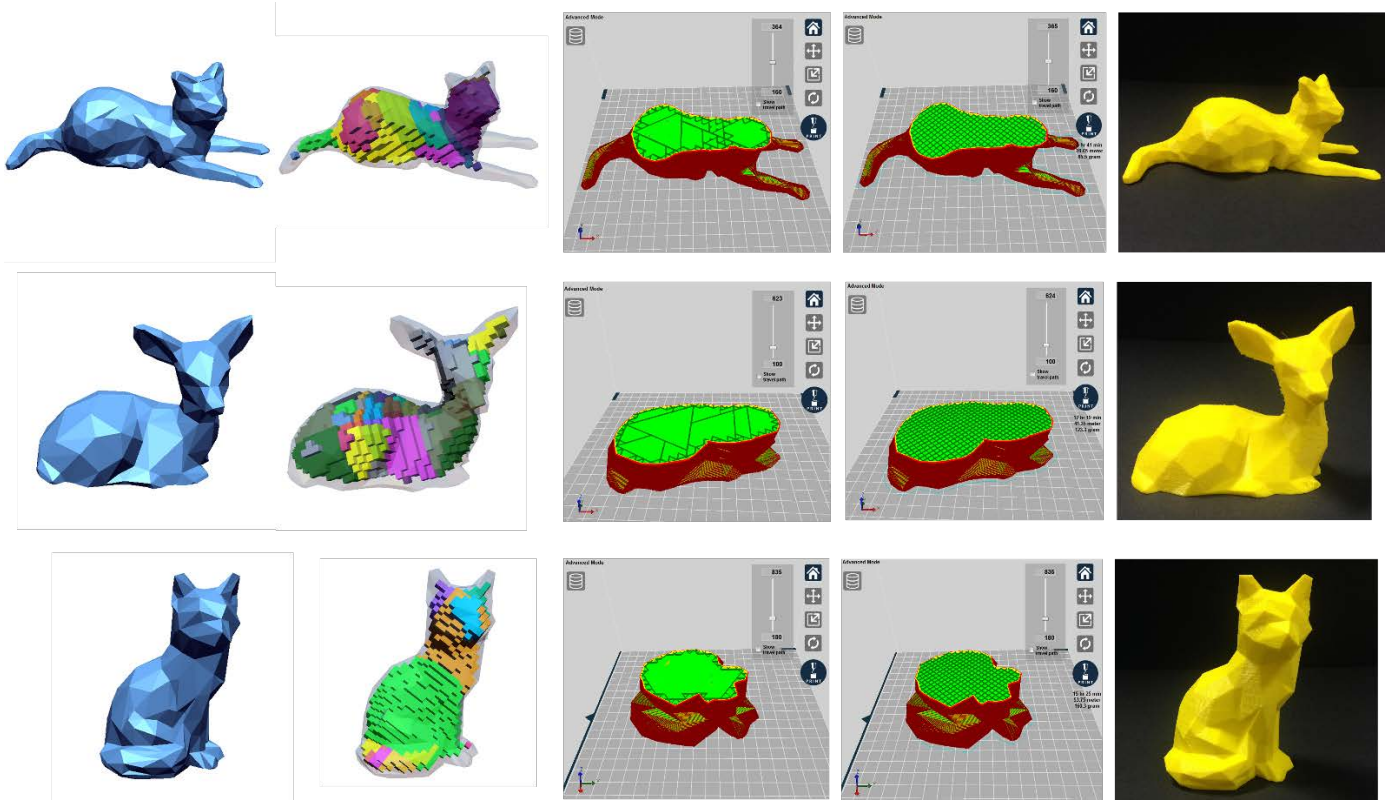


Fig. 5.2 Results for various models used in the second experiment. Top to bottom: cat, fawn, and fox. Left to right: original model, inner combined blocks, the tool path for M2, the tool path for 2D grid, and the printed model

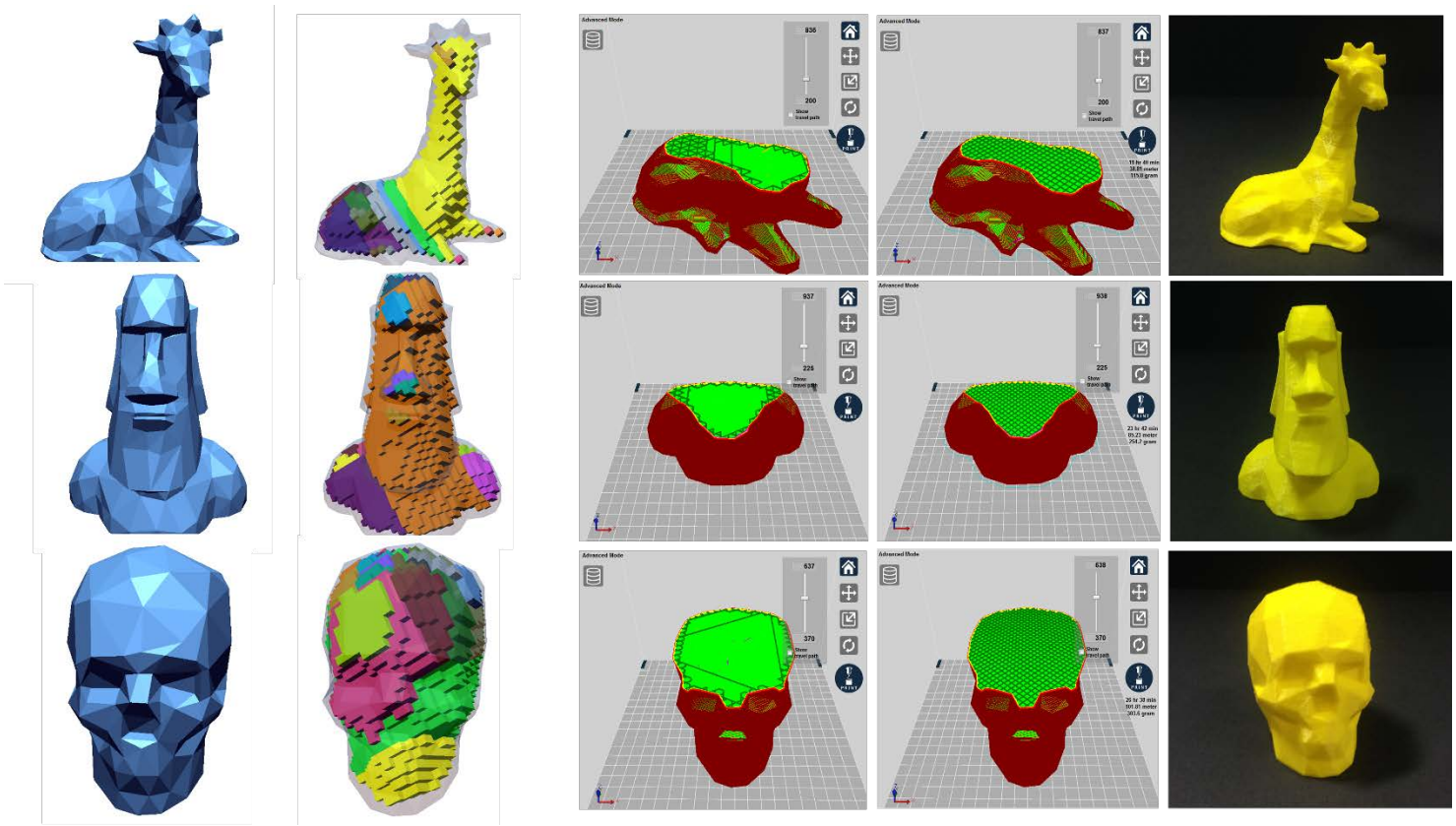


Fig. 5.3 Results of various models used in the second experiment. Top to bottom: giraffe, moai, and skull. Left to right: original model, inner combined blocks, the tool path for M2, the tool path for 2D grid, and the printed model

In the second experiment, different efficiencies were observed depending on the shape, and so it was necessary to analyze how the shape information affects infill structure generation. For a more detailed analysis, infill structure generated by M1 and M2 method was classified into 3 types: infill_{roof}, $\text{infill}_{boundary}$, and infill_{inner}. Infill_{roof} is an infill structure created by unit blocks around the roof region (RUBs), $\text{infill}_{boundary}$ is an infill structure created by unit blocks on the boundary of the shape (BUBs) and also contains infill_{roof}, and infill_{inner} is the infill generated in the splitting step.

In the third experiment, a simple box model was used to analyze the efficiency depending on the volume; the box size was increased at the same infill density (20%). The parameter settings were the same as in Table 5-13.

Table 5-5 Skin information and geometry of the box models.

Model	Geometry					Skin	
	Width (mm)	Length (mm)	Height (mm)	Volume (mm^3)	Area (mm^2)	Manufacturing Time (minutes)	Material Usage (grams)
Box1	40	40	40	64000	9600	45	9.2
Box2	60	60	60	216000	21600	102	21
Box3	80	80	80	512000	38400	181	37.5
Box4	100	100	100	1000000	60000	283	58.8
Box5	120	120	120	1728000	86400	408	84.8
Box6	140	140	140	2744000	117600	556	115.6

Table 5-6 Comparison of manufacturing times according to the scale of the box models.

Model	Manufacturing Time: minutes (percentage)						
	Grid	M1			M2		
		Total	Infill_{roof}	Infill_{inner}	Total	Infill_{bound}	Infill_{inner}
Box1	94	**21(22)**	7(33)	14(67)	**50(53)**	41(82)	9(18)
Box2	328	**63(19)**	21(33)	42(76)	**137(41)**	111(81)	26(19)
Box3	775	**116(15)**	29(25)	87(75)	**261(33)**	205(78)	56(22)
Box4	1539	**228(15)**	61(27)	167(73)	**404(26)**	294(72)	110(28)
Box5	3658	**339(12)**	69(20)	270(80)	**594(22)**	428(72)	166(28)
Box6	4235	**515(12)**	123(24)	392(76)	**925(21)**	687(74)	238(26)

Table 5-7 Comparison of material usage according to the scale of the box models.

Model	Material Usage: grams (percentage)						
	Grid	M1			M2		
		Total	Infill_{roof}	Infill_{inner}	Total	Infill_{bound}	Infill_{inner}
Box1	94	**4.1(19)**	1.3(31)	2.8(69)	**9.1(42)**	7.3(80)	1.8(20)
Box2	328	**12.3(16)**	4.2(34)	8.1(66)	**25.9(34)**	20.5(79)	5.4(21)
Box3	775	**23.1(12)**	5.6(24)	17.5(76)	**49.1(27)**	37.2(75)	11.9(25)
Box4	1539	**46.5(13)**	12.4(26)	34.1(74)	**75.5(21)**	53.6(70)	21.9(30)
Box5	3658	**69.7(11)**	13.5(19)	56.2(81)	**110.3(17)**	76.3(69)	34(31)
Box6	4235	**107.5(10)**	25.6(23)	81.9(77)	**177.4(17)**	126.8(71)	50.6(29)

In the grid pattern method, the amount of material used to produce the inner support structure is proportional to the volume of the object because of the uniform distribution of the support structure. By contrast, with the proposed methods, the amount of material used is proportional to the surface area of the object or the roof region. This is because the inner support structure is typically generated based primarily on the BUBs. Therefore, an increased time and material efficiency over the conventional method is observed as the size of the object to be manufactured increases.

There is no significant change in the ratio of infill_{inner} to infill_{roof} in M1, it can be seen that it is not influenced by the volume, as is the case with M2.

In the next experiment, the amount of infill structure generated by varying the density of the same object was measured to check the efficiency according to the infill density.

Table 5-8 Comparison of manufacturing times according to the density of the inner support structure.

Model	Grid	Manufacturing Time: minutes (percentage)					
		M1			M2		
		Total	Infill_{roof}	Infill_{inner}	Total	Infill_{bound}	Infill_{inner}
Box4	516	**142(27)**	63(44)	79(56)	**299(57)**	255(85)	44(15)
Box4	1035	**170(16)**	47(27)	123(73)	**356(34)**	272(76)	84(24)
Box4	1532	**228(15)**	61(26)	167(74)	**404(26)**	295(73)	109(27)
Box4	2030	**236(11)**	46(19)	190(81)	**395(19)**	270(68)	125(32)
Box4	2544	**276(10)**	59(21)	217(79)	**434(17)**	294(67)	140(33)
Box4	3046	**298(9)**	49(16)	249(84)	**433(14)**	285(65)	148(35)

Table 5-9 Comparison of material usage according to the density of the inner support structure.

Model	Grid	Material Usage: grams (percentage)					
		M1			M2		
		Total	Infill_{roof}	Infill_{inner}	Total	Infill_{bound}	Infill_{inner}
Box1	118.4	**29.1(24)**	13.1(45)	16(55)	**61.6(52)**	52.4(84)	9.3(16)
Box2	236.9	**34.1(14)**	9.3(27)	24.9(73)	**69.1(29)**	51.8(74)	17.3(26)
Box3	355.4	**46.5(13)**	12.4(26)	34.1(74)	**75.5(21)**	53.6(70)	21.9(30)

Box4	473.7	**47.4(10)**	8.4(17)	39(83)	**69(14)**	43.6(63)	25.4(37)
Box5	592.1	**55.8(9)**	11.4(20)	44.4(80)	**74.6(12)**	46.3(62)	28.3(38)
Box6	711.2	**59.9(8)**	8.8(14)	51.1(86)	**71.9(10)**	42.2(58)	29.7(42)

Tables 5-8 and 5-9 compare the results achieved for different densities of inner support structure. To adjust the density, the grid spacing was modified for the conventional method and the size of the unit blocks was modified for the proposed method. The manufacturing time and material used increased as the density increased for the grid pattern method but showed much less variation for the proposed methods.

For the proposed methods, an increase in density means that the size of the unit blocks decreases, and as this happens, the number of roof points in the combined block consequently increases, thus the size of the inner support structure also increases. However, because the size of the inner support structure created based on the BUBs simultaneously decrease, the total change in the size of the inner support structure is small.

It was also confirmed that the ratio of infill_{inner} increases as the infill density increases because, as the infill density increases, the size of the unit blocks becomes smaller and infill_{roof} becomes similar to the surface area of the roof region.

In conclusion, the proposed methods are not significantly affected by the choice of density of the inner support structure. For an object with dimensions of 100 mm × 100 mm × 100mm with 30% density, M1 reduced the amount of material used to approximately 13% and the manufacturing time to approximately 15% compared with the grid pattern method, and M2 reduced the amount of material used to approximately 21% and the manufacturing time to approximately 26%. In addition, the efficiency improved as the volume of the object and the density of the inner support structure increased.

5.1.1 Comparison with the extended method

Extension of dimensions

To verify the effect of the extended method, the material usage and the manufacturing time were checked while increasing the scale of the box model. In Tables 5-10 and 5-11, the figures in parentheses in the Total, Extended M1 Total and Extended M2 Total columns are the ratios of each recorded value (without parentheses) with their respective M1 Total or M2 Total (highlighted) expressed as percentages.

Table 5-10 Comparison of manufacturing times of the extended methods according to the scale of the box models.

Model	Manufacturing Time: minutes (percentage)							
	M1 Total	Extended M1			M2 Total	Extended M2		
		Total	Infill_{roof}	Infill_{inner}		Total	Infill_{bound}	Infill_{inner}
Box1	21	**15(71)**	6(40)	9(60)	50	**42(84)**	36(85)	6(15)
Box2	63	**42(66)**	20(47)	22(53)	137	**123(89)**	105(85)	18(15)
Box3	116	**92(79)**	47(51)	45(49)	261	**257(98)**	221(85)	36(15)
Box4	228	**107(46)**	32(29)	75(71)	404	**315(77)**	247(78)	68(22)
Box5	339	**192(56)**	74(38)	118(62)	594	**539(90)**	437(81)	102(19)
Box6	515	**302(58)**	134(44)	168(56)	925	**839(90)**	688(82)	151(18)

Table 5-11 Comparison of material usage of the extended methods according to the scale of the box models.

Model	Material Usage: grams (percentage)							
	M1 Total	Extended M1			M2 Total	Extended M2		
		Total	Infill_{roof}	Infill_{inner}		Total	Infill_{bound}	Infill_{inner}
Box1	4.1	**2.7(65)**	0.9(33)	1.8(67)	9.1	**7.9(86)**	6.8(86)	1.1(14)
Box2	12.3	**8.2(66)**	3.5(42)	4.7(58)	25.9	**23.6(91)**	20.0(84)	3.6(16)
Box3	23.1	**18.6(70)**	9(48)	9.6(52)	49.1	**51.1(104)**	43.7(85)	7.4(15)
Box4	46.5	**20.7(44)**	4.9(23)	15.8(77)	75.5	**59.4(78)**	45.4(76)	14.0(24)
Box5	69.7	**38.6(55)**	13.2(34)	25.4(66)	110.3	**104.3(94)**	83.1(79)	21.2(21)
Box6	107.5	**62.2(57)**	25.9(41)	36.3(59)	177.4	**168.4(95)**	136.7(81)	31.7(19)

As shown in Tables 5-10 and 5-11, the extended method is more efficient than the original method; in particular, the Infill_{inner} was significantly reduced. This is because the direction of the supporting plane is extended in four directions, and the tree structure can be generated more efficiently. In extended M2, Infill_{bound} accounts for a large portion of the total infill, so the manufacturing time may increase, but overall it improved because of savings in Infill_{inner}. In conclusion, by extending the method, the number of supporting-plane candidates can be increased so that the amount of infill generated in the splitting step can be further reduced.

Parallelepipeds as unit blocks

For the box2 model, the manufacturing times were compared when using regular hexahedrons and parallelepipeds as unit blocks. In this case, the infill density is 30%, and other parameters are shown in Table 5-13.

When cubes were used, the production time was 11 hours and 52 minutes, and when parallelepipeds were used, it took 13 hours and 59 minutes. The reason for this is that as the unit block moves away from the spherical shape, inefficiency occurs in expressing the space and in the space division.

Extension to SLA and SLS

The infill structure for SLA and SLS discussed in Chapter 3 was created using FDM, as shown in Figs. 5.4 and 5.5. When a closed structure suitable for FDM was fabricated using FDM, it took 2 hours 47 minutes and used 24.8 grams of material. On the other hand, when an open structure suitable for SLA and SLS was fabricated with FDM, a similar amount (24.2) grams of material were used, but the production time increased to 4 hours and 42 minutes.

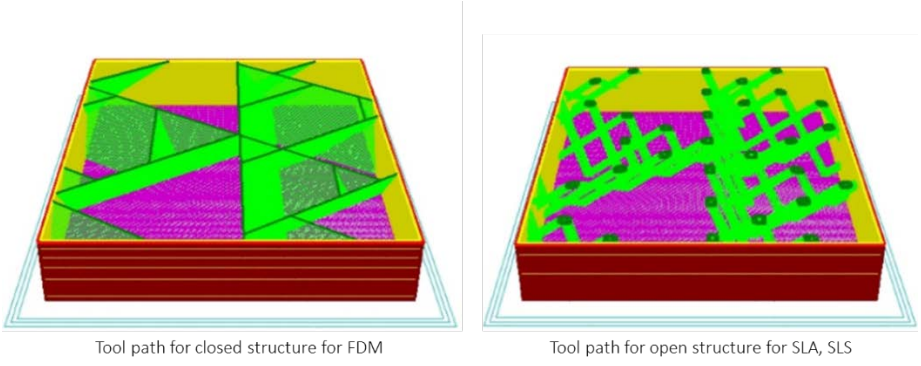


Fig. 5.4 Tool paths for closed and open structures

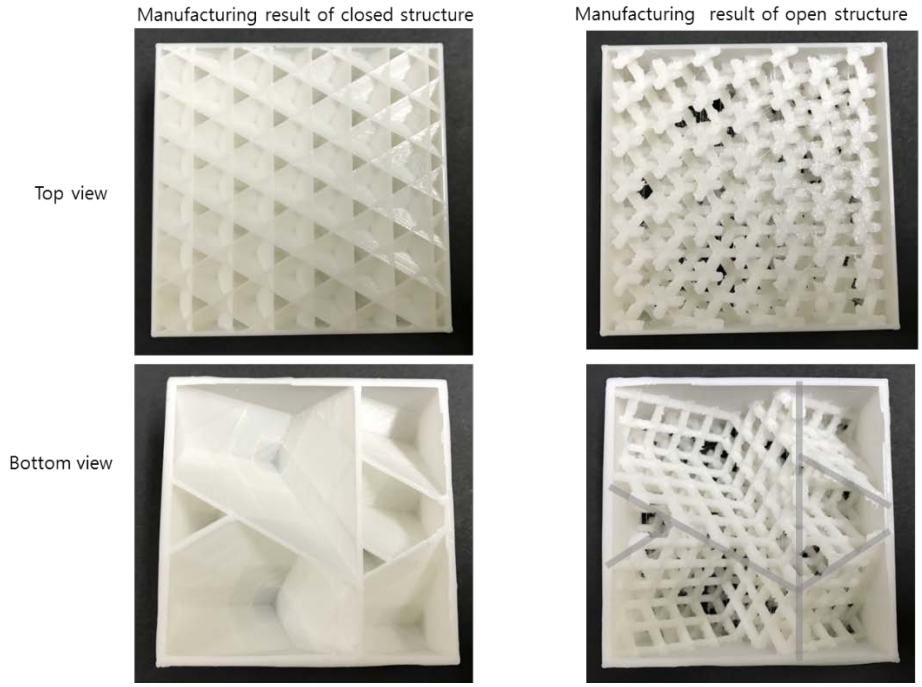


Fig. 5.5 Manufacturing results for closed and open structures

5.1.2 Comparison with related works

Honeycomb-like interior structure

It is difficult to compare directly with other studies because their objectives were different from the ones in this study. Research mentioned in Chapter 2 [5, 8, 9] focused on strength-to-weight aspects, but the focus of this study is on an object's printability and the self-supporting characteristic of its inner support structure. In particular, when using a truss-like structure [8, 9], either an additional support structure or a different printing method like SLS is required, whereas the proposed methods can be implemented on non-specific general FDM-type printers. Therefore, it can be viewed as being superior in terms of printability.

In addition, although the honeycomb-like interior structure [5] does not guarantee printability, it is easier to stack honeycomb-like structures than truss structures using FDM. Therefore, the results between the studies can be compared in terms of the amount of material used and manufacturing time. In an earlier comparison with the conventional 2D grid pattern method, which ensures printability, the efficiency of the proposed methods were checked. To perform experiments under similar conditions to those that produced Lu's results [5], the density was set at 40% because honeycomb-like cells occupy 60% of the model.

Table 5-12 Comparison between the various inner support structure generation methods for the kitten model.

Method	Total manufacturing time: minutes	Total material usage: grams
Grid	180 (100%)	34.2 (100%)
Lu's Honeycomb	243 (135%)	34.1 (99%)
M1	84 (46%)	14.3 (41%)
M2	117 (65%)	20.1 (58%)

Table 5-12 and Fig. 5.7 show the results achieved with the different methods of constructing the inner support structure. The proposed methods are more effective than Lu's honeycomb-like structure method in terms of the amount of material used and manufacturing time. Although the amount of material used for Lu's honeycomb-like structure is the same as that for the grid pattern method, the manufacturing time takes longer. This is due to the complexity of the tool path.

Uniform 3D infill pattern

Commercial software from Materialise [58] can provide a variety of 3D unit patterns, as shown in Fig. 5.6, to create a lightweight and robust inner support structure. The manufacture of objects using unit patterns from Materialise is

inefficient because they are generated uniformly, and the manufacturing time is lengthy due to the complicated tool path, as can be confirmed in Table 5-13. The results confirm that the material usage per hour is less than with a straighter tool path. In addition, the amount of material used is more than the proposed method.

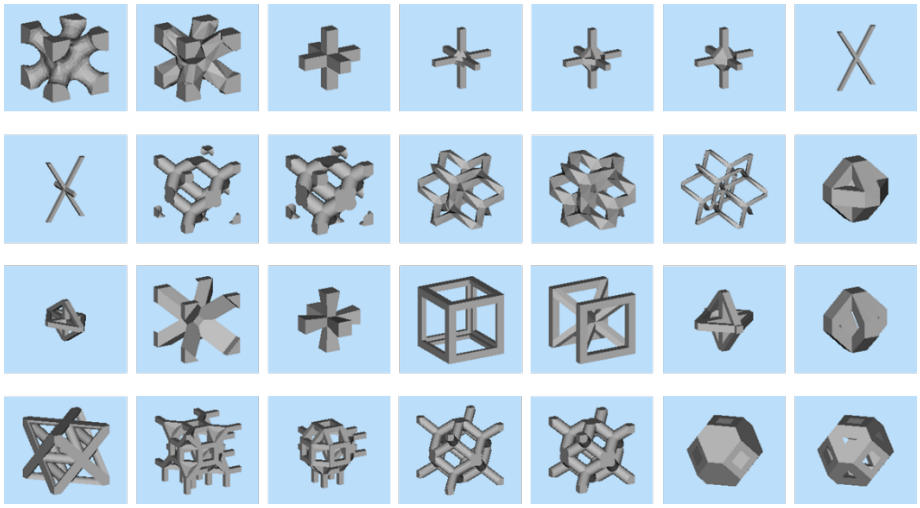


Fig. 5.6 Materialise unit patterns

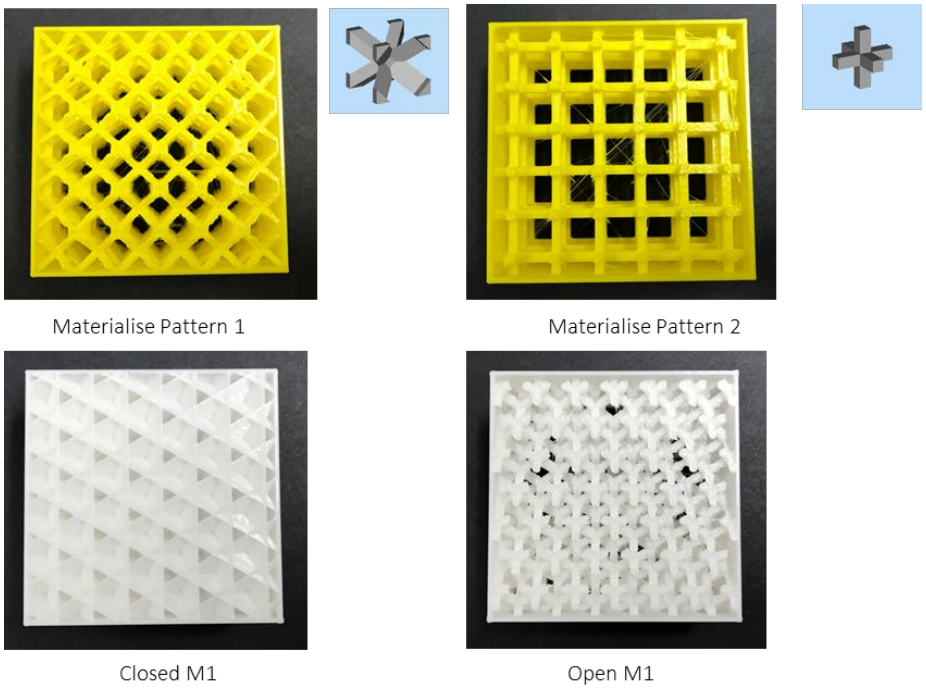


Fig. 5.7 Manufacturing results for Materialise patterns 1 and 2, closed structure M1, and open structure M1

Table 5-13 Comparison between grid, Materialise patterns 1 and 2, and the proposed model with closed and open structures for the 60mm×60mm×30mm box model.

Method	Manufacturing time (minutes)	Material usage (grams)	Gram / hour
Grid	196	31.4	9.6
Materialise Pattern 1	302	30.9	6.1
Materialise Pattern 2	254	30.2	7.1
Closed M1	155	23.5	9.1
Open M1	228	21.3	5.6

Fig. 5.8 The cross-sections of the kitten model for each method. Left to right : grid pattern, honeycomb-like interior structure, M2, and M1

5.2 Compression testing

Strengthening of an object is one of the important functions of infill, and so compression testing was carried out to analyze the effect of the proposed method on the strength of an object. An Instron 5582 universal testing machine [55] was used and the load was increased to 20 N/s (see Fig. 5.9).

Fig. 5.9 Instron 5582 universal testing machine

Since M1 is a method for improving the production speed, a compression test on it is meaningless. Therefore, M2, which is a method that reinforces infill strength, was compared with the grid method.

5.2.1 Isotropic property

In order to perform compression testing in various directions, a cube with a length of 50 mm was used. To perform the experiment under the same conditions as the 2D grid pattern method, the infill density was adjusted to have the same mass; when the densities of boxes made by the M2 method and the 2D grid pattern method were 30% and 12%, respectively, the mass was 31.4 grams. When the Z axis is referred to as the stacking direction, the strain is measured while increasing the load on all three axes. If the strain changed suddenly, the object was considered to have been destroyed.

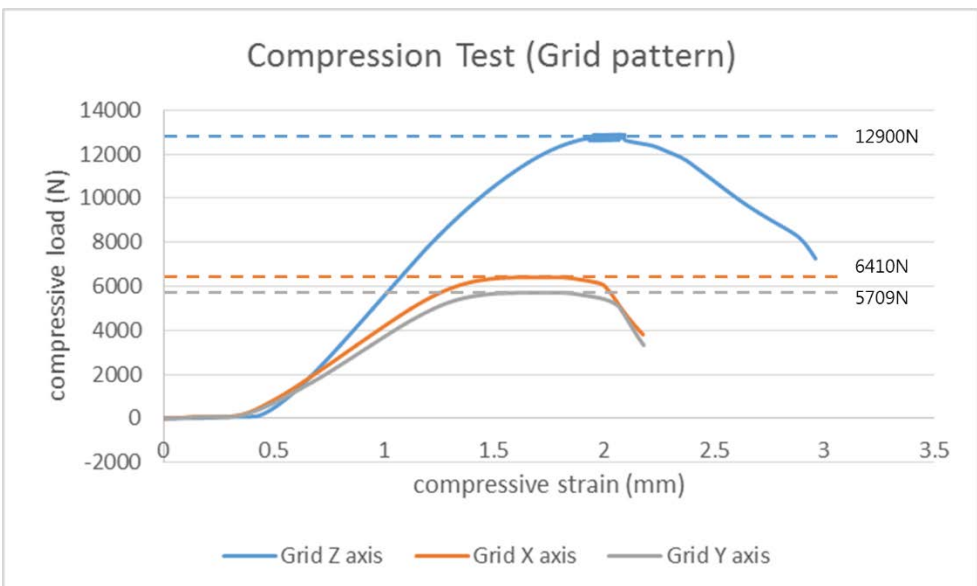


Fig. 5.10 Compression test results for the box manufactured using the grid pattern

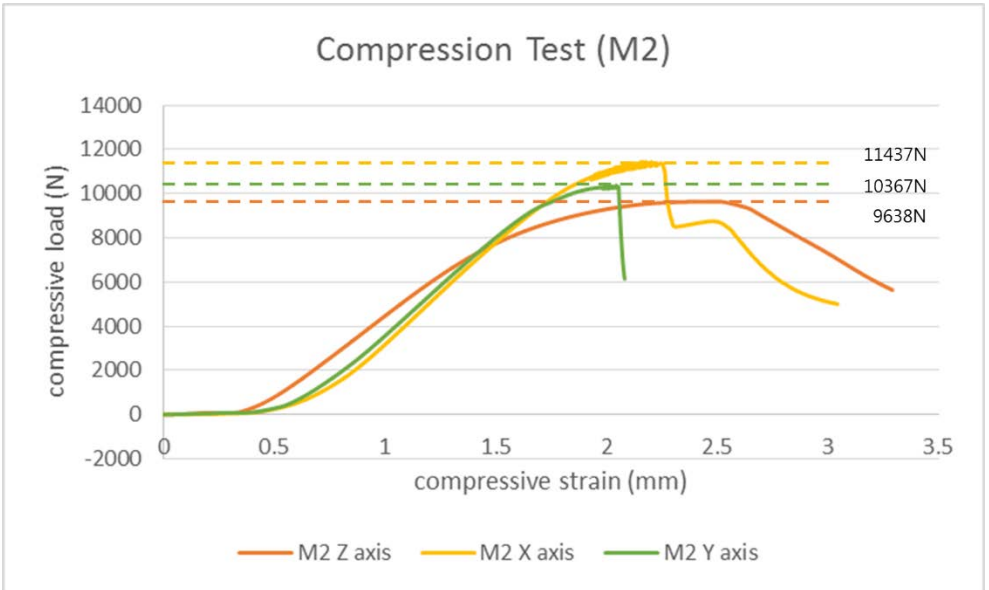


Fig. 5.11 Compression test results for the box manufactured using M2

As shown in Fig. 5.10, the infill structure generated by the 2D grid pattern method is strong in the stacking direction with 12900 N yield stress (Z axis), but weak in the X and Y axes directions with 6410 N and 5709 N yield stress, respectively. On the other hand, the compression test results of the infill structures generated using the M2 method are about 9638 N ~ 11437 N yield stress, which is weaker in the Z axis direction of the 2D grid pattern method but stronger in the X and Y axes directions (see Fig. 5.11).

In conclusion, the 2D pattern method is robust against compression in the stacking direction, but has weak isotropic characteristics in the others. However, the proposed method has good isotropic characteristics for compression regardless of direction.

5.2.2 Optional reinforcing

The proposed method can improve the strength of a specific part of an object. Through the stress analysis, more infill structure can be generated in an area where a strong load is applied, as shown in Fig. 5.12. The 2D grid pattern method increases the infill density in order to strengthen a specific part, which is inefficient because it strengthens the whole object. The sandglass shape is used to confirm the effect of optional reinforcement of proposed method M3. When subjected to stress, the hourglass is expected to be preferentially destroyed when the load is applied from above.

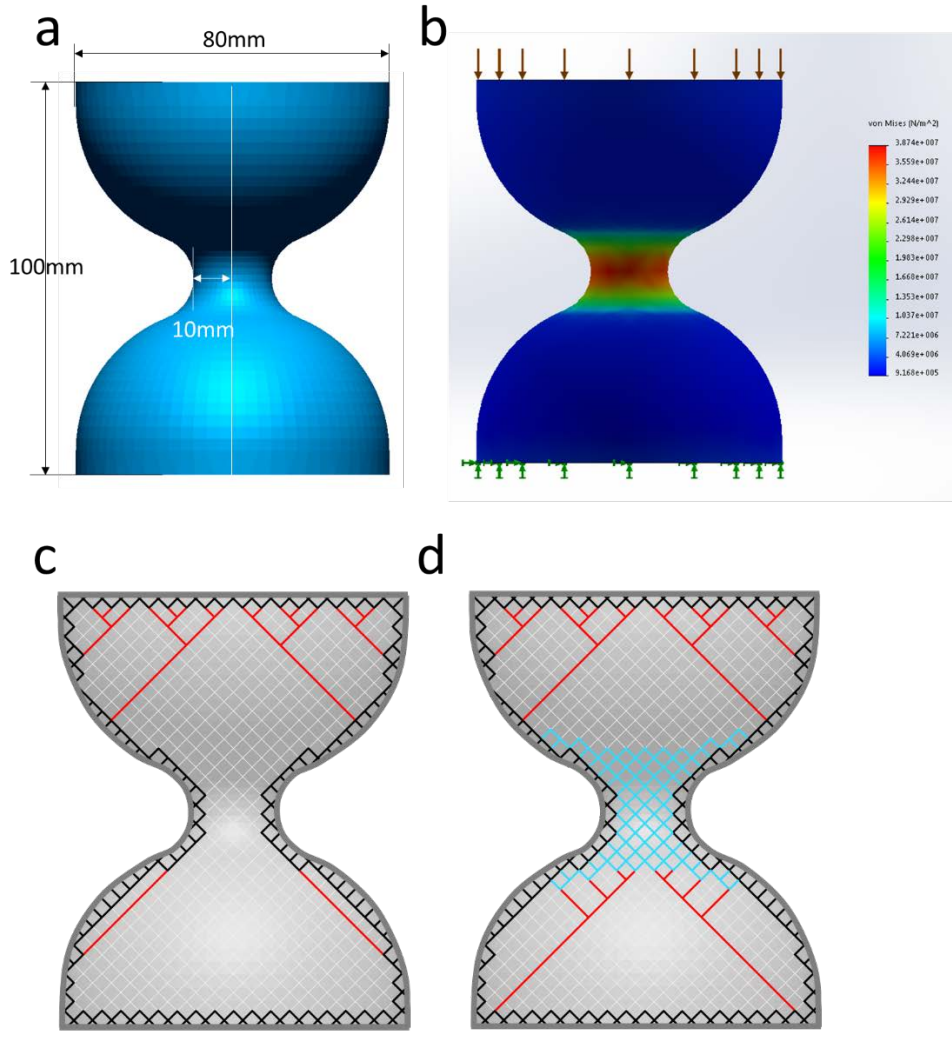


Fig. 5.12 Sandglass model: (a) geometry of the model, (b) stress analysis, (c) M2, and (d) M3.

As in the previous experiments, the mass of the sandglass generated using the 2D grid pattern method and M3 was equal; when the densities of the M3 method and the 2D grid pattern method were 20% and 12.3%, respectively, the mass was 81.7 grams.

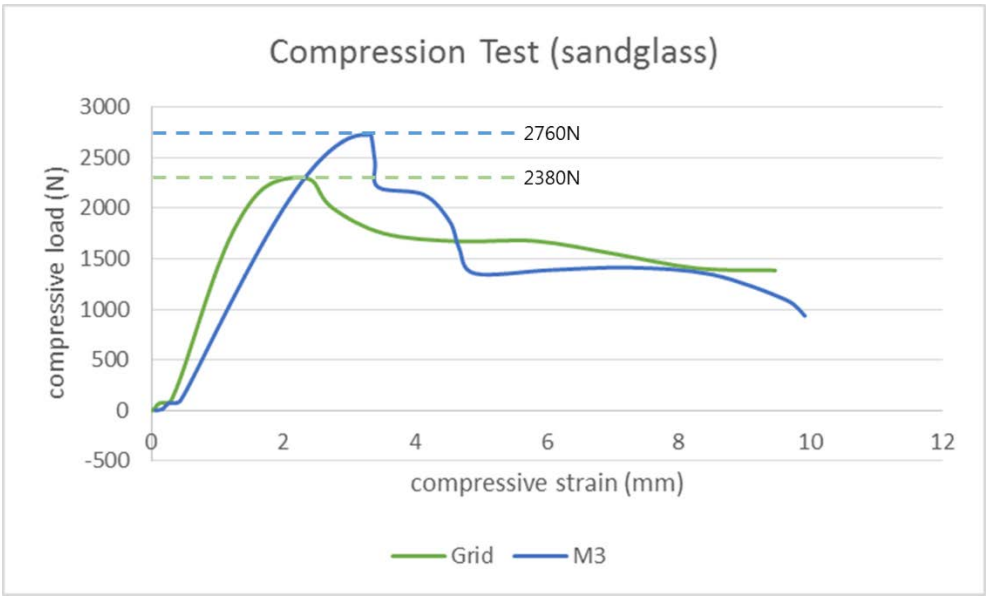


Fig. 5.13 Compression test results of the sandglass manufactured using the grid pattern and M3

As shown in Fig. 5.13, it can be seen that the yield stress of the sandglass shape generated by the M3 method is 2760 N, which is stronger than the 2D grid pattern method with a yield stress of 2380 N. This proves that the proposed method can optionally reinforce weak regions.

5.3 Tensile strength testing

It is not possible to conclude that the proposed structure has isotropy with respect to the load only from the compression test results, and so tensile strength testing was carried out after producing specimens with the dimensions shown in Fig. 5.14. 10 specimens (five of each) were made to consider infill type, orientation, and stacking direction, the details of which are shown in Fig. 5.15. The Instron 5584 [57] was used to carry out the tensile strength testing using the parameters in Table 5-14.

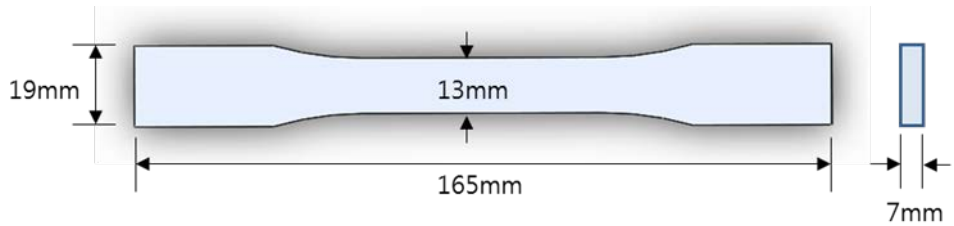


Fig. 5.14 Specimen dimensions

Fig. 5.16 shows the maximum tensile strength of each specimen from the results of the tensile strength testing. Specifically, it can be seen that the stacking angle and the area of the cross-sectional area affect the maximum tensile load rather than the infill structure. Therefore, the proposed method cannot be said to be isotropic under a tensile load, only under compression.

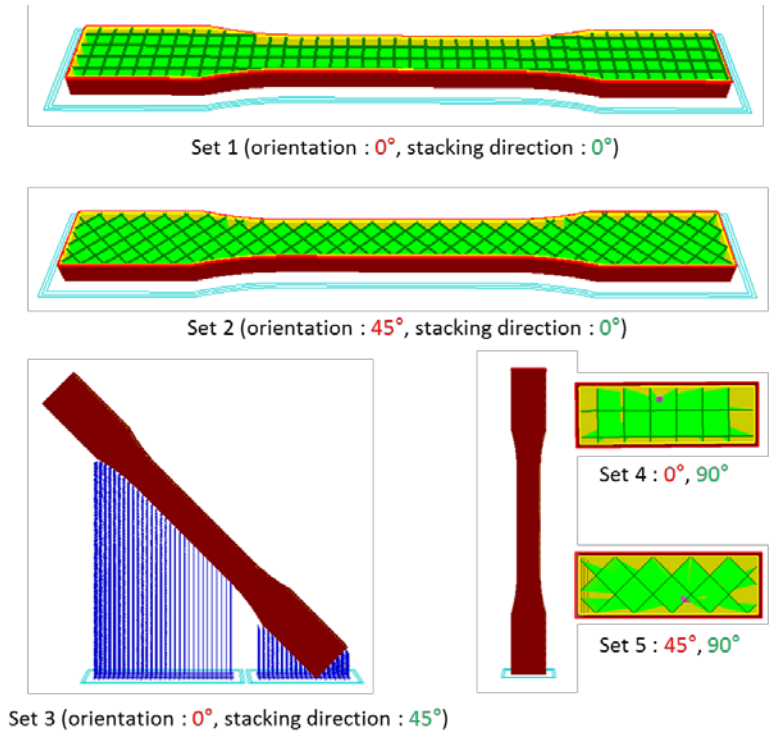


Fig. 5.15 Manufacturing conditions for the specimens

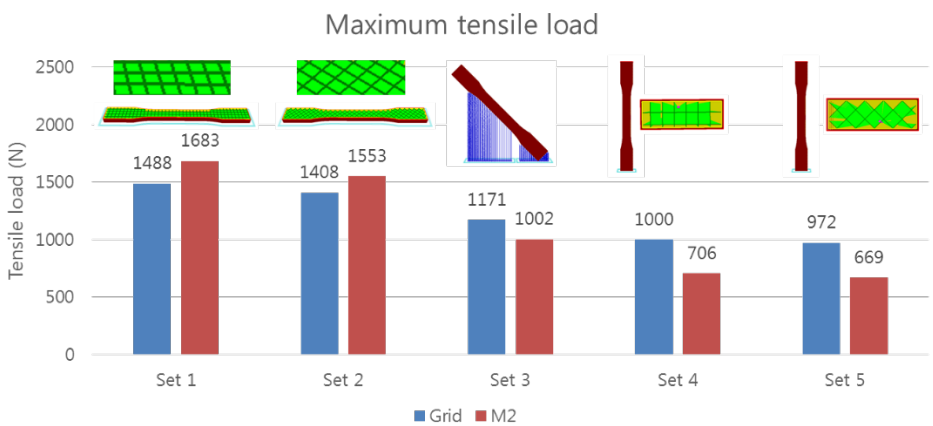


Fig. 5.16 Results of tensile strength testing

Table 5-14 Parameter settings for tensile strength testing.

Parameter	Value	Parameter	Value	Parameter	Value
Basic parameters		Solid Infill Bottom	on	Lowering Bed during Retraction (mm)	0
Layer Height (mm)	0.2	Travel Speed (mm/s)	180	Retraction Combing	All
Wall Thickness (mm)	0.8	First Later Speed (mm/s)	10	**Raft parameters**	
Enable Retraction	on	**Cooling parameters**		Raft margin (mm)	2
Bottom/Top Thickness (mm)	0.8	Minimal layer time (sec)	5	Line Spacing (mm)	3
Fill Density (%)	20	Enable Cooling Fad	on	Base Thickness (mm)	0.3
Print Speed (mm/s)	40	Fan Height at Maximum Speed	0.6	Base Line Width (mm)	1
Nozzle Temperature ©	200	Minimum Fan Speed (%)	50	Interface Thickness (mm)	0.27
Bed Temperature	60	Maximum Fan Speed (%)	50	Interface Line Width (mm)	0.4
Filament Material	PLA	Minimum Speed (mm/s)	10	Airgap (mm)	0
Filament Diameter (mm)	1.75	Minimum wait period for Lowering Bed	off	First Later Airgap (mm)	0.25
Flow Compensation (%)	100	**Retraction parameters**		Surface Layers Amount	2
Infill and speed parameters		Retraction Speed (mm/s)	30	Surface Layer Thickness (mm)	0.2
Nozzle Size (mm)	0.4	Retraction Length (mm)	6	Surface Line Width (mm)	0.4
Infill Overlap (%)	15	Minimal Travel (mm)	1.5	Enable Raft Outline Inset	off
Solid Infill Top	on	Minimal Extrusion Before Retraction (mm)	0.02		

CHAPTER 6.

Conclusions

In this chapter, conclusions on the research are drawn by summarizing the discussion of the results. The aim of the study was to create an inner support structure with reduced manufacturing time and material usage while taking into consideration the features of FDM-type printer operation and the shape of objects in terms of their isotropic characteristics for the load.

Since the control of the nozzle in an FDM-type printer is by a motor, the manufacturing time is proportional to the material usage. Furthermore, the manufacturing time is also affected by the complexity of the nozzle path. Therefore, in order to shorten the manufacturing time, which is a critical disadvantage of the FDM-type, the inner support structure using the most material was decreased by reducing the amount of material used, and the path of the inner support structure was generated so as to have a low complexity (simple) path.

The main roles of the inner support structure are largely the printability of the roof area and strength reinforcement of the object. A commonly used inner support structure is uniformly generated regardless of the shape using a 2D pattern method, and in this case, the printability and strength of an object depend on the infill density. In this study, different internal support structures were created in accordance with the main purpose of the inner support structure. They can be classified as the M1 method where objects can be made quickly and stably by only considering the possibility of lamination, the M2 method of reinforcing the overall strength of the

shape while retaining its isotropic characteristics with respect to the load, and the M3 method of reinforcing the strength of a specific region. The roles of support of the roof region and strength reinforcement of the object can be separated and used in conjunction with the purpose of the object.

It was necessary to simplify the problem of creating an inner support structure that satisfies the above-mentioned conditions for various shapes. To do this, infill was redefined using the concept of blocks, similar to the concept of voxels which are also used to represent 3D shapes. Using the concept of blocks, the complex problem of 3D shape generation was transformed into a 1D problem of ordering the supporting planes, which is an NP-hard problem in which the current state affects the next state and is the minimum solution until all of the cases have been examined. To solve this problem efficiently, a splitting method was proposed which takes into consideration only the minimum value of the current state (splitting algorithm B) and an implicit method which considers the minimum value of the current state and the next state (splitting algorithm C). Splitting algorithm B can be quickly calculated by a simple principle, but there are exceptions, and splitting algorithm C can be calculated without exceptions for various shapes but has a large computational cost which is nonetheless reasonable and is calculated within a few seconds.

To create a unit block, a reference plane is required. Since a unit block of a cube is generated using three dividing planes, the algorithm was extended by adding a reference plane to create an inner supporting structure using different unit blocks generated by the additional reference plane. In this study, the extension possibilities

of the algorithm were proved by creating an inner support structure using 3 and 4 reference planes. In addition, the characteristics of each internal support structure were analyzed experimentally.

The inner support structures generated by the proposed methods were compared with the conventional 2D pattern method by each performance criterion and shape. It can be seen that the proposed methods (M1 and M2) have lower production times and material usage than the 2D pattern method for the same object and infill density. This efficiency was experimentally confirmed as the larger the scale of the object and the higher the infill density, the better. Through compression testing of objects created using the M2 and 2D pattern methods, it can be seen that the 2D pattern method's object is strong only in the stacking direction, but the M2 method's object is isotropic in all directions. In addition, using the M3 method, it was shown that the strength of a specific region can be reinforced for an object of the same mass.

References

[1] Stava, O., Vanek, J., Benes, B., Carr, N., & Měch, R. (2012). Stress relief: improving structural strength of 3D printable objects. *ACM Transactions on Graphics (TOG)*, 31(4), 48.

[2] Telea, A., & Jalba, A. (2011, July). Voxel-based assessment of printability of 3D shapes. *In International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing* (pp. 393-404). Springer Berlin Heidelberg.

[3] Zhou, Q., Panetta, J., & Zorin, D. (2013). Worst-case structural analysis. *ACM Transactions on Graphics*, 32(4).

[4] Shapeways (2016) Tutorial. http://www.shapeways.com/tutorials/

[5] Lu, L., Sharf, A., Zhao, H., Wei, Y., Fan, Q., Chen, X., & Chen, B. (2014). Build-to-last: strength to weight 3D printed objects. *ACM Transactions on Graphics (TOG)*, 33(4), 97.

[6] Kindinger, J. (2001). Lightweight structural cores. *ASM Handbook of Metals* Vol. 21-Composites.

[7] Chen, Y. (2007). 3d texture mapping for rapid manufacturing. *Computer-Aided Design and Applications*, 4(6), 761-771.

[8] Wang, W., Wang, T. Y., Yang, Z., Liu, L., Tong, X., Tong, W., ... & Liu, X. (2013). Cost-effective printing of 3D objects with skin-frame structures. *ACM Transactions on Graphics (TOG)*, 32(6), 177.

[9] Zhang, X., Xia, Y., Wang, J., Yang, Z., Tu, C., & Wang, W. (2015). Medial axis tree-An internal supporting structure for 3D printing. *Computer Aided Geometric Design*, 35, 149-162.

[10] Wu, J., Dick, C., & Westermann, R. (2016). A system for high-resolution topology optimization. *IEEE transactions on visualization and computer graphics*, 22(3), 1195-1208.

[11] Wu, J., Aage, N., Westermann, R., & Sigmund, O. (2016). Infill Optimization for Additive Manufacturing-Approaching Bone-like Porous Structures. *arXiv preprint* arXiv:1608.04366.

[12] Li, D., Dai, N., Jiang, X., & Chen, X. (2016). Interior structural optimization based on the density-variable shape modeling of 3D printed objects. *The International Journal of Advanced Manufacturing Technology*, 83(9-12), 1627-1635.

[13] Wu, J., Wang, C. C., Zhang, X., & Westermann, R. (2016). Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design*, 80, 32-42.

[14] Wu, J., Kramer, L., & Westermann, R. (2016). Shape interior modeling and mass property optimization using ray-reps. *Computers & Graphics*, 58, 66-72.

[15] Huang, P., Wang, C. C., & Chen, Y. (2014). Algorithms for layered manufacturing in image space. *ASME Press*.

[16] Schmidt, R., & Umetani, N. (2014, July). Branching support structures for 3D printing. *In ACM SIGGRAPH 2014 Studio* (p. 9). ACM.

[17] Vanek, J., Galicia, J. A., & Benes, B. (2014, August). Clever support: Efficient support structure generation for digital fabrication. *In Computer Graphics Forum*(Vol. 33, No. 5, pp. 117-125).

[18] Dumas, J., Hergel, J., & Lefebvre, S. (2014). Bridging the gap: automated steady scaffoldings for 3D printing. *ACM Transactions on Graphics (TOG)*,33(4), 98.

[19] Qiu, J., Wu, L., & Mao, Y. (2015, August). A novel supporting structure generation scheme to 3D printing. *In Proceedings of the 7th International Conference on Internet Multimedia Computing and Service* (p. 69). ACM.

[20] Huang, X., Ye, C., Wu, S., Guo, K., & Mo, J. (2009). Sloping wall structure support generation for fused deposition modeling. *The International Journal of Advanced Manufacturing Technology*, 42(11-12), 1074-1081.

[21] Strano, G., Hao, L., Everson, R. M., & Evans, K. E. (2013). A new approach to the design and optimisation of support structures in additive manufacturing. *The International Journal of Advanced Manufacturing Technology*, 66(9-12), 1247-1254.

[22] Hu, K., Zhang, X., & Wang, C. C. (2015, August). Direct computation of minimal rotation for support slimming. *In 2015 IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 936-941). IEEE.

[23] Luo, L., Baran, I., Rusinkiewicz, S., & Matusik, W. (2012). Chopper: partitioning models into 3D-printable parts.

[24] Yao, M., Chen, Z., Luo, L., Wang, R., & Wang, H. (2015). Level-set-based partitioning and packing optimization of a printable model. *ACM Transactions on Graphics (TOG)*, 34(6), 214.

[25] Vanek, J., Galicia, J. A., Benes, B., Měch, R., Carr, N., Stava, O., & Miller, G. S. (2014, September). PackMerger: A 3D print volume optimizer. *In Computer Graphics Forum* (Vol. 33, No. 6, pp. 322-332).

[26] Hu, R., Li, H., Zhang, H., & Cohen-Or, D. (2014). Approximate pyramidal shape decomposition. *ACM Trans. Graph.*, 33(6), 213-1.

[27] Wei, X. R., Zhang, Y. H., & Geng, G. H. (2016). No-infill 3D Printing. *3D Research*, 7(3), 24.

[28] Ezair, B., Massarwi, F., & Elber, G. (2015). Orientation analysis of 3D objects toward minimal support volume in 3D-printing. *Computers & Graphics*, 51, 117-124.

[29] Hu, K., Jin, S., & Wang, C. C. (2015). Support slimming for single material based additive manufacturing. *Computer-Aided Design*, 65, 1-10.

[30] Umetani, N., & Schmidt, R. (2013). Cross-sectional structural analysis for 3D printing optimization. *SIGGRAPH Asia*, 5, 1-4.

[31] Zhang, X., Le, X., Panotopoulou, A., Whiting, E., & Wang, C. C. (2015). Perceptual models of preference in 3D printing direction. *ACM Transactions on Graphics (TOG)*, 34(6), 215.

[32] Christiansen, A. N., Schmidt, R., & Bærentzen, J. A. (2015). Automatic balancing of 3D models. *Computer-Aided Design*, 58, 236-241.

[33] Prévost, R., Whiting, E., Lefebvre, S., & Sorkine-Hornung, O. (2013). Make it stand: balancing shapes for 3D fabrication. *ACM Transactions on Graphics (TOG)*, 32(4), 81.

[34] Bächer, M., Whiting, E., Bickel, B., & Sorkine-Hornung, O. (2014). Spin-it: optimizing moment of inertia for spinnable objects. *ACM Transactions on Graphics (TOG)*, 33(4), 96.

[35] Wang, L., & Whiting, E. (2016, May). Buoyancy Optimization for Computational Fabrication. *In Computer Graphics Forum* (Vol. 35, No. 2, pp. 49-58).

[36] Schumacher, C., Bickel, B., Rys, J., Marschner, S., Daraio, C., & Gross, M. (2015). Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics (TOG)*, 34(4), 136.

[37] Kantaros, A., Chatzidai, N., & Karalekas, D. (2016). 3D printing-assisted design of scaffold structures. *The International Journal of Advanced Manufacturing Technology*, 82(1-4), 559-571.

[38] Bickel, B., Bächer, M., Otaduy, M. A., Lee, H. R., Pfister, H., Gross, M., & Matusik, W. (2010, July). Design and fabrication of materials with desired deformation behavior. *In ACM Transactions on Graphics (TOG)* (Vol. 29, No. 4, p. 63). ACM.

[39] Brennan-Craddock, J., Brackett, D., Wildman, R., & Hague, R. (2012). The design of impact absorbing structures for additive manufacture. *In Journal of Physics: Conference Series* (Vol. 382, No. 1, p. 012042). IOP Publishing.

[40] Zhou, Y., Sueda, S., Matusik, W., & Shamir, A. (2014). Boxelization: folding 3D objects into boxes. *ACM Transactions on Graphics (TOG)*, 33(4), 71.

[41] Mueller, S., Mohr, T., Guenther, K., Frohnhofen, J., & Baudisch, P. (2014, April). faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. *In Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (pp. 3827-3834). ACM.

[42] Xin, S., Lai, C. F., Fu, C. W., Wong, T. T., He, Y., & Cohen-Or, D. (2011, August). Making burr puzzles from 3D models. *In ACM Transactions on Graphics (TOG)* (Vol. 30, No. 4, p. 97). ACM.

[43] Song, P., Fu, Z., Liu, L., & Fu, C. W. (2015). Printing 3D objects with interlocking parts. *Computer Aided Geometric Design*, 35, 137-148.

[44] Calì, J., Calian, D. A., Amati, C., Kleinberger, R., Steed, A., Kautz, J., & Weyrich, T. (2012). 3D-printing of non-assembly, articulated models. *ACM Transactions on Graphics (TOG)*, 31(6), 130.

[45] Bächer, M., Bickel, B., James, D. L., & Pfister, H. (2012). Fabricating articulated characters from skinned meshes. *ACM Trans. Graph.*, 31(4), 47-1.

[46] Ou, J., Dublon, G., Cheng, C. Y., Heibeck, F., Willis, K., & Ishii, H. (2016, May). Cilllia: 3D Printed Micro-Pillar Structures for Surface Texture, Actuation and Sensing. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 5753-5764). ACM.

[47] Laput, G., Chen, X. A., & Harrison, C. (2015, November). 3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles. *In*

Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (pp. 593-597). ACM.

[48] Lee, C. S., Kim, S. G., Kim, H. J., & Ahn, S. H. (2007). Measurement of anisotropic compressive strength of rapid prototyping parts. *Journal of materials processing technology*, 187, 627-630.

[49] The Stanford 3DScanning Repository (2014) 3D mesh model database.

[50] Cormen, T. H. (2009). Introduction to algorithms. MIT press.

[51] CuraEngine, (2015): Open source.https://github.com/Ultimaker/CuraEngine

[52] Ultimaker2, (2014): Desktop 3d printer. https://ultimaker.com

[53] DP200, (2014): Desktop 3d printer. http://www.oasindoh.co.kr

[54] Pinshape, (2016): Free 3d printables and designs.https://pinshape.com/

[55] Instron 5582, (2012): Universal tester: www.instron.com

[56] Principles of 3D printers: http://kocoafab.cc/tutorial/view/487

[57] Instron 5584, (2012): Universal tester: www.instron.com

[58] Materialise, (2016): Commercial 3D print software: www.materialise.com/

ABSTRACT (KOREAN)

FDM 방식 3D 프린터를 위한 내부 지지 구조의 최적 생성

이 주 성

서울대학교 대학원

기계항공공학부

3D 프린터 중에서, FDM 방식은 저렴한 비용과 간단한 작동 방식으로 널리 사용된다. 하지만 제작 시간이 길다는 단점이 있다. 형상의 특징을 고려하지 않고 균일하게 생성되는 내부 지지 구조가 긴 제작 시간의 주요한 원인이다. 따라서 본 논문에서는 FDM 방식 3D 프린터의 제작 시간 및 재료 사용량 절감을 위한 새로운 내부 지지 구조 생성 알고리즘을 제안한다. 제안하는 내부 지지 구조는 효율성, 적층 가능성, 독립 적층 가능성, 단순 경로, 등방성의 특징을 지니고 있다.

제안하는 방법은 기존의 2차원 패턴 방식을 확장하여 3차원의 블록으로 분할하는 방법이다. 물체를 단위 블록으로 분할 한 뒤에 내부 지지 구조의 목적에 따라서 단위 블록들을 병합 한다. 병합 된 블록의 독립 출력 가능성과 단순 경로를 확보하기 위해서 평면으로 분할한다.

유닛 블록의 병합 방법에 따라 효율성을 최대로 높이는 M1 방법, 전체 강도를 보강하는 M2 방법, 선택적으로 강도를 보강하는 M3 방법으로 나눌 수 있고, 각 방법의 성능은 실험적으로 확인하였다. 또한 단위 블록의 차원을 확장하여 제안하는 알고리즘의 확장 가능성과 특징을 살펴보았다. 결과적으로 제안하는 방법을 이용하여 생성된 내부 지지 구조는 제작 시간 및 재료 사용량 측면에서 상당히 효율적이며, 항상 안정적으로 출력 가능하고, 압축 하중에 대해서 등방성 특징을 지니며, 선택적 강도 보강을 할 수 있는 특징을 가진다.

주요어: 3D 프린터, 내부 지지 구조, 효율성, 적층 가능성, 강도 보강, 등방성, Fused deposition modeling(FDM)

학 번 : 2010-20708