Fashion and Textiles

RESEARCH

Open Access

Support structure tomography using per-pixel signed shadow casting in human manikin 3D printing

Jin Young Jung[1†], Seonkoo Chee[2†] and In Hwan Sul[1*]

†Jin Young Jung and Seonkoo Chee equally contributed to this work

*Correspondence: snowman0@kumoh.ac.kr

[1] G463, Dept. of Materials Design Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea
[2] D607, Dept. of IT Convergence, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea

Abstract

This study proposes an advanced algorithm for predicting the optimal orientation in human manikin 3D printing. We can print the manikin mesh data on any scale depending on the user's needs. Once the 3D printing scale was determined, the manikin data were dissected based on the 3D printer's maximal printing volume using our previous work. Then, we applied the newly proposed algorithm, designated as "per-pixel signed-shadow casting," to each dissected manikin part to calculate the volumes of the object and the support structure. Our method classified the original mesh triangles into three groups—alpha, beta, and top-covering—to eliminate the need for special hardware such as graphic cards. The result is shown as a two-dimensional bitmap file, designated as "tomograph". This tomograph represents the local support structure distribution information on a visual and quantitative basis. Repeating this tomography method for the three rotational axes resulted in a four-dimensional (4D) box-shaped graph. The optimal orientation of any arbitrary object is easily determined from the lowest-valued pixel in the 4D box graph. We applied this proposed method to several basic primitive shapes with different degrees of symmetry and complex shapes, such as the famous "Stanford Bunny". Finally, the algorithm was applied to human manikins using several printing scales. The theoretical values were compared with those obtained from analytical or g-code-based experimental volumes.

Keywords: Human manikin, 3D printing, Optimal orientation, Signed shadow casting, Support structure, Tomograph

Introduction

Three-dimensional (3D) printing or additive manufacturing (ASTM, 2013) is one of the most popular keywords of this era. In the apparel industry, 3D printing is widely used in the field of conventional garment display manikins, as well as for high-value applications, such as car crash test dummies (Rueda-Arreguín et al., 2021), patient-specific surgery models (Badash et al., 2016), and thermal manikins (EMPA, 2018). For example, thermal manikins are expensive and typically cost approximately a couple of hundred thousand US dollars. However, the costs reflect that it is necessary to evaluate garment thermal insulation properties. Our long-term research goal is to provide non-professionals with a simple and economical way of generating human manikins, especially using FDM-type 3D printers.

A large object such as a human manikin requires two types of pre-processing. First, it is dissected to fit the 3D printer's maximal printing volume. Second, each dissected part is aligned to its optimal orientation to reduce the total support structure volume. Our team has addressed the dissection in previous research (Jung et al., 2021) and focuses on the alignment in this paper. Calculating the support structure volume is analogous to measuring an object's shadow under sunlight. We define this as shadow casting throughout the paper. Prior studies on predicting the amount of support structure used a similar approach (Das et al., 2017; Ezair et al., 2015; Wang et al., 2017) and yielded satisfactory results. However, they required special hardware such as a graphics card unit (GPU). We provide an improved algorithm, designated as "signed-shadow casting," which does not require a GPU.

The central methodology categorized the original object's mesh into three groups for the plus/minus sign of the Z coordinate of the triangular normal vector. Then, each triangular group was converted to pixels to calculate the volume information. Previous studies only provided the total amount of support structure. By contrast, our method provides quantitative, local, and visual information by which the user can easily identify where the support structure arises. The proposed algorithm was written in commercial software, Wolfram Mathematica®, for easy conversion to any programming language or platform. It was evaluated not only for simple geometric shapes, such as "Sphere," "Cone," and "Pyramid," which represent the typical symmetry classes, but also for complex shapes, such as "Stanford Bunny," and human manikin data.

Problem statement and literature review

Minimizing support structure in 3D printing

An overhang structure with an inclination angle above the critical angle requires a support structure for the extruded filaments to be solidified. The support structure is eliminated after mechanical cutting. However, the amount of support structure should be minimized because it increases the printing time and leaves a dirty surface. The melting and cooling processes of 3D printing filaments are a mixture of complex rheological phenomena. There are several factors related to the minimization of the support structure (Brooks, 2021) including:

- Factor (i) Increasing fan cooling parts
- Factor (ii) Reducing layer height
- Factor (iii) Incorporating a chamfer into the model
- Factor (iv) Decreasing printing speed
- Factor (v) Controlling the printing temperature
- Factor (vi) Decreasing layer width
- Factor (vii) Splitting the model into multiple parts
- Factor (viii) Altering the orientation.

However, changing the option values on factors (i)–(vi) is not recommended for non-professional users because the printing can continue erroneously. Factors (vii) and (viii) are feasible options that can be considered for manikin 3D printing.

Numerous studies have been conducted over the last two decades to split the model into multiple parts (factor vii). Among them, (Luo et al., 2012) and (Chen et al., 2015) proposed an objective-function-based algorithm considering the number of parts, connector feasibility, finite-element-method (FEM)-based structural soundness, and even aesthetics. Their study is versatile and can be applied to any geometry. However, hundreds of thousands of iterations are required to build a binary search tree, and they do not have an objective function for support structure minimization. Meanwhile, Wang et al. (2017) decomposed any geometry into so-called "genus-zero" shapes such as cones, pyramids, ellipses, and spheres and succeeded in dropping the filament usage to 66.56% for the famous "Dragon" mesh. However, this method does not include factors related to the optimal orientation.

It would be optimal to generate g-codes for each possible direction to find an object's optimal orientation in 3D printing (factor viii). Commercial slicing software already provides analysis functionality. However, the service is limited only to the direction of the global axes because repetitive g-code generation requires considerable time.

Therefore, alternative methods have been developed to replace the actual g-code generation for optimal orientation prediction. (Ezair et al., 2015) calculated the support structure volume (V_{ss}) simply from the object's top cover volume (V_{tc}) and actual volume (V_o) as in Eq. (1) and Fig. 1.

$$V_{ss}(\mathrm{y,p,r}) = V_{tc}(\mathrm{y,p,r}) - V_o(\mathrm{y,p,r}) \tag{1}$$

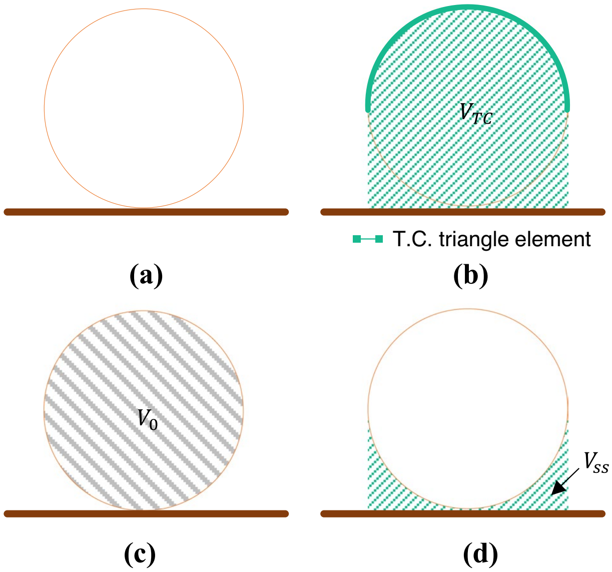


Fig. 1 *Shadow-casting* based support structure calculation scheme (Ezair et al., 2015). **a** Target object, **b** Calculating top-cover volume using GPU, **c** calculating object volume, **d** final support structure volume (=V_{tc}−V_0)

In this equation, V_{ss} is the desired support structure volume when the initial orientation is (y,p,r). V_{tc} is the volume between the top-covering (TC) triangles and the bottom plate. V_o is the volume of the object and can be learned algebraically from the triangular mesh vertices.

V_{tc} creates a shadowed-like volume as shown in Fig. 2b, which corresponds to the support structure in 3D printing. This is the crucial factor in Eq. (1) and difficult to calculate because two-dimensional triangle-to-triangle collision tests are needed for every triangle pair of the whole mesh. (Ezair et al., 2015) adopted the GPU's depth test. The depth test is a procedure to find the nearest elements to the camera for shadow rendering or hidden surface removal in 3D rendering (Hughes et al., 2014). The elements with highest depth values correspond to the TC triangles of our problem and the GPU can find them in a brief time using parallel computation. Other groups (Das et al., 2017; Wang et al., 2010) voxelized the input mesh and calculated both V_{tc} and V_o. However, these methods require a high-priced GPU and a considerable amount of memory as the size of the voxel decreases.

Meanwhile, Morgan et al. (2016) projected each triangle and calculated the triangular pillar volume to obtain V_{tc} and V_o values. This method does not require GPU and memory because it is not voxel-based. However, it is relatively slow and does not yield local position information of the support structure.

Shape symmetry

An arbitrary object's orientation is described by three rotation angles, generally known as aircraft principal axes, including yaw, pitch, and roll. It is computationally expensive to consider all the cases. Fortunately, some objects have axis symmetry, as shown in Table 1 (Chaouch & Verroust-Blondet, 2009). We selected representative shapes for each symmetry code and analyzed the effects of shape symmetry on the optimal orientation search.

Shape symmetry is an essential topic in many areas, such as in the fields of art and engineering. There have been many studies on shape symmetry and its automatic detection. Interested readers can refer to symmetry-descriptor-based shape matching (Kazhdan et al., 2004), point data sampling-based methods (Mitra et al., 2006), multi-scale-based symmetry detection (Xu et al., 2012), and user threshold-based approximate symmetry plane finding (Korman et al., 2015). However, the automatic detection of shape symmetry is beyond the scope of this study. We assumed that shape symmetry in the human body or manikin was manually input because the user could easily identify it.

Modeling

Shadow-analogy based support structure tomography

The support structure's volume in 3D printing is similar to an object's shadow volume by the imaginary light source from the infinite +Z direction (Fig. 1d). We designate this idea as "shadow-analogy". However, the support structure and shadow are generated by the TC triangles (Fig. 1b). The support structure can have different local volumes depending on the overhang's relative position and height from the bottom plate. Our idea represents the local

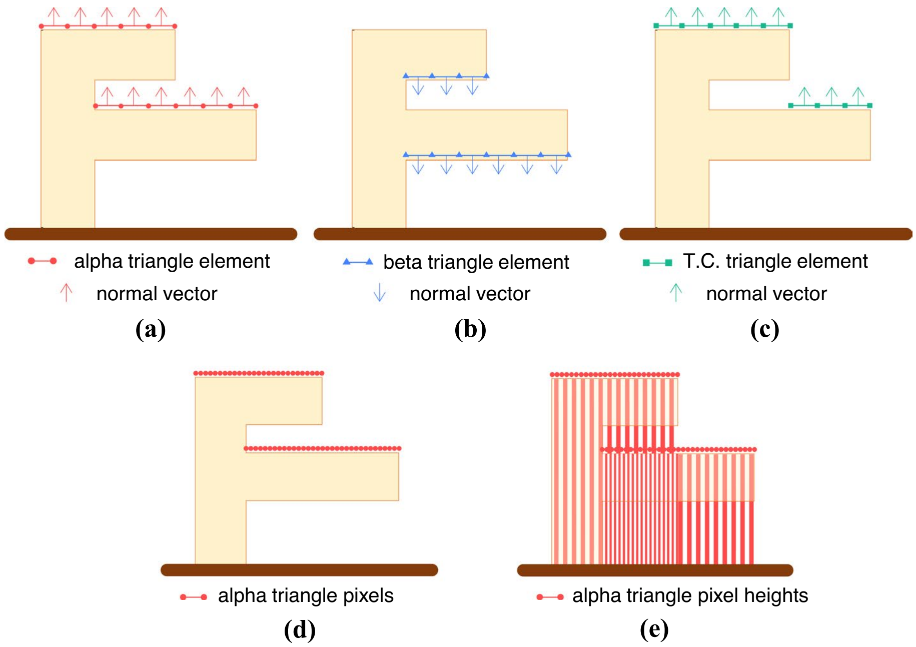


Fig. 2 Signed-shadow casting algorithm overview for a simple "F" character shape (lateral cross-sectional view). **a** alpha triangle group, **b** beta triangle group, **c** TC triangle group triangles, **d** pixelization of alpha triangles, **e** integration of angle triangle pixels' height values

support structure information as a two-dimensional (2D) bitmap file, which we named the "*support structure tomograph*". This bitmap was acquired using the following procedures.

The first procedure involves the classification of mesh triangles. We aim to calculate V_{tc} without the assistance of additional hardware, such as a GPU. Therefore, we classified the input mesh triangles into three groups, alpha, beta, and top-cover (TC). If a triangle's normal vector faces the$+Z$ direction, it belongs to the alpha group (Fig. 2a). The remaining ones whose normal vector faces the $-Z$ direction belong to the beta group (Fig. 2b). The beta triangles originate from the overhang structure. Some of them generate a support structure, while others are canceled by alpha triangles. Finally, the uppermost positioning

Table 1 Classes of shape symmetry. Modified from (Dubrovin et al., 1984) and (Chaouch & Verroust-Blondet, 2009)

Code	Notes	Examples
G_D	Multiple mirror planes pass through a fixed axis with one mirror plane perpendicular to the axis (dihedral symmetry)	Hourglasses, prisms, cylinders, spheres
G_C	Multiple mirror planes pass through a fixed axis (cyclic symmetry)	Cones, tables
G_R	Five platonic solids (rotational symmetry)	tetrahedron, cube, octahedron, dodecahedron, and icosahedron
G_U	Only one plane reflection exists (unique symmetry). Usually found in human-man objects	Airplanes, animals, A- or T-posed manikin full body with 1:10 scale or more
G_Z	No plane reflection symmetry exists (zero symmetry)	Plants, trees, asymmetrically posed manikins with 1:10 scale or more

triangles among the alpha triangles belong to the TC triangle group (Fig. 2c), which gives essential information for V_{tc}.

The second procedure involves calculating the volume between each triangular group and the bottom plate. The difference between our method and the previous method, that is, Morgan et al. (2016)'s work is that we decompose each triangle into pixels (Fig. 2d). Then, the distance between the pixel and the bottom plate is integrated over the triangle surface, as shown in Fig. 2e. The total volume from the alpha triangle group (V_α) is expressed like Eq. (2),

$$V_\alpha(\mathrm{y,p,r}) = \sum_i^I V_{\alpha,i} \tag{2}$$

$V_{\alpha,i}$ is the integrated sum of the heights of the *i*th triangle. The total volume from the beta (V_β) and TC groups (V_{tc}) is expressed in the same way as using Eqs. (3) and (4).

$$V_\beta(\mathrm{y,p,r}) = \sum_j^J V_{\beta,j} \tag{3}$$

$$V'_{tc}(\mathrm{y,p,r}) = \sum_k^K V_{tc,k} \tag{4}$$

I, *J*, and *K* denote the number of alpha, beta, and TC triangles, respectively.

The object's volume (V_o) has the following relationship.

$$V_o(\mathrm{y,p,r}) = V_\alpha(\mathrm{y,p,r}) - V_\beta(\mathrm{y,p,r}) = \sum_i^I V_{\alpha,i} - \sum_j^{\mathrm{J}} V_{\beta,j} \tag{5}$$

Equation (1)'s V_{tc} has the same value as that in Eq. (4)'s V'_{tc}. By substituting all of the equations, the final Eq. (6) is generated.

$$\begin{aligned} V_{ss}(\mathrm{y,p,r}) &= V_{tc}(\mathrm{y,p,r}) - V_o(\mathrm{y,p,r}) \\ &= V'_{tc}(\mathrm{y,p,r}) - [V_\alpha(\mathrm{y,p,r}) - V_\beta(\mathrm{y,p,r})] \\ &= \sum_k^K V_{tc,k} - (\sum_i^I V_{\alpha,i} - \sum_j^J V_{\beta,j}) \end{aligned} \tag{6}$$

Finding optimal orientations

Section Shadow-analogy based support structure tomography shows the support structure information for a specific orientation as a bitmap image. To obtain the optimal orientation, the tomography operation is repeated for each possible orientation. The results are displayed in a four-dimensional box-shaped graph.

Representation of object orientation

Here, we explain the notation of "V_{ss} (y,p,r)" in the final 4D box graph. We represent the object orientation (y, p, r) as a combination of the rotational matrices—$R_X(y)$, $R_Y(p)$, and

$R_Z(r)$—on the input mesh, M_0, as in Eq. (7). For example, $R_X(y)$ means a counter-clockwise rotational transform about the global X-axis by angle y. Quaternion is more accurate way of representing rotations, but we chose the conventional matrix form because it is easier to implement for nonprofessional 3D printer users. Figure 5 shows several examples of this notation. The initial input data are assumed to be always laid on the XY plane facing the + Z-axis, and this orientation is written as "(0°,0°,0°)" (Fig. 5a). "V_{ss} (y, p, r)" means the support structure volume for the give (y,p,r) orientation afterward.

$$M_1 = R_Z(r).R_Y(p).R_X(y).M_0 \tag{7}$$

Experimental

Input mesh data

Among the test object mesh data, "Stanford Bunny" was downloaded from the Stanford 3D scanning repository (Levoy et al. 2005). The simple geometries of "Sphere," "Cone," "Pyramid, "Chair," and "PixelMan," were drawn using AutoDesk TinkerCAD ® software. The manikin data ("Masha," 6690 vertices, 13,672 triangles) were obtained in OBJ format and its height was assumed to be 171.3 cm.

Segmentation of manikin parts

Since the target manikin mesh ("*Masha*") is much larger than the volume of the conventional 3D printer, it should be segmented in advance. Automatic segmentation of 3D printing objects has been extensively researched (Luo et al., 2012). Any type of segmentation method can be used. However, in the apparel industry, it is common to use human feature point information. Therefore, we used our previous algorithm (Jung et al., 2021). This method automatically detects the human feature point information from the input mesh. It then cuts the body part in three steps to fit the printing volume. The user can also provide cutting constraints via feature points.

Slicing software for 3D printing

The theoretical values obtained from our algorithm should be compared with the experimental values for verification. Instead of actual 3D printing, we used the filament volume shown in the g-code generation in the slicing software to reduce the possible error from the manual removal of the support structures. We used Sindoh's 3DWOX software as a slicing software because it provides both slicing and partial optimal orientation information. Sindoh's DP-202 printer (Republic of Korea, maximum printing volume of 200 × 200 × 189 mm^3) was assumed to be the output printer. Some of the g-code options were changed in filling density (100%), support structure (everywhere, 100% density, critical angle 1°), bed filling (none), and internal filling (linear type). The other options were unchanged.

Computation

The source codes were written using Wolfram® Mathematica software (V12.1) on a Microsoft Windows 10 PC (3.2GHz 6-core Intel i7-8700 CPU). Files with .STL or .OBJ format were read using the *Import* function and then converted to triangular vertices and elements using the *TransformRegion* and *MeshRegion* functions. Most of the subroutines were written with the *Module* and *ParallepMap* functions. The data were stored in a structure-of-array style vector format (Sim et al. 2012) to maximize the CPU's parallel computation.

Results and Discussion

Our algorithm consists of two steps. First, a support structure tomograph was acquired for a given input mesh data and its initial orientation. Second, the sum of the tomograph pixel values was plotted in the 4D box graph to determine the optimal orientation. The algorithm was applied to several shapes, and the results are discussed here. In section "Per-pixel visual representation of support structure distribution", the tomography-based method was validated using both simple exact shapes and complex shapes. In section "Searching optimal orientation in 4D box graph", some factors on the 4D box graph, such as shape symmetry, initial orientation, and angle intervals, are analyzed.

Per-pixel visual representation of support structure distribution

The support structure tomography method is similar to taking a medical X-ray picture, which gives a single gray-scale picture and shows a distribution of internal organs such as bone or flesh. Instead, our final tomograph requires three pictures as alpha, beta, and TC tomographs in the calculation. Compositing these three pictures using Eq. (6) gives the final support structure tomograph. The sum of the pixel values of the support structure tomograph is equal to the amount of support structure volume in the given orientation. To verify this idea, simple shapes, whose volumes can be known geometrically, were used. In addition, we used the following assumptions to simplify the calculations.

- Assumption 1: Material property of 3D printing polymeric filaments was not considered.
- Assumption 2: The internal structure, surface, and support structure use the same filament.
- Assumption 3: Internal volume of a given object is perfectly filled.
- Assumption 4: Every overhang has a support structure regardless of its angle, which is, the critical angle is always 1°.

Simple primitive solids

To check that our volume integration scheme shown in Eqs. (2)–(6) is valid, "Sphere" mesh with symmetry code G_d was used. From the input mesh, three triangular groups of alpha, beta, and TC were classified and rendered in different colors. Then, each group's triangles were converted to pixels, and the height values were plotted in a 2D graph. Finally, these three pictures were composited based on Eq. (6), and we obtained the final support structure.

Our support structure tomograph provided support structure information both visually and quantitatively. However, we found errors in volume calculation. The "Sphere" meshes with equal mesh configuration and varying radii were evaluated to find the error source. The theoretically predicted volumes were compared with the true volumes, which can be calculated geometrically. Table 2 reveals that average edge length affects the final support structure volume. Our pixelization step converts real-valued mesh data into integer-valued pixels, such as the famous Bresenham algorithm (Bresenham, 1965). The boundary edges of each triangle seemed to lose some pixels during this process. The actual amount of V_{tc} and V_o error may be different depending on the input shape's geometry. However, we found the common tendency that the error of V_{ss} decreased with increasing mean edge length. The error can be minimized by increasing the dimension of the input object, which can increase the number of pixels and the total calculation time.

The above "Sphere" example was not sensitive to the initial orientation or support structure because of its simple geometry and high symmetry. Another example is "Cone," which has less symmetry. First, the "Cone" was converted to tomography. There was no support in the final tomograph. Tomograph-based volume measurements were performed again with various dimensions, as shown in Table 3. The errors were as high as 25.0% for a small object (R/H = 10/20) and then dropped to 1–2% in both V_{tc} and V_o. Since this "*Cone*" had no overhangs and support structure, the V_{ss} had zero values.

The "Cone" was inverted vertically, and its tomograph was calculated again. The quantitative analysis results are presented in Table 4. The errors in V_{tc} and V_o decreased with increasing dimensions. The error of V_{ss} showed some fluctuation, yet its maximum value was only 3.3%. This result indicates that our algorithm (Eqs. (2)–(6)) successfully operates on simple geometry.

Arbitrary shapes

Further, we verified our method for a more general shape with few symmetries. The famous "*Stanford Bunny*" data were used, and their final tomographs were acquired, as

Table 2 Volume error estimation for "Sphere"

Input mesh				**Volume (*True*†/predicted [error])**			**Calculation time (seconds)**
nE/nV	**Radius (mm)**	**Mean element area (mm^2)**	**Mean edge length (mm)**	**V_{tc}**	**V_o**	**V_{ss}**	
320 /162	50	31,415.9	8.418375	*6.54e+5/* *5.24e+5* [2.9%]	*5.24e+5/* *5.24e+5* [0.1%]	*1.31e+5/* *1.22e+5* [7.3%]	0.417
	100	125,663.7	29.93818	*5.24e+6/* *5.10e+6* [2.8%]	*4.19e+6/* *4.13e+6* [1.5%]	*1.05e+6/* *9.67e+5* [8.2%]	1.666
	150	282,743.3	44.90727	*1.77e+7/* *1.72e+7* [2.5%]	*1.41e+7/* *1.38e+7* [2.4%]	*3.53e+6/* *3.44e+6* [2.8%]	5.513
	200	502,654.8	59.876358	*4.19e+7/* *4.08e+7* [2.8%]	*3.25e+7/* *3.26e+7* [2.8%]	*8.64e+7/* *8.04e+7* [2.6%]	8.853
	300	1,130,973.4	89.81454	*1.41e+8/* *1.38e+8* [2.5%]	*1.13e+8/* *1.10e+8* [2.7%]	*2.83e+7/* *2.78e+7* [1.6%]	13.178

"nE/nV" means # of triangle elements/vertices. "True †" volume were calculated analytically; $V_o = (4/3)\pi R^3$, $V_{tc} = V_o/2 + \pi R^2 \times R = (5/3)\pi R^3$, $V_{ss} = V_{tc} - V_o = (1/3)\pi R^3$

Table 3 Volume error estimation for "Cone"

Input mesh information			Volume (*True*†/Theoretical, in mm^{3,} [error in %])			Calculation time (seconds)
nE/nV	R/H (mm)	Mean edge length (mm)	V_{tc}	V_o	V_{ss}	
46/25	10/20	899.2	*2.09e+3/ 2.15e+3* [2.8%]	*2.09e+3/ 2.79e+3* [25.0%]	*0/ 0* [0.0%]	0.093
	30/60	2697.7	*5.65e+4/ 5.58e+4* [1.3%]	*5.65e+4/ 6.06e+4* [6.8%]	*0/ 0* [0.0%]	0.160
	50/100	4496.2	*2.62e+5/ 2.56e+5* [2.2%]	*2.62e+5/ 2.69e+5* [2.7%]	*0/ 0* [0.0%]	0.335
	70/140	6294.6	*7.18e+5/ 7.12e+5* [0.9%]	*7.18e+5/ 7.39e+5* [2.7%]	*0/ 0* [0.0%]	0.639
	90/180	8093.1	*1.53e+6/ 1.51e+6* [0.9%]	*1.53e+6/ 1.56e+6* [2.0%]	*0/ 0* [0.0%]	0.983

R/H means Radius/Height, $V_{tc} = 0$, $V_o = (1/3)\pi R^2 H$, $V_{ss} = 0$

Table 4 Volume error estimation for "Inverted Cone"

Input mesh information			Volume (*True*†/Theoretical, in mm^{3,} [error in %])			Calculation time (seconds)
nE/nV	R/H (mm)	Mean edge length (mm)	V_{tc}	V_o	V_{ss}	
46/25	10/20	899.2	*6.28e+3/ 6.84e+3* [8.9%]	*2.09e+3/ 2.63e+3* [25.8%]	*4.19e+3/ 4.21e+3* [0.4%]	0.076
	30/60	2697.7	*1.70e+5/ 1.68e+5* [0.8%]	*5.65e+4/ 5.76e+4* [2.0%]	*1.13e+5/ 1.11e+5* [2.3%]	0.189
	50/100	4496.2	*7.85e+5/ 7.71e+5* [1.9%]	*2.62e+5/ 2.64e+5* [0.7%]	*5.24e+5/ 5.07e+5* [3.3%]	0.366
	70/140	6294.6	*2.16e+6/ 2.14e+6* [0.6%]	*7.18e+5/ 7.27e+5* [1.3%]	*1.44e+6/ 1.41e+6* [1.6%]	0.726
	90/180	8093.1	*4.58e+6/ 4.55e+6* [0.7%]	*1.53e+6/ 1.53e+6* [0.1%]	*3.05e+6/ 3.02e+6* [1.1%]	1.087

$V_{tc} = 0$, $V_o = (1/3)\pi R^2 H$, $V_{ss} = 0$

shown in Fig. 3. Similar to the F-like character example in Fig. 2, the initial input mesh triangles (Fig. 3a) were classified (Fig. 3b). The three classified groups were converted into pixels and integrated (Fig. 3c). The final support structure tomography reveals that the support structure arises mainly around two ears of the "*Stanford Bunny*" (Fig. 3d).

Table 5 shows the quantitative evaluation of the algorithm for the "Stanford Bunny." In this case, we cannot calculate the true V_{ss} value via simple geometric calculation like the previous "Cone." To obtain an experimental value of V_{ss}, it is easy to measure the 3D-printed object. However, the problem is that removing the support structure is not

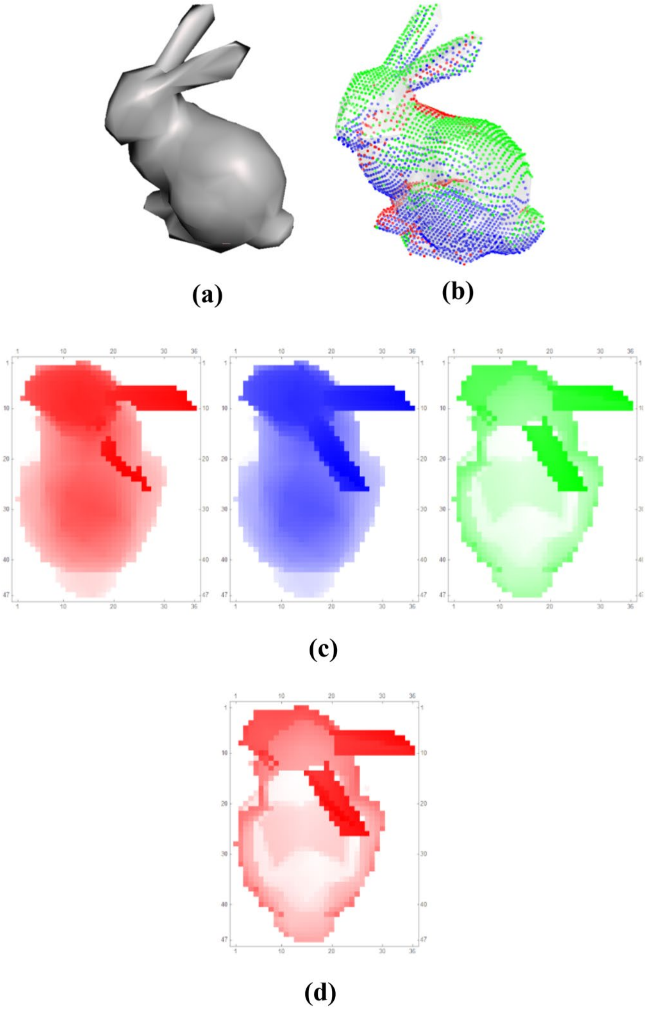


Fig. 3 Support structure tomography result for a sphere (symmetry code G_Z). **a** Original object ("Stanford bunny"), **b** alpha/beta/TC triangle groups, **c** alpha/beta/TC triangle tomography results, and **d** final support structure tomography picture

Table 5 Volume error estimation for "*Stanford Bunny*" with initial orientation (0°,0°,0°)

Input mesh		Measured volume (from slicing SW)			Predicted volume ([error])			Calculation time (seconds)
nE/nV	Mean edge length (mm)	V_0+V_{ss}	V_o	V_{ss}	V_{tc}	V_o	V_{ss}	
510 /257	5.4	2.94e+4	2.04e+4	8.95e+3	2.84e+4 [n/a]	2.74e+4 [25.4%]	1.09e+3 [723.8%]	0.247
	10.8	2.28e + 5	1.62e+5	6.65e+4	2.28e+5 [n/a]	1.80e+5 [10.1%]	4.82e+4 [38.1%]	0.750
	16.3	7.66e+5	5.45e+5	2.21e+5	7.62e+5 [n/a]	5.78e+5 [5.7%]	1.84e+5 [20.3%]	0.841
	21.7	1.81e+6	1.29e+6	5.15e+5	1.81e+6 [n/a]	1.33e+6 [8.5%]	4.75e+5 [8.5%]	1.517

"†Measured volume" were calculated from slicing SW's total filament lengths manual and can be another source of error. Therefore, we compared our theoretical value with filament lengths from the slicing software's g-code generation.

Searching optimal orientation in 4D box graph

Effect of shape symmetry to 4D box graph

An object's support structure for a given orientation is described with four variables: (y,p,r) and the support structure's volume (V_{ss}). Therefore, the final graph for all possible orientations has a four-dimensional box shape. We varied the angles y, p, and r from 0° to 360° with a certain angle interval (default 30°). V_{ss} for each (y, p, r) orientation was measured, and its value was plotted. Then, trilinear interpolation was applied to make the final continuous 4D box-shaped graph. To find the optimal orientation, the user simply has to find the lowest V_{ss} point (rendered in blue). Figure 4 shows the results for several shapes with various symmetry codes. As the symmetry becomes stronger (G_Z to G_D), the box graph shows more regular patterns, especially in the roll direction. For example, the "Sphere" (symmetry code G_D) has the same V_{ss} value in all directions, and thus shows no peak points at all. "Cone" (code G_c) and "Pyramid" (code G_R) have less symmetry but still show similar cross-sections to the roll direction. This means that objects of this symmetry type do not have to search for the roll direction (rotation about the Z-axis). This phenomenon can effectively reduce total computation. Other real-life objects, such as "Chair" (code G_U), "PixelMan", and "Stanford Bunny" (code G_Z) have little or even no symmetry. In this case, we must search the entire (y,p,r) space. In each 4D box graph, a red sphere was manually inserted to emphasize the optimal orientation, but its location was occasionally adjusted for better visibility, such as (0°, 270°, 0°) to (360°, 270°, 370°). Each 4D box required tomograph calculation time multiplied by $(2\times\pi/\text{ angle interval})^3$. For example, the "Cone" 4D box took 41.92 s when the angle intervals for yaw, pitch, and roll axes were all 30°.

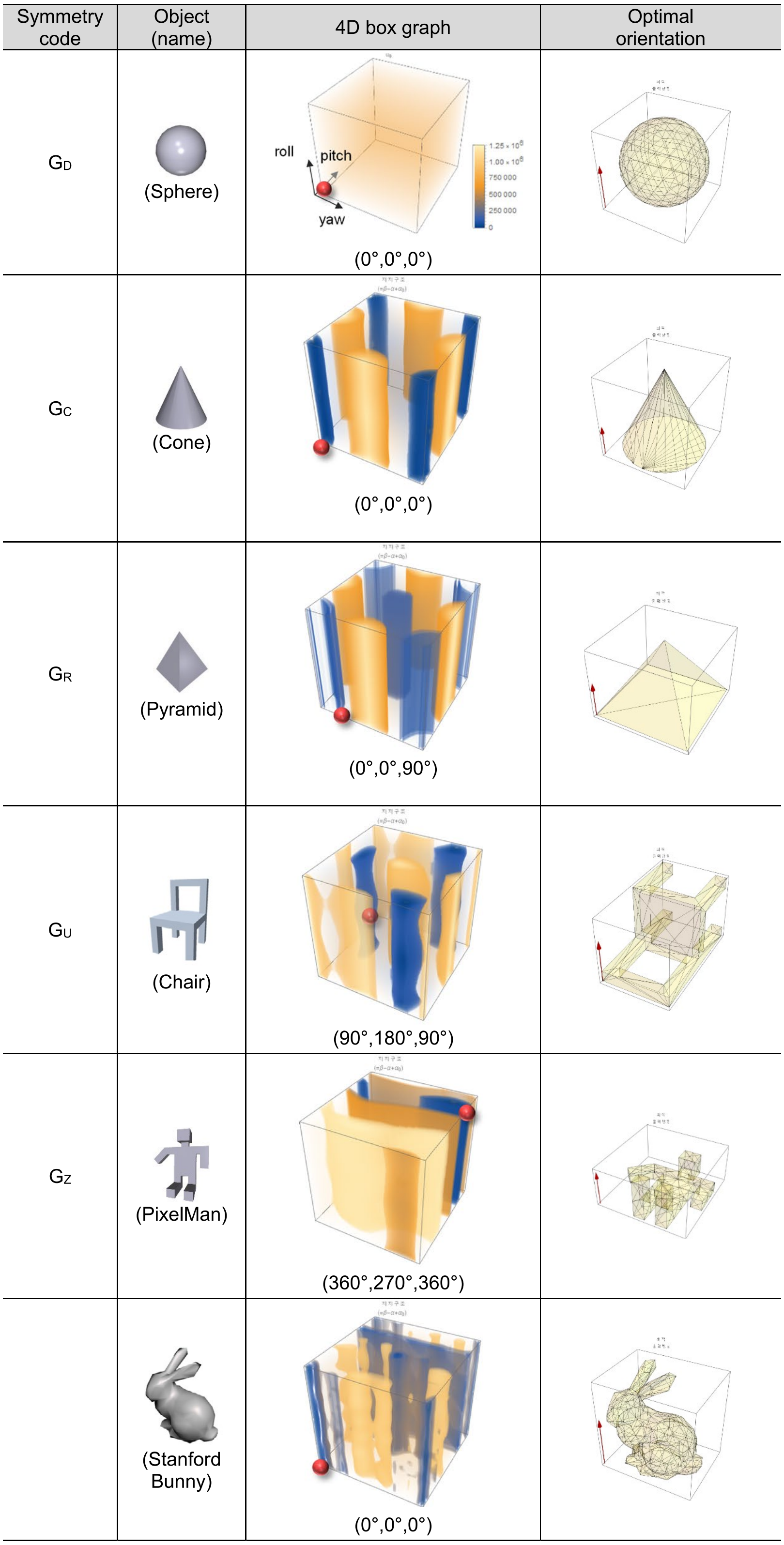
Effect of initial orientation

If our algorithm is reliable, the same input mesh with different initial orientations should always generate the same result. To verify this, the "Stanford Bunny" data were reused, as shown in Fig. 5. The six 4D graphs show a slightly different distribution, but the final optimal orientation is the same. The volume information for these six cases is shown in Fig. 6. The dashed line indicates the theoretical V_{ss} value, which was calculated from V_{tc} and V_o (dashed boxes). The solid line represents the experimental V_{ss} value from the g-code generation. Because there is no way to calculate V_{tc} in slicing software, the support structure generation option was turned off and the resultant volume was recorded as Vo. Then, the option was turned on and its value was recorded as V_o+V_{ss} in Fig. 6. Although the theoretical and experimental values do not exactly match, they have the same trends. This indicates that our algorithm is also quantitatively reliable.

Application to human manikin 3D printing

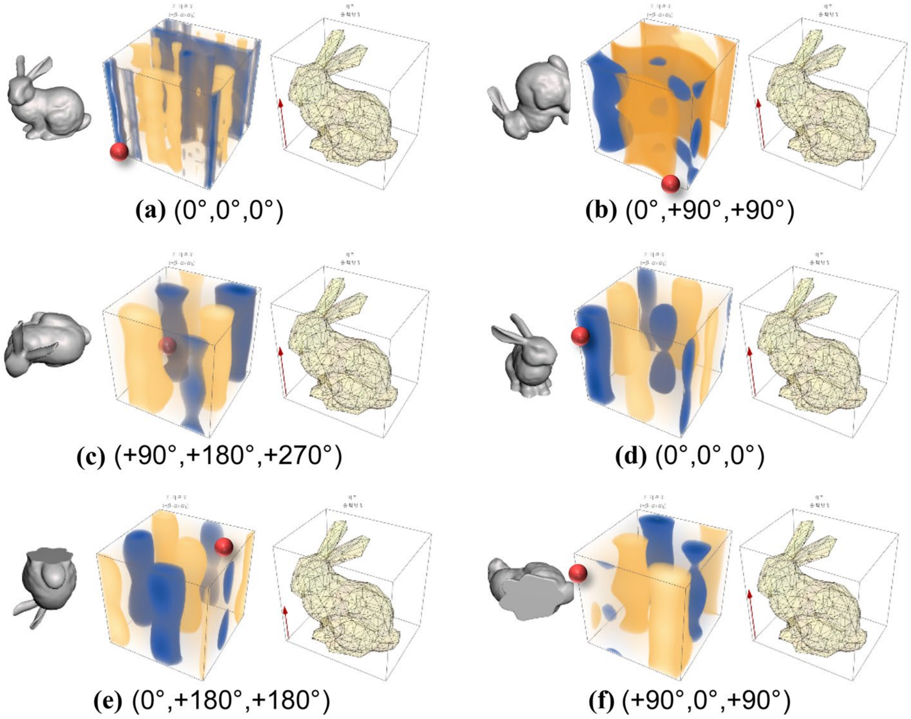
The proposed algorithm was applied to the final shape of a human manikin. As shown in Fig. 7, there are several types of shape symmetries in the human body parts.

The full-size body with a standard A-pose on a 1:10 scale belongs to the symmetry code G_U (Fig. 8a). It shows the same cross-section to the roll axis as the "Chair"



Symmetry code	Object (name)	4D box graph	Optimal orientation
G_D	(Sphere)	(0°,0°,0°)	
G_C	(Cone)	(0°,0°,0°)	
G_R	(Pyramid)	(0°,0°,90°)	
G_U	(Chair)	(90°,180°,90°)	
G_Z	(PixelMan)	(360°,270°,360°)	
	(Stanford Bunny)	(0°,0°,0°)	

Fig. 4 4D box graph representation of support structure amount for each shape symmetry (blue color means the lowest amount of support structure volume. Angle interval = 90°)



(a) (0°,0°,0°)

(b) (0°,+90°,+90°)

(c) (+90°,+180°,+270°)

(d) (0°,0°,0°)

(e) (0°,+180°,+180°)

(f) (+90°,0°,+90°)

Fig. 5 Effect of initial orientation to 4D box graph (note that left-hand-side figures are the input meshes, the middle ones are 4D box graphs, and the right-hand-side ones are objects aligned to the final optimal orientations)

mesh in Fig. 4. The 4D graph analysis indicates that the optimal orientation occurs when the manikin lies on its back or its stomach. If we print the manikin without an orientation change, the V_{ss} value increases more than five times. This matches our previous experimental result (Kim & Sul, 2018), where the upright 1:10 manikin and the lying-on-the-back 1:10 manikin needed 13.5 g and 10.7 g of the support structure, respectively. Notably, the errors between the experimental and theoretical values may be owing to our assumptions. If the manikin is extremely misaligned such as 360°, 45°, 270°) to the diagonal corner direction, Vss increases approximately nine times. The computation time for Fig. 8a is 35.56 s without any further optimization. The speed can be accelerated if other efficient programming languages, such as C/C++, are used.

If we use a scale lower than 1:10, the manikin should be segmented in advance. The segmented parts, except for the terminal parts, such as the head, hands, and feet, generally have cylindrical shapes and symmetry codes G_U or G_C. For example, the 1:5 waist part had the optimal orientation in the (0°, 0°, 90°) direction. The 1:5 upper arm part had a slanted initial orientation, and its optimal orientation was (0°, 315°, 135°).

We can easily identify the optimal orientation of the symmetric body parts without calculation. However, terminal parts such as the 1:1 ft (Fig. 8d) cannot be easily predicted. These shapes belong to the symmetry code G_Z, and their 4D graphs have an irregular shape. Thus, users cannot predict the optimal direction. The shell structure is an even more complex case. Some manikins, such as thermal manikins, need to contain heat generation devices and sensors inside the body. In this case, we must

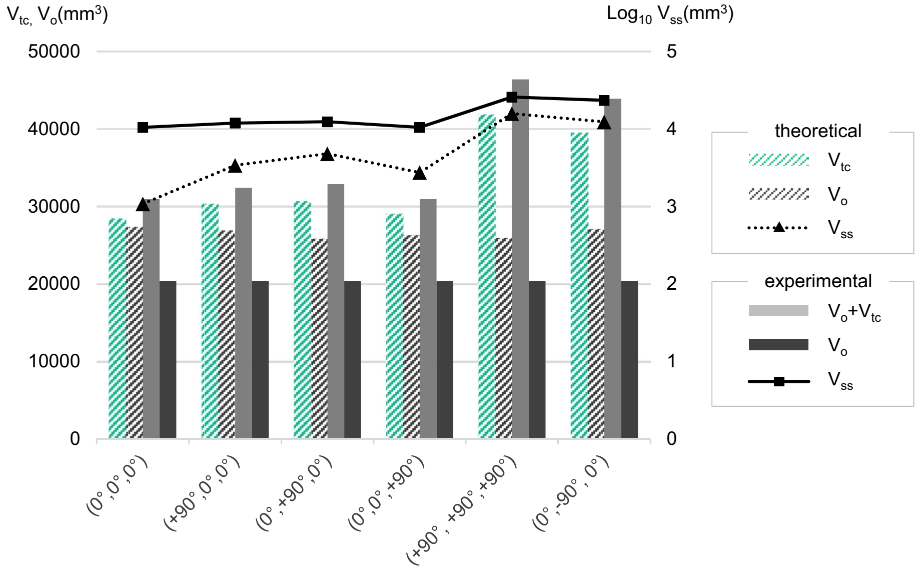


Fig. 6 Comparison of experimental and theoretical amount of support structure for *"Stanford Bunny"* for the six initial orientation

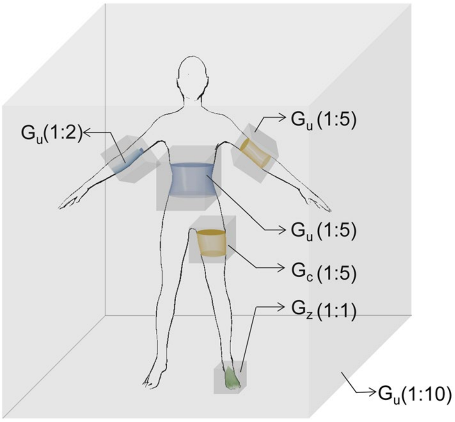


Fig. 7 Various symmetry found in human body parts for 1:1, 1:2, 1:5, and 1:10 printing scales (each cube represents the maximal printing volume of the 3D printer in its printing scale)

print only the shells of the manikin. Figure 8e shows the shell of the 1:2 elbow part. Our algorithm still generated a reliable result for this type of complex shape.

Conclusions

This study proposed an advanced algorithm to predict the optimal object orientation in 3D printing. Without any special hardware, such as a GPU, this algorithm calculates the object and structure volumes effectively. The input data had .STL or .OBJ file format with triangular meshes. Then, the triangles were classified into three groups, namely alpha, beta, and top-covering, depending on their normal vector direction. Each triangular group was pixelized, and the sum of pixels in the vertical direction

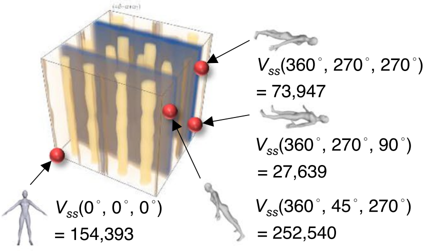
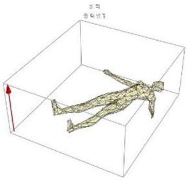
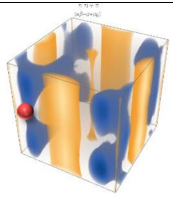
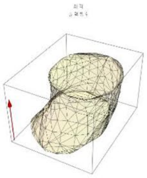
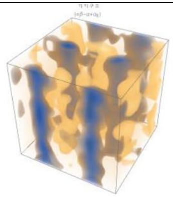
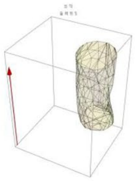
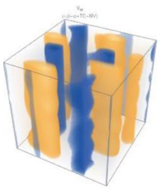
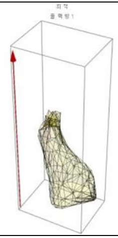
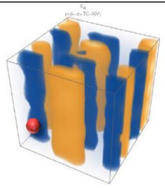
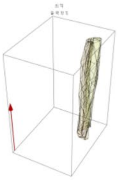
Scale and part name (Symmetry code)	4D box graph	Optimal orientation
 (a) 1:10 full body (G_U)		
 (b) 1:5 waist (G_U)	 (0°,0°,180°)	
 (c) 1:5 upper arm (G_U)	 (0°, 315°,135°)	
 (d) 1:1 foot (G_Z)	 (90°,0°,360°)	
 (e) 1:2 arm lower half shell (G_Z)	 (45°,180°,180°)	

Fig. 8 4D box graph representation of body parts in a several printing scales for the "Masha" manikin data (angle interval = 45°)

was shown in a 2D bitmap file, "tomograph". Compositing the alpha, beta, and TC tomographs led to the final support structure tomography, in which the pixel color meant local support structure volume. The sum of all the pixel values of the support structure tomograph was equivalent to the total support structure volume for a given orientation. Repeating this procedure for the three rotational axes generates the final 4D box graph to find the final optimal orientation.

Our method was verified using simple geometry such as "Sphere", "Cone", "Pyramid" , and "Chair" shapes both visually and quantitatively. Complex shapes such as the "Stanford Bunny" mesh were used. The initial orientation of the input mesh did not affect the final result. The angle interval in the yaw, pitch, and roll axes did not significantly affect the final 4D graph. There were some errors in the volume measurement due to the pixelization error. Therefore, resizing the input mesh to a larger dimension is recommended if the calculation time does not matter. The other source of error was our two assumptions. We assumed that the interval volume was 100% filled and that the overhang's critical angle was always 1° for ease of calculation. Further work is required to correct these absolute volume errors. However, the relative tendency for the support structure volume matched the experimental results. The algorithm was finally applied to the human manikin mesh, and we found several different shape symmetries. Most of the human body parts are cylindrical, and their optimal orientation lies in the longitudinal direction. The remaining terminal components or shell structures must construct a 4D graph to determine the optimal orientation. Therefore, our method is expected to be applied to the 3D printing of human manikin data and even more complex industrial objects such as heat exchangers.

Acknowledgements
Not applicable.

Authors' contributions
JYJ conducted experiments and analyzed the results with SC and IHS. SC collected literature and developed theoretical models. IHS developed programming source codes. JYJ and SC equally contributed to this work. All authors read and approved the final manuscript.

Author's information
JYJ, Graduate Student, Dept. of Materials Design Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea.
SC, Associate Professor, Dept. of IT Convergence, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea.
IHS, Associate Professor, Dept. of Materials Design Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea.

Funding
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1F1A1040883, No.2022R1A2C1010072).

Availability of data and materials
The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 14 November 2021 Accepted: 28 January 2022
Published online: 05 July 2022

References

ASTM, I. (2013). *ASTM52915-13, Standard specification for additive manufacturing file format (AMF) version 1.1* (Vol. 52915, p. 2013). ASTM International.

Badash, I., Burtt, K., Solorzano, C. A., & Carey, J. N. (2016). Innovations in surgery simulation: a review of past, current and future techniques. *Annals of Translational Medicine, 4*(23), 453. https://doi.org/10.21037/atm.2016.12.24

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal, 4*(1), 25–30. https://doi.org/10.1147/sj.41.0025

Brooks, M. (2021, October, 10). *3D Printing Overhang: Can You 3D Print Overhangs*? m3dzone. https://m3dzone.com/3d-printingoverhang

Chaouch, M., & Verroust-Blondet, A. (2009). Alignment of 3D models. *Graphical Models, 71*(2), 63–76. https://doi.org/10.1016/j.gmod.2008.12.006

Chen, X., Zhang, H., Lin, J., Hu, R., Lu, L., Huang, Q.-X., Benes, B., Cohen-Or, D., & Chen, B. (2015). Dapper: decompose-and-pack for 3D printing. *ACM Trans. Graph., 34*(6), 213:1–213:12. https://doi.org/10.1145/2816795.2818087

Das, P., Mhapsekar, K., Chowdhury, S., Samant, R., & Anand, S. (2017). Selection of build orientation for optimal support structures and minimum part errors in additive manufacturing. *Computer-Aided Design and Applications, 14*(sup1), 1–13. https://doi.org/10.1080/16864360.2017.1308074

Dubrovin, B., Novikov, S., & Fomenko, A. (1984). *Modern Geometry: methods and applications. Part I. The Geometry of Surfaces, Transformation Groups, and Fields.*

EMPA. (2018, 29–31 Aug 2018, St. Gallen, Switzerland). 12th International Manikin and Modelling Meeting (12i3m). In L. f. B. M. a. T. Empa - Federal Laboratories for Materials Science and Technology, 12th International Manikin and Modelling Meeting(12i3m), St. Gallen, Switzerland. https://www.empa.ch/web/12i3m

Ezair, B., Massarwi, F., & Elber, G. (2015). Orientation analysis of 3D objects toward minimal support volume in 3D-printing. *Computers & Graphics, 51*, 117–124. https://doi.org/10.1016/j.cag.2015.05.009

Hughes, J. F., Van Dam, A., Foley, J. D., & Feiner, S. K. (2014). *Computer graphics: principles and practice*. Pearson Education

Jung, J. Y., Chee, S., & Sul, I. H. (2021). Automatic segmentation and 3D printing of A-shaped Manikins using a bounding box and body-feature points. *Fashion and Textiles, 8*(1), 1–21. https://doi.org/10.1186/s40691-021-00255-8

Kazhdan, M., Funkhouser, T., & Rusinkiewicz, S. (2004). Symmetry descriptors and 3D shape matching. Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, https://doi.org/10.1145/1057432.1057448

Kim, S., & Sul, I. H. (2018). Minimization of support structure using cone-shaped body segmentation in fused deposition modeling type small size human manikin 3D Printing. *Textile Science and Engineering, 55*(3), 194–201. https://doi.org/10.12772/TSE.2018.55.194

Korman, S., Litman, R., Avidan, S., & Bronstein, A. (2015). Probably approximately symmetric: Fast rigid symmetry detection with global guarantees. *Computer Graphics Forum, 34*(1), 2–13. https://doi.org/10.1111/cgf.12454

Levoy, M., Gerth, J., Curless, B., & Pull, K. (2005). The Stanford 3D scanning repository. http://wwwgraphics.stanford.edu/data/3dscanrep. Accessed 10 Oct 2021.

Luo, L., Baran, I., Rusinkiewicz, S., & Matusik, W. (2012). Chopper: partitioning models into 3D-printable parts. *ACM Transactions on Graphics, 31*(6), 129–137. https://doi.org/10.1145/2366145.2366148

Mitra, N. J., Guibas, L. J., & Pauly, M. (2006). Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (ToG), 25*(3), 560–568. https://doi.org/10.1145/1179352.1141924

Morgan, H., Cherry, J., Jonnalagadda, S., Ewing, D., & Sienz, J. (2016). Part orientation optimisation for the additive layer manufacture of metal components. *The International Journal of Advanced Manufacturing Technology, 86*(5), 1679–1687. https://doi.org/10.1007/s00170-015-8151-6

Rueda-Arreguín, J. L., Ceccarelli, M., & Torres-SanMiguel, C. R. (2021). Impact Device for Biomechanics of Human Head-Neck Injuries. *Mathematical Problems in Engineering*, Article 5592673. https://doi.org/10.1155/2021/5592673

Sim, J., Dasgupta, A., Kim, H., & Vuduc, R. (2012). A performance analysis framework for identifying potential benefits in GPGPU applications. *Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*. https://doi.org/10.1145/2145816.2145819

Wang, C. C., Leung, Y.-S., & Chen, Y. (2010). Solid modeling of polyhedral objects by layered depth-normal images on the GPU. *Computer-Aided Design, 42*(6), 535–544. https://doi.org/10.1016/j.cad.2010.02.001

Wang, W., Liu, Y.-J., Wu, J., Tian, S., Wang, C. C., Liu, L., & Liu, X. (2017). Support-free hollowing. *IEEE Transactions on Visualization and Computer Graphics, 24*(10), 2787–2798. https://doi.org/10.1109/tvcg.2017.2764462

Xu, K., Zhang, H., Jiang, W., Dyer, R., Cheng, Z., Liu, L., & Chen, B. (2012). Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (ToG), 31*(6), 1–11. https://doi.org/10.1145/2366145.2366200

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.