AR based ornament design system for 3D printing

Article · December 2014

DOI: 10.1016/j.jcde.2014.11.005

Jun Mitani
University of Tsukuba

Yoshihiro Kanamori
University of Tsukuba

Yukio Fukui
Happy Science University, Chosei, Chiba, Japan

HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Design and Engineering 2 (2015) 47–54

www.elsevier.com/locate/jcde

AR based ornament design system for 3D printing

Hiroshi Aoki, Jun Mitani*, Yoshihiro Kanamori, Yukio Fukui

University of Tsukuba, 1-1-1 Tennoh-dai, Tsukuba, Ibaraki 305-0006, Japan

Received 11 June 2014; received in revised form 18 September 2014; accepted 19 September 2014
Available online 6 December 2014

Abstract

In recent years, 3D printers have become popular as a means of outputting geometries designed on CAD or 3D graphics systems. However, the complex user interfaces of standard 3D software can make it difficult for ordinary consumers to design their own objects. Furthermore, models designed on 3D graphics software often have geometrical problems that make them impossible to output on a 3D printer. We propose a novel AR (augmented reality) 3D modeling system with an air-spray like interface. We also propose a new data structure (octet voxel) for representing designed models in such a way that the model is guaranteed to be a complete solid. The target shape is based on a regular polyhedron, and the octet voxel representation is suitable for designing geometrical objects having the same symmetries as the base regular polyhedron. Finally, we conducted a user test and confirmed that users can intuitively design their own ornaments in a short time with a simple user interface.

Keywords: 3DCG; Modeling; Augmented reality; 3D printing; Voxel; Octet truss

1. Introduction

In recent years, 3D printers have attracted attention as a means of outputting computer-designed 3D shape data to the real world. The price of 3D printers is in a downward trend, and some are now even within the means of ordinary consumers. When 3D printers have become commonplace household items, what will be the sort of item they used to produce most often? Household ornaments is one possibility. People find geometric three-dimensional shapes like those shown in Fig. 1 both interesting and appealing when used in items such as key holders, straps or Christmas tree decorations, or simply as ornaments in their own right. In this study, we propose a system that supports ordinary users in the design of ornaments that include symmetrical structures.

Conventional 3D graphics software is difficult for non-specialists to use as a means of creating three-dimensional shapes. Furthermore, the high cost of 3D printer materials and the long time needed to print objects mean that the feature of previewing designed object with quick trial-and-error process is important.

Barriers to using 3D software for shape modeling include the difficulty of grasping a 3D space via a planar display, and the fact that the typical mouse and keyboard user interface is not intuitive. There have already been widespread efforts aimed at making a user interface more amenable to novice users. For example, the use of AR (augmented reality) has often been proposed. AR is a technique involving the display of computer-generated images superimposed on a real-world space captured by a video camera. By synchronizing the real and virtual coordinate systems based on the use of markers called AR markers, it is possible to make computer-generated objects and characters appear just as if they were present in the real world. By using a shape modeling interface in conjunction with stereoscopic equipment such as head-mounted displays and pointing devices, it is possible to get the same visual effect as when the object really is present. Furthermore, by allowing the user to perform shape modeling actions directly by hand or with hand-held tools instead of with a mouse and keyboard, it is possible to give the user the feeling of actually creating something.

With conventional 3D graphics software, not only is it difficult to manipulate objects, but there is also a problem in

*Corresponding author. Tel.: +81 29 853 5388; fax: +82 29 853 2333.
E-mail address: mitani@cs.tsukuba.ac.jp (J. Mitani).
Peer review under responsibility of Society of CAD/CAM Engineers.

http://dx.doi.org/10.1016/j.jcde.2014.11.005
2288-4300/

Fig. 1. Examples of ornaments with geometric patterns and symmetric features (designed with the proposedsystem).

that the 3D shapes made in this way are often inappropriate for output to a 3D printer. For example, it is not possible to print objects that have holes or self-intersections in the surface. Such defects often have to be corrected manually. To address this problem, two possible approaches can be considered. One is to provide a means to correct the problems in data after it has been created, and the other is to provide users with a system that is only capable of producing shapes that are printable on a 3D printer. For the former approach, the automatic restoration method proposed by Bischoff et al. [1] could be applied. A number of research studies on mesh reconstruction are found in the survey papers [2,3]. However, the latter approach, i.e., provide a system that is incapable of designing shapes that cannot be printed, seems to be more appropriate for users lacking in specialist knowledge of 3D modeling.

Therefore in this study we propose a 3D modeling system with the following characteristics:

- Allows ornamental shapes having symmetrical shape characteristics to be designed via an intuitive AR-based modeling interface.
- Can only produce shapes that are printable on a 3D printer.

An advantage of this system is that there is no need to correct the data so that it can be output by a 3D printer. Further, it is possible to preview a designed 3D object superimposed on the real world in AR before it is actually output by a 3D printer.

Our AR-based 3D modeling system uses an air-spray interface. Although AR is critically flawed by the lack of force feedback, an air-spray has essentially no force feedback and therefore allows us to hide this drawback of AR.

To ensure that it can only design solid models capable of being produced by a 3D printer, the proposed system uses an octet truss structure made of octahedrons and tetrahedrons to represent the internal structure that holds the shape data. We propose a shape representation method called *octet voxels*, whereby each of these octahedrons and tetrahedrons stores a value used to distinguish between the inside and outside of the three-dimensional shape.

Related studies are discussed in Section 2 of this paper, and in Section 3 we describe the properties of octet voxels and the proposed system. The results of user tests are discussed in Section 4. Finally, our conclusions are presented in Section 5.

2. Related work

The 3DM interface developed by Butterworth et al. is strongly related to our study as an attempt at improving the 3D modeling interface [4]. This system uses a head-mounted display to perform 3D modeling in a virtual reality space. This system does away with the mouse and keyboard interface that is hard for beginners to use, and instead uses a hand-held pointer device to manipulate the menus and cursor shown in a VR space.

Surface Drawing is an early study of the combination of AR and 3D computer graphics [5]. In this system, the movements of a glove worn by the user are tracked by special sensors to generate band-shaped polygonal surfaces along its locus in a virtual reality workspace, thereby allowing the user to form three-dimensional shapes intuitively. Using various tools with built-in sensors, the user can perform operations such as deforming or deleting objects. Although this system uses expensive large-scale hardware to implement AR, Cheok et al. have proposed a modeling system with approximation functions that uses a cheaper marker system [6]. Studies where AR markers are used to make simple 3D models include a study by Sano et al., who developed a system that creates a full scale box model by generating textures from photographic information [7], and a study by Schlaug where shapes can easily be formed by modeling in the same way as with clay [8].

For the input device of our system, we referred to a study by Jung et al. [9], where an air-spray interface is used to perform rough 3D computer modeling in an AR space. This involves a 3D design process whereby a skeleton shape is first created by a pointer device, and then the air spray is used to blow particles onto this shape. Although this process is limited to shapes without pronounced roughness, it can form a variety of shapes. An air compressor is connected to the air-spray held by the user, and releases puffs of air when the user performs spray operations. The sound of these puffs and the feel of the spray actuator provide the user with sensory feedback. Our proposed system uses a similar air spray interface, but without the air compressor, and uses a simple AR marker as a mouse.

To ensure that the designed shapes can be output by a 3D printer, the shapes have to be a solid. Owada et al. proposed a voxel based approach with sketch interface [10]. On the other hand, Rivers et al. proposed a CSG based tool [11]. One heavily related to our system is the Hirose et al.'s system [12]. This is a design support system tailored specifically for geometrical toys called Sphericon. In this system, the space is partitioned into small cells based on a conical shape, and three-dimensional shapes are represented as sets of these cells. In recent years, several papers related to digital fabrication with 3D printers were published. Prévost et al. proposed a method to modify shape of objects so that they stand [13].

Symmetry is essential for attractive design, and was sometimes used as a constraint of design system [14,15]. To efficiently design the shape of ornaments with symmetry, we decided to represent our 3D design space as a set of triangular pyramids, with each pyramid being represented as a set of octahedra and tetrahedra, like the structure of an octet truss.

The Spatial Sketch system developed by Willis et al. [16] supports the creation of lamp shades. The user moves a hand-held operating device freely in space, and a 3D model of a lamp shade is generated based on the locus of this device. Although this system does not use AR, it facilitates intuitive operations as an input device for the user's hand movements. This allows the user to discover interesting shapes by making new forms through trial and error, even without a preconceived idea of how the finished item should look. Similarly, our system also supports the discovery of new forms through trial and error.

3. Proposed system

This section describes our proposed system in more detail. First, we describe how the proposed octet voxels are used to represent three-dimensional shapes, and then we describe the shape design user interface that uses these voxels.

3.1. Representing shapes with octet voxels

To design symmetrical 3D shapes in the proposed system, these shapes are made based on regular polyhedra (also called Platonic solids) as shown in Fig. 2. Regular polyhedra constitute one of the simplest sets of symmetrical three-dimensional objects, and have fascinated people since ancient times. Due to the useful engineering qualities of these shapes, they are often used in buildings and are also widely used in the structure of ornaments. Regular polyhedra exhibit symmetry not only in their facets, which are all regular polygons, but also in the three-dimensional structure whereby the facets connected to each vertex are arranged uniformly.

Our system works with structures that have the same symmetry as one of the regular polyhedra. As shown in Fig. 3, the space inside a regular polyhedron is partitioned into multiple triangular pyramids, and the target shape is designed in these triangular pyramid spaces. If P is a regular polyhedron with M facets comprised of regular N-gons, then P is partitioned into M pyramids with their apex at the center of the regular polyhedron and their base in each facet. Next, these pyramids are used to make N triangular pyramids by splitting them through planes that are perpendicular to the base and intersect with the apex and the vertices in the base facet. With the above operations, it is possible to represent a regular polyhedron P as a set of N times M triangular pyramid spaces. These triangular pyramid spaces are called “basic triangular pyramid spaces”.

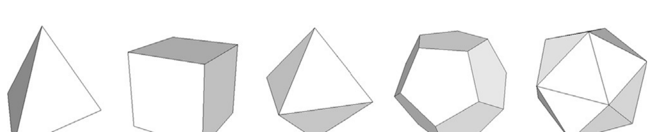

Fig. 2. The five types of regular polyhedra.

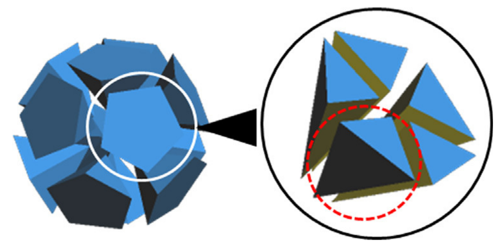

Fig. 3. Division of a regular polyhedron into basic triangular pyramid spaces (triangular pyramid surrounded by dashed line). A regular dodecahedron is divided into 60 basic triangular pyramid spaces.

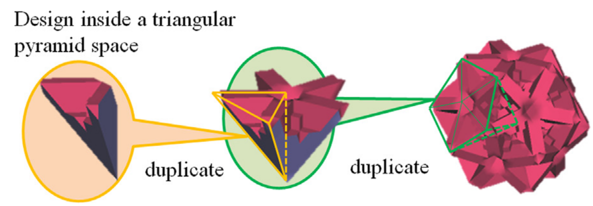

Fig. 4. Construction of 3D object by replication. By duplicating a design in triangular pyramid space, a solid with symmetry is constructed.

The user performs shape editing in a single triangular pyramid space. This shape is duplicated into other basic triangular pyramid spaces to built a new 3D object according to the procedure shown in Fig. 4. That is, the user can automatically obtain a shape having the same symmetry as the regular polyhedron on which it is based, simply by creating a design inside one triangular pyramid space.

Assuming that the design is to be formed by a 3D printer, it is preferable that when modeling the shape of the basic triangular pyramid space, a solid model with a strictly defined interior and exterior can be easily constructed. For this sort of objective, it is appropriate to use a voxel representation where the space is divided into small cells, and each cell has a value to distinguish between the inside and outside of the 3D shape. In an ordinary voxel representation, the space is partitioned into multiple cubical cells aligned with the x, y and z axes. However, this method is unsuitable for our work space, which is formed from triangular pyramids. Therefore, as shown in Fig. 5, we divide the basic triangular pyramid space into octahedral and tetrahedral spaces by using the same structure as an octet truss. Of the spaces obtained after subdivision, the cells with octahedral shapes are called *octa-cells*, and the cells with tetrahedral shapes are called *tetra-cells*. However, since there are two types of tetrahedron with different orientations, in the following we will use a prime symbol (′) to distinguish between those that are oriented in the same way as tetrahedra on the first layer “tetra-cell” and those that are oriented in the opposite direction “tetra′-cell” (Fig. 6).

A 3D shape is defined by making each cell hold a value for distinguishing between the inside and outside of the shape. Specifically, a value of 1 corresponds to the interior of the shape, and a value of 0 corresponds to its exterior. This method is the same as ordinary voxels, so the shape representation method we propose here is referred to in the following as an

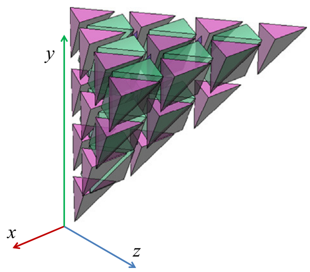

Fig. 5. Structure of octet-voxels constituting a basic triangular pyramid space.

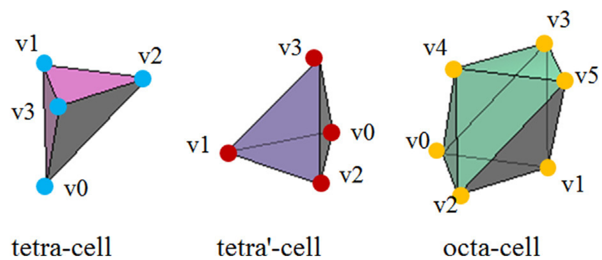

Fig. 6. Three types of cells.

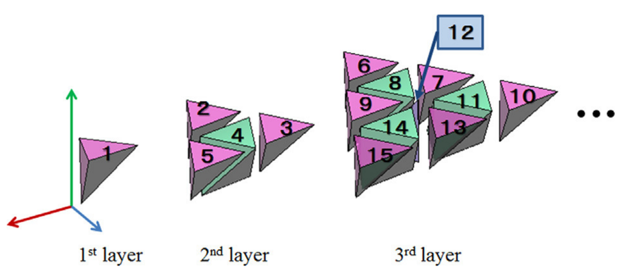

Fig. 7. Order of cell arrangement.

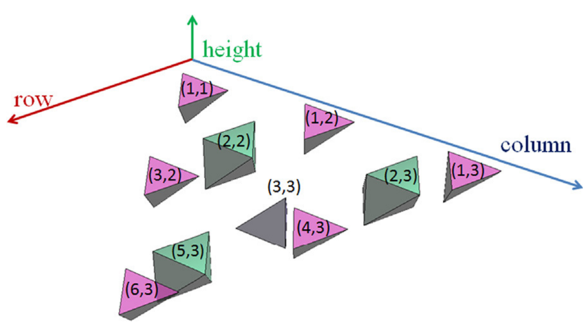

Fig. 8. Relationship between a cell's position in a single layer (row index, column index) and the cell type.

octet voxel method. Below, we discuss the properties of octet voxels according to the coordinate system shown in Fig. 5.

The basic triangular pyramid space is divided into multiple layers in the y-axis direction (the number of layers is referred to as the resolution in the following). If the layers are numbered sequentially from 1 starting with the lowermost layer, then layer k is partitioned by planes parallel with the bounding planes of the basic triangular pyramid space, thereby forming the tetrahedral and octahedral cells (Fig. 7). As shown in Fig. 8, if the cell positions in each layer are indicated by two integers representing the row and column, then the following formulae can be used to express the numbers of tetra-cells n_m^{tetra}, tetra′-cells $n_m^{tetra'}$ and octa-cells n_m^{octa} in the m-th column:

$$n_m^{tetra} = m$$

$$n_m^{tetra'} = \begin{cases} m-2(m \geq 2) \\ 0(m<2) \end{cases}$$

$$n_m^{octa} = \begin{cases} m-1(m \geq 1) \\ 0(m<1) \end{cases}$$

With the process shown in Algorithm 1, it is possible to determine the types of cells in the l-th row and m-th column. This judgment is applicable to any layer.

Algorithm 1. : Determination of cell type

Function: Determine the cell type
Input: l/* column index */, m/* row index */
if l % $3=1$ or $l=n_m^{tetra}+n_m^{tetra'}+n_m^{octa}$ **then**
    type ← tetra
else if l %$3=2$ **then**
    type ← octa
else
    type ← tetra′
return type

The total number of cells constituting the k-th layer N_k is represented by the formula shown below. When constructing the cell information on a computer, move to the next layer above when the number of voxels arranged in each layer has reached this value. N_k^{tetra}, $N_k^{tetra'}$, N_k^{octa} represent the total numbers of tetra-cells, tetra′-cells and octa-cells in layer k, respectively.

$$N_k = N_k^{\text{tetra}} + N_k^{\text{tetra}'} + N_k^{\text{octa}}$$

$$N_k^{tetra} = \frac{k(k+1)}{2}$$

$$N_k^{tetra'} = \begin{cases} \frac{(k-1)(k-2)}{2} & (k \geq 1) \\ 0 & (k=0) \end{cases}$$

$$N_k^{octa} = \frac{k(k-1)}{2}$$

The coordinates of the constituent vertices of a cell in the k-th layer at the l-th row and m-th column (v0 through v5 in Fig. 6) are calculated as follows based on the cell type. As shown in Fig. 9, W, H, and D are the width, height and depth of a single tetra-cell, and the cell type can be determined by Algorithm 1. n^{tetra}, $n^{tetra'}$ and n^{octa} are the numbers of cells of the same type included in the 1st through (l-1)th rows of the m-th column in the k-th layer.

tetra-cell

$$v0\left((2n^{tetra}-m)W, kH, mD\right)$$

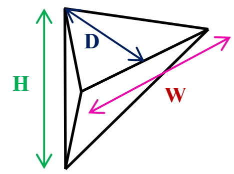

Fig. 9. Dimensions of a tetra-cell.

$$\mathrm{v1}\left((2n^{tetra}-m)W,(k+1)H,mD\right)$$

$$\mathrm{v2}\left((2n^{tetra}-m-1)W,(k+1)H,(m+1)D\right)$$

$$\mathrm{v3}\left((2n^{tetra}-m-1)W,(k+1)H,(m+1)D\right)$$

tetra′-cell

$$\mathrm{v0}\left((2n^{tetra'}-(m-2)-1)W,kH,mD\right)$$

$$\mathrm{v1}\left((2n^{tetra'}-(m-2)+1)W,kH,mD\right)$$

$$\mathrm{v2}\left((2n^{tetra'}-(m-2))W,kH,mD\right)$$

$$\mathrm{v3}\left((2n^{tetra'}-(m-2))W,(k+1)H,mD\right)$$

octa-cell

$$\mathrm{v0}\left((2n^{octa}-(m-1))W,kH,mD\right)$$

$$\mathrm{v1}\left((2n^{octa}-(m-1)-1)W,kH,mD\right)$$

$$\mathrm{v2}\left((2n^{octa}-(m-1)+1)W,kH,mD\right)$$

$$\mathrm{v3}\left((2n^{octa}-(m-1)-1)W,(k+1)H,mD\right)$$

$$\mathrm{v4}\left((2n^{octa}-(m-1)+1)W,(k+1)H,mD\right)$$

$$\mathrm{v5}\left((2n^{octa}-(m-1))W,(k+1)H,(m+1)D\right)$$

When partitioning the space into cells, by adding references to the neighboring cells to each octa-cell, it is possible to extract only the surface of the 3D object after the value of each cell has been determined with the computational complexity $O(N)$. There are only two possible combinations of adjoining cells – {octa-cell, tetra-cell} and {octa-cell, tetra′-cell} – because a tetra-cell and tetra′-cell cannot be adjoining. It is therefore possible to construct the adjacency relationships of all the cells by scanning only the adjacency relationships of octa-cells. Two mutually adjoining cells are separated by a triangle facet only when the sum of the values held by these cells is 1, in which case this triangle is displayed on the screen.

Based on the abovementioned properties of octet voxels, we created a program to read and write data and display it on the screen. In implementing this program, we opted to read and write a value of 0 or 1 for each cell starting with the first, in the order shown in Fig. 8. When a cell has a value of 1, according to the voxel type judgment results, a cell is placed in the basic triangular pyramid space. As mentioned above, the surface of the 3D object is only shown based on the values of mutually adjoining cells. The facets corresponding to the boundaries of the basic triangular pyramid space are not drawn when they are replicated, because this would mean they are ultimately contained within the 3D object.

3.2. User interface

To implement an AR interface with CG superimposed on video of the real-world space, we developed a system with a Vuzix STAR1200XL head-mounted display. The system was implemented in C++, with the OpenGL graphics library.

The AR interface was built using a popular AR toolkit [17]. We used two types of AR marker – one to determine the position and orientation of the 3D object being edited (held in the left hand), and an editing tool to provide air-spray interface (held in the right hand) as shown in Fig. 10. The editing tool was fitted with a wireless mouse in such a way that the mouse's button and scroll wheel could be used for editing operations. To provide the user with visual feedback using changes of color to indicate the air spray functions and operational state, a semi-transparent cone is displayed with its vertex at the position of the air spray marker, as shown in Fig. 11. A light blue sphere displayed at this tip position indicates the effect and range of the air spray.

The AR marker is shown on the screen, and the light blue sphere at the tip of the air spray is moved through the basic triangular pyramid space while holding down the mouse's left button. If the cell next to the sphere is adjacent to a cell that

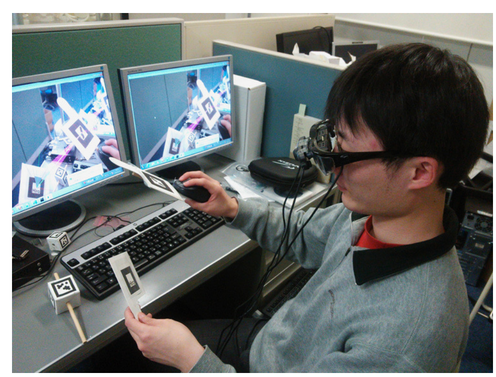

Fig. 10. The user interface of the proposed system. The user wears a head-mounted display, and holds a handle and an editing tool with AR markers.

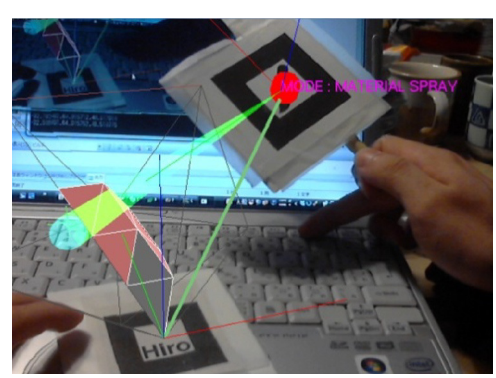

Fig. 11. Modeling in progress. Virtual material is blown onto a design shape from the AR marker held by the right hand. The octet-voxels in a triangular pyramid space is displayed at the maker held by the left hand.

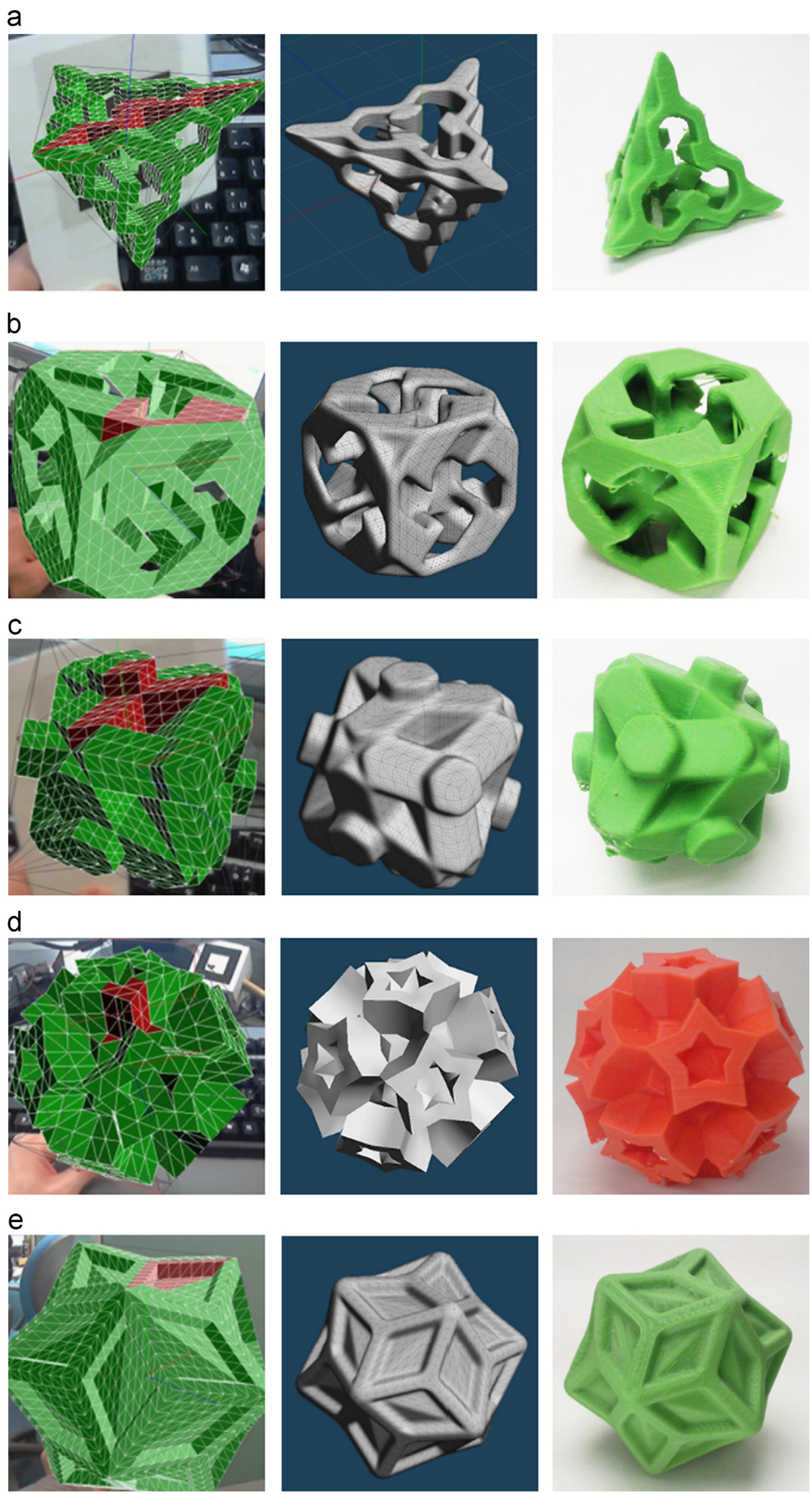

Fig. 12. Designed objects (left), 3D models for printing (center), and the printed results (right). The base regular polyhedra of (a) to (e) is tetrahedron, cube, octahedron, dodecahedron, and icosahedron, respectively.

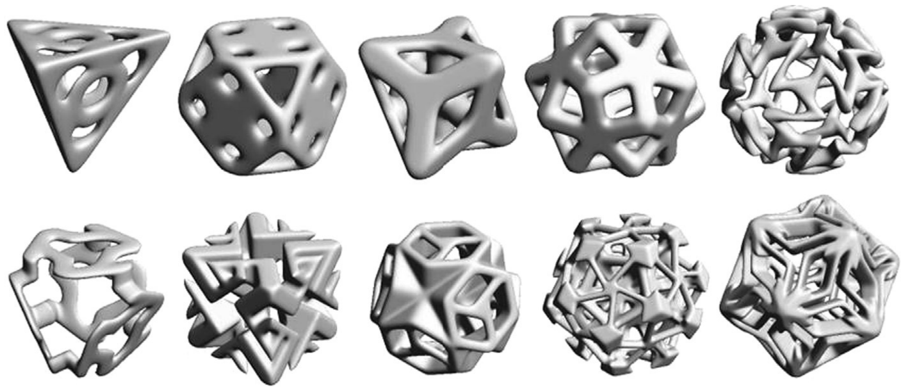

Fig. 13. 3D shapes of printed examples shown in Fig. 1. Smooth surfaces were obtained by applying mesh subdivision.

has a value of 1, this cell's value is also changed to 1. This gives the impression of working on a shape while blowing material onto it with the air spray. In the initial state, only the first layer of cells is assigned a value of 1. The right mouse button is used for an erasing air spray, which will change the value of its neighboring cell to 0 if it is equal to 1. Also, the range over which the effects of the air spray are exerted can be changed over a total of 10 levels by rotating the mouse's scroll wheel. It is also supported changing the base regular polyhedra in editing process. It means that users can preview five different objects simply by switching base regular polyhera.

4. Result

Using the proposed system, we designed various shapes. The octet voxel representation made it easy to produce designs having the same symmetry structures as five types of regular polyhedra (Fig. 12). We printed these shapes in PLA (polylactide), using a CubeX Duo 3D printer made by 3D Systems, and in Nylon (polyamide), using a commercial 3D printing service (Fig. 1). To smooth out the surfaces of these printed shapes, we applied a mesh subdivision method to the polyhedrons obtained from the octet voxels (Fig. 13).

To verify the usability of this system, we performed the following informal user test.

- For the first five minutes, we explained how to operate the system and allowed the user to practice.
- Next, the user was asked to design an ornament using this system. The user was allowed to select a preferred polyhedron from the base polyhedra.
- The cell resolution (number of layers) of a base triangle space was set to a small scale of $8 \times 8 \times 8$.
- The user was allowed to continue working until satisfied with the results.

This user test was performed with test subjects comprising four graduate students. The models produced by these test subjects are shown in Fig. 14. The test subjects completed the task in the range from 4 min to 7 min. In a free response questionnaire, we received favorable comments from the test subjects, who enjoyed the ease with which 3D objects could be created from scratch, and found it interesting to construct unexpected shapes from a single part. Using the proposed system to discover attractive ornamental shapes through trial and error, the users found that symmetrical 3D objects could be made one after another like symmetrical patterns in a kaleidoscope. On the other hand, there were also comments pointing out drawbacks of the air-spray interface, such as the difficulty of making shapes matching one's intentions. Furthermore, since the designs were too small when shown at the same dimensions as the objects actually produced by the 3D printer, the models were displayed at a size of several tens of centimeters during the design process, but it was pointed out that detracted from the value of using AR to project CG video in real space.

There were also calls for making it possible to perform editing directly even in parts made by duplication, so as to enable the creation of symmetrical forms even in a basic triangular pyramid space. This feedback is a source of important reference opinions for system improvement.

5. Conclusions

In this study, we developed a 3D computer graphics modeling system for the design of ornaments based on new data structures called octet voxels. From the results of user tests, we have confirmed that it is possible to produce a wide variety of forms by simple operations, that it is possible to create satisfactory shapes in a short time, and that the steps up to 3D printing can be performed smoothly. By implementing an AR user interface that mimics an air spray device, using replication to create symmetrical structures automatically, and creating data tailored for output to a 3D printer, we have made it possible for users without specialist skills to make 3D objects from readily available symmetrical shapes.

Based on the octet-voxel structure, not only the regular polyhedrons, but also other well known important polyhedrons such as Archimedean solids, and Kepler–Poinsot polyhedrons will be handled on the same framework. We believe our system is able to be easily extended without changing the basic data structure. The current system does not have the feature to handle multiple objects at a time, but after designing multiple

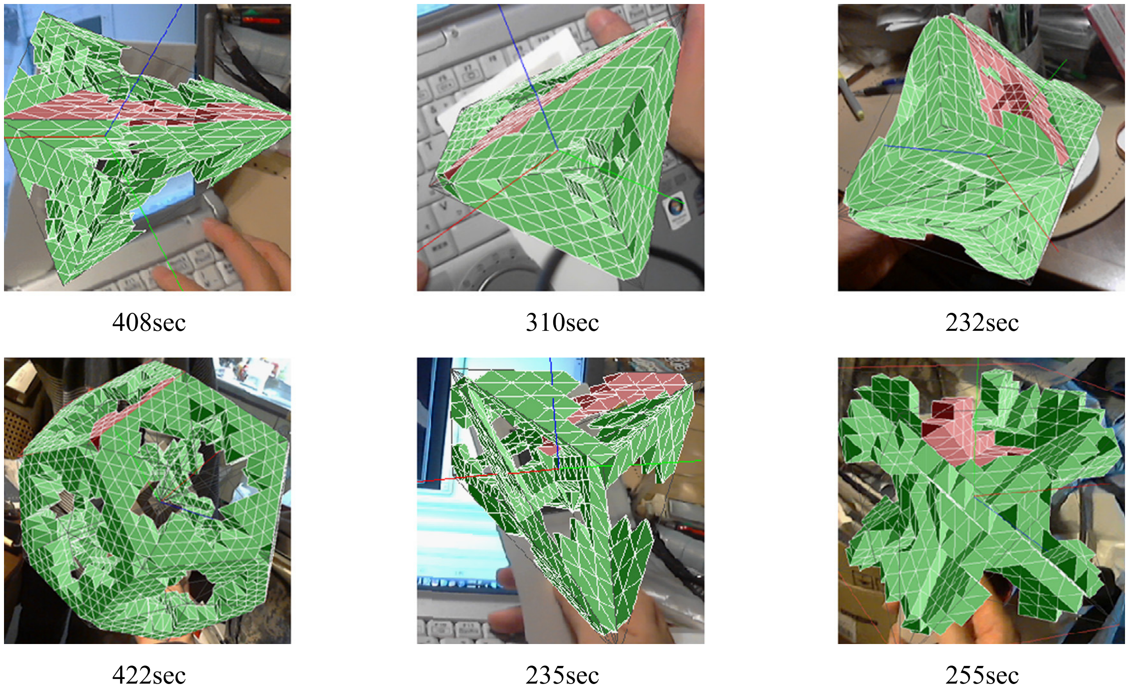

Fig. 14. Examples of 3D shapes produced by test subjects. The operation time is shown under each image.

objects independently, merging them will make it possible to assemble a larger object.

Since it was proven that making the intended shape and detailed structure using the implemented interface is difficult, we will also consider providing additional features. To cooperate with the air spray user interface, virtual mask or stamp tools would be a possible solution for adding straight lines or predefined shapes.

Conflict of interest statement

The authors have no conflict of interest to report.

References

[1] Bischoff S, Pavic D, Kobbelt L. Automatic restoration of polygon models. *ACM Transactions on Graphics* 2005;**24**(no. 4)1332–52.

[2] Ju T. Fixing geometric errors on polygonal models: a survey. *Journal of Computer Science and Technology* 2009;**24**(no. 1)19–29.

[3] Attene M, Campen M, Kobbelt L. Polygon mesh repairing: an application perspective. *ACM Computing Surveys* 2013;**45**(no. 2)15:1–33.

[4] Butterworth J, Davidson A, Hench S, Olano MT. 3DM: a three dimensional modeler using a head-mounted display. In: I3D '92 Proceedings of the 1992 Symposium on Interactive 3D Graphics; 1992; p. 135–138.

[5] Schkolne S, Pruett M, Schröder P. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In: CHI '01 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems; 2001; p. 261–268.

[6] Cheok Adrian David, Edmund Neo Weng Chuen, Wee Eng Ang. Inexpensive non-sensor based augmented reality modeling of curves and surfaces in physical space. In: ISMAR '02 Proceedings of the 1st International Symposium on Mixed and Augmented Reality; 2002; p. 273–274.

[7] Sano A. An application for creating full-scale augmented reality content without 3d modeling skills. In: RDURP '11 Proceedings of the 2011 ACM Symposium on the Role of Design in UbiComp Research & Practice; 2011; p. 19–24.

[8] Schlaug F. 3D Modeling in Augmented Reality, Final Thesis in Institutionen för systemteknik Department of Electical Engineering, LITH-ISY-EX-ET-10/0379-SE; 2011.

[9] Jung H, Nam T, Lee H, Han S. Spray modeling: augmented reality based 3D modeling interface for intuitive and evolutionary form development. In: Proceedings of International Conference on Artificial Reality and Telexistence; 2004.

[10] Owada S, Nielsen F, Nakazawa K, Igarashi T. A sketching interface for modeling the internal structures of 3D shapes, smart graphics 2003. *Lecture Notes in Computer Science (LNCS)* 2004;**2733**:49–57.

[11] Rivers A, Durand F, Igarashi T. 3D modeling with silhouettes. *ACM Transactions on Graphics* 2010;**29**(no. 4)109:1–8.

[12] Hirose M, Mitani J, Kanamori Y, Fukui Y. An interactive design system for sphericon-based geometric toys using conical voxels. In: SG'11 Proceedings of the 11th International Conference on Smart Graphics; 2011; p. 37–47.

[13] Prévost R, Whiting E, Lefebvre S, Sorkine O. Make it stand: balancing shapes for 3D fabrication. *ACM Transactions on Graphics* 2013;**32**(no. 4) 81:1–0.

[14] Mitra N, Guibas L, Pauly M. Symmetrization. *ACM Transactions on Graphics* 2007;**26**(no. 3)63:1–8.

[15] Jiang N, Tan P, Ping C, Loong F. Symmetric architecture modeling with a single image. *ACM Transactions on Graphics* 2009;**28**(no. 5)113:1–8.

[16] Willis Karl DD, Lin J, Mitani J, Igarashi T. Spatial sketch: bridging between movement & fabrication. In: TEI '10 Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction; 2010; p. 5–12.

[17] Kato H. ARToolKit: Library for Vision-based Augmented Reality, Technical report of IEICE. PRMU, 101, no. 652; 2002; p. 79–86.