

Augmented Reality for Zome Construction

Gouta Hayashi and Hideki Tsuiki

Graduate School of Human and Environmental Studies, Kyoto University
Hayashi.gouta.33z@st.kyoto-u.ac.jp, tsuiki@i.h.kyoto-u.ac.jp

Abstract

Zometool is widely used for mathematical constructions, and can also be used for constructions of complicated 3D models [3]. For complicated constructions, this is not an easy task and the user can easily become confused. We developed an AR (augmented reality) system for constructing complicated 3D Zome models. With this system, the target Zome model appears on a smartphone screen overlapped to the image of the physical Zome model under construction, and one can easily see how to connect the struts to complete the model.

Introduction

AR (augmented reality) is a technology to enhance the real world with computer-generated information. Among its applications is the support of assembling physical objects by displaying the target model overlapped to the real-world image of the object under construction and visually giving instructions step by step. Smartphones are an effective platform for AR systems because one can move the device around the physical object and match it with the virtual target object from many different angles. A smartphone uses information given through cameras and various sensors to overlay a virtual object on its camera view as if the object exists physically at a fixed location of the real-world. Recently, AR toolkits for smartphones like ARKit[1] and ARCore were released to ease the development of AR software.

Zometool is a popular assembly tool for constructing mathematical models [8]. It has a simple structure consisting of a single type of node called a ball and a small set of different edges called struts. It is mathematically designed based on the symmetry of a regular icosahedron. Therefore, one can construct objects with mathematical structures like projections of 4D H4-polytopes [2] symmetrically following their structures, and one usually does not need design drawings.

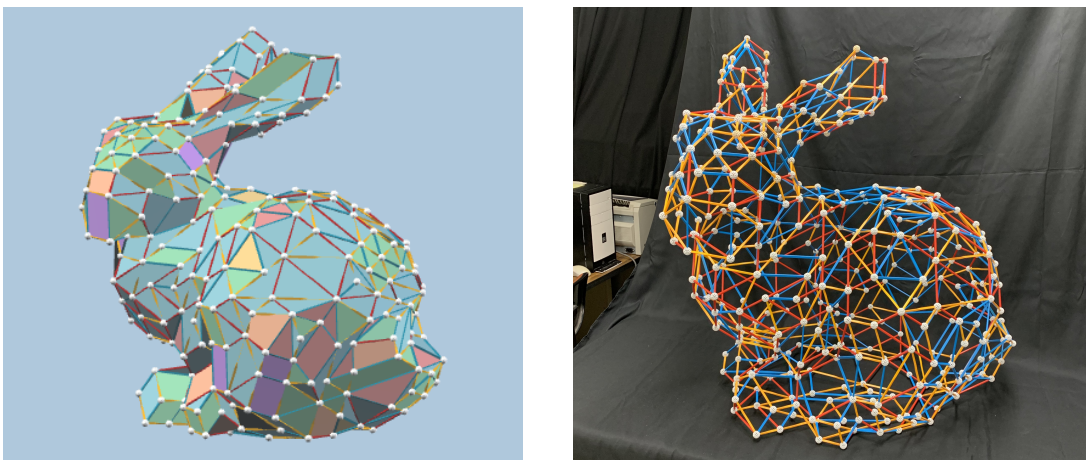


Figure 1: *Stanford Zome-Bunny [3] in vZome (left) and the Zome model we assembled (right).*

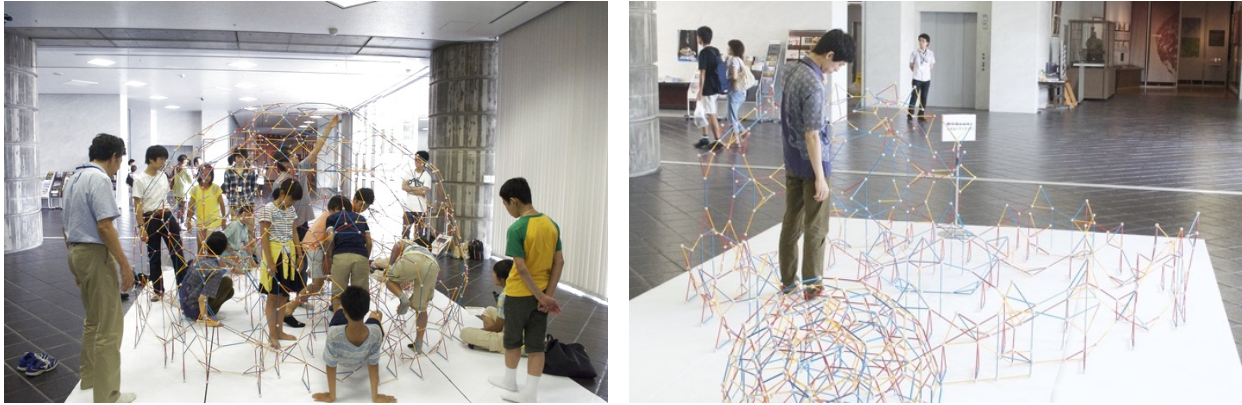


Figure 2: *Truncated 120-cell dome and the lower part of its construction [4].*

On the other hand, Zometool can also be used to construct a complicated model without mathematical structure. Henrik Zimmer, et al. designed software to make a Zometool model based on a given input shape [6, 7]. On their homepage [3], some Zometool models approximating physical objects like Stanford Zome-Bunny (Figure 1) are available in the Zome Mesh format and the formats of Scott Vorthmann's vZome [5], and one can download and display them with vZome. In order to physically construct such an object, one needs to go back and forth between a computer-supported design drawing and the partially completed physical model. During this process, the builder can easily get confused.

We give another example of a complicated Zome construction. In 2015, our research group had a workshop headed by Yasuyuki Tsukamoto to construct a huge walk-in dome that is almost the upper 2/3 of a truncated 120-cell [4] (Figure 2; left). It was great fun to enter into a huge Zome structure and have a look from the center of the object. Usually, assembling a whole truncated 120-cell with smaller struts is not a difficult task with the understanding of its structure, because one can construct it from the center and use the symmetry of the structure as a guide on what to do next [2]. However, it was a very difficult task to construct this dome from the cross-section on the floor to the top (Figure 2; right).

This paper is about the development of smartphone AR systems which support this kind of Zome constructions. We used the iPhone Xs Max with ARKit version 3 and developed two experimental systems. The first one recognizes a half-constructed Zome physical model and displays a virtual target model adjusted to the physical model. It succeeded for some experiments but the authors feel that we need a lot of improvements to make it a practical system. The second one is a simpler AR system in which one needs to adjust the virtual model to the physical model by hand, but one can easily do this with the user interface of smartphones. It worked very well for our Zome constructions and we assembled Zome-Bunny (Figure 1) and some part of the Zome-dome mentioned above with this system.

AR system with image recognition

We first constructed an AR system such that, when a half-constructed Zome physical model is given to the camera, (1) it recognizes Zome nodes; (2) converts them to the Zome coordinate system; (3) matches them with the target Zome shape; and (4) displays the virtual target model overlapped to the physical model.

To recognize a ball, we used the Hough transform, which is a classical method to detect lines and circles from images. The system extracts high brightness parts of the image to capture only white balls, and applies the Hough transform to identify balls in the image. Then, it calculates their 3D positions according to the output of the Hough transform and the depth information given by the ARHitTest mechanism of the ARKit,

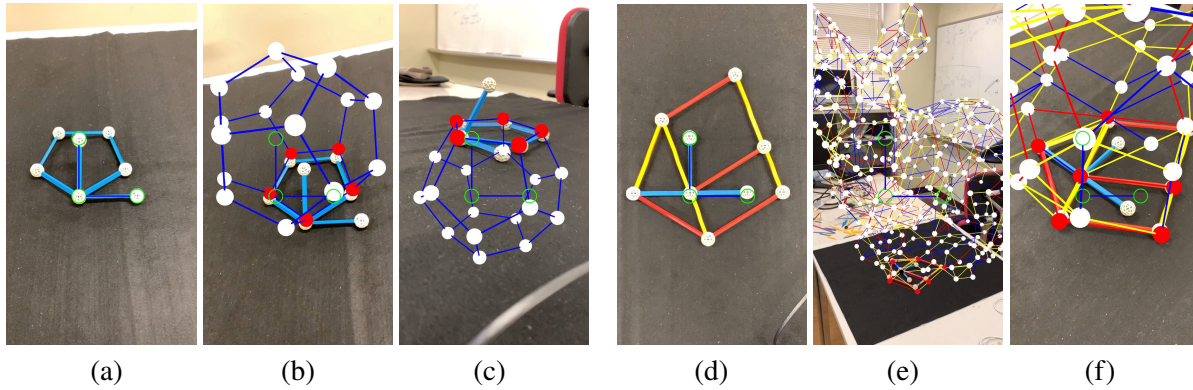


Figure 3: (a) a part of a dodecahedron; (b) virtual dodecahedron overlapped to (a); (c) virtual dodecahedron overlapped to the flipped image of (a); (d) a part of the bunny; (e, f) virtual bunny model overlapped to (d).

and judges whether two balls are connected by some struts through the analysis of the image.

From the information of the 3D positions of the balls and struts, the system guesses its structure as a Zome model. It is known that, with some axis of coordinates, all the balls of a connected Zome model are located at (x, y, z) where $x, y, z \in \mathbb{Z} + \mathbb{Z}\varphi$ for φ the golden ratio $\frac{1+\sqrt{5}}{2}$ and \mathbb{Z} the set of integers. According to the strut connections, the system assigns such Zome coordinates to balls. In order to make the calculation simple and accurate, a user is required to select the origin ball and add two orthogonal blue struts corresponding to axes, and adjust them to the guide line appearing on a smartphone while the system is detecting this structure. Once it assigns consistent Zome coordinates to the balls, it checks whether it is a part of the final model, by checking all the changes of the origin and all the rotations of symmetry of the Zome system. If it succeeds, it shows the virtual image overlapped to the physical one. If there are more than one candidate and the user thinks that the displayed one is not correct, he can choose another one. If there is no candidate, the system captures a new image and repeats this procedure until it succeeds.

This system worked for some examples (Figure 3), but failed for complicated ones even under controlled environments. We had black background to make the image recognition easy, but the detection of balls and struts becomes difficult if there are some balls and struts in the background not connected to the origin. The system could not find the Zome structure for three-dimensionally complicated models; one reason would be that the depth information given through the ARHitTest mechanism does not have enough precision.

In addition, there is a gap between the virtual model that appears on the smartphone and the physical model, and the virtual model slowly moves around as we proceed with assembling the model. Since the virtual model is fixed to the world-coordinate system defined in a smartphone, it means that the world-coordinate system moves as we move a smartphone and it is a limitation of an AR with smartphones. To solve this problem, we first planned to detect the image frequently, but it is difficult as we mentioned above. Therefore, we added a mechanism to adjust the virtual model with the physical model by hand, and it leads to another system, described in the next section.

AR system without image recognition

The second system does not do image recognition and only has the adjustment mechanism. When the new system is started, the virtual model appears in front of the smartphone and is fixed to the camera coordinates. Therefore, it moves as the camera moves and a user can adjust it to the physical model under construction by moving the smartphone. Then, the user will tap the release button to fix it to the world coordinate. After

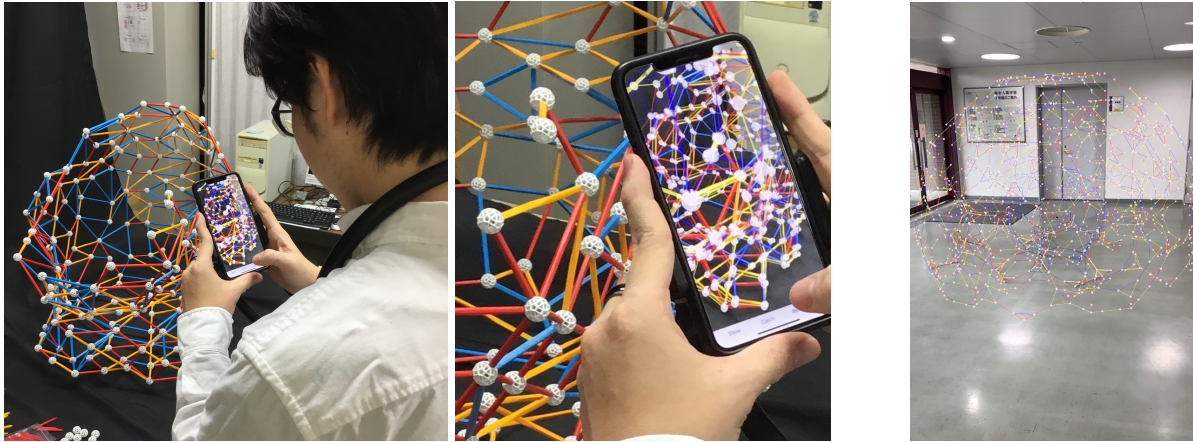


Figure 4: AR Zome construction of Zome-bunny and a part of Zome-dome

that, the user may move around and find a gap between the physical model and the virtual model. Then, the user will attach the virtual model to the camera coordinate by tapping a capture button. After that, the user will move the smartphone to make another adjustment and release it. When one is working with a big sparse model, the physical model is distorted according to gravity and it is impossible to adjust the whole physical model to the virtual image. This system works even for such a case because one only needs to adjust the part one is working on. This catch and release mechanism would be a general method that can be used for object layout in virtual space in general, not only for Zome constructions.

Experiments and Conclusions

With the second system¹, we assembled the Zome-bunny (Figure 1) and a part of the Zome-dome we explained in the introduction (Figure 2). With a phone neck holder, one can use both hands for assembling while looking at the object through the smartphone. Note that this system can be used for assembling a big model with several people.

References

- [1] ARKit homepage. <https://developer.apple.com/augmented-reality/>
- [2] G. Hart, "Four-Dimensional Polytope Projection Barn Raisings," in Proceedings of International Society of Art, Math, and Architecture 2007, Texas A&M, May, 2007.
- [3] L. Kobbelt. Zometool Software page. <https://www.graphics.rwth-aachen.de/software/zometool/>.
- [4] Hideki Tsuiki. Truncated 120-cell Dome. <https://www.i.h.kyoto-u.ac.jp/users/tsuiki/600cell/truncated120/index-e.html>
- [5] vZome homepage. <https://vzome.com/home/>
- [6] H. Zimmer and L. Kobbelt, "Zometool Rationalization of Freeform Surfaces," in IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 10, pp. 1461-1473, Oct. 2014.
- [7] H. Zimmer, F. Lafarge, P. Alliez, L. Kobbelt "Zometool Shape Approximation." Graphical Models Volume 76, Issue 5, 2014, pp. 390-401
- [8] Zometool homepage. <https://www.zometool.com>

¹This system is available from <https://github.com/gou-hayashi/ARZome>.