Automatic Generation of 3D Typography

Suzi Kim Sunghee Choi
Korea Advanced Institute of Science and Technology

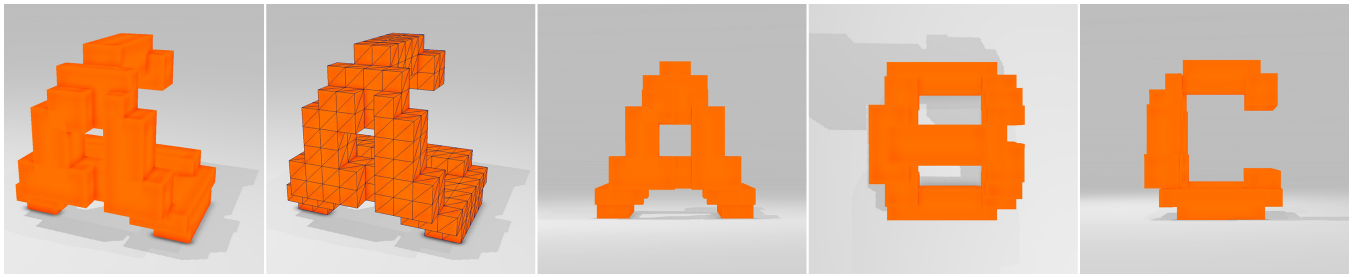

Figure 1: *Our system automatically creates a 3D typography model for given letters. The model shows different letter depending on the view direction. The figure is an example of the 3D typography model from letters 'A' (front view), 'B' (top view), and 'C' (side view).*

Keywords: typography, 3D typography, 3D CAPTCHA, 3D text model

Concepts: •Computing methodologies → Shape modeling;

1 Introduction

Three-dimensional typography (3D typography) refers to the arrangement of text in three-dimensional space. It injects vitality into the letters, thereby giving the viewer a strong impression that is hard to forget. These days, 3D typography plays an important role in daily life beyond the artistic design. It is easy to observe the 3D typography used in the 3D virtual space such as movie or games. Also it is used frequently in signboard or furniture design. Despite its noticeable strength, most of the 3D typography is generated by just a simple extrusion of flat 2D typography. Comparing with 2D typography, 3D typography is more difficult to generate in short time due to its high complexity.

To the best of our knowledge, there has been no attempt to generate a 3D typography automatically. This paper introduces a method to design a 3D typography from given input letters automatically. We focus on the optical illusion technique to generate an object which shows different texts depending on the viewer's perspective. We suggest a method satisfying two conditions: 1) it must create an 3D model which provides different contents in accordance with different perspectives and 2) the 3D model must have orthogonal projections which are identical with the corresponding shape of letters. Afterword, this paper applies the 3D typography to 3D CAPTCHA.

2 Our Approach

We use a molecular construction strategy [Miller 1996] to generate a 3D typography. Molecular construction is a technique to build a whole typography from smaller units. We divide the cube into smaller blocks, each of which is a smaller unit cube and remove some blocks to express letters.

First of all, we created Latin alphabet patterns which will be carved to the cube. Any font can be tailored to pixel-based pattern. After preparing images which contain each letter from A to Z using a particular font, $n \times n$ grid is drawn on the images, where n is the edge length of a cube. If the cell is covered over 50% by the letter, we mark the cell as pattern and repeat this process for all the cells in the grid.

Pattern sculpting is a basic operation to sculpt the pattern on a face of the cube. As a cube consists of many blocks, a pattern can be sculpted on a cube by removing of blocks which obstruct the recognition of a pattern. To sculpt multiple letters on a single cube, each extrusion of letter is merged to one model from different directions one by one. Since the previous letter could be spoiled by later letter, verification is necessary to check whether current carving might spoil the shapes of previous letters or not. If the synthesis fails in every combination of patterns with rotating or flipping of patterns, we regard the combination of letters as impossible to synthesize. To increase the success rate of a 3D typography generation for arbitrary combinations, the original patterns are replaced with candidate patterns which compromise the legibility but succeed in synthesis of 3D typography.

CAPTCHA [Von Ahn et al. 2003] is a test that distinguishes between human and computer. The most commonly used CAPTCHA is based on the humans' ability to read distorted 2D letters. However, through the development of machine learning, computer can recognize the alphabet quickly. For this reason, 3D CAPTCHA [Imsamai and Phimoltares 2010; Susilo et al. 2010] was designed to minimize the legibility. However, previous studies hardly represent the complete 3D CAPTCHA because they just enumerate the 3D objects extruded from 2D text with depth on a single direction. We suggest a new 3D CAPTCHA which allows human to recognize the text at some certain angles from 360 degrees of freedom.

To extend 3D typography to 3D CAPTCHA, we build an undirected multigraph $G = (V, E)$ whose vertex set V is the set of all blocks in the cube and edge set E represents the connection between vertices. Multiple edges are used to represent the physical connection between blocks (C-edge) and the connectivity of blocks in terms of projection direction (X-edge, y-edge, and Z-edge). One vertex can have a maximum of six C-edges because a block has six faces. X-edge, Y-edge and Z-edge represent the connectivity of blocks in

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4371-8/16/07
DOI: http://dx.doi.org/10.1145/2945078.2945099

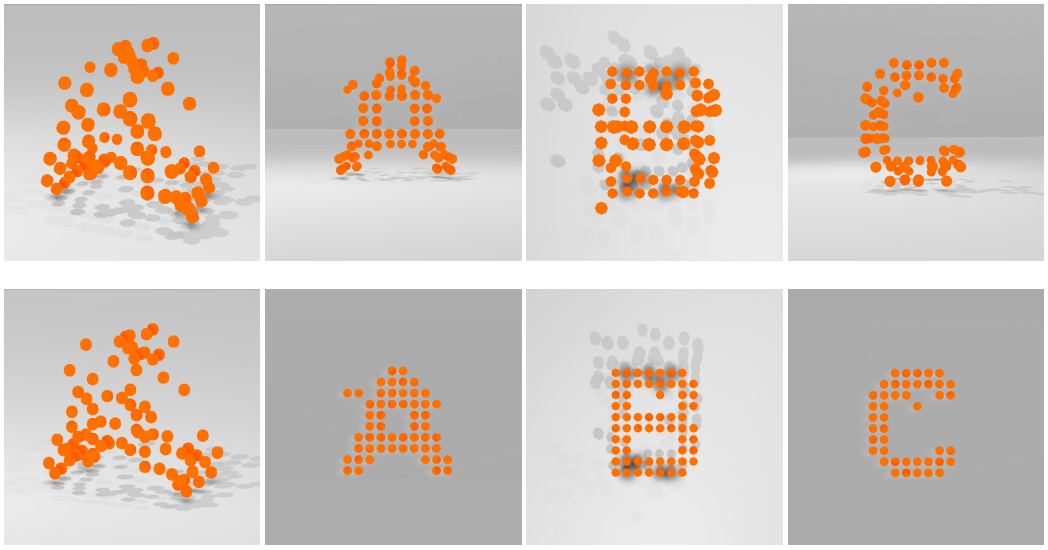

Figure 2: *3D CAPTCHA captured on perspective projection (top) and orthogonal projection (bottom).*

terms of projection direction x, y, z respectively. It connects the blocks which generate same projection result on the screen.

In order to make our 3D CAPTCHA difficult to be recognized by a machine, our 3D typography model goes through three additional stages. The first stage is the removal of redundant blocks, which have no influence to the recognition of text, to minimize the number of blocks in the cube. This stage disturbs the recognition of texts in an arbitrary direction except some directions which are perpendicular to the faces of the cube. During the removal of blocks, the arrangement of blocks must be balanced in the cube. If the blocks are concentrated in some areas, there is a chance to detect the shape of the letter in that area regardless of humans' perspectives. Thus, we score the blocks and remove them in descending order of score until every block is decided once to be removed or not. The scoring function S of vertex v is defined as follows:

$$
\begin{aligned}
S(\boldsymbol{v}) &= \alpha \cdot \frac{\sum_{\boldsymbol{g} \in N} E(\boldsymbol{g})}{\|N\|} + \frac{\sum_{\boldsymbol{x} \in X} E(\boldsymbol{x})}{\|X\|} \\
&+ \frac{\sum_{\boldsymbol{y} \in Y} E(\boldsymbol{y})}{\|Y\|} + \frac{\sum_{\boldsymbol{z} \in Z} E(\boldsymbol{z})}{\|Z\|} \\
G_{\boldsymbol{\kappa}} &= \{\boldsymbol{g} |\ |\boldsymbol{v} - \boldsymbol{g}| \leq |\boldsymbol{\kappa}|, \boldsymbol{g} \in U\} \\
N &= G_{(k,k,k)}, \quad X = G_{(xDepth,0,0)}, \\
Y &= G_{(0,yDepth,0)}, \quad Z = G_{(0,0,zDepth)} \\
E(\boldsymbol{c}) &= \begin{cases} 1 & \text{if there is a block at } \boldsymbol{c} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

where N is a set of neighbor blocks in range of k and X, Y, Z represent the set of blocks connected to $\boldsymbol{v}$ by X-edge, Y-edge, and Z-edge, respectively. The first term is for the dispersion of blocks in the cube. It is a density ratio of neighbors around center block $\boldsymbol{v}$. It counts the number of neighbor blocks in the range of k. The remaining terms represent the importance of block itself in the row, column, and depth respectively. For example, if there are enough number of blocks in the same row, the block is easy to be removed because other block in the same row still makes the shadow on the same position. α is a constant to balance between dispersion and importance terms.

Second, random noise is added to the cube in order to jitter the data. The noise makes the 3D CAPTCHA legible for human but difficult for machine. In case of the 2D CAPTCHA, horizontal line or distortion plays the same role of noise. However, too much noise makes human impossible to recognize the text also. We add 0.5% of the total number of blocks in the initial cube to the synthesis result randomly. To make 3D CAPTCHA robust to edge detection, we replace the blocks with small spheres. It prevents 3D CAPTCHA attacks from vanishing point detection which computes the perspective of the object automatically by edge detection. We adopt the scatter type [Baird and Riopka 2005] method by reducing the radius of sphere. The scatter type method breaks the letters into a large number of small pieces to make segmentation of the letters .

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A09060399).

References

BAIRD, H. S., AND RIOPKA, T. P. 2005. Scattertype: a reading captcha resistant to segmentation attack. In *Electronic Imaging 2005*, International Society for Optics and Photonics, 197–207.

IMSAMAI, M., AND PHIMOLTARES, S. 2010. 3d captcha: A next generation of the captcha. In *Information Science and Applications (ICISA), 2010 International Conference on*, IEEE, 1–8.

MILLER, J. A. 1996. *Dimensional Typography:: Words in Space: Kiosk Report# 1*. No. 1. Princeton Architectural Press.

SUSILO, W., CHOW, Y.-W., AND ZHOU, H.-Y. 2010. Ste3d-cap: Stereoscopic 3d captcha. In *Cryptology and Network Security*. Springer, 221–240.

VON AHN, L., BLUM, M., HOPPER, N. J., AND LANGFORD, J. 2003. Captcha: Using hard ai problems for security. In *Advances in Cryptology-EUROCRYPT 2003*. Springer, 294–311.