

DESIGNING A NEW TOY TO FIT OTHER TOY PIECES - A shape-matching toy design based on existing building blocks -

Yuki IGARASHI¹ and Hiromasa SUZUKI²

¹The University of Tokyo, Japan / JSPS research fellow

²The University of Tokyo, Japan

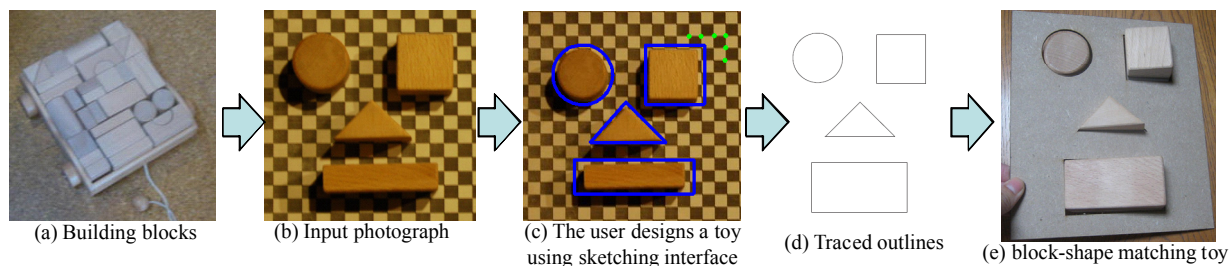


Figure 1: Overview of our method

ABSTRACT: Shape-matching toys are popular items for infants, and consist of boxes with many holes in different shapes along with corresponding blocks of the same shapes. To play with the toy, an infant finds and inserts a block matching the shape of a particular hole. It is very difficult to design such toys because the toy body O_{new} must closely fit the shape of the existing blocks $O_{existing}$. The designer usually takes the measurements of the blocks $O_{existing}$ and draws corresponding feature sizes, then makes a construction diagram for the toy body O_{new} . We propose a computer-based method for novices to enable the design of a new toy to fit other toy pieces. The method involves making a construction diagram for O_{new} on a photograph of the blocks $O_{existing}$ using a computer. The user first takes a photograph of the blocks on a checkerboard, then designs the form of the new toy O_{new} on the photograph using a sketching interface. From the photograph on the checkerboard, the system automatically fits the designed toy O_{new} to the actual measurements and exports the results in vector form. The system supports SVG and DXF formats. Finally, the user cuts the shapes from a wooden board using a cutting plotter or laser printer.

Keywords: design, user interface, craft

1. INTRODUCTION

Shape-matching toys are popular items for infants, and consist of boxes with many holes in different shapes along with corresponding blocks of the same shapes. To play with the toy, an infant finds and inserts a block matching the shape of a particular hole. It is difficult to design new shape-matching toys based on existing blocks. We assume that the user performs such design as shown in Figure

1 (e) based on existing building blocks like those shown in Figure 1 (a). The construction of the toy body can be roughly divided into three steps: gather the parts, lay them out on a wooden board and trace them using a pencil, and saw the wooden board. This manual method is straightforward, but errors cannot be rectified and it is also unsuitable for mass production.

Accordingly, we propose the use of a laser

cutter (e.g., Commax Laser System [1]) or a cutting plotter (e.g., Craft ROBO [2]). Today, services are available that allow the user to send a vector dataset to a company and have the corresponding wooden board returned to them [3]. This means that the user can obtain a new toy body simply by providing shape data in vector form. The construction of a toy using a laser cutter can be roughly divided into five steps: gather the parts, measure them, implement design using existing software that allows the user to input actual measurements (e.g., Adobe Illustrator [4]), output the design in vector form and input the data to the laser cutter. However, the labor-intensive nature of this process means that it is very difficult to design a new toy body O_{new} to fit existing blocks $O_{existing}$. The user usually takes the measurements of the blocks $O_{existing}$ and draws corresponding feature sizes, then creates a construction diagram for the toy body O_{new} .

To enable the design of new toys based on existing pieces and other similar products, we propose a computer-based method for novices to enable the design of a new toy body to fit existing toy blocks. The method involves creating a construction diagram for O_{new} on a photograph of the blocks $O_{existing}$ using a computer (Fig. 1). The user first takes a photograph of the existing blocks $O_{existing}$ on a checkerboard, then designs the form of the new body O_{new} on the photograph using a sketching interface. From the photograph on the checkerboard, the system automatically fits the designed body O_{new} to the actual measurements and exports the results in vector form. The system supports SVG and DXF formats. Finally, the user cuts the shapes from a real wooden board using a cutting plotter or laser printer.

2. SYSTEM OVERVIEW

Figure 1 shows a real toy made using the proposed method. Our goal was to design a new shape-matching toy body (Fig. 1 (e)) from existing building blocks (Fig. 1 (a)). Figure 2 shows the proposed user interface.

The user first prints a checkerboard and takes a photograph of the existing building blocks $O_{existing}$ on it. The checkerboard consists of squares with 10-mm edges.

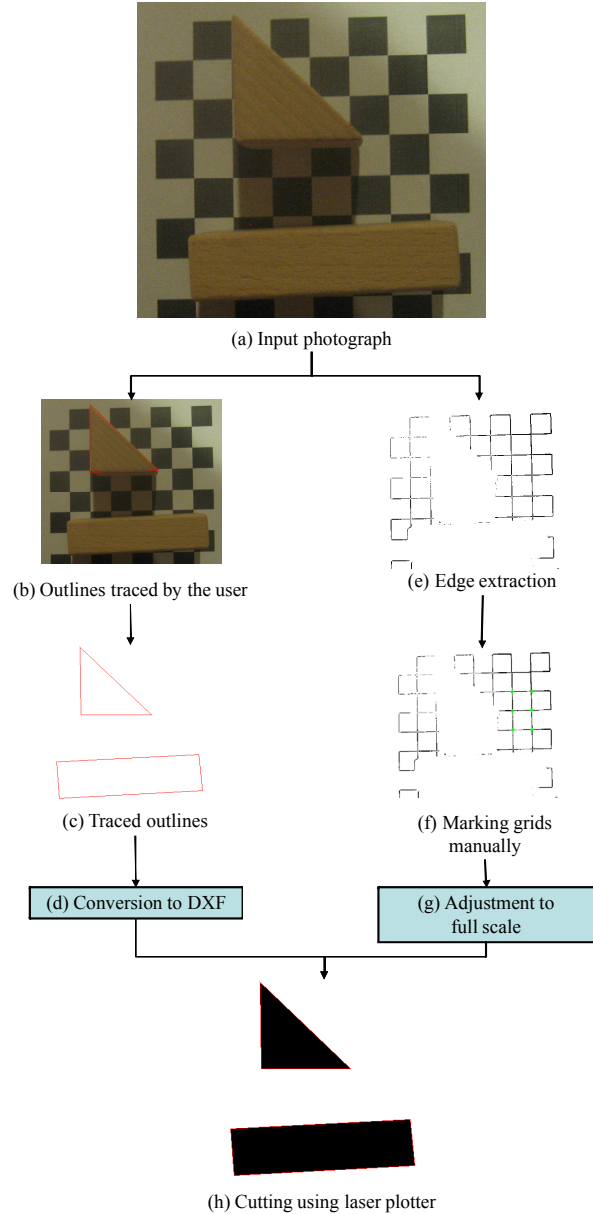


Figure 2: Overview of our algorithm

The user draws free-form strokes on the input photograph to trace the outlines of the blocks using the sketching interface (Fig. 2 (b)). The system also provides a number of basic shapes (line, rectangle and ellipse), and exports the user-input strokes in vector form as shown in Fig. 2 (c). In line mode,

user-input strokes can be made to snap to existing strokes by holding down the Shift key. In rectangle mode, the user can draw squares by holding down the Shift key. In ellipse mode, the user can draw circles by holding down the Shift key. The system finally exports the use-input strokes in vector form as shown in Figure 2 (c).

The checkerboard is used for scale adjustment. Although various methods have been proposed for feature point extraction in 2D [5], [6], [7], no method currently enables precise extraction. Accordingly, we adopt manual input by the user. The system first extracts the checkerboard grid using edge detection, and the user can easily mark the corners of the grid using the extracted edge image (Fig. 2 (e)). The user has to mark corners at more than two unbroken points (Fig. 2 (f)). The system calculates the ratio r of the image to the actual size (Fig. 2 (g)).

The system exports the designed shape (Figure 2 (c)) in vector form using the ratio r , and supports SVG and DXF formats. The user cuts the shapes from a real wooden board using a laser printer [1] or a cutting plotter [2]. In this way, the user can design a toy body O_{new} to fit existing blocks $O_{existing}$.

3. ALGORITHM

Our prototype system was implemented as a Java program with the user-input strokes represented as polylines. The system employs internal vector representation.

3.1 Edge detection

Edge detection here uses a convolution filter. The system first transforms the input image into gray-scale. To obtain the output shown as Image I' (Figure 3, bottom) from Image I (Figure 3, top), we prepare four Sobel operators (two for the perpendicular direction and two for the horizontal direction). The system first applies each Sobel operator to the

input image, then inverts each image and adds it. The system finally obtains Image I'.

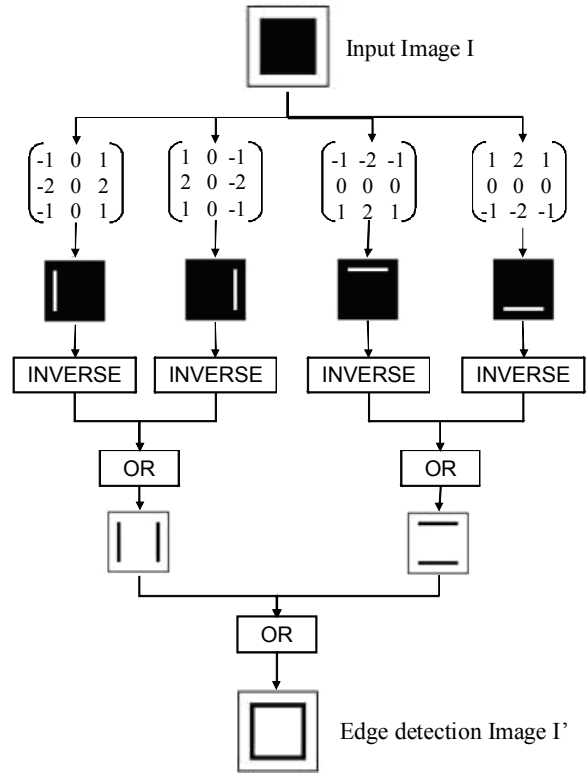


Figure 3: Edge detection algorithm

3.2 Adjustment to actual scale

After edge detection, the user marks the corners of the grid manually as shown in Figure 2 (f). The user has to mark corners connecting one side of a slot on the checkerboard (Figure 2 (f)). The system then computes the average distance for the sides on the grid. Specifically, it first finds the closest marker point for each point P_i and calculates the distance d_i to its closest point. The system then calculates the average value of these computed distances.

4. RESULTS

The system runs in real time on a 1.1-GHz Pentium-M PC. We designed two shape-fitting toys using our approach and created real toys, one of which is shown in Figure 4. This example shows the design of a new block-shape-matching toy body based on

existing building blocks. The other is shown in Figure 5. This example shows the design of a new puzzle cover based on existing pieces. The holes of the cover designed fit the relevant positions on the base puzzle. We export the final design data in DXF using our system and perform laser cutting [1].

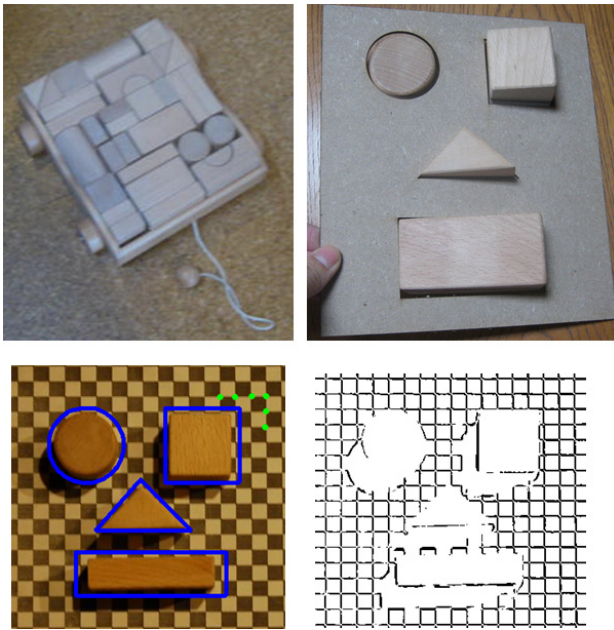


Figure 4: Example of designing a shape-matching toy body based on existing building blocks using our method

5. EVALUATION

We tried the system with three test-users, who designed a shape-matching toy as shown in Fig. 1. Graph 1 shows a comparison of the approach and its implementation using existing software (Adobe Illustrator). In our method, the user first takes a photograph of the building blocks $O_{existing}$ on a checkerboard, then draws strokes to create a diagram for the shape-matching toy body O_{new} on the photograph and marks its corners. With the existing approach, the user first measures the building blocks $O_{existing}$ and draws a construction diagram for the shape-matching toy body O_{new} . We skip the remaining processes of exporting the designed shape in vector form and formation using a cutting plotter, as both methods follow the same

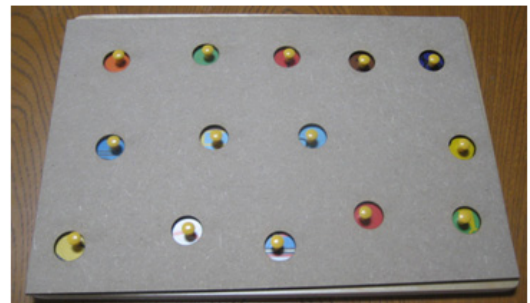
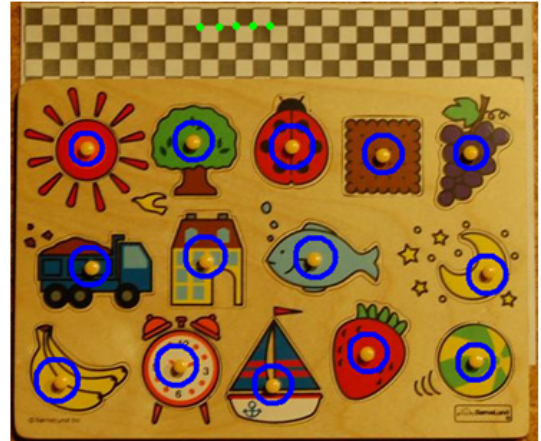


Figure 5: Example of a puzzle cover design based on an existing puzzle.

process after shape design. As indicated by the results in Graph 1, new toy design using our method is relatively easy.

We also check the number of markers the system needs. It is only necessary to mark 5 – 6 points according to the results of the user study as shown in Fig. 4. This took about 10 – 13 seconds, and the test-users reported that they found the system easy to use.

Graph 1: Comparison of timing data

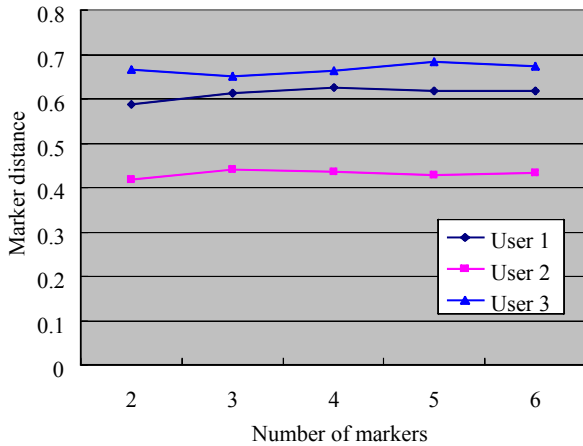
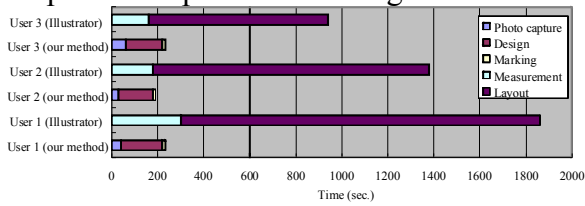


Figure 6: Number of markers

6. CONCLUSION AND FUTURE WORK

We propose a computer-based method for novices to enable the design of new toy bodies to fit existing pieces. The approach allows simple design of shape-matching toys through tracing with a pencil. From the photograph on the checkerboard, the system automatically fits the designed toy body O_{new} to the actual measurements taken.

One limitation of this method is that care is required with taller pieces. For instance, if a photograph is taken from a height of 40 cm and the target toy is 4 cm tall, the shape designed using the system is 10% smaller than the user requires. This problem was resolved here by taking the photograph from a higher position. Although various methods have been proposed for 3D modeling from multiple images, they remain very difficult for novices to use as they require image calibration and other advanced techniques. It is difficult to construct a faithful 3D model from a single image, but the results here indicate the feasibility of practical

measurement using our method.

This research shows that an amateur user can design a new shape-matching object (e.g., a toy) based on existing objects using a computer. This is very helpful for non-professional users seeking to design functioning real-world objects. We plan to apply this approach to various other problems. As an example of extension in 3D usage, a user may wish to perform design to replace a broken handle on an existing teapot.

ACKNOWLEDGMENTS

We would like to thank Prof. Jun Mitani of Tsukuba University, Japan, for providing the source code for DXF export. We are also grateful to the participants of the user study, and to the JST ERATO Igarashi Design Interface project for providing the laser cutter used here. This work was partially supported by a Grant-in-Aid for JSPS Fellows.

REFERENCES

- [1] Commax Co., Ltd., *Commax Laser System Value Direct 7050-60W*.
- [2] GRAPHTEC, *CraftROBO*.
<http://www.graphteccorp.com/craftrobo/>.
- [3] Pololu Robotics & Electronics, *Custom Laser Cutting*.
<http://www.customlasercutting.com/>
- [4] Adobe Illustrator.
- [5] Carlson, S., *Sketch based coding of gray level images*. Signal Processing, Vol. 15, pp. 57-83 (1988).
- [6] Elder, J. H. and Goldberg, R. M., *Image Editing in the Contour Domain*. IEEE Trans. on Pattern Analysis and Machine Intelligence, **23** (3), pp. 291-296, 2001.
- [7] Bradski, G., Kaehler, A. (2008), *Learning OpenCV: Computer Vision with the OpenCV Library*.

ABOUT THE AUTHORS

1. Yuki IGARASHI is a Ph.D student at the University of Tokyo and a JSPS research fellow. Contact details: yukim@acm.org; Graduate School of Engineering, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, 153-8904 Tokyo, Japan
2. Hiromasa SUZUKI Dr.Eng. is a professor at the University of Tokyo, Japan. Contact details: suzuki@den.rcast.u-tokyo.ac.jp; Graduate School of Engineering, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, 153-8904 Tokyo, Japan