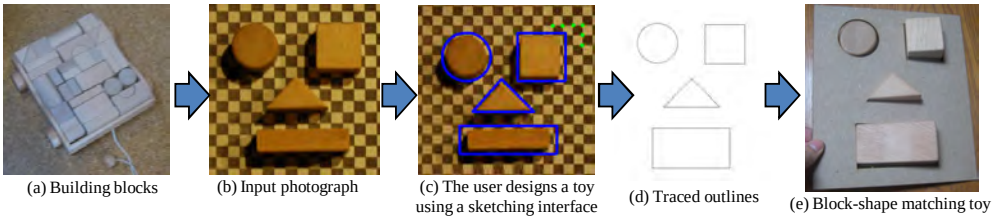
Designing a New Toy to Fit Other Toy Pieces
- A shape-matching toy design based on existing building blocks -

Yuki Igarashi*
University of Tsukuba / JSPS Research Fellow

Hiromasa Suzuki
The University of Tokyo



(a) Building blocks (b) Input photograph (c) The user designs a toy using a sketching interface (d) Traced outlines (e) Block-shape matching toy

Figure 1: *Overview of our method*

1 Introduction

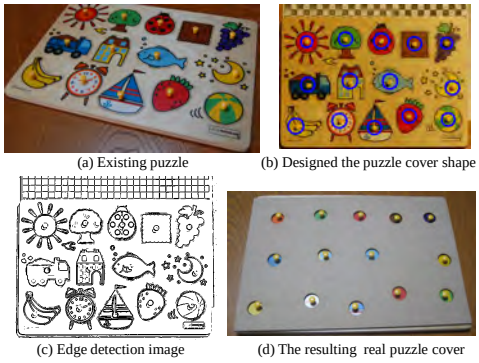
Shape-matching toys are popular items for infants, and consist of boxes with many holes in different shapes along with corresponding blocks of the same shapes. To play with the toy, an infant finds and inserts a block matching the shape of a particular hole. It is difficult to design new shape-matching toys based on existing blocks. We assume that the user performs such design as shown in Fig. 1 (e) based on existing building blocks like those shown in Fig. 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) or a cutting plotter (e.g., Craft ROBO). 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.

We propose a computer-based method for novices to enable the design of a new toy to fit another existing toy. The method involves designing a construction diagram for new toy on a photograph of the existing toy using a computer (Fig. 1 b). The user first takes a photograph of the existing toy on a checkerboard, then designs the form of the new toy on the photograph using a sketching interface. To take the photograph on the checkerboard, the system automatically fits the designed toy to the real 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 cutter plotter or laser printer.

2 System Overview

Figure 1 shows a real toy made using the proposed method. Our goal is to design a new shape-matching toy (Fig. 1 e) from existing building blocks (Fig. 1 a). The user first prints a checkerboard and takes a photograph of the existing building blocks $O_{existing}$ on it. The user draws free-form strokes on the input photograph shapes (line, rectangle and ellipse), and exports the user-input strokes in vector form as shown Fig. 1(d).

The checkerboard is used for scale adjustment. Although various methods have been proposed for feature point extraction in 2D [Carlson 1988; Bradski and Kaehler 2008], no method currently enables precise extraction. Accordingly, we adopt manual input by the user. The system first extracts the grid of the checkerboard, and the user can easily mark the corners of the grid using the extracted edge detection image (Fig. 2 c). The user has to mark corners at more than two unbroken points (Fig. 2 b). The system then calculates the ratio of the image to the actual size. We applied the method to the design of a new toy using existing real toys (Fig. 1 and 2).



(a) Existing puzzle (b) Designed the puzzle cover shape (c) Edge detection image (d) The resulting real puzzle cover

Figure 2: *Example of a puzzle cover design based on an existing puzzle*

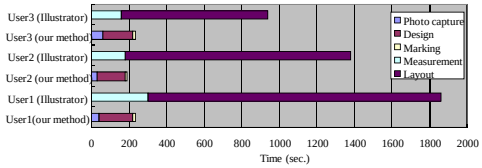


Figure 3: *Comparison of timing data*

3 Evaluation

We tried the system with three test-users who designed a shape-matching toy as shown in Fig.1. Figure 3 shows a comparison of our method and existing software (Adobe Illustrator). In our method, the user first takes a photograph of the building blocks on a checkerboard, then draws strokes to create a diagram for the shape-matching toy on the photograph and marks its corners. In the existing method, the user first measures the building blocks and draws a construction diagram for the shape-matching toy. It is only necessary to mark $5-6$ points from the results of the user study. It took about $10-13$ seconds, and the test users reported that they found the system easy to use.

References

BRADSKI, G., AND KAEHLER, A., 2008. Learning openCV: Computer vision with the openCV library.

CARLSON, S. 1988. Sketch-based image coding of gray-level images. *Signal Processing 15*, 57 – 83.

*e-mail:yukim@acm.org