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

Yuki IGARASHI[1] and Hiromasa SUZUKI[2]
[1]The University of Tokyo, Japan / JSPS research fellow
[2]The University of Tokyo, Japan

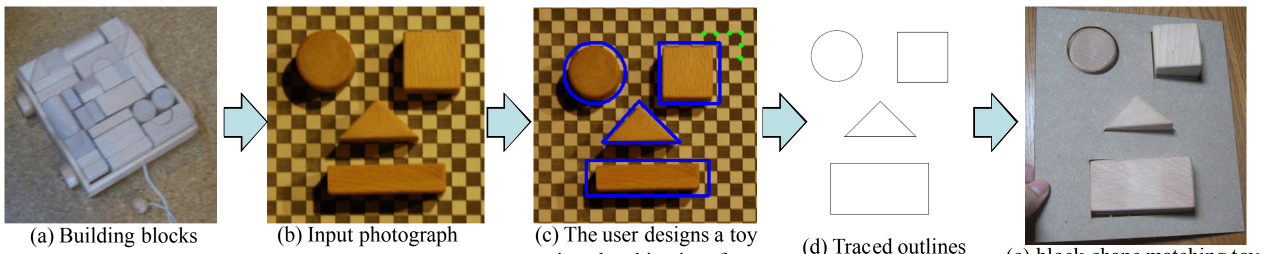

(a) Building blocks
(b) Input photograph
(c) The user designs a toy using sketching interface
(d) Traced outlines
(e) block-shape matching toy

Figure 1: Overview of our method

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 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.

System Overview Figure 1 shows a real toy made using the proposed method. Our goal was to design a new shape-matching toy (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.

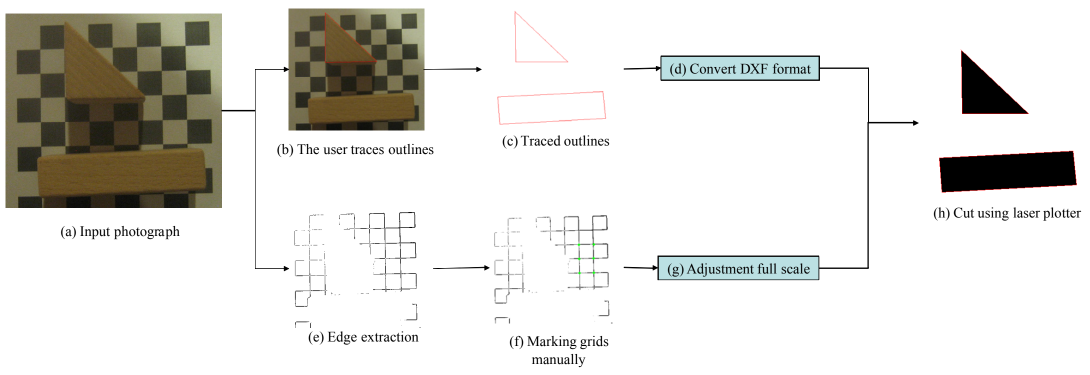

(a) Input photograph
(b) The user traces outlines
(c) Traced outlines
(d) Convert DXF format
(e) Edge extraction
(f) Marking grids manually
(g) Adjustment full scale
(h) Cut using laser plotter

Figure 2: Our algorithm

The user draws free-form strokes on the input photograph using the sketching interface (Fig. 2 (b)). The system also provides some basic shapes (line, rectangle and ellipse), and exports the user-input strokes in vector form as shown Fig. 2 (c).

The checkerboard is used for scale adjustment. Although various methods have been proposed for feature point extraction in 2D [1], [2], [3], 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 (e)). The user has to mark corners at more than two unbroken points (Fig. 2 (f)). The system calculates the ratio of the image to the actual size (Fig. 2 (g)). We applied the method to the design of a new toy using existing real toys (Figs. 1 and 3).

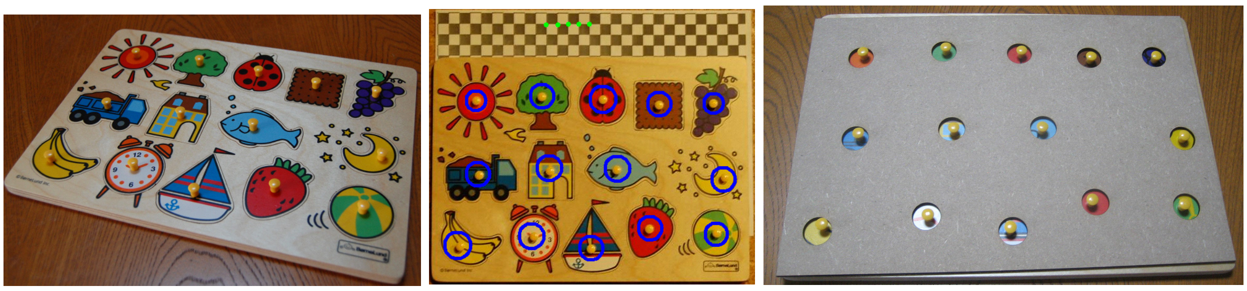
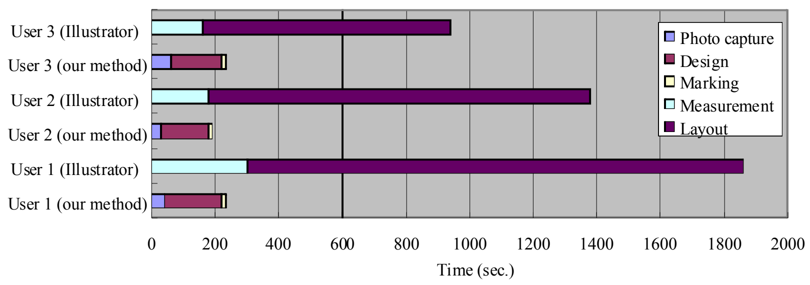


Figure 3: Example of a puzzle cover design based on an existing puzzle

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 our method and 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 O_{new} on the photograph and marks its corners. In the existing method, the user first measures the building blocks of $O_{existing}$ and draws a construction diagram for the shape-matching toy O_{new}. It is only necessary to mark 5 – 6 points from the results of the user study as shown in Fig. 4. It took about 10 – 13 seconds, and the test users reported that they found the system easy to use.

Graph 1: Comparison of timing data



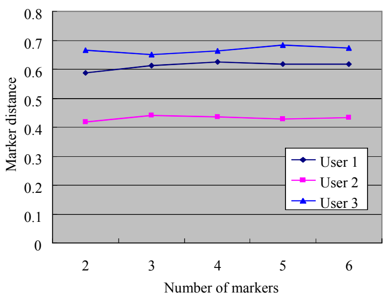


Figure 4: Number of markers

Keywords: design, user interface, craft

References

[1] S. Carlson, “Sketch-based image coding of gray-level images,” Signal Processing, Vol. 15, pp. 57 – 83, 1988.
[2] J. H. Elder, R. M. Goldberg, “Image Editing in the Contour Domain,” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 23, No. 3, pp. 291 – 296, March 2001.
[3] Bradski, G., Kaehler, A. (2008), Learning OpenCV: Computer Vision with the OpenCV Library.