Computer-Generated Papercutting

Jie Xu   Craig S. Kaplan
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{jiexu,csk}@cgl.uwaterloo.ca

Xiaofeng Mi
Department of Computer Science
Rutgers University
Piscataway, NJ, USA 08854-8019
xmi@cs.rutgers.edu

Abstract

The craft of papercutting is part of the folk art traditions of cultures all over the world. From the point of view of computer graphics, papercutting can be seen as a method of composing bi-level images under a set of geometric connectivity constraints. In this paper, we present a technique for composing digital paper-cut designs. The elements of a design may be images, which are processed via a multi-layer thresholding operation, or they may be procedurally-generated arrangements of shapes. Elements are composed using a set of boolean operators that preserve connectivity. The resulting designs are well suited to being cut by a new generation of inexpensive computer peripherals.

1. Introduction

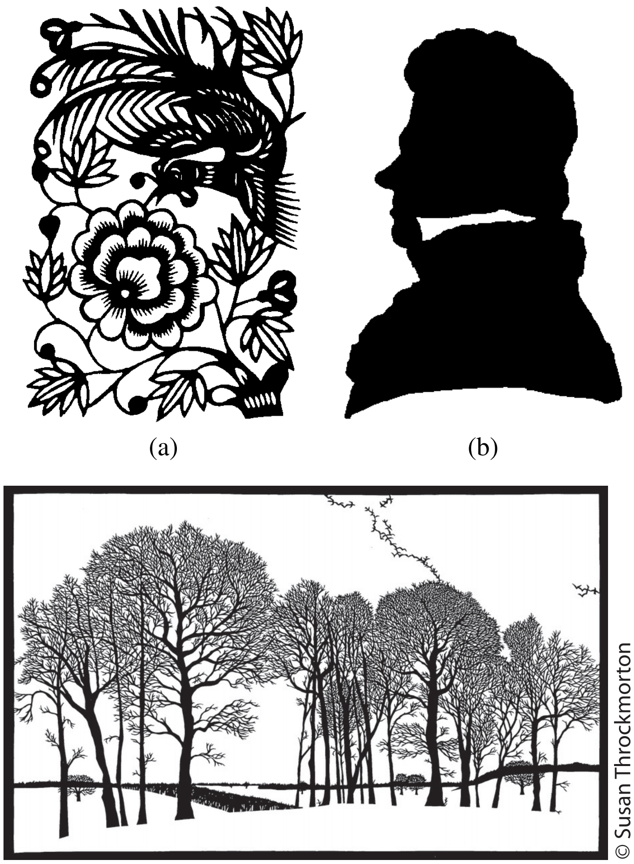
Papercutting, which originated in China 2000 years ago, is today part of the folk art traditions of cultures all over the world [8]. It is still a popular decorative art in China [12] (where it is known as Jianzhi), and is practised in distinct styles in Japan (Kirigami), Germany (Scherenschnitte), Poland (Wycinanki), Mexico (Papel Picado) and Jewish culture. In some of these cases, designs play a symbolic role in rituals or festivals; other uses are more purely decorative or artistic.

As a loose collection of traditions, there is no single recipe for constructing paper-cut designs. But if we restrict ourselves to a subset of all human-made examples, we begin to see some recurring mathematical features. In this work we consider the common case where the design is a connected shape formed by cutting holes into a piece of paper. If the image being depicted is represented by the paper itself, we call it a "positive paper-cut"; the image may also be represented by the holes left in the paper, which we call a "negative paper-cut". In either case the design is a connected subset of the plane which, except for the simplest silhouettes, will contain holes. Such shapes, which can feasibly be cut from a piece of paper, will be called "valid paper-cut designs".

Figure 1. A paper-cut design based on the Stanford Dragon model, created using the technique presented in this paper.

Motivated by this simple mathematical description, in this paper we examine the problem of constructing valid paper-cut designs with computer assistance. In particular, we develop a set of tools for constructing simple designs, each of which is a valid paper-cut, and then give a set of binary operations that allow paper-cuts to be combined while preserving validity. In our system, the user has access to high-level controls for creating and combining designs, and the computer handles the geometric details. An example of a design created using our system is given in Figure 1.

We derive primitive paper-cut designs from two sources. In Section 3, we show how to derive designs from images, and in Section 4 we discuss techniques for synthesizing ge-



(a) (b) (c)

Figure 2. Three examples of papercutting: a traditional phoenix and flower design in (a), a silhouette portrait in (b), and a modern paper-cut by Susan Throckmorton in (c).

ometric and ornamental designs from scratch. We discuss our validity-preserving boolean operators in Section 5.

2. Related work

In Figure 2, we show just a few examples from thousands of years of papercutting tradition. The traditional Chinese design in (a) is used to express wishes of good fortune for the new year. Cut-out silhouettes like those in (b) enjoyed significant popularity as a style of portraiture in 18^{th} century Europe. Today, artists such as Susan Throckmorton, whose work is shown in (c), produce beautiful and highly detailed paper-cut scenes that are excellent examples of both art and craft [20].

The use of computers to design cutting patterns for paper has become a popular research topic, driven in part by the simplicity and ready availability of computer-controlled cutting tools. Previous research has investigated papercraft sculpture [15], origami architecture [14] and pop-up books [3]. Within the art-math community, paper has been cut into interlocking grids depicting mathematical shapes [19] and polyhedral sculptures [6].

In computational geometry, some researchers have investigated the algorithmic aspects of cutting paper. Demaine et al. showed that a single straight-line cut can remove any shape from a suitably folded piece of paper [2].

On the other hand, very little research in computer science has addressed the problem of papercutting in the traditional folk art sense. The cut-out Islamic star patterns of Kaplan and Salesin [9] can be seen as a family of purely geometric paper-cut designs. Liu et al. [13] studied the cyclic and dihedral symmetries of different annuli in paper-cut designs, and showed how to synthesize new designs with different rotational orders. Recently, Li et al. [11] presented a design tool for annotating animated 3D surfaces with holes derived from traditional papercutting motifs.

Insofar as the designs we study in this paper are binary images, we can see papercutting as a form of halftoning. Existing research in non-photorealistic halftoning, such as pen-and-ink rendering [22], might therefore be seen as related. On the other hand, even if a pen-and-ink algorithm might be constrained to produce a connected final result (for example, the maze designs of Xu and Kaplan [24] are connected by construction), such designs are likely too detailed to be cut by a human, and possibly even by a machine. In our study of traditional papercutting, we find very few examples with hatching or any other form of continuous tone reproduction. Papercutting tends to use a more stylized representation of shape.

Gooch et al. showed how to use a difference-of-Gaussians method to create convincing black-and-white facial illustrations [4]. These illustrations could easily be transformed into paper-cut designs. Xu and Kaplan created halftoned depictions of images by assembling deformed letters [23]. Using a stencil font in their system would guarantee the validity of the resulting packing.

3. Image-based paper-cut designs

We would like to have access to images as a source of paper-cut designs. As was mentioned in Section 2, there are many algorithms for converting continuous-tone images to black-and-white. For the purposes of papercutting, we must impose the additional constraint that the resulting halftoned representation be connected. In this section we therefore propose a two step conversion process: we threshold images, and then run an image-based algorithm to enforce connectivity.

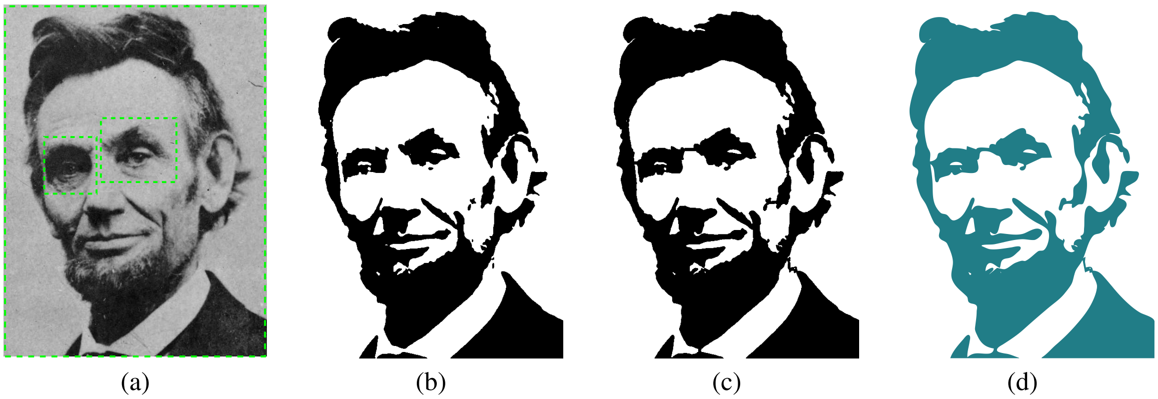


Figure 3. Multilayer thresholding applied to a portrait of Lincoln. The green rectangles in (a) outline three layers chosen by the user. The contents of the rectangles are thresholded and merged to form the black and white image in (b). Additional pixels are coloured in to force all regions to be connected in (c). The final vectorized paper-cut design in shown in (d).

3.1. Multilayer thresholding

In our interactive application, we first provide the user with a standard set of tools for foreground extraction, based on lazy snapping [10] and intelligent scissors [16]. We then blur the foreground object to remove high frequencies.

We are now ready to threshold the image. Given an interval of intensities (chosen by the user), we set every image pixel to black if its value lies in that interval, and white otherwise. In our case, we compute the value of a pixel as the maximum of its red, green, and blue components.

For many real-world images, a single threshold interval does not provide a satisfactory result. Therefore, we allow the user to decompose the image into layers and assign separate intervals to each layer. The individual thresholded layers can then be merged together using the boolean operators AND, OR, or XOR. This procedure is illustrated in Figure 3. Additionally, we provide a layer in which we place the result of edge detection on the original image. Image edges can help fill in outlines that disappear during thresholding (see Figure 4).

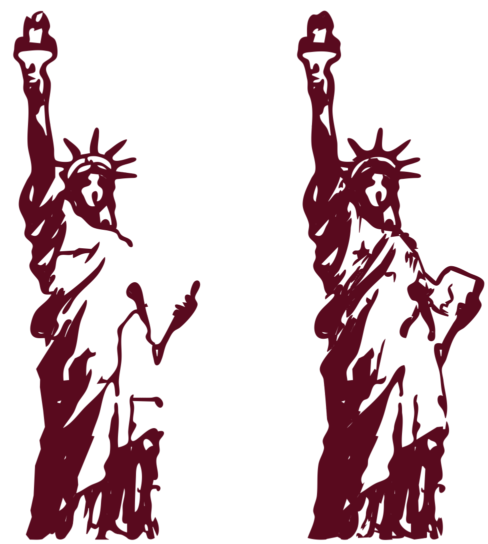


Figure 4. A demonstration of adding detected edges to better depict object outlines.

3.2. Enforcing connectivity

The thresholding operation can produce many disconnected regions. We use a simple image-based algorithm to colour additional pixels black in order to yield a single connected component.

We first compute all the connected components in the thresholded image, and identify the smallest one. To create a valid paper-cut, we will need to connect this component to some other component via a path of pixels. We would like this path to be as dark as possible (measured as the sum of the values of its pixels). We can find such a path by running Dijkstra's algorithm outward from every pixel on the boundary of the component, using pixel values as edge weights. We treat the component as a single source vertex connected to its boundary pixels, and stop the algorithm as soon as we encounter a pixel on the boundary of another component. Figure 5(a) shows an example of a path discovered using this search.

Of course, a one-pixel-wide path is unattractive, and probably too narrow to be practical in a papercutting con-

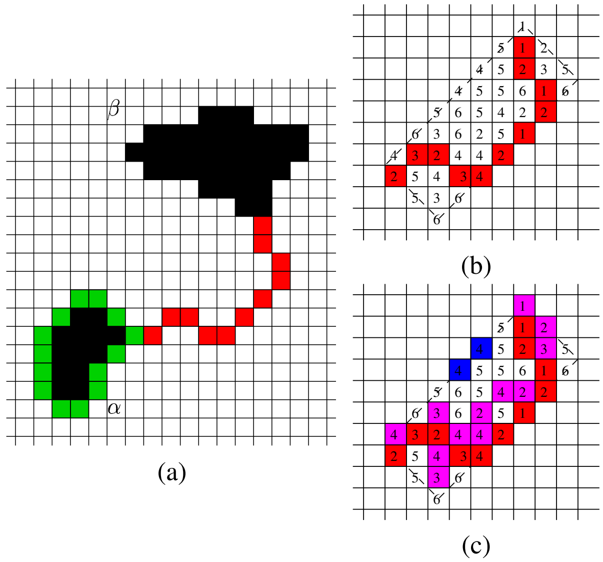


Figure 5. In (a), two components α and β are connected by a path of red pixels from the boundary of α (in green) to β. The path is surrounded by an oriented bounding box in (b). The numbers are the intensities of the corresponding pixels in the source image. We threshold all pixels inside the bounding box with the maximum intensity along the path (4 in this example). The result is shown in (c). The pixels connected to the path (shown in magenta) are added. The disconnected pixels (shown in blue) are discarded.

Figure 6. Positive (a) and negative (b) paper-cut designs of an image of a chrysanthemum. In both cases, the red paper area is connected.

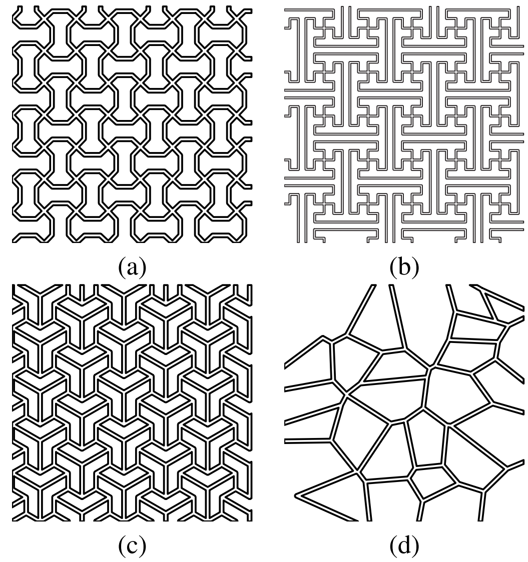


Figure 7. Paper-cut designs based on simple geometric patterns. Isohedral tilings are shown in (a), (b), and (c); the latter two were taken from Chinese papercutting and latticework. A Voronoi diagram is given in (d) (patterns similar to Voronoi diagrams occasionally appear in Chinese latticework).

text. We therefore provide a means of thickening the path based on intensities in the source image. We compute an oriented bounding box from the 2D covariance matrix of all pixels on the path, as in the stroke analysis method of Barla et al. [1, Section 2.1]. The two eigenvectors of this matrix are the axes of the bounding box, and we project all pixels to these axes to get the box's dimensions (see Figure 5(b)). We define a threshold I as the lightest intensity of the source image pixels on the path. We then set every pixel in the bounding box to black if it is darker than I, and white otherwise. Finally, we keep those black pixels that belong to the same connected component as the path itself, and discard any others (see Figure 5(c)). This process can be seen as locally relaxing the thresholding operation to let in more pixels. In practice we have found that it adequately thickens paths, making satisfactory connections between components.

The pathfinding step merges the smallest component with some other component. We repeat this process until there is only one connected component left. We fill small holes introduced during thresholding and pathfinding by applying two rounds of the morphological Close operation, and then use the well known Autotrace library [21] to recover vector paths for the design.

Note that if we invert the binary image before computing connectivity, we can construct a negative paper-cut design instead of a positive one. Connectivity can then be enforced on the negative binary image, as before. Figure 6 shows a comparison of positive and negative designs.

Figure 8. Predefined ornamental patterns, based on conventionalized motifs from the Chinese papercutting tradition.

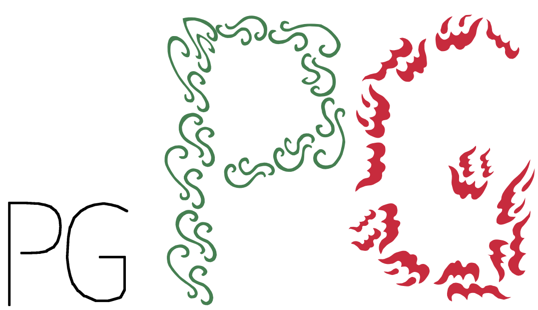


Figure 9. An example of synthesizing the WATER **and** FIRE **patterns along two letterforms.**

4. Pattern synthesis

Traditional paper-cuts frequently feature decorative ornament and stylized or geometric patterns. We support an extensible set of patterns that can be combined with each other and with image-based designs.

Geometric patterns reminiscent of latticework are a popular papercutting device. Aside from their decorative function, they provide a connected substrate into which other objects can be embedded. We support a wide range of geometric patterns in the form of isohedral tilings of the plane [5]. The edges in any tiling can be thickened to produce a valid paper-cut design. Figure 7 shows several of our geometric patterns, inspired by traditional Chinese designs. There are many other potential sources of geometric designs for papercutting, such as Islamic star patterns [9].

We also support the synthesis of freeform arrangements of stylized ornamental motifs. We have experimented with a small set of conventionalized motifs inspired by Chinese papercutting (see Figure 8), though of course many others are possible. In our system, the user selects a pattern type and draws a stroke. We use the stroke pattern synthesis method of Barla et al. [1] to place motifs along the stroke, and deform the motifs to fit the stroke's path in a manner similar to skeletal strokes [7]. The user can also control the density and sizes of motifs placed along strokes. An example of our stroke-based pattern synthesis is shown in Figure 9.

Note that synthesized patterns are generally disconnected, and hence not valid paper-cut designs. When necessary, we consider a pattern to be a set of holes cut out from a sufficiently large rectangle.

5. Compositing Paper-cuts

The techniques of the previous two sections can be used to construct a wide variety of paper-cut designs. We would also like to be able to combine individual designs into finished scenes. Here we encounter an interesting mathematical problem: given two valid paper-cut designs, how may they be combined to produce a valid result? In other words, we wish to define a set of binary operations on connected sets that preserve connectivity.

Let A be a valid paper-cut design, i.e., a non-degenerate connected set in the plane. We may regard A as partitioning the plane into three disjoint regions: A itself, A_i, the set of holes contained entirely within the outer boundary of A, and A_o, the rest of the plane. Similarly, a second design B partitions the plane into B, B_i, and B_o (see the top row of Figure 10). When A and B are superimposed, the pairwise intersections of these sets partition the plane into nine regions. Any binary operation on A and B can be specified by assigning a boolean value to each of these nine regions – **true** if the region is part of the result, and **false** if it is not. Therefore, there are 512 possible operations. We can limit the possibilities by assuming that operations do not add paper that did not originally belong to A or B. In other words, the regions A_oB_o, A_oB_i, A_iB_o and A_iB_i must be set to **false**, leaving us with five boolean choices and 32 operations.

For many operations, it is natural to assume additionally that AB is non-empty, for otherwise it would be very difficult to enforce connectivity in the result. Working through the 32 possible operations under this additional assumption, we find eight that guarantee validity. Inspired by the compositing operators introduced by Porter and Duff [18], we summarize our operations in the table below. We name each one and define it in terms of the boolean values for the five regions. Figure 10 gives illustrations for each.

	AB	AB_i	AB_o	A_iB	A_oB
A	T	T	T	F	F
B	T	F	F	T	T
A UNION B	T	T	T	T	T
A AND B	T	T	F	T	F
A OVER B	T	T	T	F	T
A UNDER B	T	F	T	T	T
A WITHIN B	T	T	F	T	T
A WITHOUT B	T	T	T	T	F

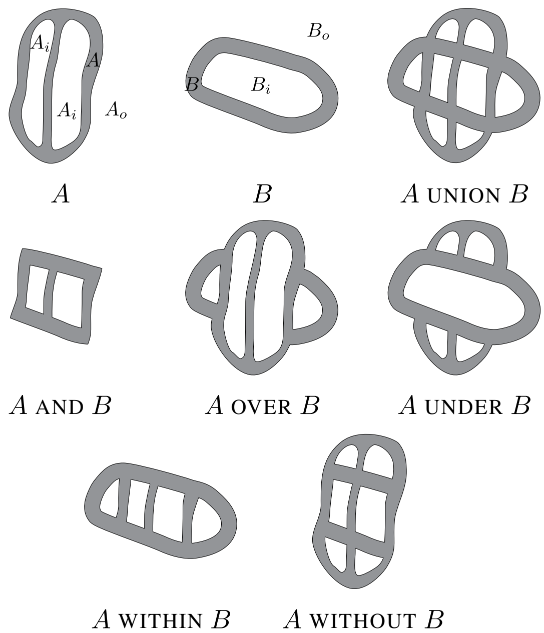


Figure 10. The eight binary operations on paper-cut designs that preserve validity, as explained in Section 5.

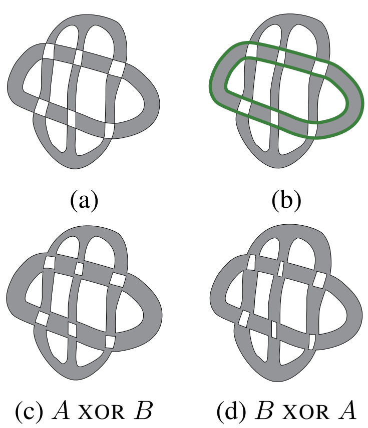


Figure 11. The XOR operation on paper-cut designs. The simple XOR in (a) does not preserve validity. If we thicken the edges of B as in (b), we can restore validity as in (c). The design in (d) thickens the edges in A instead.

Based on observations of traditional paper-cut designs, we would also like to support an XOR operation, in which the intersection of the two shapes is cut out. This operation allows two shapes to coexist in the same space, with the edges of both discernible. Unfortunately, XOR is not

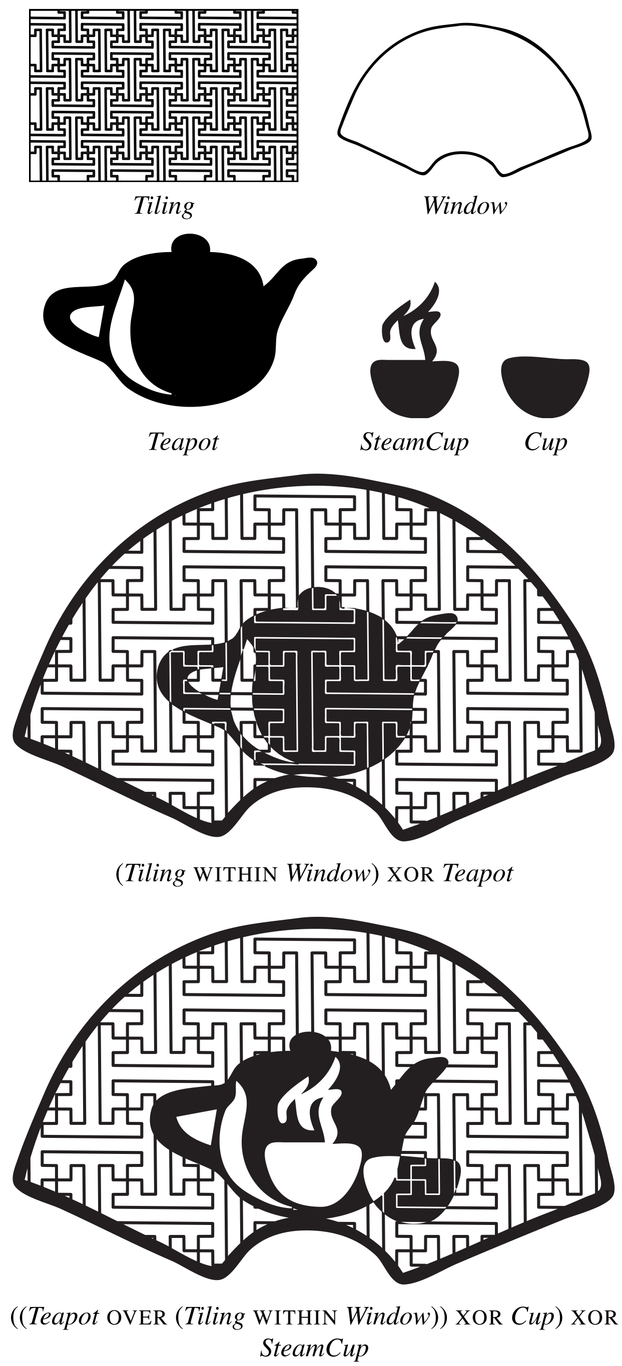


Figure 12. A demonstration of boolean operations on paper-cut designs. Five source designs are shown in the top two rows. In the first finished design, the XOR operation is used to make the teapot appear to be behind the screen. The teapot is placed OVER the screen in the second design, and a couple of cups are then added with XOR.

guaranteed to produce connected results. We can fix this by thickening the edges of one of the shapes so that it overlaps the second, as shown in Figure 11. Because we can choose to thicken the edges of A or B, our XOR operation is asymmetric. Figure 12 shows an example of composing a complex paper-cut design from a set of primitive elements using the eight valid operations and XOR.

Many other set-theoretic operations, such as $A \cap B$ and $A \setminus B$, produce correct results in some contexts but not others. For instance, we can cut a pattern of holes into a shape A by intersecting A with a large rectangle containing the holes, as mentioned in Section 4. We permit the user to perform these "unsafe" operations, but must check whether the result is connected.

6. Results and future work

We have created a prototype implementation of our technique using C++ and Gtkmm. We represent paper-cut designs as polygons with holes, and use the GPC library [17] to compute boolean operations on them. Our interface lets the user process images, create stylized patterns, and perform boolean operations on designs. The output is a PDF file of cutting paths. The PDF can be displayed directly by filling the paths, or it can be used to create actual paper-cuts using a variety of computer-controlled manufacturing devices. We have experimented with the QuicKutz Silhouette digital craft cutter, a small and very inexpensive knife cutter intended for home use (see Figure 13(d)). Finer and more precise results could be obtained using a laser cutter. Some results are shown in Figures 13, 14, and 15.

There are several directions we would like to explore in the future. Many traditional paper-cuts feature a central design surrounded by annuli with different cyclic or dihedral symmetries [13]. We would like to automate the construction of symmetric designs.

It would also be interesting to extract more information from images when transforming them into paper-cut designs. It might be possible to decompose an image into overlapping regions that can then be assembled using XOR. Another challenge would be to automatically select patterns from the library to approximate details in the image, or better yet to develop conventionalized vector patterns directly from image features.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We are also very grateful to Kwan-Liu Ma for providing a copy of his paper, to Susan Throckmorton for permission to print the paper-cut design in Figure 2(c).

References

[1] P. Barla, S. Breslav, J. Thollot, F. Sillion, and L. Markosian. Stroke pattern analysis and synthesis. In *Computer Graphics Forum (Proc. of Eurographics 2006)*, volume 25, 2006.

[2] E. D. Demaine, M. L. Demaine, and A. Lubiw. Folding and one straight cut suffice. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 891–892, Baltimore, Maryland, January 17–19 1999.

[3] A. Glassner. Interactive pop-up card design, part 1. *IEEE Computer Graphics and Applications*, 22(1):79–86, 2002.

[4] B. Gooch, E. Reinhard, and A. Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph.*, 23(1):27–44, 2004.

[5] B. Grünbaum and G. C. Shephard. *Tilings and patterns*. W. H. Freeman & Co., New York, NY, USA, 1986.

[6] G. W. Hart. Modular kirigami. In *Proceedings of Bridges 2007*, 2007. See also `http://www.georgehart.com/kirigami/Hart-Modular-Kirigami.html`.

[7] S. C. Hsu and I. H. H. Lee. Drawing and animation using skeletal strokes. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 109–118, New York, NY, USA, 1994. ACM Press.

[8] R. Jablonski. *The Paper Cut-Out Design Book*. Stemmer House Publishers, Inc., Owings Mills, Maryland, 1976.

[9] C. S. Kaplan and D. H. Salesin. Islamic star patterns in absolute geometry. *ACM Trans. Graph.*, 23(2):97–119, 2004.

[10] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[11] Y. Li, J. Yu, K.-L. Ma, and J. Shi. 3D paper-cut modeling and animation. *The Journal of Computer Animation and Virtual Worlds*, 18:to appear, 2007. (Also published in Proceedings of CASA 2007).

[12] C. Liang. *The Techniques of Traditional Chinese Papercutting*. Gansu People's Publishing House, Lanzhou, Gansu, China, 2006.

[13] Y. Liu, J. Hays, Y.-Q. Xu, and H.-Y. Shum. Digital papercutting. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 99, New York, NY, USA, 2005. ACM Press.

[14] J. Mitani and H. Suzuki. Computer aided design for Origamic Architecture models with polygonal representation. In *Proceedings of Computer Graphics International 2004*, pages 93–99. IEEE Press, 2004.

[15] J. Mitani and H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 259–263, New York, NY, USA, 2004. ACM Press.

[16] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. pages 191–198. ACM Press, 1995.

[17] A. Murta. GPC: General Polygon Clipper library. `http://www.cs.man.ac.uk/~toby/alan/software`, 2007.

[18] T. Porter and T. Duff. Compositing digital images. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, New York, NY, USA, 1984. ACM Press.

[19] J. Sharp. *Sliceforms: Mathematical Models from Paper Sections*. Tarquin, 1999.

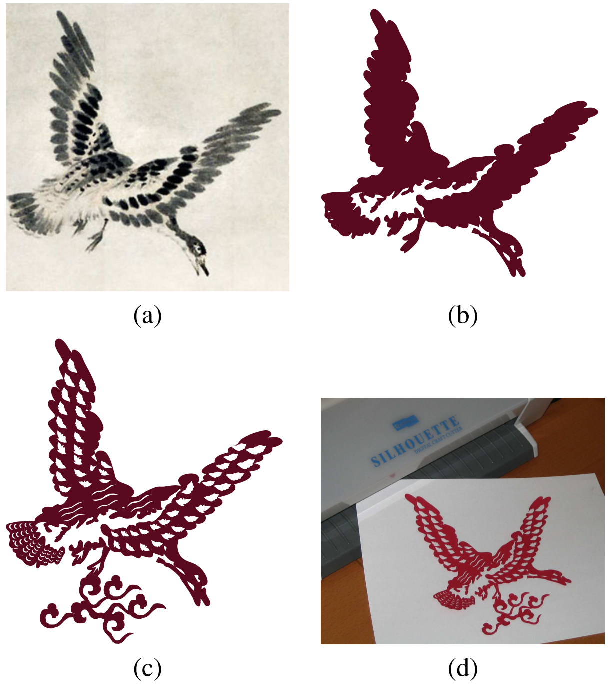


Figure 13. The construction of a paper-cut based on a painting of a goose by Zhu Da (c.1626 – c.1705). The original image is shown in (a). Our image-based technique extracted the design in (b), to which some synthesized patterns were added in (c). In (d), we show the design cut from cardstock using an inexpensive computer-controlled craft cutter.

[20] S. L. Throckmorton. Wycinanki: original papercuttings by Susan L. Throckmorton. `http://www.papercuttings.waw.pl`, 2007.

[21] M. Weber. Autotrace - converts bitmap to vector graphics. `http://autotrace.sourceforge.net/`, 2007.

[22] G. Winkenbach and D. H. Salesin. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100, New York, NY, USA, 1994. ACM Press.

[23] J. Xu and C. S. Kaplan. Calligraphic packing. In *GI '07: Proceedings of the 2007 conference on Graphics Interface*, pages 43–50, 2007.

[24] J. Xu and C. S. Kaplan. Image-guided maze construction. In *SIGGRAPH 2007: Proceedings of the 34th annual conference on Computer graphics and interactive techniques*, 2007. to appear.

Figure 14. A paper-cut design based on the Big Wild Goose pagoda (Xi'an, China).

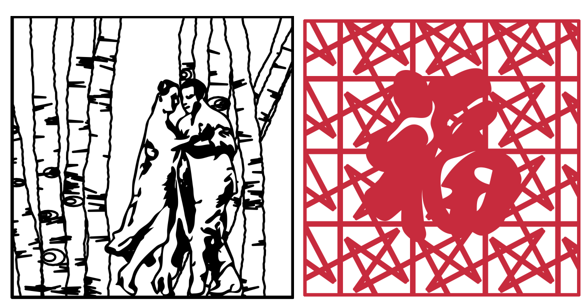


Figure 15. Two paper-cuts. The left design is a couple dancing among procedurally generated trees, and the right is the Chinese character "Fu" (good fortune) embedded in a geometric pattern.