Interactive Pop-Up Card Design

Andrew Glassner

Microsoft Research Technical Report TR 98-03
16 January 1998
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

Pop-up cards are fun to make and receive. Traditionally they have been designed by a process of trial and error, requiring the creator to repeatedly draw, cut, glue, and test each piece until it is correctly positioned and moves as desired. The process can quickly become complicated and ultimately frustrating as the card increases in complexity. To address this problem, we have developed stable analytic solutions for the locations of important points as the card folds and unfolds. We give the essential geometry for the three important pop-up techniques of single-slit, asymmetric single-slit, and V-fold mechanisms. We have implemented our solutions in a small design program.

Keywords: Origami, Pop-up, Constraints

1 Introduction

Pop-up cards are enjoyable 3D constructions that change shape when a page or card is opened. They are fun to design, make, and share. Generally the construction techniques are easy, and children armed with blunt scissors and non-toxic glue can make their own. Adults enjoy pop-up cards too; Siggraph used a pop-up in its promotional materials for the 1995 conference.

Unfortunately, designing a pop-up and then executing that design is much harder. The problem is that finding the right shapes for all the pieces is usually a process of trial and error: the designer cuts out pieces, folds and glues them, waits, and then tests the result. If it doesn't look quite right, the next step is to cut new pieces with slightly different shapes, fold, glue, wait, and try out the result again. That's a lot of cutting, gluing, and waiting, which can add up to boredom and frustration. A typical card-in-progress is shown in Figure 1a; this is the tenth version of that design (the final version was the fourteenth). An interactive system could relieve this tedium and provide the opportunity for immediate feedback and responsive design, potentially improving the interactive process.

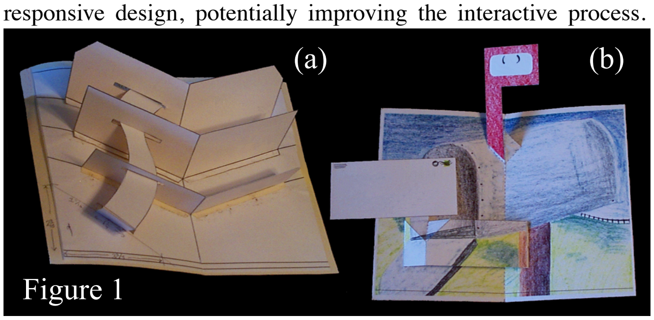

Figure 1

One approach to building such a system would be to use constraint systems [6]. But constraint systems are complex, and can be slow and numerically delicate. The geometry of pop-up cards is structured enough to let us find analytic solutions which are fast, easy to code, and robust.

There are many possible constructions that move as a card is opened, and they could all be called pop-up designs. In this paper we will restrict ourselves to mechanisms which are based on moving planes driven by an opening pair of pages. These are fundamental pop-up techniques, and the ones that cause the most physical labor through trial and error. The other popular mechanisms [2] are relatively straightforward to lay out and execute.

2 Prior Work

Most prior work on paper techniques has focused on origami [1] and paper-folding [3] [5]. Origami methods generally do not address the motion of pieces as other pieces move, which is central to pop-ups. The only technical publication we have found on pop-up cards is [7]. This work found a geometric invariant among the planes involved in a pop-up mechanism called a V-fold design, but it was not clear how to use that result for interactive design. Popular books on construction techniques abound; two of the best are [2] and [4]. There are also many books available with templates and patterns for cards of different types.

3 Single-Slit Mechanisms

The simplest pop-up is called the *single slit* design; it's the triangular shape at the base of the red mailbox flap in Figure 1b. There are many possible variations on this design, but for our purposes they all boil down to the same thing: as the card closes, two hinged planes rise toward the reader. Figure 2 shows the essential geometry behind all single-slit designs.

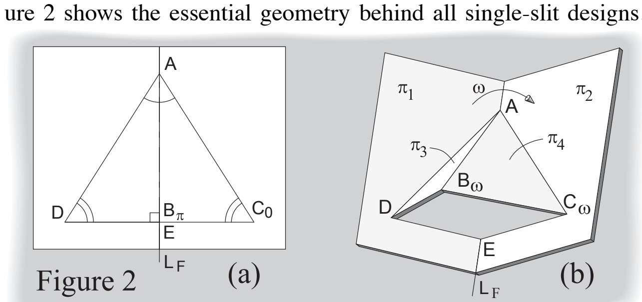

Figure 2

Figure 2 also establishes the conventions we will use in this paper. The two outside planes are called π_1 and π_2; together they form the *card*. Their line of intersection is the crease along which the card is folded; we call that line the *support crease* and label it L_F. The angle between the planes is ω. Planes π_3 and π_4 form the pop-up itself, and move as ω chages. Our convention is that π_1 is held constant while π_2 is moved; thus π_2 rotates around L_F

while π_1 is fixed. This also means that points A, D, and E are fixed, and point C simply rotates around L_F by $\omega - \pi$, the negative complement of card angle ω. We write the position of points B and C for a given fold angle ω as B_ω and C_ω. Thus B_π and B_0 are the positions of B when the card is open and closed, respectively. We want a general expression for arbitrary B_ω. Note that in Figure 2a, E and B_π mark the same point. We call edge AB_ω the *central pop-up crease*, and edges AD and AC_ω the *side pop-up creases*.

Figure 2 shows the simplest form of the single-slit geometry, where the cut DC_0 is perpendicular to L_F. If this cut is at an angle to the fold line, then point B_ω is simply a point on the pop-up edge rather than its terminus.

Note that the single-slit mechanism is at its most extreme position when the card angle is at $\omega = \pi/2$, so we call this a "right-angle" mechanism. When the card is fully open, the pop-up returns to the plane of the card. We now turn to finding B_ω, the location of B as the card folds.

4 Finding B_ω

From Figure 2 and our conventions we see that points A, D, and E are fixed, and C_ω is simply C_0 rotated around L_F by $\pi - \omega$. Given these points, and the original location of B_0, where is B_ω?

Observe that the lengths $|AB_\omega|$, $|DB_\omega|$, and $|C_\omega B_\omega|$ do not change as the card folds; these distances are always the same as when $B_\omega = B_0$. Thus B_ω is simultaneously on the surface of three spheres with these radii and centers A, D, and C respectively.

Three mutually intersecting spheres meet in two common points. The value of those points when plugged into the implicit equation for each sphere is zero. Imagine a plane through the centers of the three spheres, as in Figure 3. Now we have the simpler 2D problem of finding the point M which has the same value with respect to the three circle equations. The line perpendicular to this plane through M (dashed in Figure 3) contains the intersections of the spheres, one of which is B_ω.

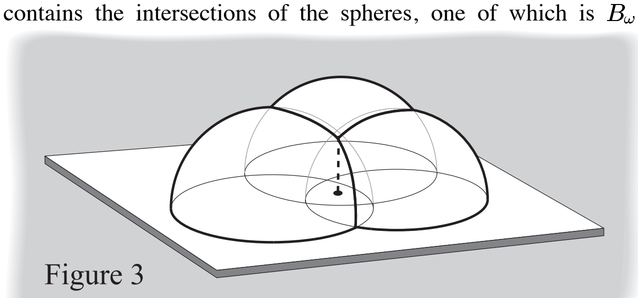

Figure 3

To find M, we first find the locus of all points that have the same value with respect to two circles. The implicit function for a circle C is $C(x, y) = (x - C_x)^2 + (y - C_y)^2 - C_r{}^2$. We want to find all points P on two circles U and V, so $U(P) = V(P)$, or $U(P) - V(P) = 0$. For a generic point (x, y) we find

$$\begin{aligned} 0 &= U(x,y) - V(x,y) \\ &= 2x(V_x - U_x) + 2y(V_y - U_y) + U(0,0) - V(0,0) \\ &= Ax + By + C \end{aligned}$$

Thus all points (x, y) which have the same value with respect to two circles lie on a line, which is called the *radical axis*.

Now suppose that we have three mutually intersecting circles of different radii, labeled U, V, and W, as in Figure 4b. Circles U and V meet in two points, P_{UV} and Q_{UV}. Since both points have the value zero with respect to both U and V, these two points determine the radical axis L_{UV}. Similarly, P_{UW} and Q_{UW} determine the radical axis L_{UW}. Since we've said that the three circles are mutually intersecting, lines L_{UV} and L_{UW} meet at some point M. By construction we have $U(M) = V(M)$ and $U(M) = W(M)$. Thus $V(M) = W(M)$, which means that M also lies on the radical axis L_{VW} formed by points P_{VW} and Q_{VW}. Our conclusion is that if three circles mutually intersect, their three radical axes intersect at the unique point M.

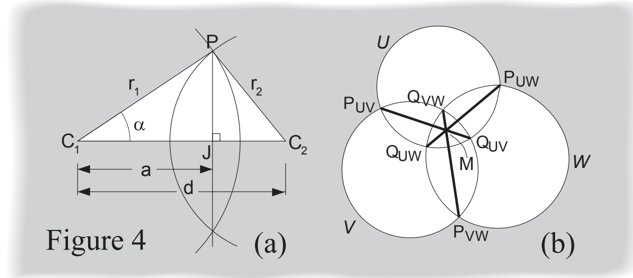

Figure 4 (a) (b)

We find M by creating a plane for each radical axis. The plane for axis L_{UV} contains that line and is perpendicular to the page. Figure 4a shows the basic geometry for finding this plane: we are given circle centers C_1, C_2, their radii r_1, r_2, and $d = |C_1 - C_2|$, and wish to find point J. From ΔPJC_1 we see $a = r_1 \cos\alpha$. To find $\cos\alpha$ we use the law of cosines with triangle ΔC_1PC_2 to find $\cos\alpha = (d^2 + r_1{}^2 - r_2{}^2)/(2r_1 d)$. So $J = C_1 + (C_2 - C_1)(a/d)$. Our plane passes through J with a normal parallel to $C_2 - C_1$. Intersecting any two of these planes gives us the dashed line in Figure 3.

Given the equation of this line, we can intersect it with any one of the spheres to find the two common points. We select for B_ω the point that is on the same side of plane π_1 as point C_ω.

If the circles do not mutually intersect, the radical axes are parallel, and there is no point of intersection [9]. Our circles are never disjoint; the extreme cases are when the card is fully open or closed and the three circles are mutually tangent.

Interactive adjustment of B_ω in the single-slit is limited to the plane that includes L_F and is perpendicular to line $C_\omega D$. Other choices for the designer are to move A, or C_ω and D together. The limits on the motion of B_ω are given by the size of the card (which constrains the other points), and the necessity that the point J found above exists. In other words, the system permits adjustment of B_ω only within the range where the three circles in the plane overlap.

Finding B_ω is the heart of the pop-up simulator. The other mechanisms use either this geometry or simple special-purpose code to position every plane in the card as it folds and unfolds.

5 Asymmetric Single-Slit Mechanisms

An important variant of the single slit is the *asymmetric slit*. Here the fold does not follow the crease of the backing card, but is at an angle to it. Figure 5 shows the essential geometry; Figure 5a is the open card and Figure 5b shows it in closed position. In Figure 5a, the central pop-up crease AB_π forms an angle β to the support crease AE. Although in action the card looks generally like Figure 2b, the central pop-up crease is rotated, creating an asymmetrical pair of triangles on each side. In Figure 5a, we are free to choose A, D, and C_π; we want to find B_π that allows the card to fold flat. In terms of angles, we have ψ, γ, and δ and wish to find α.

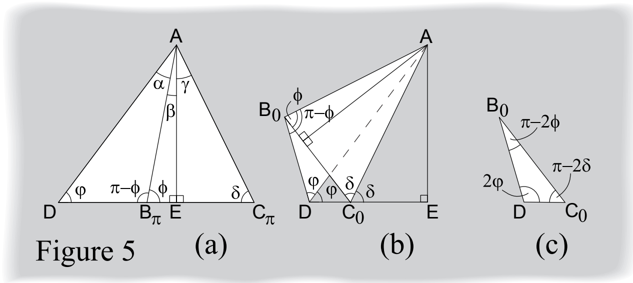

Figure 5 (a) (b) (c)

In Figure 5, we can see that as the card folds, point B_π comes up out of the plane, and eventually comes down to rest at B_0. This causes triangle ΔADB_π to become reflected, since B_ω pulls it around AD. Triangle $\Delta AB_\pi C_\pi$ is pulled along by the motion of B_ω, and comes to rest at ΔAB_0C_0 in an orientation equal to a rotation of γ around A. Because EC_π is perpendicular to the folding axis AE, point C_π moves to C_0 along line DE; this means that triangle ΔAC_0E is similar to triangle $\Delta AC_\pi E$.

To find α, we begin with ΔB_0DC_0 in Figures 5b and 5c, giving $2\psi + (\pi - 2\delta) + (\pi - 2\phi) = \pi$, or $\phi = \psi - \delta + (\pi/2)$.

From ΔADB_π in Figure 5a, write $\alpha + \psi + \pi - \phi = \pi$. With the value for ϕ found above, this becomes $\alpha = (\pi/2) - \delta$. From ΔAEC_π we find that $\delta = (\pi/2) - \gamma$. Combining these last two results, we find our goal: $\alpha = (\pi/2) - ((\pi/2) - \gamma) = \gamma$. Thus, to construct an asymmetric slit pop-up that folds flat, place B_π in Figure 5a so that $\alpha = \gamma$.

6 V-Fold Mechanisms

The *V-fold* mechanism creates a pair of free-standing slanted planes when the card is opened, as shown in Figure 6. The V-fold is one of the hardest pieces to design using paper and scissors, since one indirectly controls how much the plane leans back by changing the angle at the base of the piece when it is cut out; this angle is the V at the bottom of Figure 6a.

Because a V-fold is a separate piece attached to the backing card, it can rise out of the card plane when the card is fully open, unlike the single-slit. Thus the geometry of the V-fold is based on that of the single slit, but allows more flexibility in its design. Though B_ω still locates the central crease, there may be no paper at that point in space. For example, the apex of the fold (point E in Figure 6) need not be included; the shaded "tunnel" region in Figure 6b can be cut out of the card. Figure 6 also shows the small flaps which are scored, bent back, and then glued to the support planes.

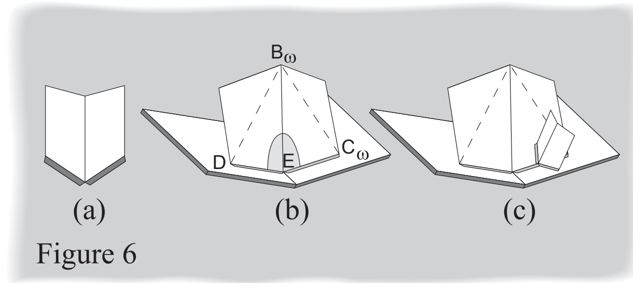

Figure 6 (a) (b) (c)

Since V-folds don't cut into the page, they may be placed on any crease, which is then treated just like the card's crease for that mechanism. Figure 6c shows a cascaded pair of V-folds. The larger one uses the card fold as its support crease, and creates EC_ω as one of its side pop-up creases. The smaller V-fold uses EC_ω as its support crease. So opening the card pops up the big V-fold, which then drives the smaller one to pop up as well.

The tabs of a V-fold must be carefully glued in the right places or the card may not open or close fully. Our system will optionally print out targets on the support planes to indicate where the tabs should be attached.

Depending on the placement of the V-fold on the support planes, it can be designed to fold either towards or away from the reader. When the planes of the V-fold become parallel to the support planes, all the folding lines become parallel to one another [7]. This configuration is sometimes called a *floating layer* [2].

7 Other Mechanisms

The *double-slit* mechanism is based on cutting away a piece of the backing card, and then folding it towards the reader rather than away. Figure 7 shows an example. The new fold is placed so that when $\omega = \pi/2$, the pop-up and the card all form right angles. In the figure, $PQ = RS$.

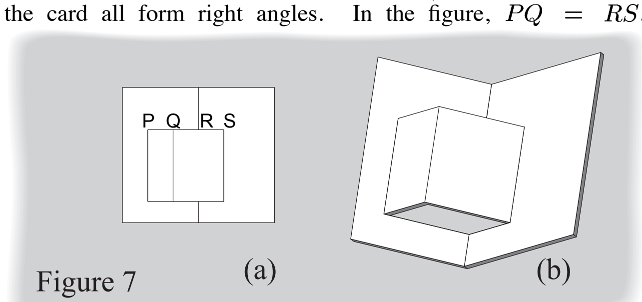

Figure 7 (a) (b)

Layered mechanisms are a variation on the double-slit; rather than being cut out of the backing card, the pieces are built independently and glued in place using tabs. They are still bound by the same requirement that the angles formed are all $\pi/2$ when the card crease is a right angle. Figure 8 shows an example of a layered mechanism using two pieces; here also $PQ = RS$.

As with V-folds, it is important to glue these pieces down in the right place. This usually requires a blend of measuring, marking, and eyeballing; the targets printed by the system guarantee that the card will work correctly.

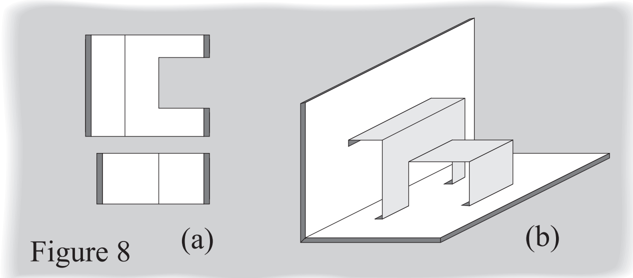

Figure 8

These techniques can be effectively combined with the constructions discussed earlier. Two of the most useful combinations are called the *strap* and the *pivot*, shown together in Figures 1b and 9. The strap is based on either a double-slit, or a layer glued to its support planes. Both the double-slit and the layer are right-angle mechanisms; when the card is fully open these pieces fold down into the layer of the card. The value of the strap is that it allows the designer to displace the central card crease to a parallel line.

In Figure 9, the fold of the strap has been combined with a single-slit mechanism (we don't actually need to cut this slit because we're using the edge of the card). Note that the strap's central pop-up crease rises toward the reader; when the single-slit construction uses this as its support crease, the pyramid created by the single slit folds away from the reader. By attaching another piece to one of the planes of the single-slit, that piece will pivot as the card is opened. Creative designers can create pivot elements that extend far outside of the card when it's opened, yet fold down and tuck away completely out of view when the card is closed.

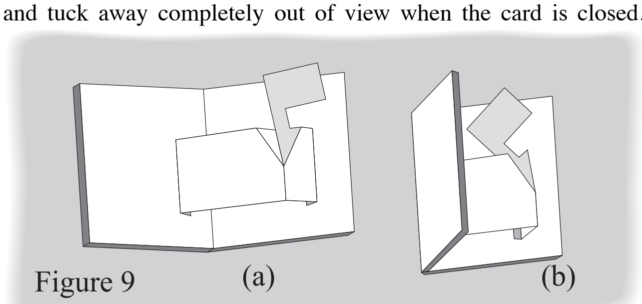

Figure 9

8 Implementation

We have built a small interactive system for the design of pop-up cards. The designer may erect new mechanisms over any fold, including edges formed by other pop-ups. In this way we can produce multi-layered designs. The implementation builds a dependency graph, and follows it from the card fold outward. Due to the physical nature of pop-up cards, this graph is inherently acyclic.

The designer may create mechanisms, open and close the card, and interactively drag points around; the rest of the mechanism is automatically adjusted as necessary. If the designer moves D or C_ω, the other point moves symmetrically. Points A and E may be moved freely along the central pop-up crease. In both types of single-slit techniques, point B_ω may move along the central pop-up crease. V-folds are more flexible, and also allow adjustment of B_ω perpendicular to support planes π_1 and π_2. Sets of points, up to and including entire mechanisms, may be moved at once.

This makes it very easy to solve the problem posed in the Introduction. To cause a V-fold piece to lean back at a shallower angle, simply select any point (such as B_ω) on the central crease, and move it interactively until it looks good.

We track the motion of pieces by representing all planes as triangles; more complex polygons use the triangles as a reference coordinate system. For example, mechanism points A, D, and C_ω are expressed in barycentric coordinates with respect to the support planes in which they lie.

Some designs include a horizon on one or more V-folds; this horizon should appear straight and parallel to the base when the card is open and viewed along a line parallel to the base. This requires a slight adjustment to the placement of the texture on the V-fold, moving the texture along the crease either towards or away from point E. The designer may apply horizon-line correction to the entire card, or only selected pieces.

A fun application of pop-ups is to use a general-purpose stereo algorithm to group objects from one or more photographs into planes. Figure 10 shows a card created with our system using manual segmentation of a single photograph, and the Siggraph logo for a background.

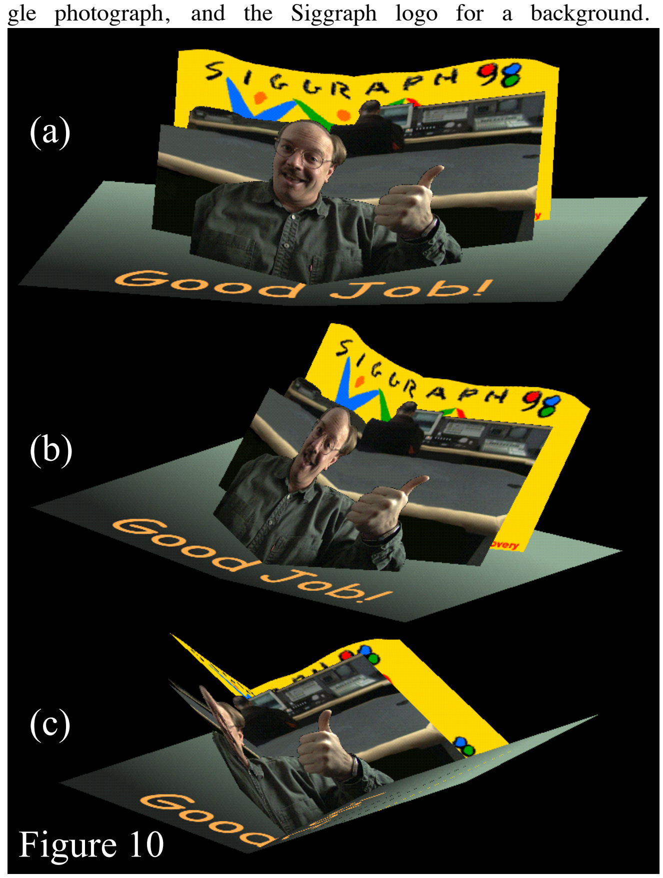

Figure 10

When the designer is satisfied with the card, the system prints out pages of templates for all the mechanisms. To build the pages, we use a greedy packing algorithm. Each page maintains a rectangle indicating its available area. As long as there are pieces remaining to pack, we select the largest one, and then look for the first page with an available rectangle large enough to hold the piece; if there is no such page then we create a new page. On the chosen page we place the piece at the top, and compute the area of the remaining rectangle below. We then place the piece at the side and find the remaining area. We choose the position that leaves the most area on the page, and move on to the next-smaller piece. Figure 11 shows the templates for the card of Figure 10.

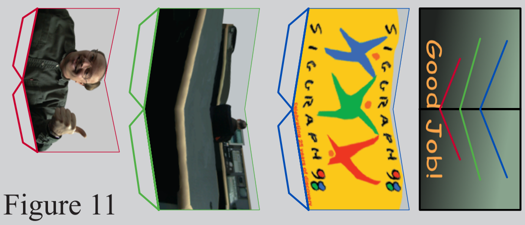

Figure 11

9 Future Work

It would be nice to include collision detection in the design program; we currently rely on the designer's eye to prevent pieces from hitting one another. We would also like to use a 2D packing algorithm like [8] to minimize wasted paper area. A complete design system should include the rest of the common pop-up-card mechanisms, including discs, straps, and boxes [2]. Although these are straightforward to design without a computer's help, it would be convenient to have all the mechanisms available in one place while creating a card. The implemention of these other devices would use a combination of specific special-purpose code and the geometry in this paper. We would also like to add other design amenities such as 3D painting directly onto the card.

Some pop-up books use the rigidity of the paper as a mechanical device to push and rotate other pieces into position; the V-folds in Figure 1a pull up the curved piece this way. We would like to investigate such techniques. Other extensions include curved paper surfaces, curved folds [3], and the use of hardware such as rubber bands and springs to cause mechanical action. We would like to see a stable design system made available so people can quickly and conveniently design their own pop-up cards for themselves and each other.

References

[1] Peter Engel. *Folding the Universe*. Vintage, 1989.

[2] Mark Hiner. *Paper Engineering*. Tarquin Publications, 1985.

[3] David A. Huffman. Curvature and creases: A primer on paper. *IEEE Trans. Comp.*, C-25(10):1010–1019, October 1976.

[4] Paul Jackson. *The Pop-up Book*. Henry Holt, 1993.

[5] Yannick L. Kergosien, Hironoba Gotoda, and Tosiyasu L. Kunii. Bending and creasing virtual paper. *IEEE Computer Graphics & Applications*, 14:40–48, January 1994.

[6] Glenn A. Kramer. *Solving Geometric Constraint Systems*. MIT Press, 1992.

[7] Y.T. Lee, B. Tor, and E. L. Soo. Mathematical modelling and simulation of pop-up books. *Computers & Graphics*, 20(1):21–31, 1996.

[8] V. Milenkovic, K. Daniels, and Z. Li. Placement and compaction of nonconvex polygons for clothing manufacture. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 236–243, 1992.

[9] Dan Pedoe. *Geometry: A Comprehensive Course*. Dover Publications, 1970.