Robotic Origami Folding

Devin Balkcom

CMU-RI-TR-04-43

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August, 2004

Committee
Matthew T. Mason (Chair)
James J. Kuffner
Doug L. James
Jeffrey C. Trinkle (Rensselaer Polytechnic Institute)

Abstract

Origami, the human art of paper sculpture, is a fresh challenge for the field of robotic manipulation, and provides a concrete example for many difficult and general manipulation problems. This thesis will present some initial results, including the world's first origami-folding robot, some new theorems about foldability, definition of a simple class of origami for which I have designed a complete automatic planner, analysis of the kinematics of more complicated folds, and some observations about the configuration spaces of compound spherical closed chains.

Acknowledgments

Thanks to my family, for everything. And thanks to everyone who has made the development of this thesis and life in Pittsburgh so great. Matt Mason. Wow! Who ever had a better advisor, or friend? Thanks to my colleagues, for all that I've learned from them. Jeff Trinkle, "in loco advisoris". My academic older siblings and cousins, aunts and uncles, for their help and guidance in so many things: Alan Christiansen, Ken Goldberg, Randy Brost, Wes Huang, Srinivas Akella, Kevin Lynch, Garth Zeglin, Yan-Bin Jia, Bruce Donald, Mark Moll, Al Rizzi, Howie Choset, Illah Nourbakhsh, Daniel Nikovsky. Yoshihiko Nakamura, for his guidance while I was in Japan. And my academic younger brothers, Sidd Srinivasa and Ravi Balasubramanian, for making the mlab such a cool place to work. For contributions to the thesis, specific and general, thanks to Jim Milgram, Erik and Marty Demaine, Nell Hana Hoffman, James Kuffner, Doug James, Brendan Meeder, and Yasumichi Aiyama. And of course, thanks to the jugglers, the swimmers, the musicians, the dancers, and so many other friends, for all the great times. Who knew that Pittsburgh would be so much fun? Thanks!

Contents

1 Introduction **8**
- 1.1 The challenge of origami . . . 8
- 1.2 Three example problems . . . 9
- 1.3 Key contributions . . . 12
  - 1.3.1 A new domain . . . 12
  - 1.3.2 An origami-folding robot . . . 14
  - 1.3.3 Complete planning for a simple class of origami . . . 14
  - 1.3.4 Kinematic models of origami . . . 14
  - 1.3.5 3D foldability results . . . 15
  - 1.3.6 Low-level paper manipulation . . . 15
- 1.4 Structure of the thesis . . . 16

2 Human and robotic origami skills **17**
- 2.1 Introduction . . . 17
- 2.2 Origami classification . . . 18
- 2.3 Pureland origami . . . 19
  - 2.3.1 Mountain and valley folding skills . . . 21
- 2.4 Flat origami: basic folds, bases, and examples . . . 22
  - 2.4.1 Basic vertex folds . . . 22
  - 2.4.2 Basic vertex-folding skills . . . 23
  - 2.4.3 Compound vertex folds . . . 29
  - 2.4.4 Folding skills for compound patterns . . . 29
  - 2.4.5 Bases for flat origami . . . 38
  - 2.4.6 Example flat crease patterns . . . 45
- 2.5 3D origami . . . 47

3 Representation and design **50**
- 3.1 Related work; properties of paper . . . 51
  - 3.1.1 Developable surfaces . . . 51

3.1.2 Representing paper by developable surfaces 54
3.1.3 Geometry of creases 55
3.1.4 Modelling paper with natural creases 57
3.1.5 Manipulation of flexible objects 58
3.1.6 Rigid multibody dynamic simulation 58
3.1.7 Cloth simulation 60
3.1.8 Haptic simulation 61
3.1.9 Fourier models 62
3.1.10 Continuum models 63
3.1.11 Origami mathematics and design 64
3.2 Rigid-body origami models 65
3.3 Line-segment origami with revolute joints 66
3.4 Faceted origami with revolute joints 68
3.5 Properties of flat origami 71
3.5.1 Local properties of flat origami 71
3.5.2 Global properties of flat origami 72
3.6 Origami with ball joints and struts 75
3.7 Bending of paper 79
3.7.1 Formulation of the model 79
3.7.2 Potential energy functions 81
3.7.3 Differential kinematics 82
3.7.4 Kinetic energy and dynamics 83
3.7.5 Force control 85
3.7.6 Evaluation 86

4 Simple origami folding **87**
4.1 Related work 88
4.1.1 Another origami-folding robot 88
4.1.2 Sheet-metal bending 88
4.1.3 Box folding 89
4.1.4 Rope handling 91
4.1.5 Planning for flexible objects 92
4.1.6 Wire bending and insertion 94
4.1.7 Manipulation of fabric 94
4.1.8 Grasping of flexible objects 94
4.2 Simple folds 95
4.3 Book folds 95
4.3.1 Necessary conditions 98
4.4 A planner for book-foldable origami 100
4.5 An origami folding machine 102

4.6 Machine evaluation; future directions . . . 105
4.6.1 Step 1 – Paper positioning . . . 105
4.6.2 Step 2 – Paper bending and friction grasp . . . 108
4.6.3 Step 3 – Dynamic crease formation . . . 109
4.6.4 Step 4 – Sweep-flattening of paper . . . 110
4.6.5 Step 5 – Paper release . . . 110
4.6.6 Paper selection; effect of humidity . . . 111
4.6.7 Sensing . . . 111
4.7 Comparison to other folding methods . . . 111
4.7.1 Comparison to sheet-metal bending . . . 112
4.7.2 Comparison to carton folding . . . 113
4.7.3 Comparison to human folding . . . 113
4.8 Experiments motivated by human folding techniques . . . 114
4.8.1 Tool design . . . 114
4.8.2 Folding the paper . . . 115
4.8.3 Bending the paper . . . 117
4.9 Experiments in folding pre-creased paper . . . 119
4.9.1 Folding creased paper . . . 120
4.9.2 Folding an envelope . . . 120
4.9.3 Evaluation . . . 121

5 Vertex folding **123**
5.1 Related work . . . 125
5.1.1 Parameterization of closed-chain mechanism c-spaces 125
5.1.2 Topology of configuration spaces . . . 125
5.1.3 Foldability of 3D structures . . . 126
5.2 Local parameterization . . . 126
5.2.1 Sequential crease angles . . . 126
5.2.2 Non-sequential crease angles . . . 128
5.3 The c-space topology of spherical n-bar linkages . . . 131
5.3.1 Four- and five-bar mechanisms . . . 131
5.3.2 Many-link mechanisms . . . 136
5.4 Self-intersection . . . 139
5.5 Multi-vertex patterns . . . 141
5.6 3D foldability . . . 142
5.6.1 The bellows theorem . . . 142
5.6.2 Can a shopping bag be collapsed? . . . 143
5.6.3 Unfolding the shopping bag . . . 149

6 Conclusion **153**

A Some notes on Morse theory **154**
A.1 Definitions 154
A.1.1 Theorems 155
A.1.2 A simple example: Morse on a sphere 156

Chapter 1

Introduction

The contributions of the thesis fall in three classes: designing for foldability, folding manipulation, and analysis of closed chains. This chapter will discuss some of the challenges presented by origami, key scientific questions motivated by origami, and the contributions and structure of the thesis.

1.1 The challenge of origami

Modelling and manipulation of flexible objects, folding manipulation, and analysis and planning for closed-chain structures are key areas on the boundaries of what we understand about manipulation science. Origami is a concrete example for study. Paper is flexible and springy, but stretches hardly at all; simulation and manipulation are hard. Complex origami involves many thicknesses of paper; each successive fold is more difficult than the last, and the volume or surface area typically shrinks with each fold. We might model creases as joints, and the uncreased regions as rigid bodies. If the creases cross, it turns out that the mechanism is a closed chain. Although robot configuration spaces are typically modelled as manifolds, we can make a mechanism with a non-manifold configuration space with just two folds of a flat piece of paper.

Origami is also a good problem to study because there is a 'ladder' of origami skills and designs, from the very simple to the complex. Although this thesis will only reach the first few rungs, the avenue of exploration is clear. Origami books provide thousands of designs, with annotated instructions describing one way to fold each. Each new design that we analyse will require and inspire a better understanding of the art of manipulation.

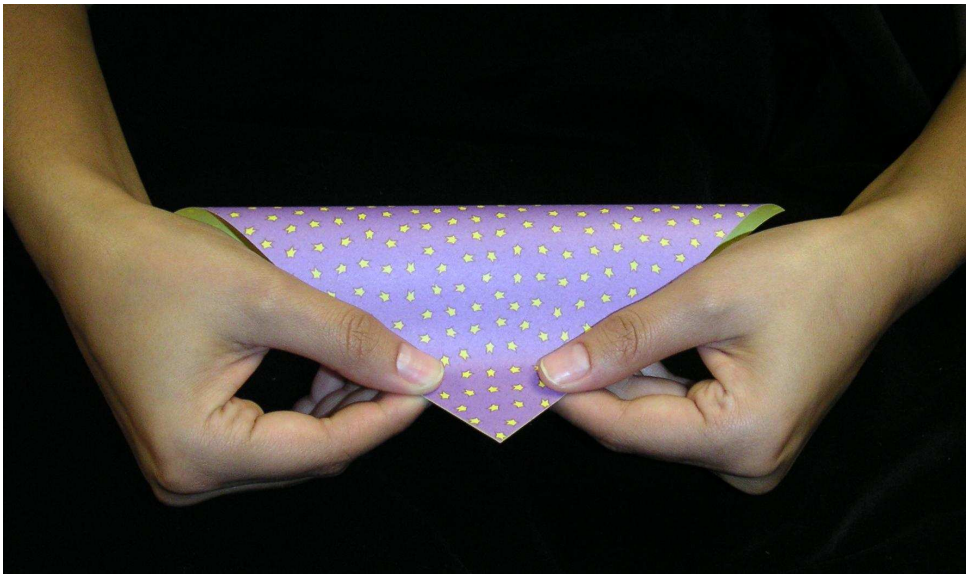


Figure 1.1: Landmarking.

1.2 Three example problems

Origami is interesting from the robotics perpsective because of the wealth of problems that it poses, from the design of low-level manipulation primitives, to mechanism configuration-space analysis, to the mathematical modelling of folding.

This section presents three motivating 'case studies', each of which will be returned to and discussed more fully later in the thesis.

One perspective from which origami folding is interesting is that of classical manipulation. How should a crease be placed precisely in a flexible sheet of paper? It is difficult to see how to measure the state of a bent paper with sensors. Occlusion, the thin-ness of the paper, and the presence of curved surfaces make a general origami vision or laser-range-finder system seem hopeless. Tactile sensors are even worse – touching the paper is likely to deform it. Do dynamics need to be measured? Representation poses an additional problem; flexible paper seems likely to require a large number of configuration variables.

In spite of these difficulties, a human folder can place certain types of fold very reliably and accurately. Figure 1.1 shows an example. The key seems to be a technique called 'landmarking', in which two corners or

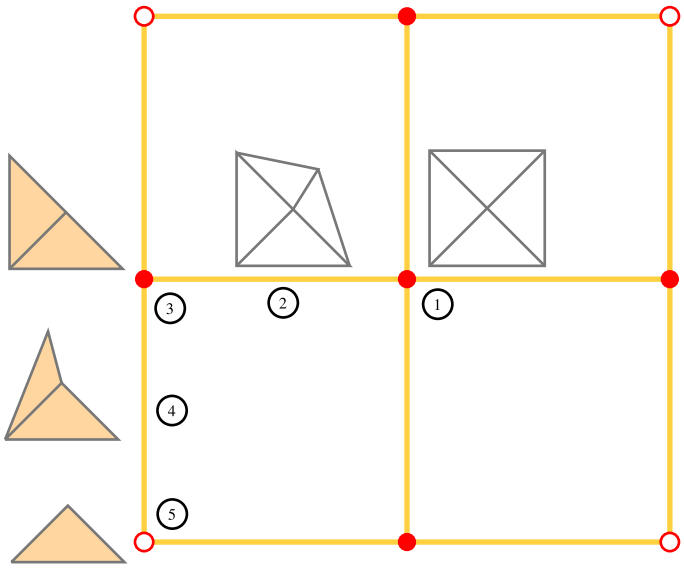


Figure 1.2: Topology of the configuration space of a simple origami design.

edges far away from the fold are precisely placed together. The paper is then flattened; the fact that the paper does not stretch forces the crease to occur in the desired location. This is a minimal-sensing approach to flexible object manipulation; once the fixed corners have been aligned, the fold can be completed quickly, apparently without much effort spent on determining the exact state of the paper.

Interestingly, there are various approaches to making folds of this type; some seem to allow or require more or less sensing as the fold is created, but each starts with the same landmarking step. Chapter 2 will survey some of these skills in more detail.

A second perspective from which origami is interesting is mechanism kinematics. Consider a piece of origami with two perpendicular crease lines. Model each of the four creases as a revolute joint, and model each piece of uncreased paper as a rigid body. What is the configuration space for this mechanism? When flat, the paper can be folded along either crease line. As soon as the paper is not flat, however, it is only possible to fold along the previously folded crease line, at least until the paper becomes flat again.

Figure 1.2 shows a graph describing the topology of the configuration space, and an example fold trajectory. The horizontal and vertical axes are the joint angles of two perpendicular creases. For the example, crease 1

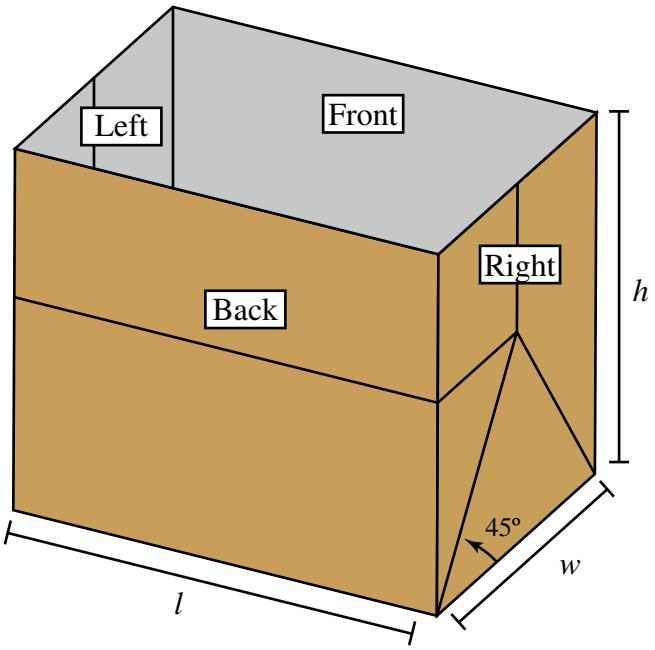


Figure 1.3: A shopping bag. The configuration space of the rigid-body model is isolated points, and the bag cannot be collapsed.

is first folded in the 'negative' direction; once the first fold has been made, crease 2 is folded in the negative direction, forming a small four-layer triangle. The nodes in the graph represended by filled circles are points where folding is possible in multiple directions; the nodes represented by open circles are points where folding in multiple directions would be possible if not for collisions between facets.

Even the simplest origami design with crossing creases may have a relatively complicated configuration space; what happens when there are high-degree vertices, and networks of vertices connected by creases? Chapter 5 explores this problem in more detail.

A third perspective from which origami is interesting is mathematical foldability. Figure 1.3 shows a paper shopping bag, like those used in department stores and grocery stores. It turns out that if we model the bag using rigid bodies and revolute joints at creases, the configuration space is isolated points: the bag can't be opened or folded. When are kinematic structures foldable? What happens if we add additional creases? Chapter 5 explores these questions in more detail.

1.3 Key contributions

How should roboticists measure research progress in a domain? Some milestones include basic modelling of the system, simple local planning or optimization, understanding the space of configurations of the system, and development of a complete planner or global optimization algorithm. For manipulation tasks, understanding the physical process or manipulation skills needed to transform the system state can be a challenge, and building a robot to execute those skills is an important milestone.

Figure 1.4 gives a rough map of the origami domain, and outlines the level of understanding reached by this thesis. The far left column shows some traditional human origami skills. I have also developed a classification of origami skills that is based on the complexity of the simplest model that can be used to represent a fold. For example, 'simple folds' reflect all paper from one side of a crease line to the other, and fold sequences can be planned using a very simple model of polygons in the plane, but origami with curved creases and facets requires a richer model.

1.3.1 A new domain

The driving application for the study of manipulation has been automated manufacturing. The types of manipulation considered most in the literature reflect this: grasping, fixturing, pushing, sorting, and feeding are elements of the manufacturing process.

New applications can give new perspectives. The study of human athletic abilities (including throwing, running, juggling, soccer, swimming, and acrobatics) has led to significant advances in our understanding of robot control, planning, dynamic manipulation, mobile manipulation, and the manipulation of flexible or high-degree-of-freedom objects.

Origami is a human manipulation art. The study of origami provides insight into folding as a class of manipulation of its own. What is folding? A perspective suggested by the current work is that folding is a series of local shape-transforming manipulations of a flexible or high-degree-of-freedom body. An unfolded piece of origami typically has a very complex structure, even if the uncreased portions of the paper are modelled as rigid bodies. Yet, complete and efficient planners can be designed to take the mechanism to interesting goal configurations, using only simple manipulation skills.

A primary contribution of this work is the analysis and classification of simple human origami. How should simple folding skills be modelled

Figure 1.4: The origami domain.

with enough precision that they can be simulated or analysed? Chapter 2 will discuss some aspects of human origami folding.

1.3.2 An origami-folding robot

I have built what appears to be the first machine designed specifically to fold origami. The heart of the system is a 4-DOF Adept SCARA arm. The arm positions the paper using a vacuum pad, and a machine similar to a sheet-metal brake makes the folds. Successful folds include a 'cushion base', where all four corners of the paper are folded to the center, a simplified version of the classic samurai hat, a paper airplane, and a paper cup.

1.3.3 Complete planning for a simple class of origami

The origami-folding machine can fold paper from one side of a desired crease line to the other; we call this type of folding *simple folding*. If the pattern of creases in the final shape is known, it is possible to enumerate all simple folds. Chapter 4 describes a graph search planner that is complete for simply-foldable origami; the planner was used to find the fold sequences that the robot used to fold the cushion base, the hat, and the airplane.

Simply-foldable origami is a subset of Pureland origami, a class of origami that will be discussed in chapters 2 and 4. The primary skill that divides simply-foldable and Pureland origami is the separation of flaps of paper, a skill the robot does not have. Separating folds that reflect paper across a crease line are often called *book folds* in the origami community. Chapter 4 models these folds, and describes a planner that is complete for book-foldable origami. The traditional samurai-hat design is book-foldable, but not simply-foldable.

1.3.4 Kinematic models of origami

The simplest model of origami takes creases as revolute joints and uncreased paper as rigid. Under this model, the kinematic structure of an unfolded origami pattern is either an open (serial) or closed chain, depending on whether or not creases meet on the interior of the paper. Around each interior vertex is a *spherical* closed chain; if there are many interior vertices we say that the mechanism is a *compound* spherical closed chain.

Origami therefore provides a motivation to study spherical closed chains. Chapter 5 will present a number of results about closed chains, including

parameterizations of simple and compound chains, and some analysis of their configuration spaces.

1.3.5 3D foldability results

The primary model of origami used in this thesis is a rigid-body model with a fixed number of creases in fixed locations on the paper. This model is sometimes sufficient to describe high-level state transitions of the paper as folds are made, but there are some cases where it is not. One example is closed structures. Connelly's *bellows theorem* states that a polyhedron with rigid facets and a finite number of hinge joints at fixed locations has constant volume. This means that it is impossible to model the collapse or inflation of a car airbag, a teabag, an origami waterbomb, or other balloon-like structures using a finite number of fixed creases.

There are other cases where the finite-crease model does not work. It will be shown in chapter 5 that a finite-crease model of the paper shopping bag with creases in the usual places cannot be collapsed; there are not enough revolute joints, and the mechanism is overconstrained. It might seem that some extension of the bellows theorem would apply, but it turns out that with the addition of a finite number of creases, it is in fact possible to collapse the bag. (This is joint work with Erik and Martin Demaine.) The procedure for collapsing the bag does *not* collapse the bag to the same flat state as the shopping bag folded in the usual way. An interesting question is then, given the usual flat state, is it possible to unfold the bag using a finite number of creases? We conjecture that it is, and present a possible pattern of creases. (Although we have built a physical model of the mechanism that seems to unfold, the proof that our conjectured pattern allows unfolding is not yet complete.)

1.3.6 Low-level paper manipulation

How should paper be grasped and manipulated? The flexibility of paper is one obvious problem. Another difficulty is that folds made in paper are typically very acute; punch-and-die folding methods like those used for bending sheet metal are problematic, because the punch must be very thin, and it is difficult to see how to remove it after folding the paper through almost 180°.

Human folders often solve these problems by lining up edges or corners of the paper far away from the intended crease, and then using the fact that

paper doesn't stretch to ensure that the crease is formed in the correct place as a fingernail presses the paper flat.

The origami-folding robot uses a different technique and friction-grips the outside of the paper before creating a crease by slamming two metal plates together, but I have also explored some aspects of more human-like creasing.

1.4 Structure of the thesis

The structure of the thesis is as follows. Chapters 1 and 2 present some background, motivation, and a very brief introduction to human origami folding. Chapters 3, 4, and 5 are the core of the thesis, and present the bulk of the results. Each of these chapters presents related work and the context of current work, gives main results, and some evaluation of the results.

Chapter 2

Human and robotic origami skills

2.1 Introduction

What is origami? There are many ways that origami is used and enjoyed, and there often seems to be no common thread. In this chapter, we classify origami into a few broad categories, and examine some of the very different skills used to fold each type. But before we turn to the classification of origami by skills, we briefly discuss some of the uses of origami.

In traditional Japanese origami, the folding of forms is considered a contemplative ritual. Traditional designs include the samurai hat, the Chinese junk, and the crane. For a purist, the paper must be square. Cutting, gluing, and the use of tools are strictly forbidden, and there seeems to be little molding or sculpting: only folds and tucking are used. Typically, the paper is transformed step-by-step, with a clear new intermediate form after each fold.

According to the educational computer software *Origami, the Secret Life of Paper* [10], each traditional design has a unique key move that is surprising and satisfying. From the perspective of robotics, these 'key moves' are very interesting. How are these moves different from the moves in the usual library of skills? Can they be automatically discovered by a planner, or do they require human intuition?

Origami is also used for education or entertainment of children. Although the rules are less strict than for traditional designs, there are still template designs and a small set of skills that are used for each fold.

Traditional and educational origami can be described as paper folding.

Modern origami should be considered paper sculpture. Wet-folded models are an example – the paper is sculpted while wet and then allowed to dry, allowing amazing free-form figures. Although there are purists and minimalists, most cutting-edge folders take a no-holds-barred approach. Origami master Robert Lang describes the use of methlycellulose coating to paper to allow better sculpting:

> And speaking of purity, the pinnacle of "pure origami is folding from one sheet with no cuts or glue" is surrounded on all sides by slippery slopes, and "judicious selective application of sizing" is rather far down one of those slopes; if not actually down at the bottom of the Valley Of Gluing, it's certainly close enough to peek through the windows at night. If applying MC and its ilk bothers you, then don't do it; if seeing it in others' work bothers you, well, don't look at my work, because I'm using it more and more. (And be prepared to expand your blinders, since I was not the first and the practice seems to be spreading.) [37]

Complex models have hundreds of creases, and each must be made precisely. On the origami e-mail list, there is a continued discussion of where to get paper with just the right properties to fold a specific model – many top folders consider it a necessity to make their own paper.

The most complicated models being designed today push the limits of the best folders, and require special paper, years of training, and sometimes special tools. Are there origami designs that cannot be folded by any human being? Thickness of paper is one limitation; it is often mentioned that no more than seven consecutive folds can be made in a piece of paper ($2^7 = 128$ thicknesses of paper!). But there may be theoretical limitations as well. Current folds only require two hands; can origami models be designed that require more?

2.2 Origami classification

What is success for an origami folding? Human 'origamists' provide one standard, and their techniques may provide inspiration and intuition. There are novice folders, and there are established masters of the craft. `http://folds.net` classifies human origami folding skills in six categories.

- Pureland: valley fold, mountain fold, turn over, rotate, book fold.

- Simple: inside and outside reverse folds, prayer fold.
- Low-intermediate: squash fold, petal fold.
- Intermediate skills: crimp, swivel fold, spread-squash
- High-intermediate skills: open sink, open double sink, closed sink.
- Complex skills: closed unsink.

Our version of the list omits some folds: pleat, radial pleat, cupboard fold, blintz fold, waterbomb base, preliminary fold, bird base, frog base, kite base, pentagon, and stretched bird base. Each omitted fold can be folded by a sequence of the listed folds; for example, a blintz fold valley folds four corners of a square into the center. These compound folds do provide an advantage that we ignore for now – they allow *landmarking* (the center of a square can be precisely located by making a blintz fold), and also form larger building blocks for origami instructions.

This list is only a beginning. Wet folding, cutting, scoring and scraping, modular origami, and folding with various tools are common skills that are not even mentioned. Some skills are considered too basic to be described, and others are so complex that they are typically taught through word-of-mouth and creative experimentation on the part of origami masters.

We will classify origami into roughly three categories: Pureland origami, flat origami, and 3D origami. The following sections will discuss each of these classes of origami, with some example patterns, and some discussion of the required skills. Since detailed origami folding instructions for each of the patterns we discuss are widely available in origami books and on the internet, we provide only the patterns; it is recommended that the reader xerox the patterns and experiment with folding each of the designs.

2.3 Pureland origami

According to the inventor of Pureland origami, John Smith, the motivations for the creation of Pureland origami were two-fold: the aesthetics of minimalist design, and making origami accessible to handicapped children. (See `http://www.users.waitrose.com/~pureland/`.) The name is from the words 'pure land' – only valley and mountain folds are permitted. For a robot, Pureland provides the easiest entry-point into the world of origami. However, even Pureland origami is human-centric; operations

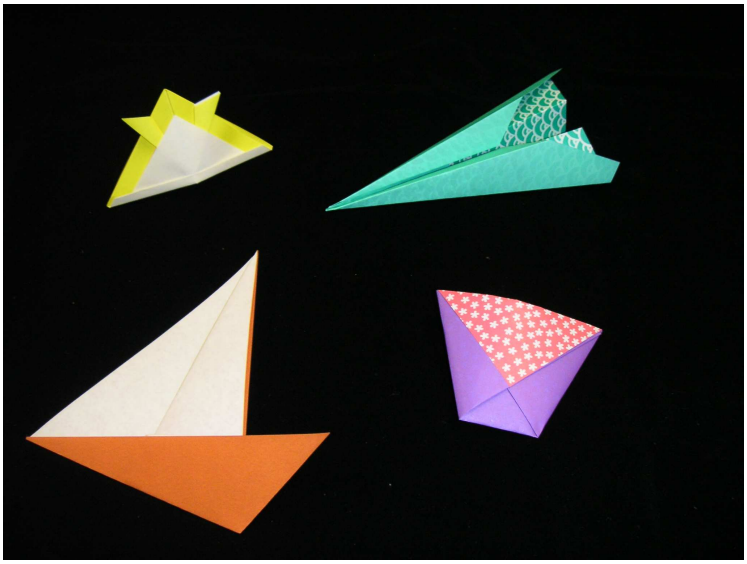


Figure 2.1: Four examples of Pureland origami: samurai hat, airplane, boat, and cup.

like flap insertion and other 3D manipulations are considered easy, and are permitted.

Pureland origami permits only the simplest type of folds: mountain and valley folds. In fact, a mountain fold is just a valley fold viewed from the other side of the paper, so we will only consider valley folds. It should also be pointed out that there is a sometimes confusing difference between a valley *fold* and a valley *crease*. *Every* crease on the crease pattern must either be a mountain or valley; when used in this way, 'valley' refers to the sign of the crease angle. However, valley *folds* fold paper across a single crease line; a valley crease is created in every folded layer of the paper. (If the origami is unfolded and looked at from one side, some of these creases may be considered mountain creases, since some layers may be 'upside down' when the valley fold is made.)

The difference between valley folds and valley creases is apparently even sometimes confusing to master origami folders and designers, as evidenced by a number of origami designs labelled as Pureland that in fact require folding actions more complicated than valley folds. Some examples include published diagrams for 'Pureland' ways of folding the waterbomb base, the preliminary base, the windmill base, and even John Smith's butterfly. Still, there are many Pureland designs that do in fact use only moun-

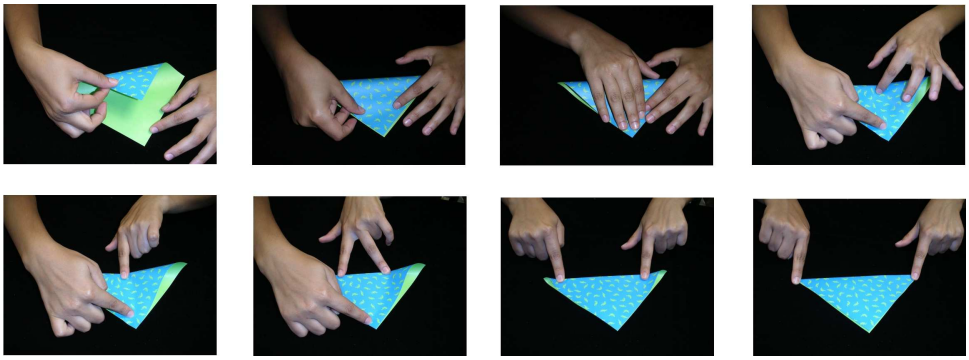


Figure 2.2: Creating a valley fold using landmarking.

tain and valley folds, and for a robot, these simplest of origami designs are the easiest entry point into the world of origami folding.

2.3.1 Mountain and valley folding skills

How should mountain and valley folds actually be made? Figure 2.2 shows an example of the creation of a diagonal valley fold on a table. First, the folder makes a bend in the paper. Then two opposite corners of the paper are aligned, and held in place. The folder slides her finger across the paper towards the intended crease location, flattening the paper along a line, and making a small initial crease in the center of the paper. (We call this step *flattening to create a crease*.) The folder then uses her fingers to extend this crease in either direction to the corners of the paper. (We will call this step *flattening to extend the crease*.)

The method used to make the fold in figure 2.2 has some good characteristics. Since the two opposite corners are far from the crease and from each other, the flexibility of the paper allows them to be manipulated essentially independently. Once the opposite corners have been aligned, the fact that the paper does not stretch ensures that the crease is created in the right place, without having to measure the paper; we call this process *landmarking*.

Not every human folder uses the same technique to make valley and mountain folds, although most seem to use some variation of landmarking. For example, many folders prefer making folds in the air, without the use of the table. It is hard to say whether one method is intrinsically more precise than another, but the different methods may provide useful inspiration for designs for origami-folding robots.

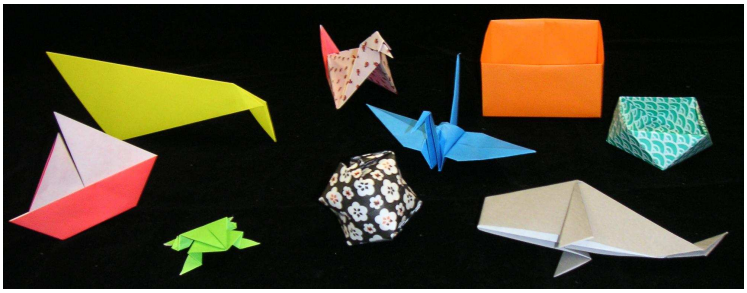


Figure 2.3: Ten examples of flat origami: pecking crow, duck, crane, box, hexagonal box, boat, frog, waterbomb, whale.

2.4 Flat origami: basic folds, bases, and examples

Flat origami is a larger class than Pureland origami. Figure 2.3 shows some examples of flat origami; some of the origami has been 'opened up' to 3D, but was flat after each fold. Kinematically, we can consider each intermediate step in a Pureland fold of a design as a serial arm with one revolute joint. Flat origami allows folding of multiple creases to occur simultaneously; kinematically, the mechanism consists of multiple revolute joints with joint axes intersecting at crease vertices.

Each intersection of creases on the interior of a crease pattern is a candidate for *vertex folding*. We may classify each vertex by the degree of the vertex (the number of creases that meet at the vertex) and by the sector angles around the vertex. In this section, we consider two types of vertex folding. *Basic* vertex folds involve only a single vertex, or multiple stacked vertices with identical patterns that can be treated as a single vertex. *Compound* vertex folds involve networks of creases and multiple vertices; the motion of the paper around a single vertex may be considered as a basic vertex fold, but it may be the case that creases around different vertices must be folded simultaneously rather than sequentially.

2.4.1 Basic vertex folds

One way to classify vertex folds is by the number of creases that intersect at the vertex. The reverse, squash, and rabbit ear folds involve only four creases; the prayer fold requires six. In this section we consider these four example folds, and briefly discuss the manipulation skills required to fold each.

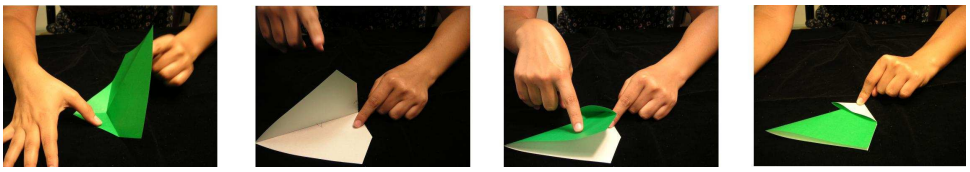


Figure 2.4: Folding a pre-creased reverse fold using two fingers and the table.

The reverse, the squash, and the rabbit ear

The pattern for the reverse fold is a single crease vertex of degree four. There are two colinear creases: one mountain, one valley. At the intersection, there are two creases making an equal angle with the first crease line, either both mountain or both valley. The sector angles are $\alpha, 180° - \alpha, 180° - \alpha, \alpha$, where α is the angle between the central crease line and one of the other creases. Figure 2.6 gives an example pattern for a reverse fold, and figure 2.4 shows the folding of a pre-creased reverse fold.

The squash fold (pattern in figure 2.7) and rabbit ear (figure 2.8) are very similar to the reverse fold from the point of view of manipulation; the pattern for each is also a degree-four vertex. The sector angles for the squash fold are $135°, 90°, 45°, 90°$; for the rabbit ear, $135°, 77.5°, 45°, 112.5°$.

The prayer fold

The pattern for the prayer fold is a vertex of degree six. Both the waterbomb base and the preliminary base are folded with a single prayer fold. The usual sector angles are $90°, 45°, 45°, 90°, 45°, 45°$.

2.4.2 Basic vertex-folding skills

How should basic vertex folds be executed? There is some variation in how humans make folds, and this is particularly apparent with vertex folds. One primary choice is whether or not to pre-crease the paper. Paper can be pre-creased using mountain and valley folds, but there are some decisions to be made during precreasing as well. For example, it is usually more difficult to create a crease that ends in a vertex rather than extending across the entire width of the paper. The reverse fold pattern can be precreased by a sequence of two folds across the entire paper, but some of the mountain folds will have to be changed to valley folds, or vice versa. One method for pre-creasing the rabbit ear involves making creases all of the way across

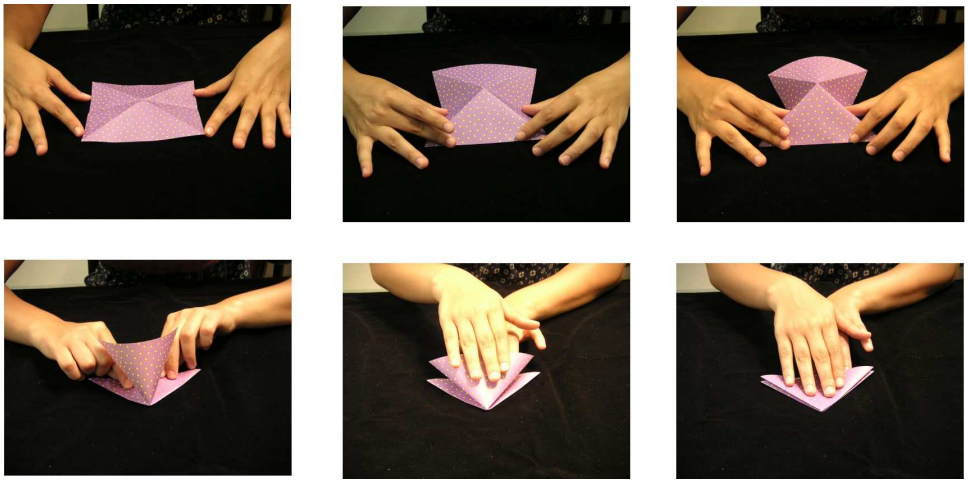


Figure 2.5: Folding a pre-creased prayer fold (waterbomb base) using two fingers and the table.

the paper, but only using the parts of the creases that are part of the rabbit ear; this method has the disadvantage that some creases are created but left unfolded in the final design.

It is also possible to make basic vertex folds without fully pre-creasing the paper, but this is often more difficult. Combinations are also possible; the four valley folds in the waterbomb base (figure 2.9) are often precreased, but the two mountain folds are not created until the prayer fold is pressed flat. It is interesting that unlike mountain and valley folding, multiple non-colinear crease lines may be created at once, and in may be necessary to control multiple flexible regions of paper during flattening.

How many fingers are necessary to make basic vertex folds? Experimentally, Nell Hana Hoffman has shown in our lab that it is possible to make each of the folds discussed using just two fingers and the table, assuming the pattern has been precreased.

Another aspect of making basic vertex folds is landmarking. It may be difficult to line up appropriate edges during pre-creasing, but each of the rabbit ear, squash fold, and waterbomb base patterns shown offer convenient opportunities for landmarking if the pattern is not completely precreased before folding.

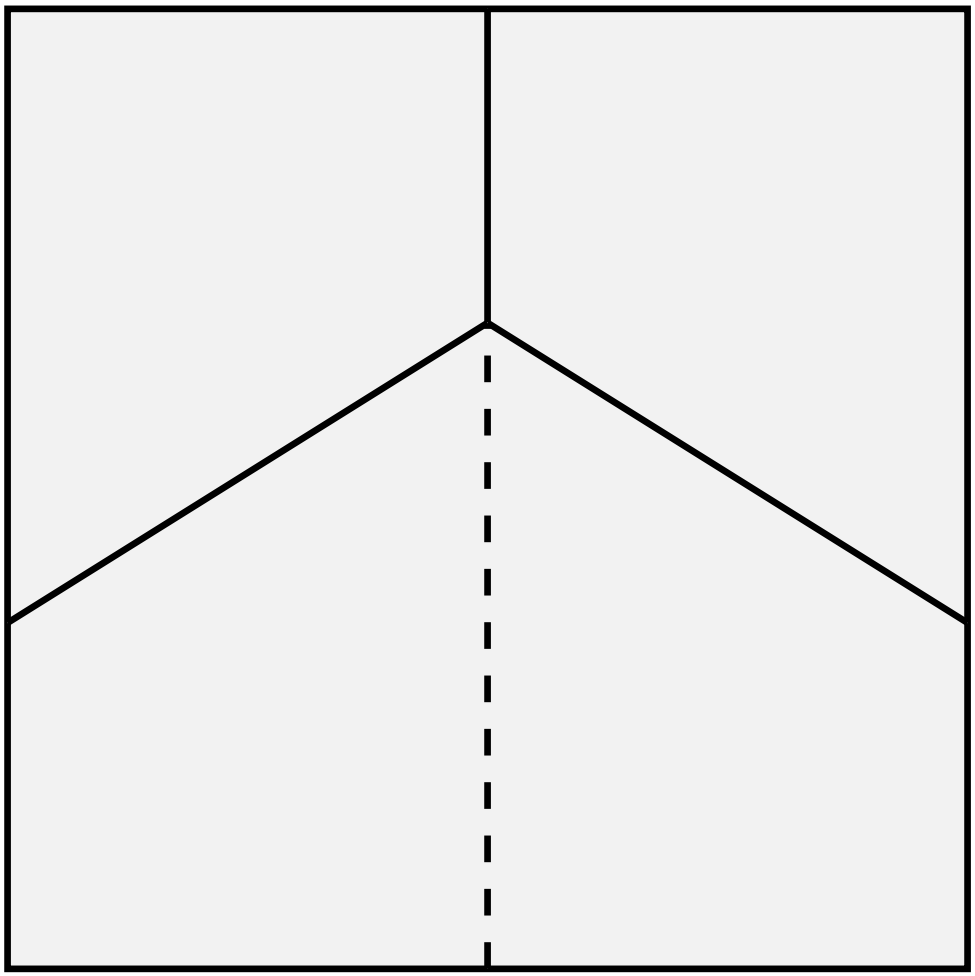


Figure 2.6: Pattern for the basic reverse fold. Relative to the side facing the viewer, this is an 'inside' reverse fold.

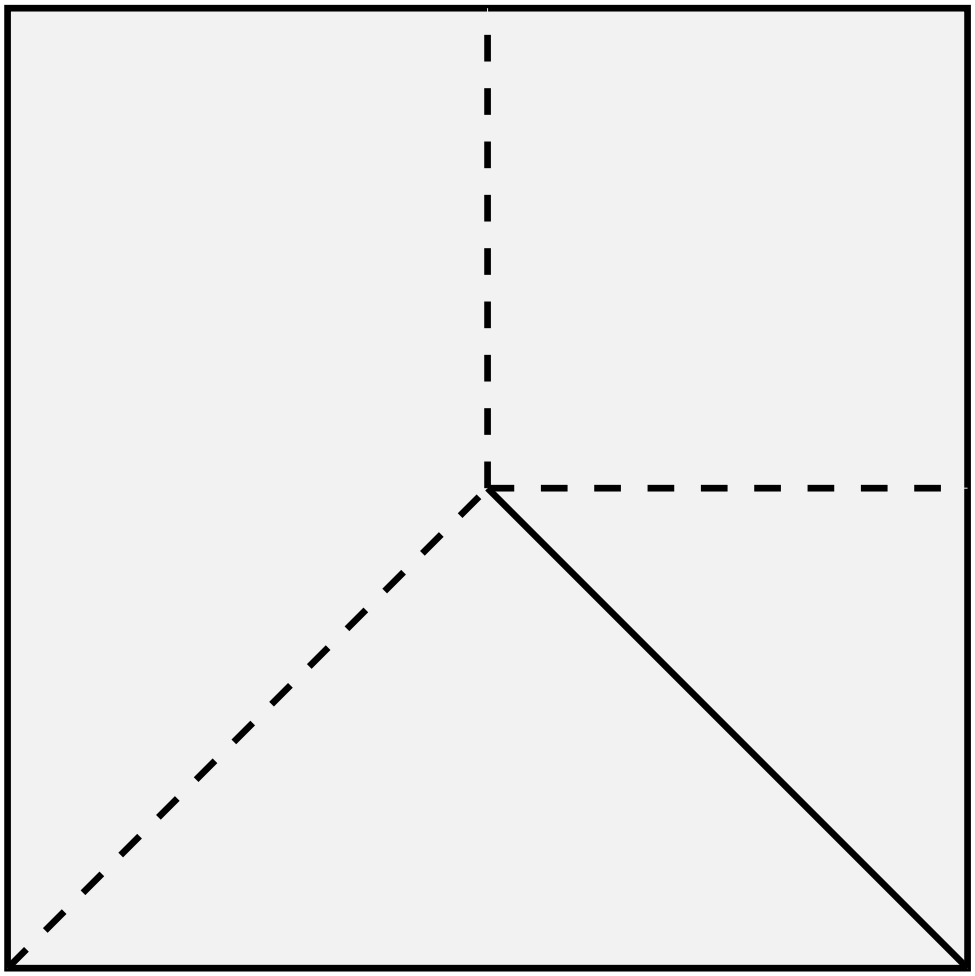


Figure 2.7: The squash fold.

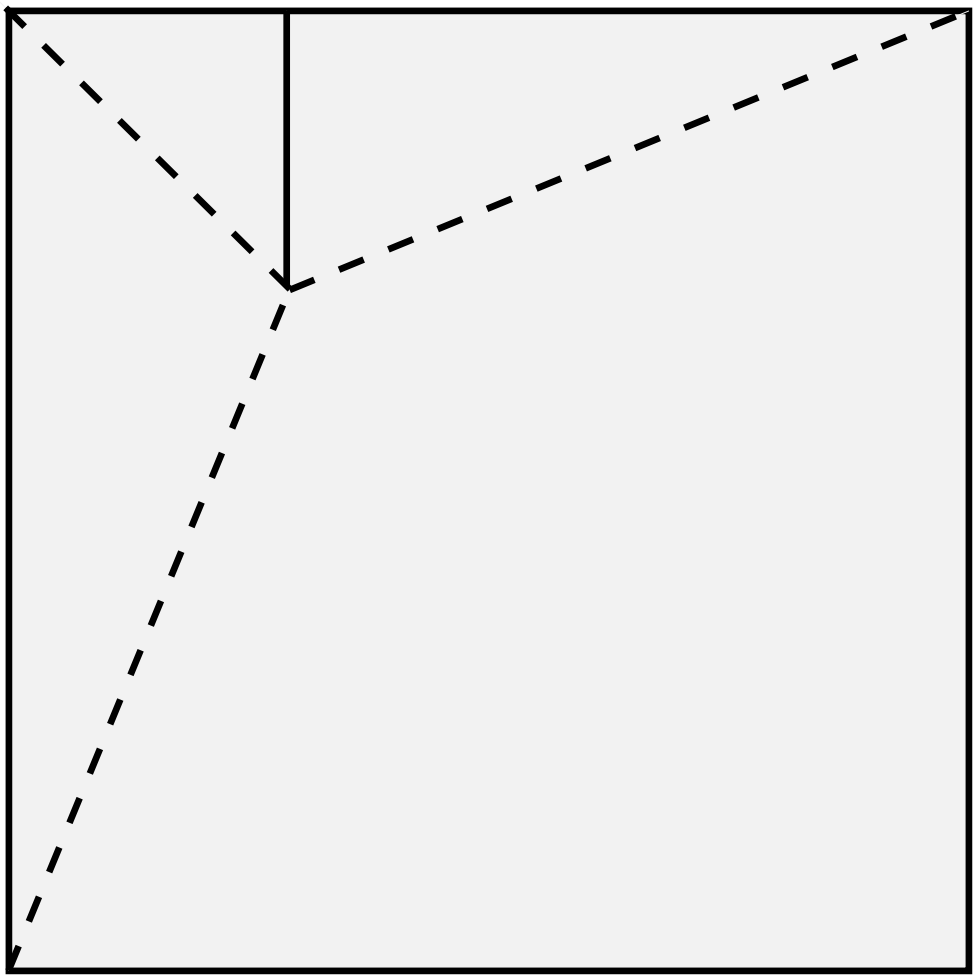


Figure 2.8: The pattern for a rabbit-ear fold.

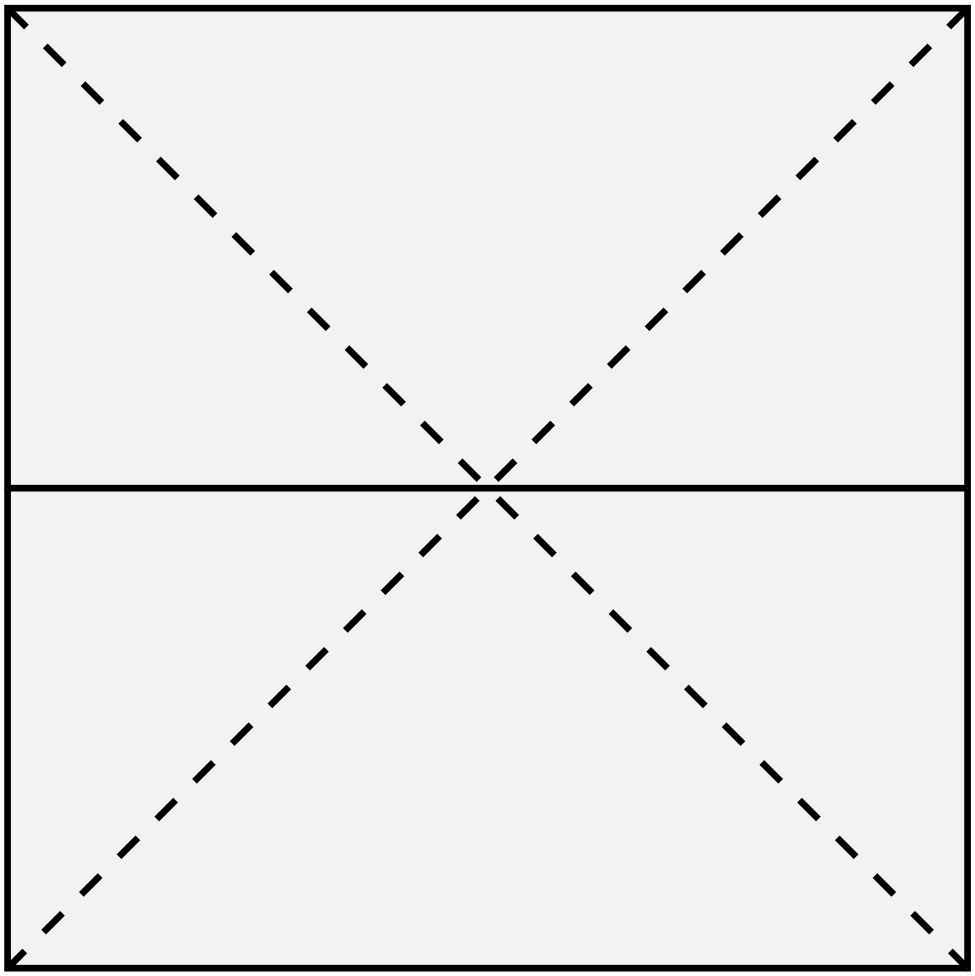


Figure 2.9: Pattern for the prayer fold.

2.4.3 Compound vertex folds

The crimp, double rabbit-ear, petal fold, spread-squash, and open sink all have patterns with multiple vertices; these vertices can be classified as reverse, squash, rabbit ear, or prayer vertices. The basic components suggest how these folds may have been designed, and are one way of analysing the behavior of these folds.

Figures 2.14, 2.15, 2.16, 2.17, and 2.18 show the patterns for these folds; the following table summarizes the components that make up each fold:

Fold	Components
Crimp	Two reverse folds
Double rabbit-ear	Reverse, 2 rabbit-ears
Petal	Squash, 2 rabbit-ears
Spread-squash	6 squashes, 2 reverses
Open sink	2 nested prayer, or 1 prayer and 6 reverses

Another way to classify crease-network patterns is by the number of creases and vertices, and the maximum vertex degree; we expect patterns with many high-degree vertices to be more difficult to fold.

Fold	# creases	# vertices	max. degree
Crimp	7	2	4
Double rabbit-ear	10	3	4
Petal	9	3	4
Spread-squash	19	8	4
Open sink	18	7	6

2.4.4 Folding skills for compound patterns

What skills are required to fold compound patterns? Although the individual vertices of compound patterns may be recognizable as basic reverse folds, squash folds, rabbit ears, or prayer folds, it may be necessary to fold all or many of the vertices in a crease pattern simultaneously. For example, the spread-squash has eight vertices of degree-four connected by creases. Since each vertex has a mobility of 1, the pattern as a whole has a mobility of 1; partially folding any crease will cause all other creases to become partially folded.

Basic vertex folds of degree four or higher always have degrees of freedom, if we model the facets as rigid links and the creases as revolute joints.

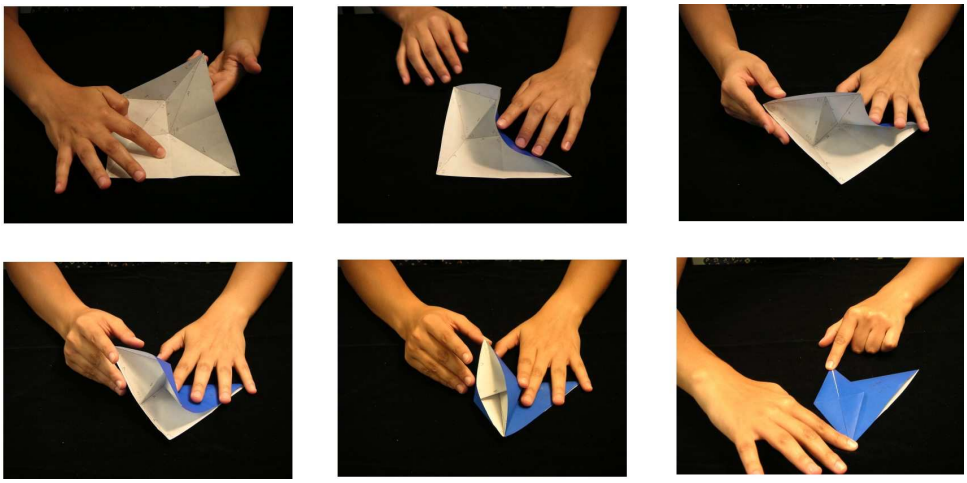


Figure 2.10: The petal fold.

But once we connect multiple vertices together, it may be possible that there are not enough degrees of freedom for the rigid-body model to be folded. Consider the pattern for the open-sink fold, shown in figure 2.18. Two of the six reverse-fold vertices have angles between the creases of $90°, 90°, 90°, 90°$; although we do not go into details here, these vertices cannot be folded simultaneously in a way that is consistent with the folding of the central prayer-fold, without bending the facets. (The proof of this would be similar to the proof that the rigid-body model of a shopping bag cannot be folded, in chapter 5.) An open question is, when is it possible to add a finite number of creases so that the facets do not have to be bent?

It is hard to say whether folding that requires bending facets is intrinsically more difficult than folding that doesn't require bending facets. One interesting observation is that the final folded shape behaves somewhat differently. Since facets tend to resist bending, the open-sink is kinematically 'locked' in place in the final folded shape, and does not unfold automatically when released. (Compare to the final step of the extended bird base, shown in the last few subfigures of figure 2.11. When released, the flaps immediately splay outwards from each other!)

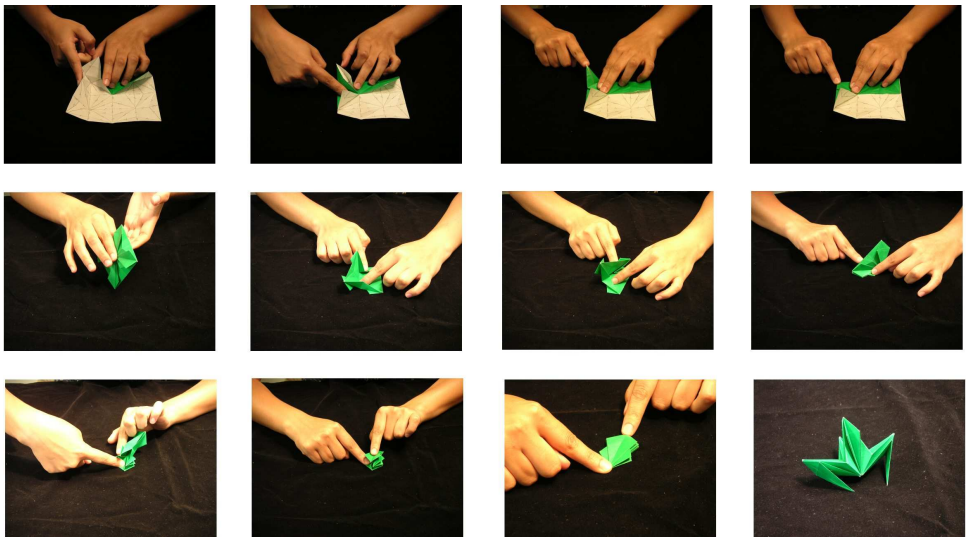


Figure 2.11: Folding the extended bird base.

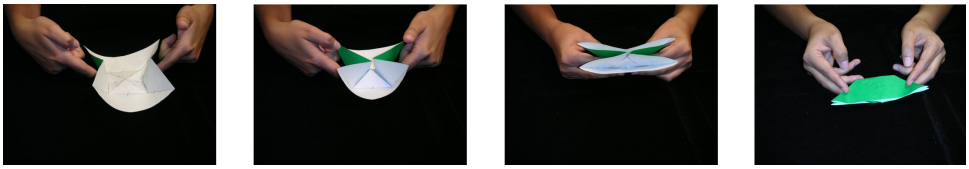


Figure 2.12: Folding the open sink.

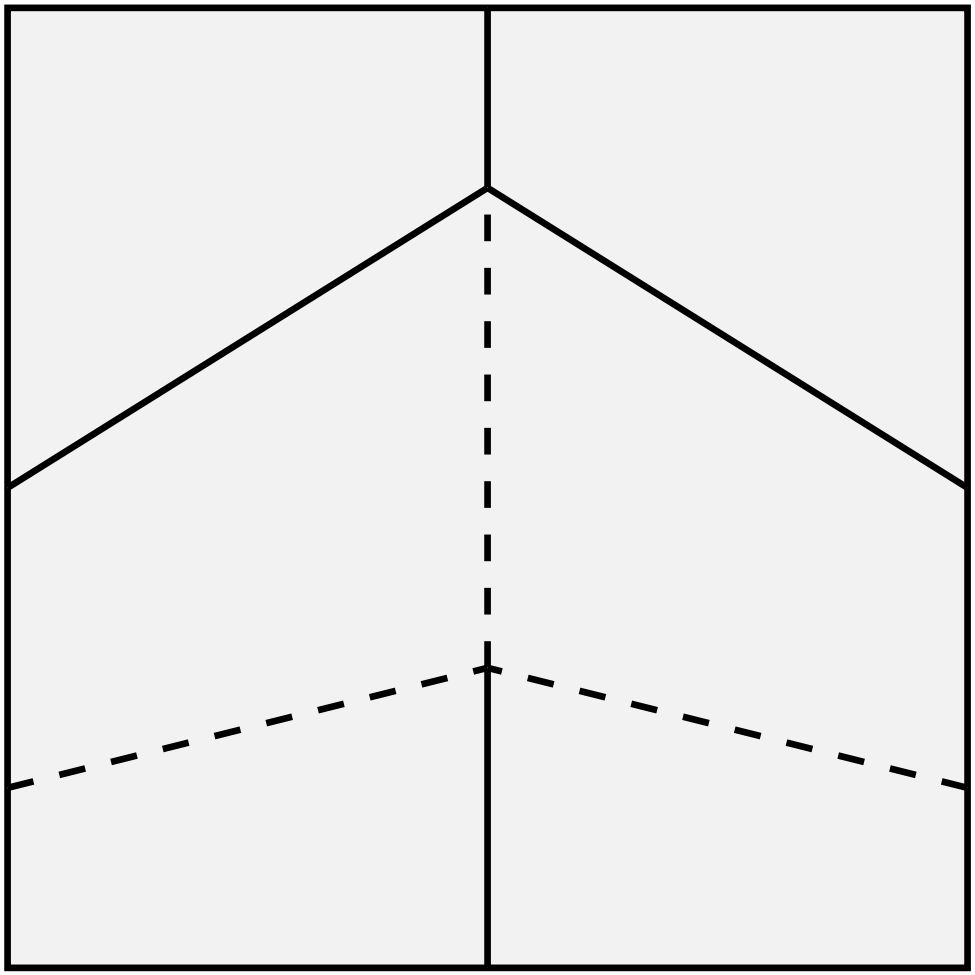


Figure 2.13: Pattern for a double reverse fold. Relative to the side of the paper facing the viewer, the top fold is an outside reverse fold, and the bottom fold an inside reverse fold.

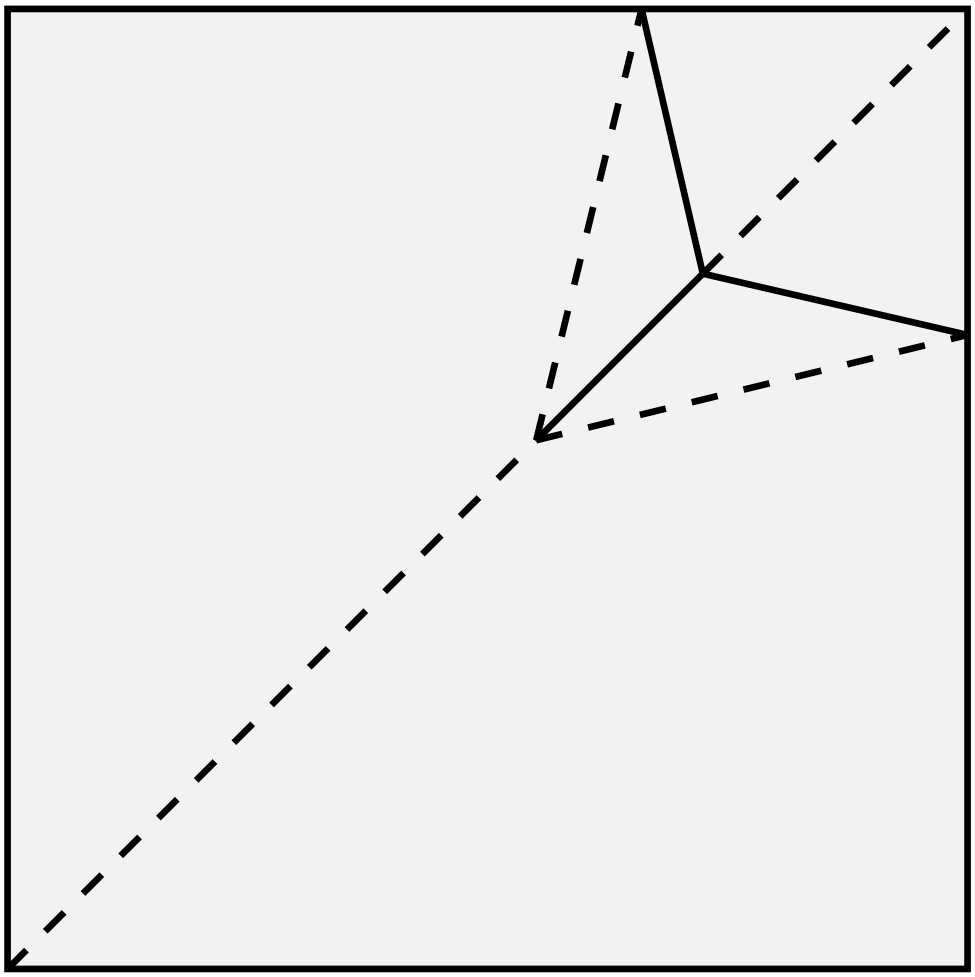


Figure 2.14: The crimp fold. The crimp fold can be seen as a pair of reverse folds, with outer creases that just touch. (Compare to figure 2.13.)

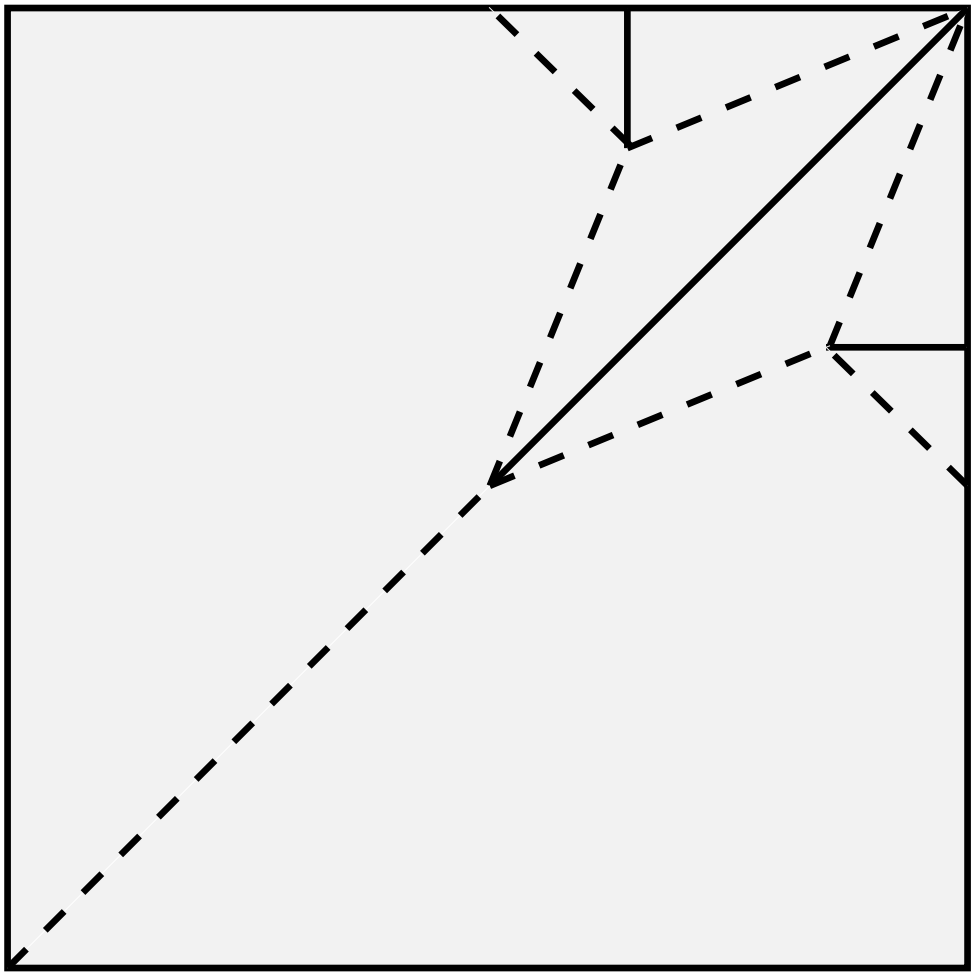


Figure 2.15: The pattern for a double-rabbit-ear fold. The pattern is built from two rabbit ear folds (upper right corner of the paper, to the left and right of the main diagonal) that substitute for part of a reverse fold.

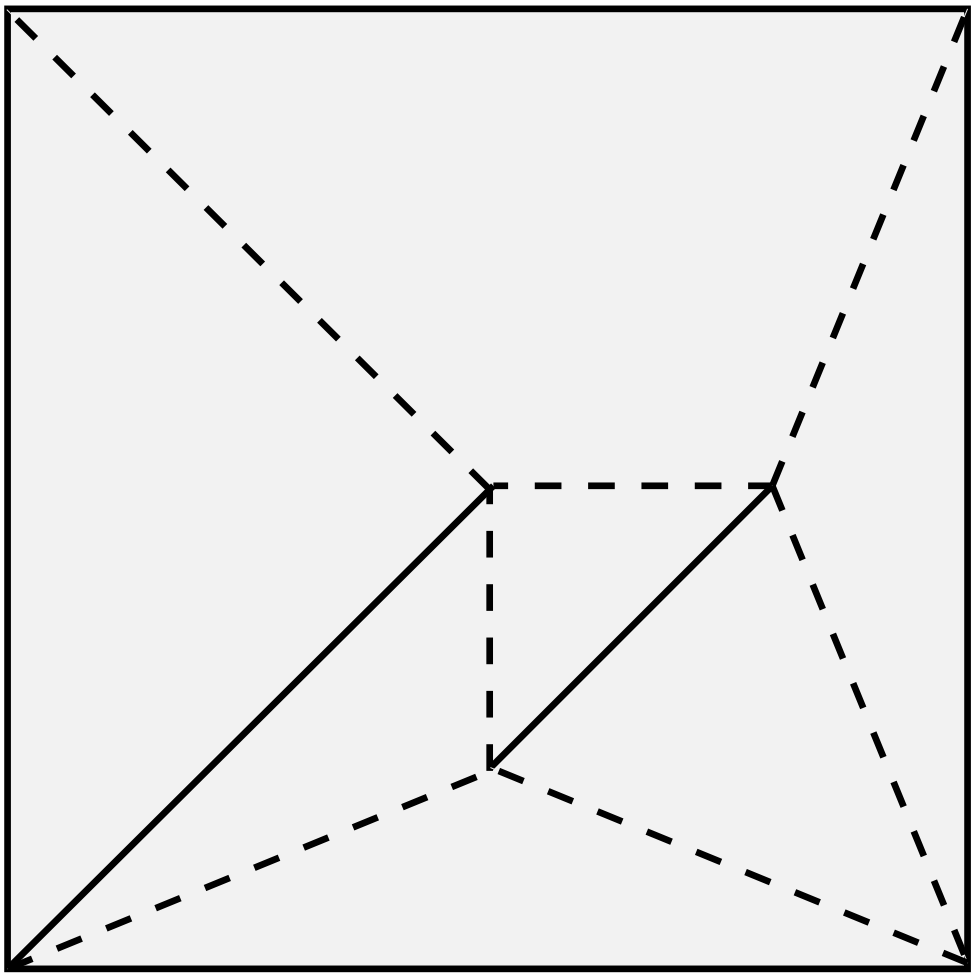


Figure 2.16: The pattern for a petal fold. The upper left vertex is a squash fold; the other two vertices are rabbit-ears.

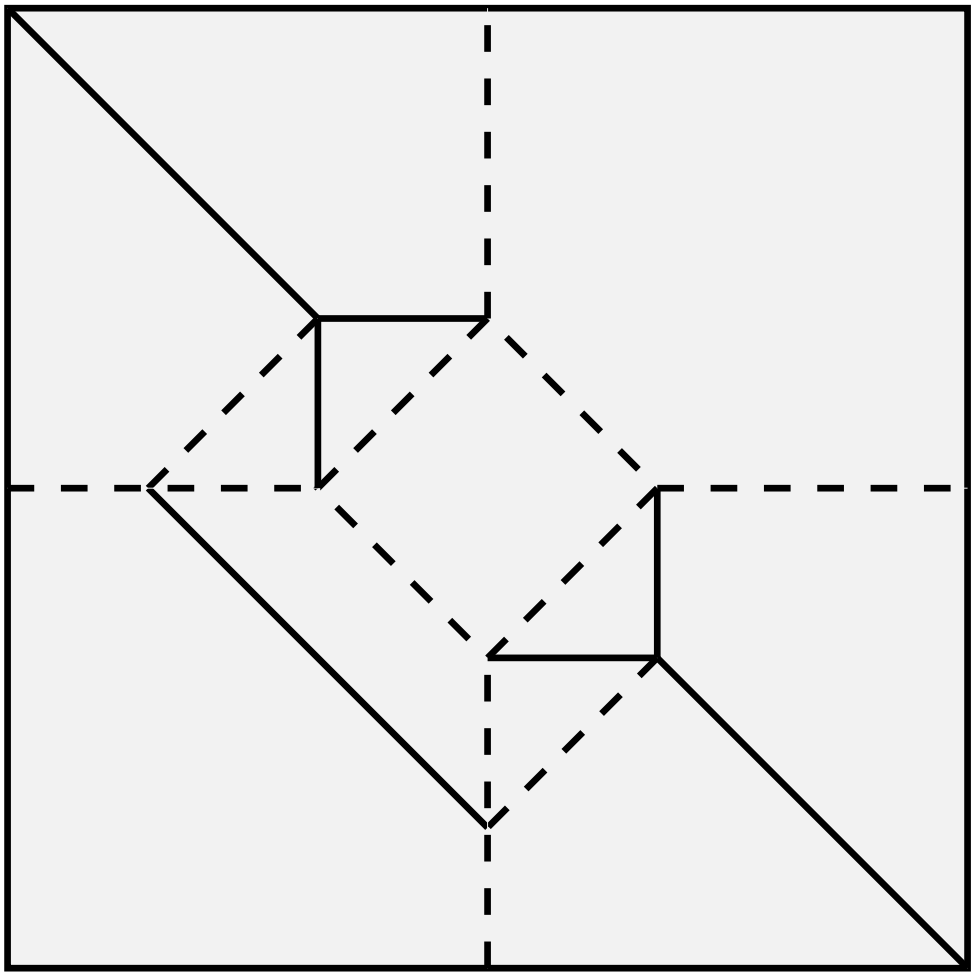


Figure 2.17: The pattern for a spread-squash fold.

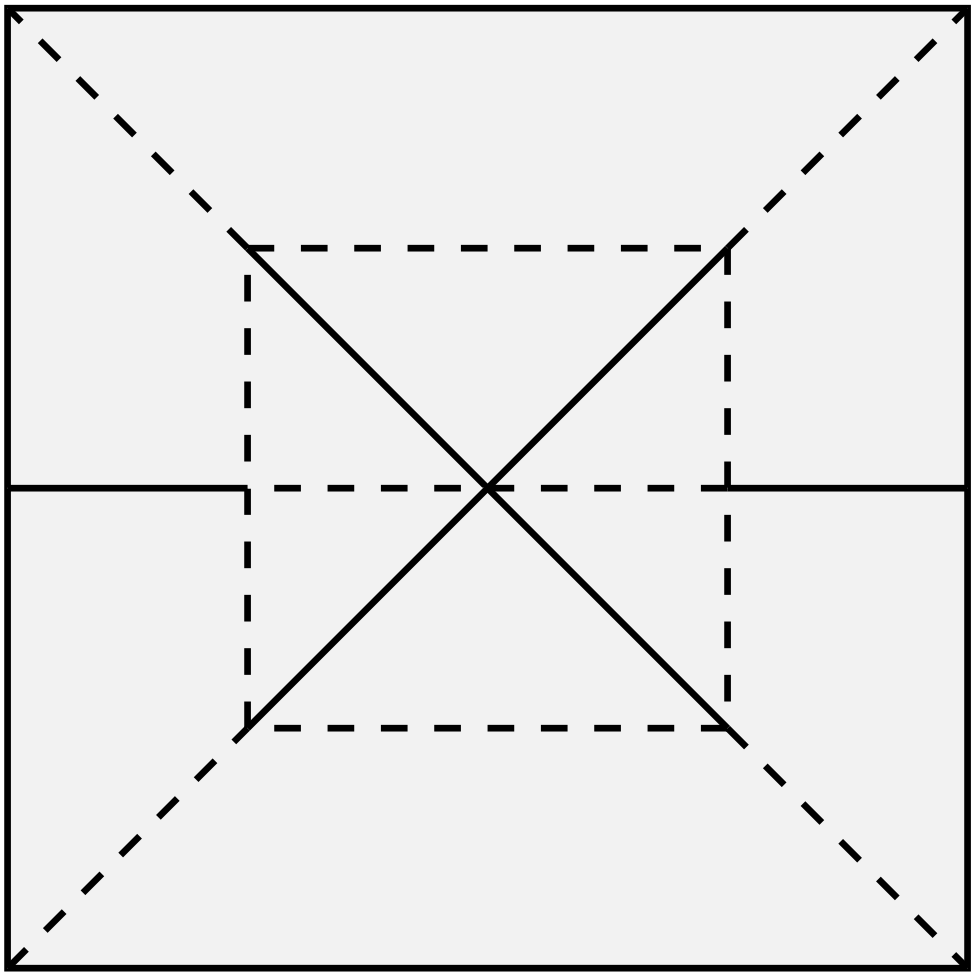


Figure 2.18: The pattern for the open-sink fold.

2.4.5 Bases for flat origami

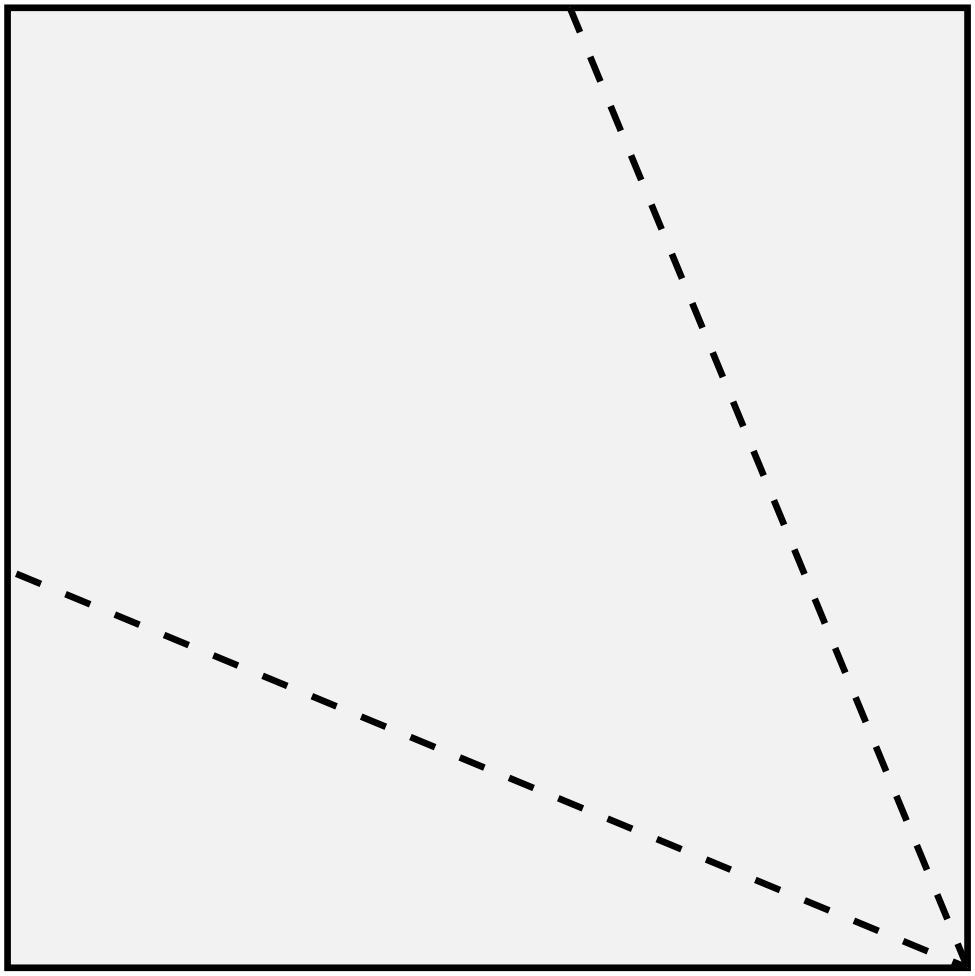


Figure 2.19: Pattern for the kite base.

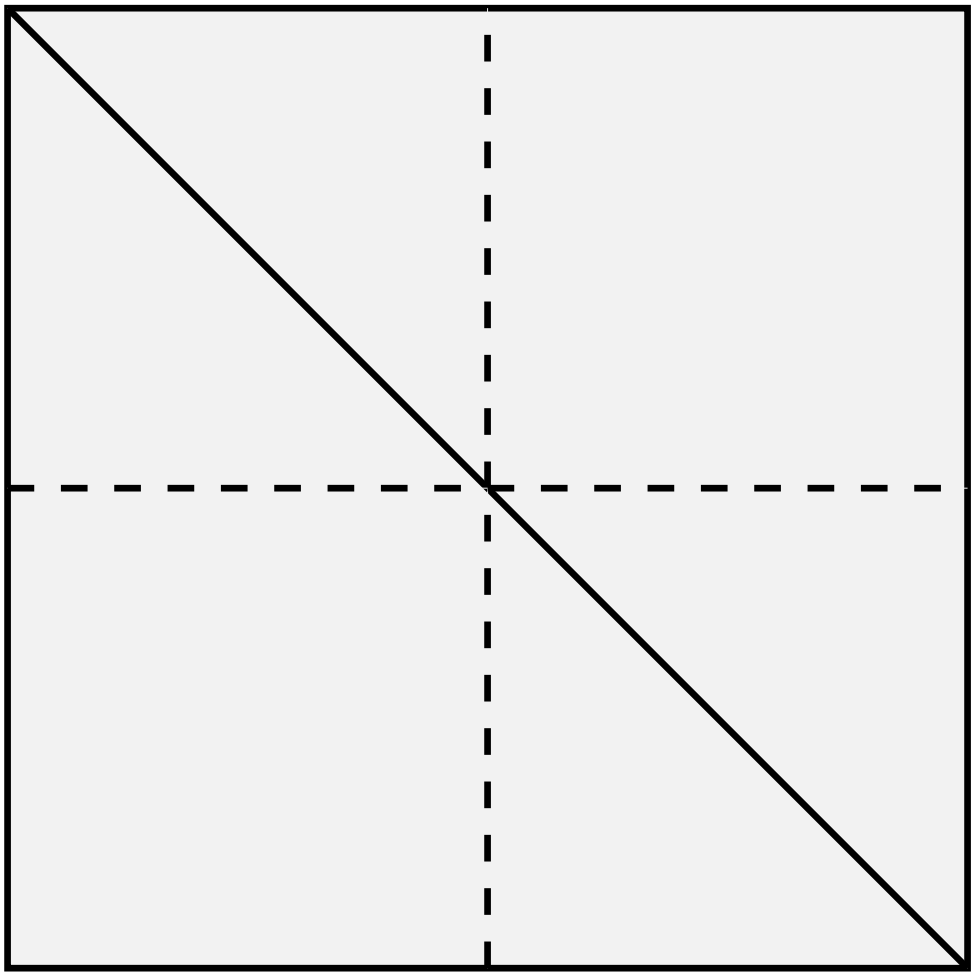


Figure 2.20: Pattern for the preliminary base.

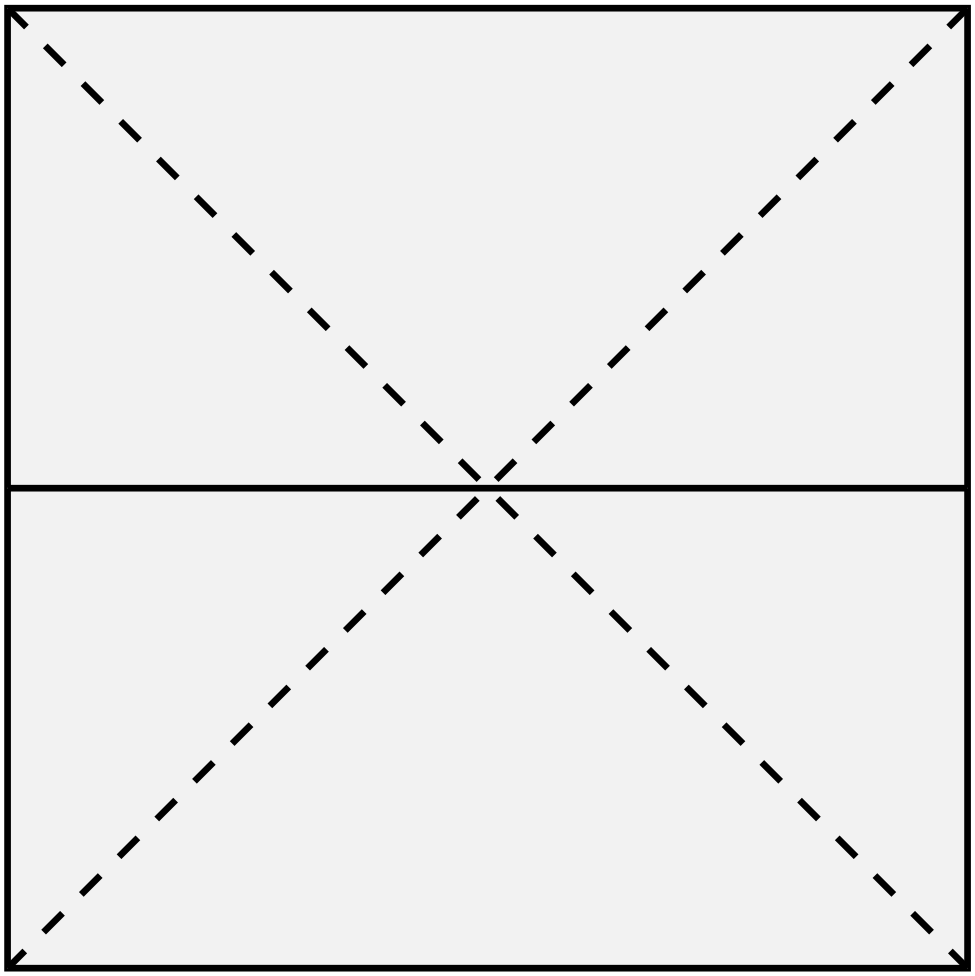


Figure 2.21: Pattern for the waterbomb base.

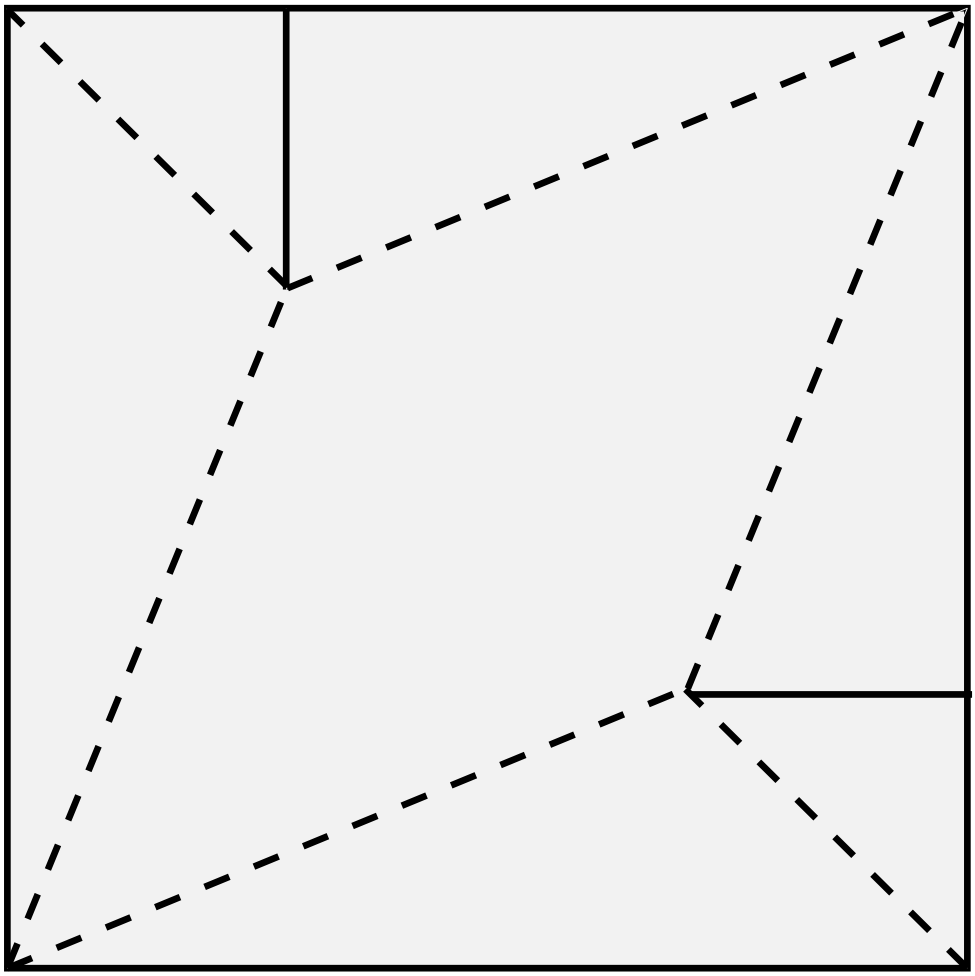


Figure 2.22: Pattern for the fish base.

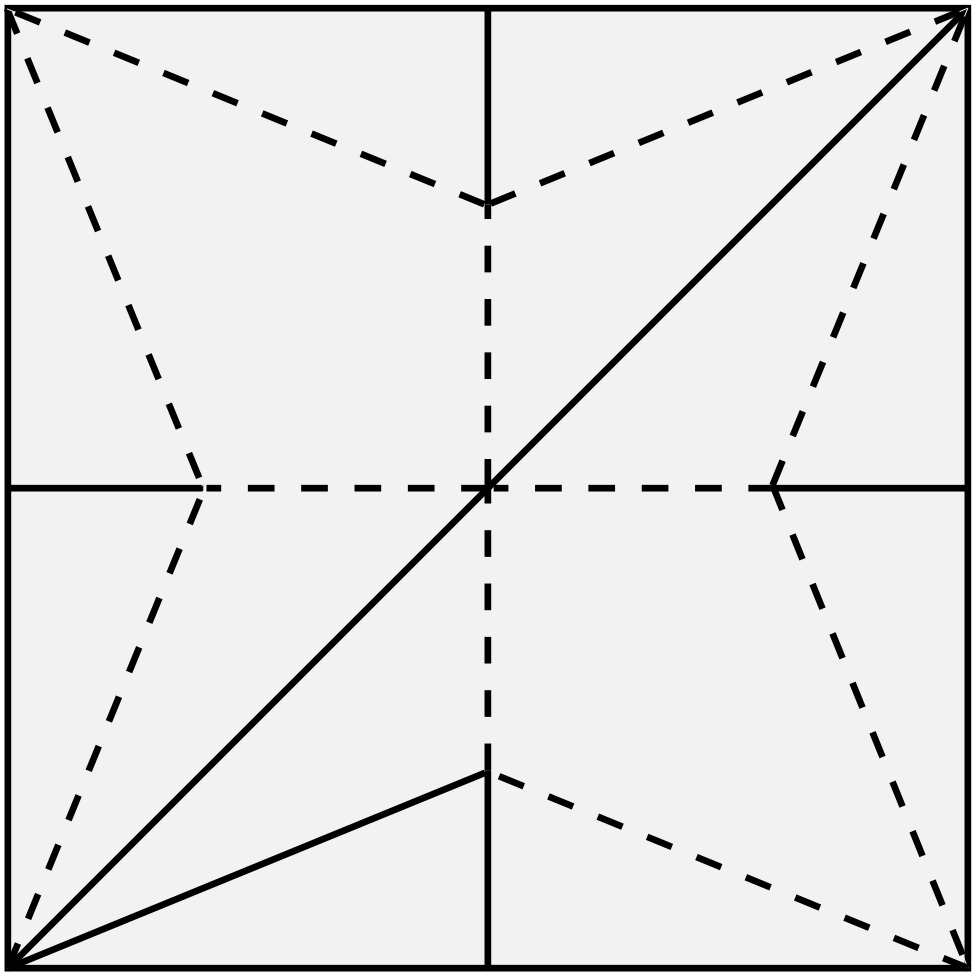


Figure 2.23: Pattern for the bird base.

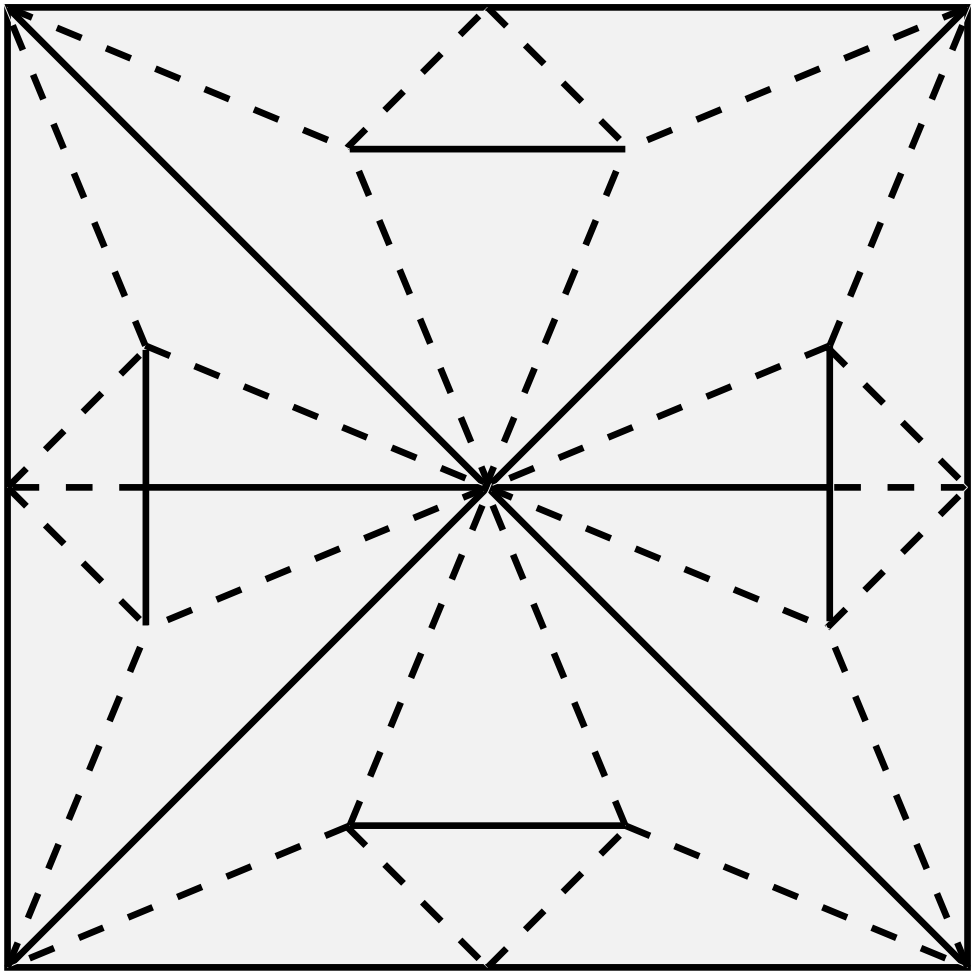


Figure 2.24: Pattern for the frog base.

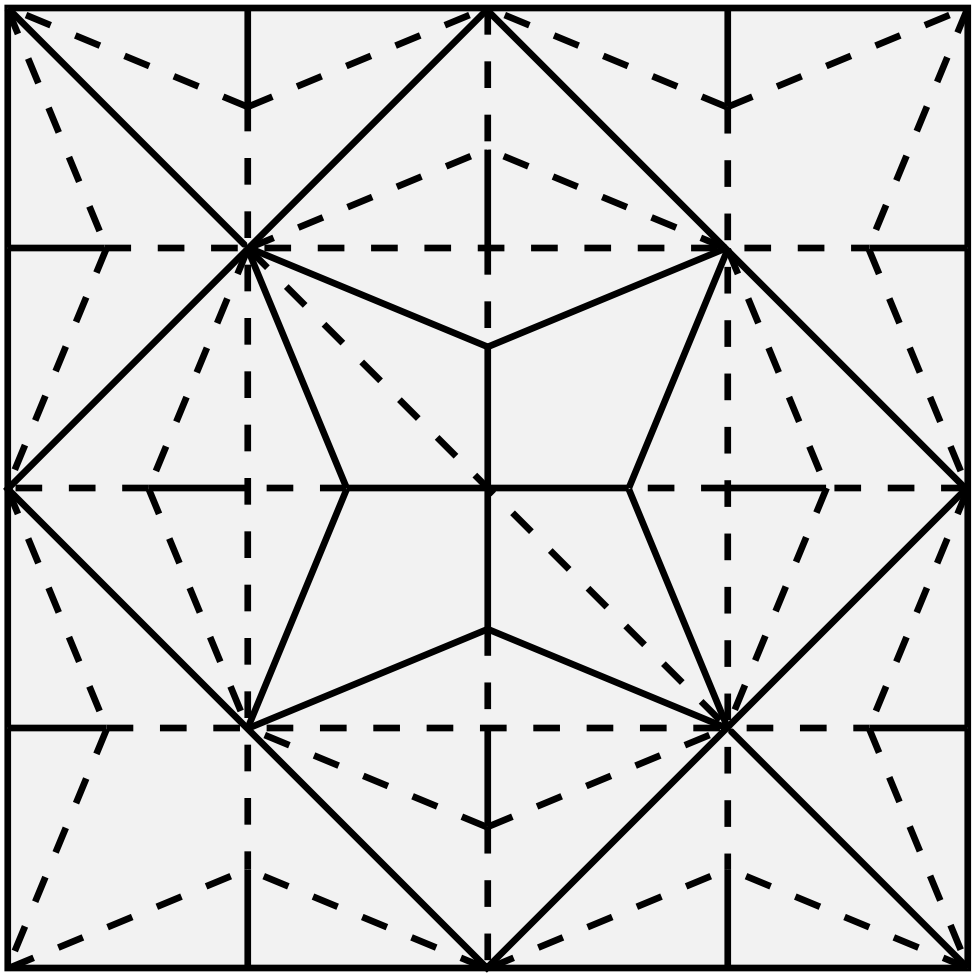


Figure 2.25: Pattern for the extended bird base.

2.4.6 Example flat crease patterns

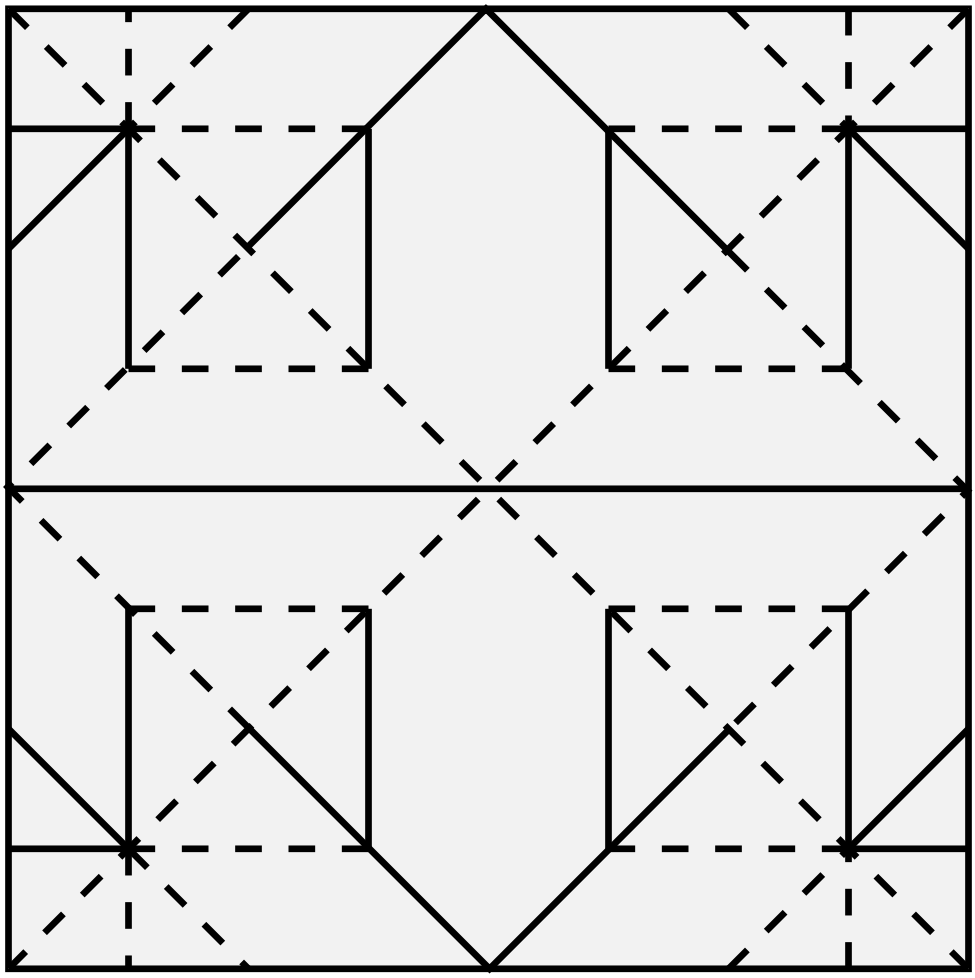


Figure 2.26: The pattern for an origami waterbomb.

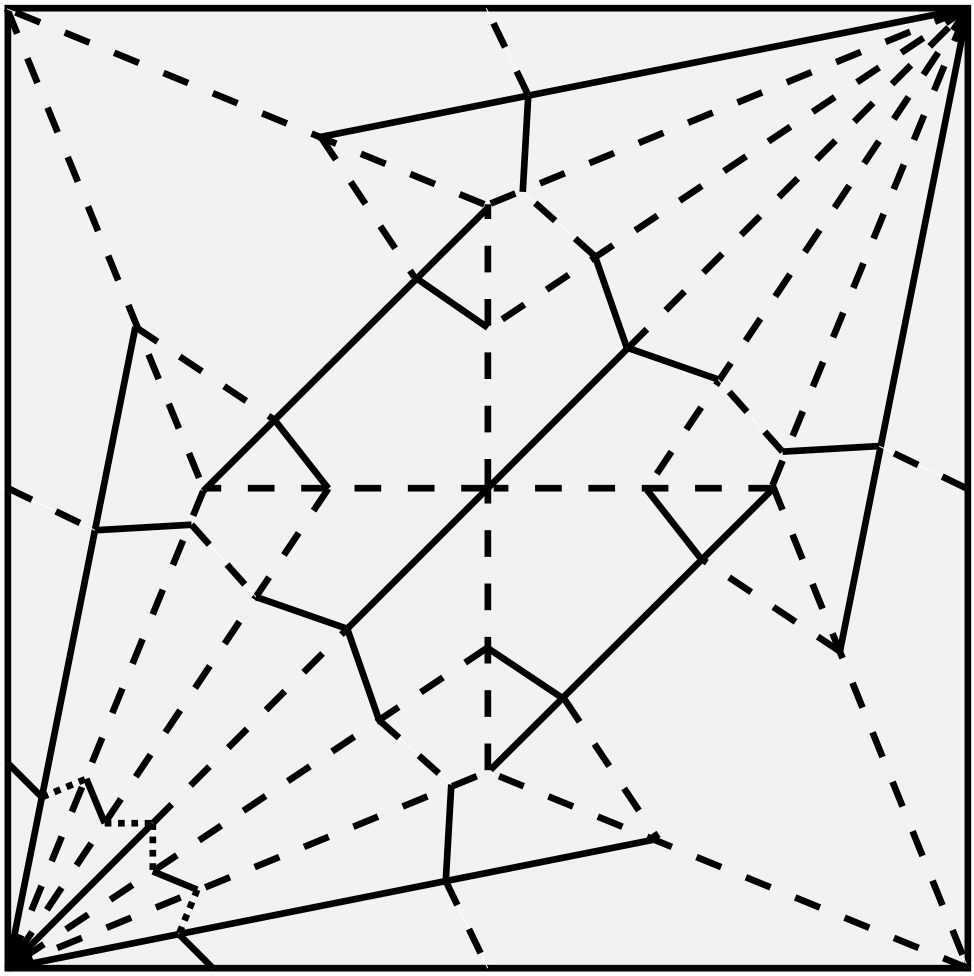


Figure 2.27: The pattern for a paper crane.

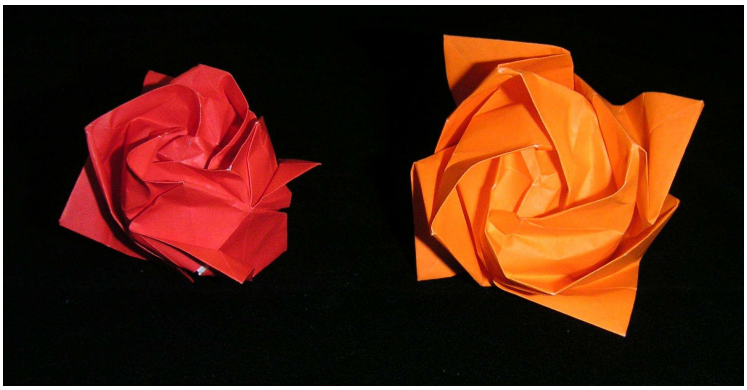


Figure 2.28: An example of 3D origami: two Kawasaki roses folded by Luis Pena.

2.5 3D origami

Although the folding of more complicated three-dimensional models of origami like those shown in figure 2.28 are largely beyond the scope of this paper, in this section we mention a few interesting observations. For more detail, the reader is referred to Robert Lang's excellent three-part article on 3D orgami folding techniques, posted online on the origami internet bulletin board [37].

One observation, confirmed by experienced origami folder Luis Pena, is that most of the creasing for a 3D model is done in an initial first stage, before any 3D folds are made. Consider the first subfigure of figure 2.29, which shows the precreasing done for the Kawasaki rose. One reason for the pre-creasing step may be that once the model becomes three-dimensional, it is much harder to add additional creases precisely, since the model is harder to grasp, and there may be no available flaps for landmarking.

How should the model be locked in its final form? As mentioned earlier in the chapter, coatings and glue are one approach. The choice of paper also seems essential, since it can be important that creases of less than 180° hold their shape. Techniques like wet-folding rely on the fact that thick paper, when folded wet, tends to stiffen as it dries.

What about other techniques? Curved creases are only possible in 3D origami, since folding curved creases bends the paper. But how should curved creases be created? One method is pre-creasing or light scoring with a knife blade. Other techniques include crumpling, and scraping one side of the paper to curl it. Although these techniques are beyond the scope

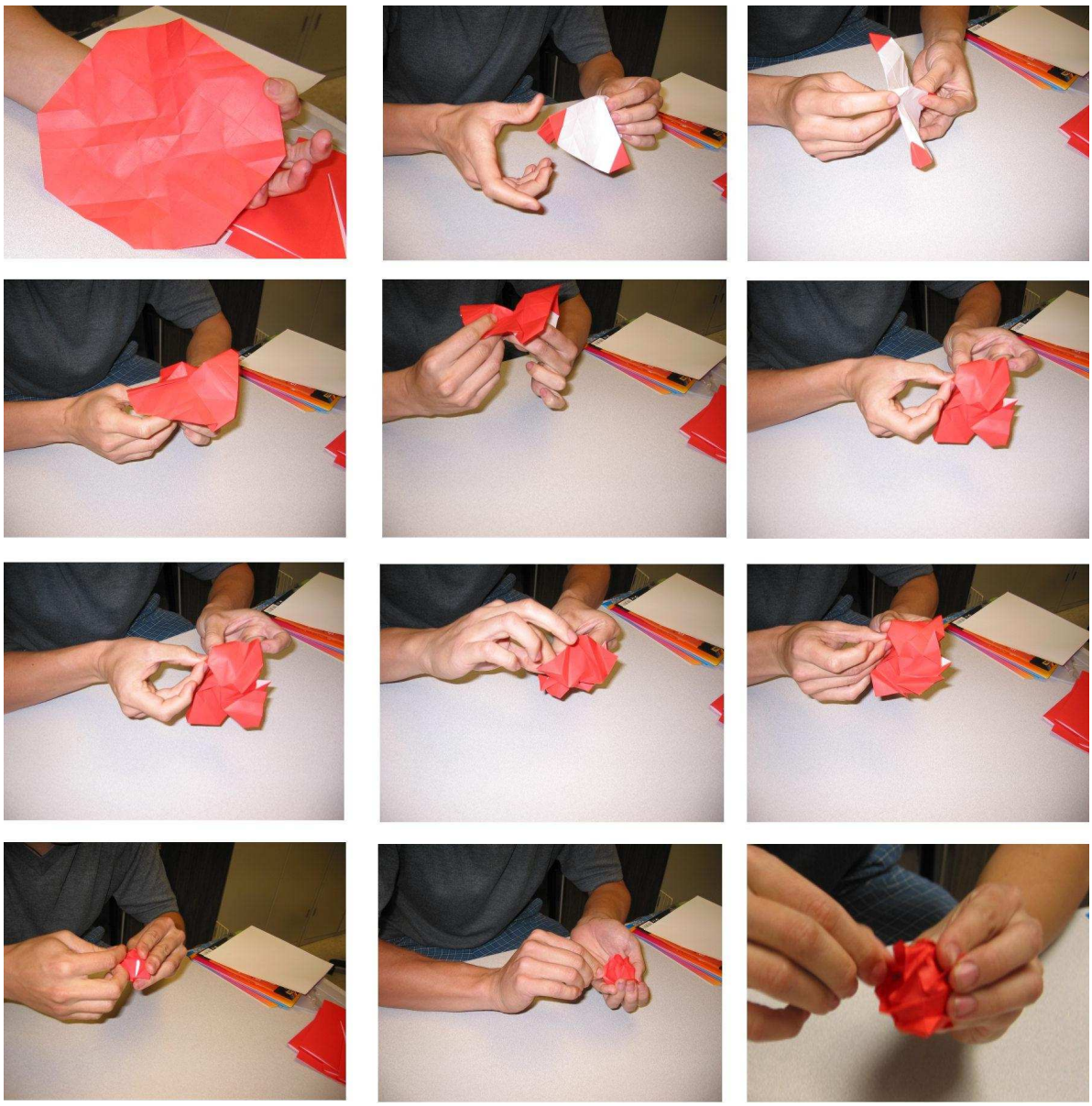


Figure 2.29: Three-dimensional paper manipulation used to fold the rose. (Folder: Luis Pena.)

of this thesis, they pose a fascinating challenge for future work.

Chapter 3

Representation and design

This chapter presents some of the simple models used in the thesis to describe origami folding, and surveys related work on modelling the state of paper, cloth, and bendable wire.

What mathematical models should be used to describe origami? There are a number of problems that make finding a single unified model of origami state difficult. Paper is thin, flexible, and not stretchy. The paper behaves like a spring when flexed, but creasing occurs when elastic limits are exceeded. Origami creases add layers of papers to each step of a fold exponentially, and natural creases (crinkling) can occur easily if too many constraints are applied to the edges of the paper. And what happens when creases are curved, or cross in the middle of the paper?

Models of paper tend to be driven by the manipulation or simulation task being considered, and can be classified as high-level or low-level models. Many origami designs are flat (essentially planar) after each fold is made, and most creases are not curved. The simplest high-level model of origami therefore uses revolute joints to model creases, and rigid bodies to model the uncreased portions of the paper. This model is the primary model used by the thesis, and allows planning over flat origami states.

The rigid-body origami model also describes the folding of pre-creased origami, a primary technique of human origami folders, and provides a beginning point for understanding more complicated origami-like mechanisms; *e.g.* paper shopping bags, teabags, airbags, and foldable mirrors.

The rigid-body model is not particularly useful for describing the creation of creases, or the dynamic motion of the flexible paper as it is actually manipulated. Therefore, this chapter also outlines some (very much more complicated) models that could provide an avenue for future exploration

of low-level folding techniques.

3.1 Related work; properties of paper

This thesis focuses on a model of paper that is very simple; uncreased facets of paper are considered to be planar rigid bodies, and creases are considered to act as revolute joints. However, for completeness, this section surveys a wide variety of work related to modelling flexible paper. Sections 3.1.1 and 3.1.4 provide necessary background material on the behavior of paper, and suggest directions in which our current model of origami could be extended. Huffman's results (section 3.1.3) are particularly relevant, and are further developed and extended in chapter 5.

Sections 3.1.5 through 3.1.10 describe work primarily concerned with the dynamics of flexible objects. Although relevant to future efforts at understanding robotic origami, dynamics are not considered in the thesis, and these sections may be skipped by most readers.

3.1.1 Developable surfaces

Paper stretches much less than materials like cloth or sheet metal, and assuming that it does not stretch at all may be a useful approximation. If paper does not stretch, the class of shapes it can assume without creasing is restricted – wrapping an initially flat piece of paper onto the surface of a sphere is impossible. The possible shapes are called *developable surfaces*. We will need some concepts from differential geometry to describe the characteristics of developable surfaces; Thorpe [62] is my reference for basic differential geometry, and Hilbert and Cohn-Vossen [25] has an extensive section on the properties of developable surfaces.

An *isometry* between two surfaces with defined dot products on their tangent spaces is defined as a continuous bijective mapping that preserves the dot products of tangent vectors. Since lengths of paths and areas of regions on a surface are defined in terms of dot products of tangent vectors, isometries preserve length and area.

The simplest isometries are rigid-body transforms (rotations and translations), but there are more complicated isometries. If an initially flat paper cannot stretch, then there must be an isometry between the flat paper and any uncreased configuration of the paper. A path drawn on the flat piece of paper will have the same length along the bent piece of paper, and a rectangle will have the same area.

Even if there is an isometry between two surfaces, it may not be possible to smoothly transform one surface into the other. A *bending* between two surfaces is a one-parameter family of isometries that continuously deforms one surface into the other. Consider a knotted piece of string. Glue the ends together to form a knotted loop. Although there is an isometry between the knotted loop and an unknotted loop, there is no bending between the two configurations that avoids self-intersection – the knot cannot be removed.

A property of a surface is said to be *intrinsic* if it is preserved under isometries. *Geodesics* are curves on a surface that have no component of acceleration tangent to the surface. The shortest paths on a surface are geodesics; on a flat piece of paper the geodesics are straight lines. Since isometries preserve length, it is not surprising that geodesics are intrinsic. So, the curves created by drawing straight lines on a flat piece of paper and then bending the paper are geodesics on the bent paper.

The *Gauss map* takes all of the (unit) normal vectors of a surface to the origin. Since all the normal vectors are of unit length, the image of a surface under the Gauss map must fall on a unit sphere centered on the origin. This unit sphere is called the *Gaussian sphere*. The image is called the *spherical indicatrix*.

If we draw a small closed path around a point p on a surface, the path encloses some area on the surface; call this area a. The image of the region within the path under the Gauss map has an area on the Gaussian sphere; call this area g. We define the *Gaussian curvature* G at p to be the limit of the ratio of these two areas as the area of the region withing the path goes to zero.

$$G(p) = \lim_{a \to 0} \frac{g}{a} \tag{3.1}$$

There is another way to find the Gaussian curvature. If we take the intersection of a surface with a plane that includes the normal at p, we get a plane curve which we call a *normal section* (or slice) at p. Define the *principle curvatures* at p to be the maximum and minimum curvatures of the normal sections, evaluated at p. The Gaussian curvature at p is the product of the two principle curvatures at p.

Surprisingly, Gaussian curvature is an intrinsic property of a surface. (Gauss' *Theorem Egregium* [19].) The Gaussian curvature of a plane is zero, since both principle curvatures are zero, and since the Gauss map takes the entire surface to a single point of zero area on the sphere. We define a *developable surface* to be a surface which is everywhere locally isometric to the plane. Because of this local isometry, the Gaussian curvature of a

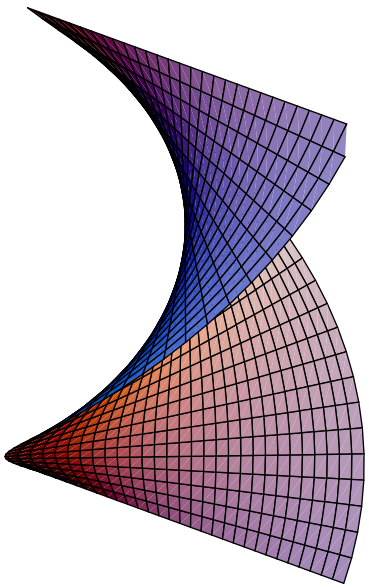


Figure 3.1: A ruled surface that is not a developable.

developable must also be zero everywhere. For example, a piece of paper can be folded into a circular cone. At any point on the paper, one principle curvature is zero (along the line from that point to the vertex), and the other is equal to the curvature of the circle formed by intersecting the cone with a plane containing the point and perpendicular to the line through the center of the cone. Since one principle curvature is zero, the product must be zero; Gaussian curvature is preserved.

For any surface with Gaussian curvature of zero, at least one of the principle curvatures must also always be zero at every point. From this it is possible to show that developable surfaces are *ruled surfaces*; through any point in the surface there is a line segment (a *ruling*) contained in the surface and extending to the boundaries of the surface. However, not all ruled surfaces are developables: consider the surface shown in figure 3.1, parameterized by (u, v), with $u \in [-1, 1]$, $v \in [0, \pi]$, and

$$x = u \cos v \tag{3.2}$$

$$y = u \sin v \tag{3.3}$$

$$z = v \tag{3.4}$$

The surface is generated by spinning and translating a segment of the x-axis of length 2 around and along the z axis. The rulings are the line segments

at each z value, but there is not an isometry between this surface and the plane.

A developable may be defined as a ruled surface for which the tangent plane is the same at any point along a line embedded in the surface. This gives an additional way to describe developable surfaces – as the envelope of a one-parameter family of tangent planes.

3.1.2 Representing paper by developable surfaces

A number of authors have used the geometric properties discussed above to derive representations of developable surfaces. Redont [55] used the zero-curvature property as well as the fact that geodesics are intrinsic to show that a developable can be described by a trajectory on the Gaussian sphere. Since the path on the Gaussian sphere gives the normals to the surface, the formulation is in terms of an ordinary differential equation, together with an initial condition. Although the differential equation usually cannot be solved analytically, Redont points out that if the trajectory on the Gaussian sphere is a circular arc, then the developable is a segment of a circular cone. Redont therefore proposes a method of approximating developable surfaces using C^1-connected circular arcs on the Gaussian sphere. Thus, the class of surfaces considered are composed of segments of right circular cones.

Sun and Fiume [60] used a representation similar to Redont's to build a geometric modelling program. The authors used their software to create models of a hanging scarf and of a bow made out of ribbon. Leopoldseder and Pottmann [38] have also explored the problem of approximating developable surfaces by right circular cones. They point out that one difference between their work and Redont's is that they are concerned primarily with approximating local properties of the general developable surface, whereas Redont's algorithm is global in nature.

Pottmann and Wallner [54] also propose an alternate representation of developable surfaces, based on the definition of a developable surface as the envelope of a one parameter family of tangent planes. Since four numbers can be used to represent a plane using homogenous coordinates, there is a duality between developable surfaces and trajectories in projective Cartesian space. The authors present metrics in the dual space, and use this to derive a method for approximating a set of tangent planes with developable surfaces of a certain class.

Weiss and Furtner [71] considered the problem of finding a developable surface that connects two space curves. The rulings of the developable are

used to connect the curves. However, arbitrarily connecting the two curves by rulings will yield a ruled surface, but not necessarily a developable; the additional constraint is that the tangent plane to the surface must be the same at each point along the ruling. The authors propose a metric measuring the extent to which the four endpoints of two adjacent rulings are co-planar. An iterative algorithm generates appropriate rulings, and thus a polyhedral approximation of a developable surface connecting the two curves.

Aumann [4] presents an important extension of Weiss and Furtner's work. Two general curves cannot always be connected by a developable – bending the edges of a piece of paper into certain shapes will lead to crinkling and creasing of the paper. Aumann considers the special case where the two curves to be connected are Bézier curves, and determines necessary and sufficient conditions for the interpolating developable patches to exist and be free of singularities.

3.1.3 Geometry of creases

The work on developable surfaces presents a detailed picture of the shapes a piece of paper can be bent into without creasing. But what happens if we crease the paper? Huffman studied this problem in [28], also from a geometric perspective. Huffman's motivating application was scene analysis. One goal of the work was to extend the generality of the models that could be considered in computer vision. Huffman wrote,

> Objects bounded by planes were reasonable ones upon which to do initial research in scene analysis... No two neighboring points on an arbitrary surface need have the same tangent plane. By contrast, all points on a plane surface have the same tangent plane. On a developable...all points on a given line embedded in the surface have the same tangent plane... [A] paper surface offers a complexity that is, therefore, in a very real sense exactly midway between that of a completely general surface and that of a plane surface. Consequently, paper surfaces constitute a class that may be ideally suited to be both richer than that of plane surfaces and more tractable analytically than that of totally arbitrary surfaces.

Huffman first examined the simpler problem of polyhedral vertices. Consider the vertex of the cube shown in the upper left of figure 3.2. The Gauss map takes each of the three faces to a point on the Gaussian sphere.

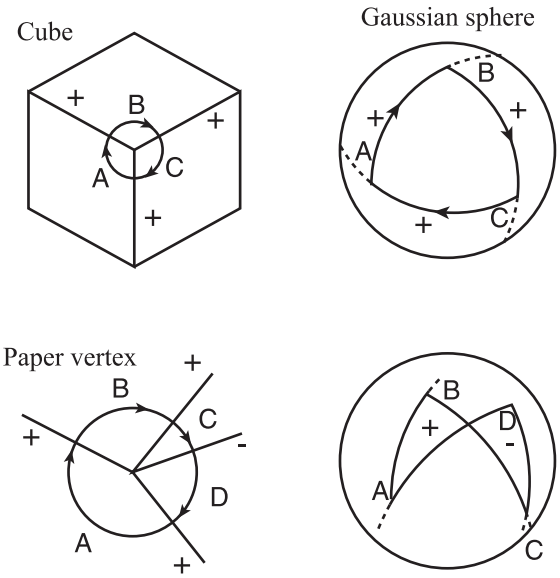


Figure 3.2: Polyhedral vertices on the Gaussian sphere. Re-drawn from [28].

We may consider the dihedral angles of the cube to correspond to edges connecting these points. Thus the Gauss map of the area of the surface enclosed by a small loop around the vertex is a spherical triangle (with edges that are segments of great circles) on the Gaussian sphere, shown in the upper right of figure 3.2. As the area of the loop shrinks to zero, the triangle on the Gaussian sphere is constant. Therefore, the Gaussian curvature at a vertex of a cube is infinite.

If we assign a direction to the loop around the vertex, we can associate a direction with each edge of the triangle on the Gaussian sphere. We consider the area of the triangle to be positive, since the area is enclosed in a clockwise fashion. (Area enclosed in a counterclockwise fashion would be considered to be negative.) The area of the triangle is $+\pi/2$. If the cube were cut along an edge and laid flat, the 'missing angle' would also be $+\pi/2$. In fact, a similar observation is true for all polyhedral vertices – the area of the region enclosed on the Gaussian sphere is equal to the 'missing' angle (positive area) or 'excess' angle (negative area) if a cut were made from the vertex along an edge and the facets were laid flat.

If we form a vertex by creasing paper, there will be no missing or excess angle. Huffman analyzed the simplest interesting example of a case where

the area on the Gaussian sphere is zero; this example is shown in the bottom of figure 3.2. Since three creases intersecting at a point will always lead to a triangle on the Gaussian sphere (with non-zero area), the simplest example must have four creases. Furthermore, the configuration must be such that either three of the creases are convex, and one concave, or vice versa.

The above discussion assumes that the faces of the polyhedron are rigid. However, if the faces are actually parts of the paper where there are no creases, the faces should be permitted to bend. For example, if a piece of paper contains only three creases which intersect at a point, then any non-flat configuration of the paper will require that the uncreased portions bend. In this case, each facet is a developable surface, and can be represented by a curve on the Gaussian sphere. Finally, Huffman considered the case of curved creases. The shape of the curve places a restriction on the orientations of the nearby tangent planes.

3.1.4 Modelling paper with natural creases

Sometimes paper is creased intentionally, and sometimes creases occur because there are constraints applied that are inconsistent with the paper remaining a smooth developable surface. We will call creasing of the second type *natural* creasing. Kergosien *et al* [35] (*Bending and Creasing Virtual Paper*) is the only work that I know of that combines the work on developable surfaces with a model of natural creasing.

In spirit, the work is most similar to that of Weiss and Furtner [71] and Aumann [4]. The location of rulings was used to describe the uncreased sections of paper; the locations of rulings were parameterized along a trajectory around the edge of each section. The locations were discretized and all bending was assumed to occur along rulings. It was shown that there is a linear constraint between the positions of the rulings and the 'developability' of the surface (the extent to which rulings share a common tangent plane at each endpoint). The authors implemented a graphical interface that allowed the user to apply spring forces to the surface. The forces typically deformed the object in a way that violated the constraint that the surface was developable. A constraint projection step was then used to ensure developability.

When the rulings of the paper began to cross, a creasing model was triggered. Thus, the paper was described by a data structure containing both developable patches and creased regions. The authors point out that the creasing patterns that may be used are non-unique; they studied both point creases and short line creases. The specific choice of crease type was

heuristic; placing the creases was posed as an optimization problem and solved using sequential quadratic programming.

3.1.5 Manipulation of flexible objects

The work on developable surfaces and creasing is concerned with what shapes paper (or other developables) *might* take, assuming that paper does not stretch. If there are enough additional constraints, this may be sufficient to determine the shape of the paper kinematically – for the purposes of high level motion planning, we could treat a piece of paper stretched tightly over a drumhead as a rigid body. Typically, however, the configuration of paper is determined by internal spring forces as well as external forces and constraints.

The simplest way of modelling these internal forces is to attach discrete springs to various parts of the flexible body. Another possibility is to use a constitutive law describing the potential energy as an integral of a continuous function representing the shape of the flexible object.

3.1.6 Rigid multibody dynamic simulation

Many of the most successful methods for simulating flexible objects treat the flexible object as a collection of a large number of rigid bodies. Therefore it is appropriate to briefly discuss efficient simulation of high-DOF rigid body systems.

The simulation techniques may be broadly classified as two types: those with implicit constraints (joint space, or generalized coordinates), and those with explicit constraints. The former techniques may make use either of Lagrangian or Newton-Euler formulations of the dynamic equations; the latter introduce Lagrange multiplier forces to maintain the constraints.

Craig [15] points out that the first algorithms used to simulate the dynamics of (serial) robot arms used a straightforwards Lagrangian formulation. The approach relied on calculating and inverting the (dense) mass matrix, and was $O(n^4)$ in the number of links. The first efficient ($O(n)$) algorithms were based on recursive Newton-Euler formulations of the dynamics, and efficient Lagrangian formulations were also eventually developed [15].

Simulating the dynamics of systems with closed loops is more difficult, and efficient solutions methods have only recently been proposed. As for arms, the methods are of two general types: those that deal with either only implicit constraints, and those that also permit explicit constraints.

Lagrangian representations with implicit constraints typically consider dynamics equations of the form

$$M\ddot{q} = f \tag{3.5}$$

where M is the mass matrix, q is the configuration of the system, and f is the vector of external and velocity-dependent forces. The structures of M and f depend on the choice of coordinates for q, and the choice of coordinates ensures that the constraints are always satisfied.

Unfortunately, it may be difficult to determine appropriate generalized coordinates to represent the constraints. Another difficulty with these approaches is that some choices of generalized coordinates lead to a system that is difficult to simulate efficiently. Fortunately, efficient dynamic simulation techniques that allow explicit constraints have also been demonstrated. In this case, the system consists of an equation similar to that of equation 3.5, together with a constraint equation of the form

$$g(q) = 0 \tag{3.6}$$

Since the constraint equation is satisfied at each time, the time derivative is also 0:

$$\frac{d}{dt}g(q) = J(q)\dot{q} = 0 \tag{3.7}$$

where J is the matrix of partials known as the *constraint Jacobian*. Simulation techniques that allow explicit constraints typically involve writing a matrix equation involving the external forces, the mass matrix, the constraint Jacobian, and the Lagrange multipliers (constraint forces). Since the mass matrix and the constraint Jacobian are sparse, sparse matrix methods can be used to efficiently solve for the Lagrange multipliers. Gleicher [20] used a conjugate gradient method; Baraff [7] presents an algorithm that is $O(n + m^3)$, where n is the number of links and m is the number of closed loops – the method is linear if there are no closed loops.

Baraff's algorithm is linear in the number of links, but cubic in the number of closed loops. Ascher and Lin [3] present an algorithm that is linear even if there are a large number of closed loops. The algorithm breaks each closed loop to form an open chain. At each time step, the open chain is simulated using a method similar to that used by Baraff [7] and others. An iterative method is used to satsify the loop closure constraints. It is shown that the number of iterations is independent of the number of links, although it is dependent on the topology of the configuration space. The algorithm

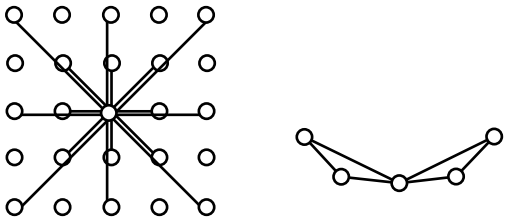


Figure 3.3: Connections between a central particle and its neighbors used by Choi and Ko [12]. Top view (left) and side view (right). Re-drawn from [12].

was applied to simulate the dynamics of a four-connected mesh with up to 1000 links (25×20). In each case only two constraint satisfaction iterations per time step were required. Simulations were conducted with a time step of .2 seconds, for a simulation time of a few seconds. No information on run-time was provided.

It is interesting that in addition to models that approximate flexible systems by high-degree-of-freedom systems of rigid bodies, there are also examples of approximating discrete high-degree-of-freedom rigid-body systems by continuous models – for example, Minksy's elephant trunk arm [49] and Chirikjian's work on the inverse kinematics of high-DOF binary manipulators [11].

3.1.7 Cloth simulation

Impressive dynamic simulations of cloth have been achieved by modelling the cloth as a high-DOF system of rigid bodies. Some of the most notable recent techniques are presented by Choi and Ko [12], and by Bridson *et al* [9]; these papers also present a more complete survey of previous cloth simulation work than is possible here. [12] focuses on the internal dynamics of the cloth, and [9] concerns itself primarily with problems of contact and friction.

Most cloth simulation algorithms do not place hard constraints on the relative motion of particles or small facet elements; rather, stiff springs are used to keep the cloth from stretching very much. As an example, figure 3.3 shows the configurations of springs and particles used by Choi and Ko [12]. Typically, these stiff springs introduce instability into the dynamic simulation. The usual solution, proposed first by Baraff and Witkin [8], is to use implicit integration techniques. Implicit integration formulations may also

ameliorate problems related to stiffness in the differential equations due to contact and friction; for example, see Stewart and Trinkle [58]. (For a discussion of implicit integration methods, see the referenced papers.)

Since there are no constraints, and since the only direct interactions between particles of the cloth are local, the mass matrix is sparse. Iterative sparse matrix methods (*e.g.*, conjugate gradient) are therefore used to solve the dynamic equations efficiently. External contact forces due to collisions, friction, and self-intersection are usually handled by attaching virtual springs at contact points.

3.1.8 Haptic simulation

Cloth simulations do not typically run in real-time; even the most efficient algorithms require minutes or hours to create a realistic-seeming dynamic simulation. Therefore, motion planning or control of flexible objects using these algorithms is problematic.

A different perspective on simulation of flexible objects is provided by a number of papers on haptic simulation; James and Pai [31] provides a good survey. Unlike the cloth simulation algorithms, haptic simulation algorithms typically use a quasistatic model. That is, it is assumed (or proven) that if forces are applied to a flexible object, then it will eventually reach an equilibrium configuration. *Green's functions* relate displacements of the material to forces applied.

James and Pai represent the surface of a flexible object by a set of discrete points, or nodes. Each node is either fixed in space, or has forces applied to it. The set of nodes that are fixed describes the problem type. (Typically, the base of a flexible object might be fixed, while the remainder of the nodes would be free. Poking at the object with a finger would introduce a new spatial constraint, and change the type.)

The authors consider linear models; models for which (by definition) there is a linear relationship between displacements and applied forces. If the model is linear, there is a linear basis for the Green's functions. This basis can be computed off-line. Once the basis has been computed, simulation can be carried out quickly by simple matrix multiplication.

The linear model also allows efficient re-use of computation if the type of the problem changes slightly (for example, if a finger pokes the object). Computing the response of the system to a set of forces or constraints requires a matrix inversion. If the type of the problem is known, this inversion can be done off-line. If the type changes slightly, *capacitance matrix algorithms* can be used to efficiently update the inverse.

3.1.9 Fourier models

Hirai *et al* [27] modelled the shape of a cross section of a piece of bent (but not creased) paper. Fourier coefficients were used to describe the shape of the paper. The model was used to find equilibrium configurations of the paper under a set of geometric constraints. The authors write an equation for the potential energy in terms of the Fourier basis coefficients, and use non-linear optimization software to minimize the energy. They used five coefficients, and demonstrated experimentally that the model predicted the behavior of a piece of copy paper reasonably well, for some simple examples. In [65] a similar approach was used to model yarn in a knitted piece of fabric.

One difficulty of the method described is the problem of bifurcation. There may be two or more possible configurations of the paper consistent with a given set of geometric constraints. Non-uniqueness of solutions is a familiar problem in physical simulation algorithms. For example, it is well-known that the problem of determining the accelerations of two contacting rigid bodies under the Coulomb friction assumption may have no solutions, one solution, or many solutions (see Painlevé [53], Lötstedt [41], Erdmann [17], and Stewart [59]).

There are various approaches to dealing with the problem of non-uniqueness of solutions for the purposes of simulation. The simplest is to modify the assumptions so that the solutions are unique; this is the approach taken by Lötstedt [41], and Anitescu and Potra [1] in designing their rigid-body simulation algorithms. For the purposes of planning, analysis of the model to determine when the solutions are non-unique or non-existent may allow plans to be generated that are guaranteed to work in spite of the uncertainty; for the rigid body contact planning problem, this is the approach taken by Erdmann [17], Trinkle *et al.* [64], and Balkcom and Trinkle [6].

Wada *et al.* [68] extended the Fourier model developed in Hirai *et al* [27] to the case of a rod in three dimensions, and considered dealing with the bifurcation problem by using an optimizer that tends to find local rather than global potential energy minima. This approach is not really satisfactory, since it is not clear what metric should be used to decide which minima are 'close' and which are 'far'; I expect that some modelling of dynamics is necessary to determine which of several local minima the system eventually reaches.

Wakamatsu *et al.* [70] considered the problem of simulating the dynamics of rodlike objects. If the system is conservative, then the trajectories it follows must minimize the integral of the difference between kinetic and

potential energies, subject to the constraints. The authors wrote equations for the kinetic and potential energies as functions of Fourier basis coefficients that were used to approximate the rod's configuration and velocity. They then used non-linear optimization software to solve for the accelerations at each discretized time step.

There is an interesting connection between the problem of finding minimum energy configurations of a rod and finding optimal (or near-optimal) trajectories for robots. Although the application is much different, the algorithm proposed by Hirai *et al.* [27] appears to be identical to that used by Fernandes, Gurvits, and Li in *A Variational Approach to Optimal Nonholonomic Motion Planning* [18].

The Fourier model just discussed uses a finite number of variables to approximate the configuration of a flexible rod. This method is similar to classical techniques used to analyse vibration. Symon's *Mechanics* [61] describes the configuration of a vibrating string by an infinite Fourier series. In this case, the series describes the x and y coordinates of each particle, rather than the angle of the tangent. As a result, there is no implicit arc length constraint, and the string can stretch. The dynamics are modelled, and the frequency of vibration is determined analytically.

3.1.10 Continuum models

There is a vast field of research on elasticity and the mechanics of continua. Only the briefest summary is possible here; Antman [2] provides a good survey. Typically, the configuration of the flexible object is represented by a parameterized function. Constitutive laws are formulated to describe the local behavior of the material. Potential and kinetic energy are described as functionals. Various techniques are then used to analyze the behavior. For example, minima of the potential energy functional correspond to equilibrium states; variational approaches attempt to solve for the equilibria directly, or, when that is not possible (the usual case), used to find properties of the equilibria (*e.g.* bifurcation points, geometric properties, *etc.*).

The method of Lagrangian dynamics can also be extended to objects whose configurations are described by continuous functions. Chapter 13 of *Classical Mechanics* by Goldstein *et al.* [21] describes this technique in detail. It turns out that Lagrange's equations yield partial differential equations describing the dynamics, rather than ordinary differential equations.

What if the flexible object is thin, like paper or string? Pai [52] considered the problem of simulating thin elastic solids that both bend and twist. Pai points out that "modelling these [objects] as 3D elastic solids

requires very fine FEM meshes to correctly capture the global twisting behavior... models using meshes of mass particles and springs have similar problems since they require a large number of particles and springs..."

Pai uses a *Cosserat* model to describe the behavior of a thin elastic rod that can twist – a *strand*. The configuration is described by the trajectory of a frame of three *directors*. One of the directors is the tangent vector to the strand; the other two describe the twisting. If the trajectory is of constant speed, then the strand may bend but does not stretch. Pai formulates (ordinary) differential equations describing equilibria for the case where one end is fixed and force is applied at the other end. He discretizes the equations, and presents a linear-time algorithm to solve the discretized equations.

Cosserat models also exist for shells and points, as well as for rods (strands). Antman [2] and Rubin [56] are standard sources. Rubin points out that an advantage of modelling flexible objects using thin, directed media is that thinness may simplify the form of the dynamic equations. In tabular form,

Model	Dynamic equations
Cosserat point	ODE in time
Cosserat rod	PDE in time, and in one spatial coordinate
Cosserat shells	PDE in time, and in two spatial coordinates
3D elastic	PDE in time, and in three spatial coordinates

Since ODEs are easier to solve than PDEs, Rubin also presents a number of methods to numerically simulate Cosserat rods, shells, and general 3D elastic materials using a collection of points.

3.1.11 Origami mathematics and design

There is a rich field of work on the mathematics of origami design. Demaine *et al.* [16] is a good survey. According to [16], the field "essentially began with Robert Lang's work on algorithmic origami design, starting around 1993." Given a desired *origami base* (an origami silhouette from a restricted class of shapes) Robert Lang's *TreeMaker* software (described in [36]) finds a crease pattern allowing the paper to be folded into the base.

Demaine *et al.* [16] classifies work in computational origami as *universality results*, *efficient decision algorithms*, and *computational intractability results*. As an example of universality results, the authors state that "any

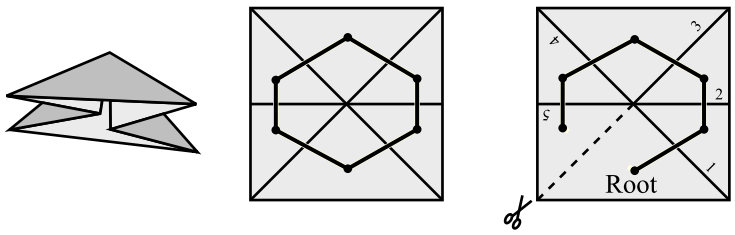


Figure 3.4: The pattern, facet graph, and a facet tree for a waterbomb base.

tree-shaped origami base, any polygonal surface, and any polyhedral surface can be folded out of a large enough piece of paper". As an example of an efficient decision algorithm, "there is a polynomial time algorithm to decide whether a ... grid of creases marked mountain and valley can be folded by a sequence of simple folds." Intractability results include that the problem of determining whether a crease pattern can be folded flat is NP-hard, and that "given a crease pattern... finding the overlap order of a flat folded state is NP-hard".

Miyazaki *et al.* [50] describes software that allows the folding of "virtual" origami by the user. The origami is treated as a collection of rigid facets connected by hinge joints. Two basic primitives are designed: *folding* and *tucking in*. During folding, facets rotate around a single hinge joint. During tucking, a pair of facets connected by a hinge joint is reflected through a plane perpendicular to the facets. The authors were able to use their system to virtually fold a crane and a paper airplane.

3.2 Rigid-body origami models

The goal of a paper folding task is to achieve some final state of the paper – a particular shape of the paper surface. Origami books usually describe paper folding tasks by a series of instructions. The instructions encode both intermediate states and the goal. As long as only common folding techniques are used, the instructions are simple to represent. The primary limitation of this method is that no origami piece can be described until the folding technique (including folding skills and the correct sequence) is known.

Planning and simulation techniques for mechanisms in robotics and computer graphics often describe the state of a mechanism as a set of floating point numbers representing joint angles. These representations work

well as long as the configuration of the mechanism is not too near a self-intersection. Unfortunately, in folding manipulation, facets of origami almost always touch.

The representation of origami that will be used for the majority of the thesis uses both continuous crease angle information and discrete values to represent the relationship between facets that almost touch. The kinematics of the mechanism are described by the locations of the creases on the unfolded paper – the origami *pattern*. Flat folded origami can be described by the pattern, the relative heights of the facets in the folded state, and the constraint that the final shape be flat. Representations of this type are useful because they allow descriptions of tasks for which the goal is known, but for which no simple set of folding instructions exists. We can then search for ways to fold the origami using only known folding actions, or even design new folding actions to suit the task.

3.3 Line-segment origami with revolute joints

This section uses a very simple class of origami to introduce some key features of origami representations. Consider a strip of paper with parallel crease lines. Cross-sections of the facets can be represented by line segments. Number the segments from left to right on the initially unfolded paper, and fix the pose of the first segment.

Origami is flat if all crease angles are 0° (unfolded), 180° (a *valley fold*), or -180° (a *mountain fold*). However, the crease angles do not fully determine the configuration of the origami; see figure 3.8.

The fold between the first two segments of our cross-sectional origami will be on the right, regardless of whether we fold segment two up or down; the fold between the next two segments must be on the left. Therefore, if we know the locations of creases on the unfolded paper (the origami *pattern*), we can calculate the locations of each facet on the folded piece. Specifically, describe the location of each segment by a real number x_k, and the length of each segment by a real number l_i. For $k \in [2 \dots n]$,

$$x_k = x_1 + \sum_{i=1}^{k-1} (-1)^{i+1} l_i \tag{3.8}$$

Assume that the top of the unfolded origami is colored. Once the origami has been folded, segments with odd indices will have the colored side up, and segments with even indices will have the white side up.

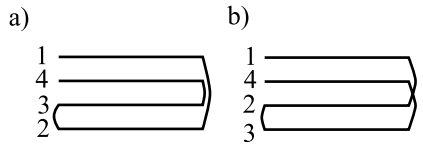


Figure 3.5: On the left, a feasible stacking. On the right, a stacking that is not feasible due to collisions between facets.

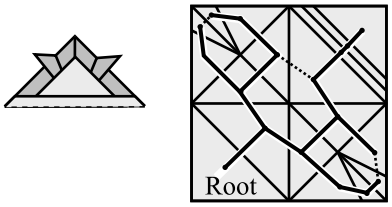


Figure 3.6: The pattern, facet graph, and a facet tree for a samurai hat.

The location and top color of each segment are independent of the folding method, order, and direction. Folding order and direction do determine the heights of each segment. We can describe the relative heights of each segment by an ordered list of segment indices. For example, the list (1 3 2) would be read as: segment two is on the bottom, segment three is on top of segment two, and segment one is on the top. We call the ordered list of segment heights an origami *stacking*. In fact, partial orderings are also possible, since two non-intersecting segments can sometimes be considered to have the same height; this will be discussed more fully below.

We can determine the crease angles using the stacking and top colors. If segment i is colored, and segment $i+1$ is on top of segment i, then fold i is a valley fold. The other three cases are similar.

How many ways are there to fold a pattern? There is an upper bound of $n!$, if we consider all possible orderings of the segments. However, there are some stackings that are not possible, for some lengths of segments. For example, consider the stacking (1 3 2 4), with segment lengths all equal. There would have to be a right-hand side fold between one and two, and another between three and four. However, as figure 3.5b shows, this configuration is not possible.

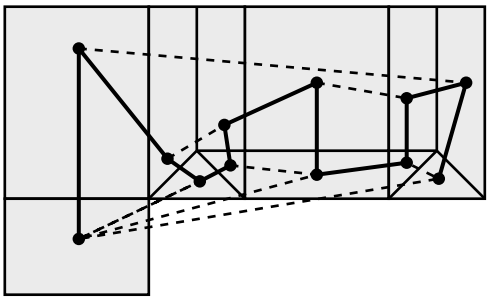


Figure 3.7: The pattern, facet graph, and a facet tree for a paper shopping bag. The bold lines show the facet tree; the dashed lines show the cut edges of facet graph.

3.4 Faceted origami with revolute joints

The model of origami that will be used for the majority of the thesis takes each facet as a rigid link and each crease as a hinge joint; this model is similar to those used by Huffman [28] (geometry of creases), Lu and Akella [43] (carton folding), and Gupta *et al* [22] (sheet-metal bending). Even for this simple model of origami, the kinematic structure may be quite complicated. If creases meet on the interior of the paper, then the structure includes closed chains.

Define the *facets* of an origami piece to be the unfolded regions of paper. In the present work, we model only polygonal facets, and treat them as rigid links. The facet edges interior to the paper are *creases*; a *crease line* is a set of colinear creases. Define the *crease pattern* to be the location of creases on unfolded origami. Creases meet at interior vertices of the pattern; if n creases meet, we say that a vertex is of *degree* n. The angles between creases around a vertex in the pattern are called *sector angles*.

Origami kinematics

Take each facet to be a node of a graph, and connect adjacent facets on the pattern with an edge; we will call this graph the *facet graph*. We say that any tree that spans the facet graph is a *facet tree*. Facet trees are easy to construct; any complete search method such as breadth-first or depth-first search is suitable. Creases not contained in a facet tree will be said to be virtually (but not necessarily physically) *cut* relative to that tree.

A facet tree implies a parent-child relationship between two facets con-

nected by a crease. We will choose the convention that all facets are described by a counter-clockwise set of points in the pattern; we will associate a unit vector with each crease such that the vector's direction agrees with the order of vertices in the child facet. We then describe the *crease angle* as the angle between a parent facet and its child; the sign is chosen to be consistent with the 'right-hand rule' applied to the crease vector.

Given a pattern and any facet tree, the crease angles associated with all uncut creases determine the kinematics of the origami mechanism – the pose of each facet and the angle of each cut crease can be determined by traversing the facet tree applying rotations to descendent facets.

Facet trees also allow the determination of the *mobility* of the system, a lower bound on the number of degrees of freedom. Assume there are n_c creases and and l loops in the facet graph. The facet tree has $n_c - l$ uncut creases. Since each loop around any vertex is a spherical closed chain, each loop closure removes an additional 2 freedoms. The mobility m is therefore

$$m = n_c - 3l. \tag{3.9}$$

For the waterbomb base, $n_c = 6$ and $l = 1$, so $m = 3$. In a generic configuration like the one shown on the left side of figure 3.4, there are three degrees of freedom. But sometimes the constraints are dependent. Consider the waterbomb pattern in an unfolded configuration (right side of figure 3.4). There are locally *four* independent directions of motion for the mechanism. It is possible to mountain or valley fold along any of the three crease lines, and also possible to 'prayer' fold, bringing creases 2 and 5 towards each other until the configuration shown on the left side of the figure is reached. (This folding is shown in figure 2.5.) We will return to this case in chapter 5.

The mobility of some crease patterns is not as important as for others. For the samurai hat, $n_c = 20$ and $l = 3$, so $m = 11$. However, as we will see in chapter 4, it is possible to fold the samurai hat using a succession of mountain and valley folds; during folding, all creases with a value not one of $\{-\pi, 0, \pi\}$ are colinear and can be treated as a single crease.

For the shopping bag with flattened pattern shown in figure 3.7, $n_c = 18$ and $l = 8$. According to the formula, $m = 18 - 24 = -6$. This implies that constraints must be dependent for any valid configuration of the shopping bag. As we will see in chapter 5, it turns out that the configurations for which enough constraints are dependent are the open and folded configurations; the configuration space of the shopping bag is just isolated points. This might be considered a design feature; facets must be bent if the bag is

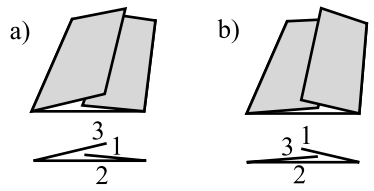


Figure 3.8: Orthogonal and cross-sectional views of two nearly flat origami pieces. Once completely folded, the crease angles will be the same, but the two origami pieces are clearly different.

in any configuration that is not fully open or fully closed, and since facets resist bending, the bag tends to stay in the open or closed configuration it is put into.

Stacking order and compound facets

Since origami is flexible and can be folded essentially flat, it is convenient to allow crease angles in the range $[-\pi, \pi]$. Flat origami illustrates a difficulty with a purely 'kinematic' model, however. Figure 3.8 shows the problem: although in the limit the crease angles of two flat origami pieces may be the same, the order in which facets are stacked is important. We will call a group of coplanar facets a *compound facet*. With each compound facet we associate a normal vector and a facet *stacking* relative to this vector. The height of a facet is its height in the stacking, and the height of a crease is the height of its child facet.

Before the paper is folded, there is a single compound facet, with normal pointing upwards; each facet height is zero. As planning or simulation takes place, compound facets break and form depending on the fold executed. Both 'simple folding' and 'book folding', defined in chapter 4, break a single compound facet into two compound facets; one of the compound facets is then flipped, and the two compounds are combined into one new compound.

Given a stacking order, it is possible to find a *minimal stacking order* that minimizes the height of each facet. Apply a bubble sort to the facets of height 1 to height n, using polygon intersection to determine whether the current ('bubble') facet moves downwards.

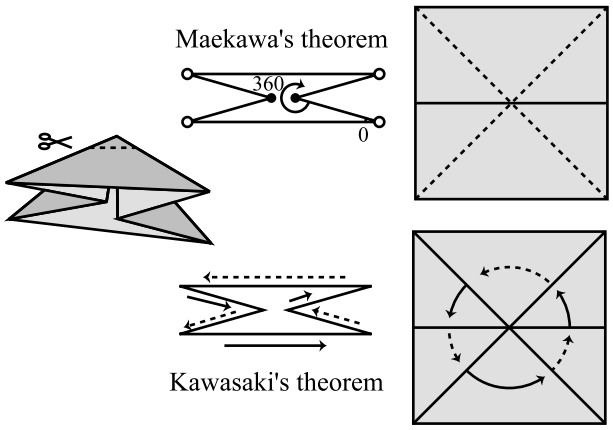


Figure 3.9: The proofs of Maekawa's and Kawasaki's theorems.

3.5 Properties of flat origami

'Flat origami' that is two-dimensional when folded has some special properties when considered from the perspective of the rigid-body model described above. This section presents well-known local properties that apply to individual crease vertices, and also some global conditions that apply to origami with multiple vertices connected by creases. The typical approach is to consider the origami in the flat, folded configuration, and determine how this state is constrained.

3.5.1 Local properties of flat origami

Two well-known theorems describe necessary conditions for origami to be flat-foldable. Maekawa's theorem describes origami in the folded state, and Kawasaki's theorem describes the unfolded crease pattern. Figure 3.9 illustrates the following proofs.

Theorem 1 (Maekawa) *In any flat-foldable origami, the difference between the number of mountain and valley creases around any vertex is two.*

Proof: [(due to Jan Siwanowicz, according to [29]):] Cut the paper near the vertex with a circle. The lengths of the arcs are proportional to sector angles on the crease pattern, and the arcs form a polygon on the circle. (Note that the polygon can be 'unbent' so that it is planar.) For any closed polygon, the sum of interior angles is $(n-2)180$, and any flat closed polygon has an even number of vertices. For each mountain fold, the interior

angle is 0, and for each valley, 360. So

$$0 + 360V = (M + V - 2)180 \tag{3.10}$$
$$M - V = 2 \tag{3.11}$$

If we flip the paper over, mountain creases become valley creases and vice versa. In this case, $V - M = 2$. ■

Theorem 2 (Kawasaki) *For any flat origami, the odd (and even) sector angles around any vertex sum to 180.*

Proof: [Thomas Hull [29]):] Cut across the paper near the vertex. Note that $\sum \text{odd} - \sum \text{even} = 0$, since odd go right, even go left, and we need to get back to the starting point. Since $\sum \text{odd} + \sum \text{even} = 360$, both the odd and the even angles must sum to 180. ■

We can generalize Kawasaki's theorem to vertices with missing angle:

Corollary 1 *For any flat origami, the odd (and even) sector angles around any vertex sum to half of the sum of the sector angles around the vertex. That is, if γ is the 'missing' or 'excess' angle,*

$$\sum \text{odd} = \sum \text{even} = (360 - \gamma)/2. \tag{3.12}$$

As an example, consider the corner of a collapsible garment box. The sector angles sum to 270°, and since the box is flat-foldable, the odd and even sector angles each sum to 135°.

Kawasaki's theorem and Maekawa's theorem are not sufficient to prove flat-foldability of even a single vertex, since neither considers the problem of self-intersection – as an example of a pattern that satisfies both theorems but is not flat foldable, see figure 3.10. (To prove that the pattern is not foldable with the mountain-valley assignments and sector angles shown, consider that the associated flat polygon self-intersects.)

3.5.2 Global properties of flat origami

Necessary and sufficient conditions for a multi-vertex pattern to be flat-foldable are not known, but it is possible to derive some further necessary conditions.

A flat origami piece can be described by the (planar) poses of each of the facets, and the stacking (relative heights) of the facets. If the origami

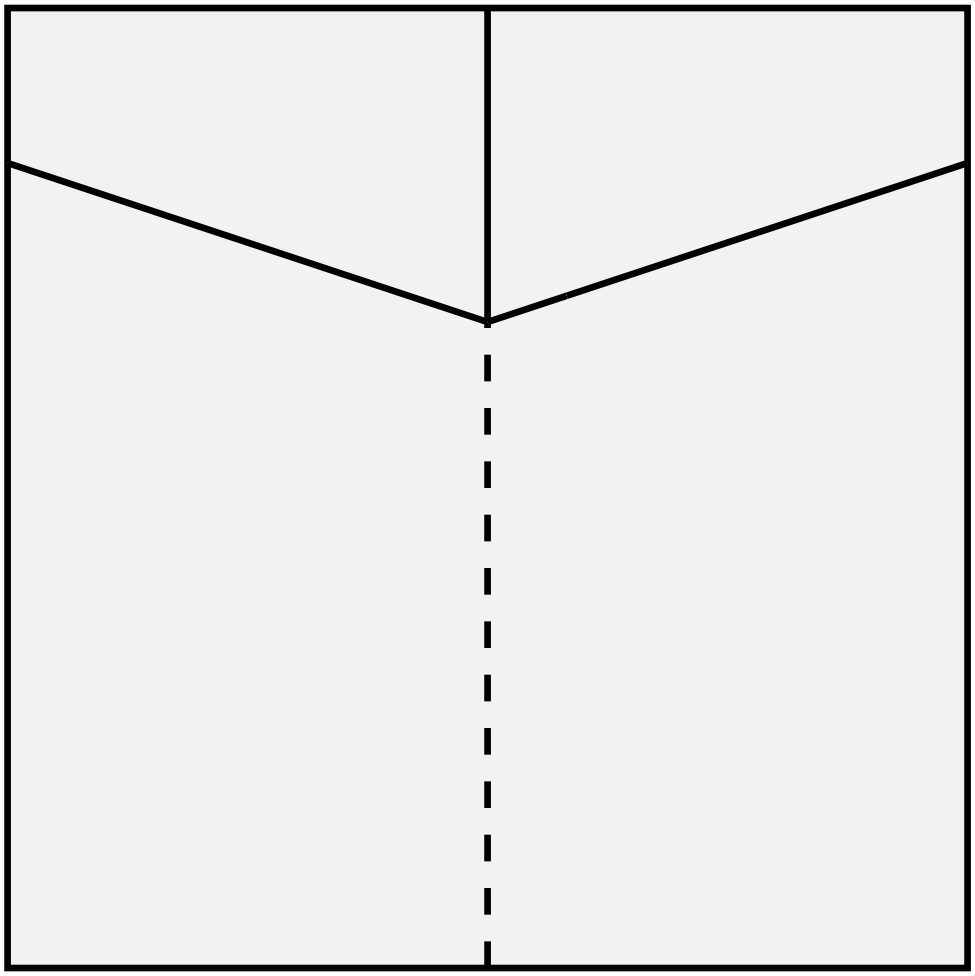


Figure 3.10: Conditions of Kawasaki's and Maekawa's theorems are not sufficient to prove flat foldability. This 'reverse fold' is impossible. Example by Nell Hana Hoffman.

is completely folded, it turns out that the poses of the facets can be determined from the pattern alone.

We first define a folding method, *edge folding*. To edge fold an origami pattern, choose a facet tree on the facet graph. (Search algorithms like breadth-first or depth-first search are suitable.) Cut every crease that is not in the facet tree; the kinematic structure of the cut origami piece is a branching chain with no loops. Fold every crease 180°. Edge folding is virtual, rather than physical, in the sense that we allow self-intersections of the origami during folding, and cut the paper as necessary to allow folding.

Theorem 3 *The relative poses and top color of the facets in a completely folded flat origami piece depend only on the origami pattern. Specifically, the poses and top color depend neither on the signs of the crease angles nor on the final relative heights of the facets.*

Proof: Edge fold the paper; note that the pose and top color of each child facet depends only on the path on the facet tree from the root to the child. Note that regardless of the physical folding method used, this path on the facet tree must be folded. ■

Because edge folding is virtual and not physical, edge folding is not a sufficient condition for flat-foldability, but does provide a useful necessary condition on the crease pattern:

Theorem 4 *If a pattern is flat-foldable, then for every edge folding of the pattern, the two sides of any cut crease must match.*

Most crease patterns contain loops in the facet graph, and cannot be edge folded without cutting some creases. However, if we allow aligned creases to be folded simultaneously, more complicated patterns can be folded without making any cuts. This will be discussed in chapter 4.

Maekawa's thereom gives a condition on crease patterns for flat foldable origami: around any vertex, the difference between the number of mountain and valley creases is two. If we know the stacking order of the final origami piece, we can do better, and determine the types of all creases for a multi-vertex pattern.

Theorem 5 *The facet graph and the goal stacking order completely determine which creases are mountain creases and which are valley creases.*

Proof: Construct a facet tree that spans the facet graph. Assign colors to each facet by walking the facet tree. Each crease connects two facets;

the sign of the crease angle (valley creases have positive crease angles, and mountain have negative) depends only on the top color of the two adjacent facets and the relative height of the two facets. Specifically, if the two colors are white (w) and black (b), and we call the two facets '1' and '2', the following table determines the type of the crease:

Color facet 1	Color facet 2	z order	type
w	b	1 2	valley
w	b	2 1	mountain
b	w	1 2	mountain
b	w	2 1	valley

■

As can be seen from the above proof, every child node has a different top color than its parent after edge folding. This observation constitutes a very simple proof of another well-known theorem about flat origami, Meguro's theorem.

Theorem 6 (Meguro) *Any flat origami must have a two-color facet graph.*

The above algorithms can be applied whenever the final shape is flat, even if the unfolded state is not flat, and even if the actual continuous folding motions are not known. In this case, the fold must be accomplished by a 3D rotation of vertices, rather than a planar reflection. Collisions between facets may be ignored if the only goal is to find the pose of the facets in the flat folded state, or to find necessary conditions on the pattern.

As an example of a case where the theorems may be useful even when continuous folding motions are not known, we point out that a polyhedral model of a paper shopping bag has a configuration space that is just isolated points (see chapter 5); there is no path between the folded and unfolded states. (Some crumpling or bending of facets must occur during folding.) However, we can still determine the folded state from the unfolded state, up to the stacking order of the facets. A cloth airbag provides another example; no finite number of creases and rigid facets is sufficent to model the continuous collapse of the bag (see chapter 5, the bellows theorem), but theorems 4 and 5 still place restrictions on the pattern and final folded state.

3.6 Origami with ball joints and struts

The physical model of origami used in the current work primarily represents origami using rigid bodies with revolute joints at the creases. It turns

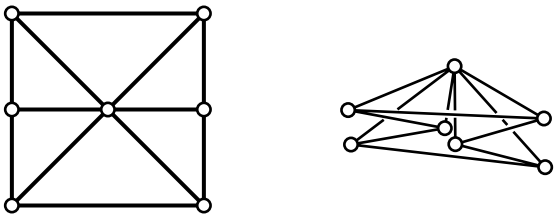


Figure 3.11: The waterbomb base modelled with struts and ball joints.

out that an alternate physical model is possible, with struts connecting ball joints placed at the vertices of the crease pattern. Consider the waterbomb base. There are six facets and six creases. The mechanism shown in figure 3.11 has twelve struts and seven ball joints; the struts are placed around the border of each facet, and the ball joints are placed wherever struts meet. Since each facet is triangular, the struts ensure that each facet remains rigid. Folding is possible along any crease, but along no other line.

Each physical model is suggestive of a mathematical model. The constraints in revolute-joint origami can be represented by a series of matrices that express the constraint that around each vertex, the crease angle must be such that the first and last facet around the loop 'line up'. In terms of Huffman's work [28], the facet normals around the vertex must enclose zero net area on the Gaussian sphere. These constraints are discussed in detail in chapter 5.

The ball-joint model suggests a mathematical model that has only pairwise distance constraints. Symbolically, the state of the system is described by the Cartesian coordinates (x_i, y_i, z_i) of each ball joint; for each pair of ball joints (a, b) connected by a strut, there is a distance constraint of the form

$$(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2 = l_{a,b}^2 \tag{3.13}$$

where $l_{a,b}$ is the length of the strut.

Each mathematical model has some advantages: the crease-angle formulation can be solved to eliminate the constraints, yielding a minimal-coordinate represenation of origami configurations (see chapter 5), and the pairwise-distance model has very simple constraints.

Unlike the constraints on a crease-angle model, the constraints for the distance-constraints model are local (depend only on coordinates of ball joints a and b), and the constraint Jacobian (that describes the tangent space to the configuration space embedded in R^{3n}) depends linearly on the coordinates. For revolute-joint origami, the joint angles depend non-quadratically

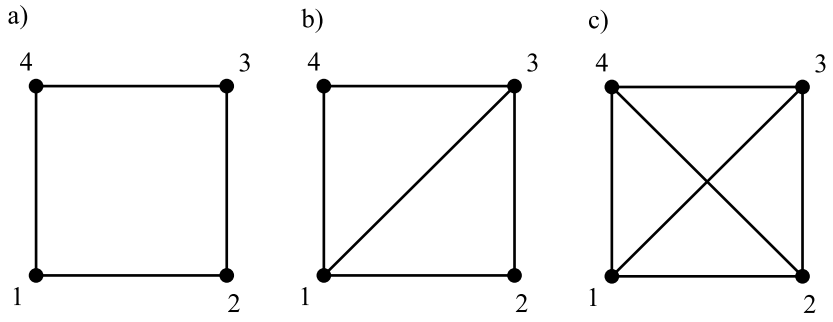


Figure 3.12: Rigidifying a four-bar linkage.

on crease angles throughout the origami, and thus the elements of the Jacobian are non-linear in the coordinates.

Chapter 5 describes the topology of configuration spaces for single-vertex origami mechanisms, and the proofs use either the crease-angle or distance-constraint model as convenient.

The minimum number of struts to make a facet rigid

If all facets are triangular, placing struts along the borders of facets and ball joints at the intersections of struts is sufficient. However, if facets have four or more edges, then this placement of struts might allow deformation of facets.

Consider the four-bar linkage shown in figure 3.12a. If we fix a base to the ground, the linkage still has one degree of freedom that allows flexing in the plane. Figure 3.12b shows a way to place a fifth strut to prevent flexing in the plane. It might seem that the mechanism shown in figure 3.12b will behave as a rigid body, but in fact it can flex out of the plane. Figure 3.12c places a sixth strut to prevent this out-of-plane flexing.

A body is *rigid* if the distance between any two points on the body is fixed. If there are two, three, or four vertices, then it is necessary to connect each pair of vertices with a strut to ensure rigidity. If there are more vertices, it may not be necessary to use n choose 2 struts.

A set of rigid bodies is *rigidly connected* if there are sufficient constraints such that the entire set of bodies must move as a rigid body. If we have a set of n_v vertices in R^3, we need at least

$$n_s = 3n_v - 6 \tag{3.14}$$

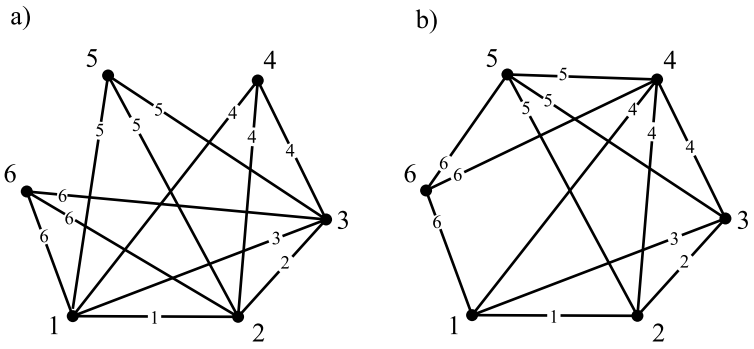


Figure 3.13: Two ways of rigidly connecting six points.

struts to rigidly connect them, since each vertex has three degrees of freedom, each strut removes at most one degree of freedom, and the rigid body has six degrees of freedom.

Two rigid bodies that share three non-colinear points are rigidly connected, since the common points describe a frame. We can use this observation to minimally rigidly connect any set of vertices, as long as there exist three non-colinear vertices. Figure 3.13a illustrates one procedure.

Procedure #1:

1. Choose an ordering for the vertices, with first three not colinear.
2. Use three struts to connect the first three vertices.
3. Use three struts to attach each new vertex to the first three.

This procedure is simple, but has a disadvantage. All of the vertices of an origami facet lie in a plane; the facet is a polygon. In order to minimize the number of struts used to form the entire origami mechanism, the number of struts interior to each facet must be minimized, so that exterior struts can be shared between facets. Figure 3.13b illustrates a procedure that rigidly connects the vertices, while ensuring that a strut is placed along each edge.

Procedure #2 (rigidly connecting a polygon):

1. Choose an ordering of the vertices from 1 to n.
2. Use three struts to connect the first three vertices.

3. Connect each of vertices $4 \ldots n_v - 1$ to its three predecessors.

4. Connect vertex n to vertices $n_v - 2$, $n_v - 1$, and 1.

Since each new vertex is rigidly connected to a rigid body, each procedure rigidly connects the vertices. Since each procedure places a single strut for each of the first three vertices, and three struts for each additional vertex, $3n_v - 6$ struts are placed, which is minimal.

3.7 Bending of paper

The models discussed so far are primarily concerned with the creasing of paper, but creased and uncreased paper behave differently. The work discussed in this section explores the behavior of paper that is bent but not creased. A primary goal of the work was to extend principles used in the manipulation of rigid bodies to a model of flexible paper.

We consider a one-dimensional "cross-sectional" model of the paper. This model is most applicable when the rulings of the paper are approximately parallel in R^3.

In order to better understand the behavior of bending paper, I implemented a method described by Hirai *et al.* [27] to find likely configurations of paper subject to a set of geometric constraints. The model used by [27] describes the configuration of the paper using a set of Fourier basis coefficients. I used the model to extend some analysis techniques typically used for rigid bodies to the flexible paper. I determined a formula describing the forwards kinematics of the paper. The principle of virtual work can also be applied to the model, yielding equations that describe the static equilbria. I derived the dynamic equations that are implied by the model, and considered the problem of force control of the paper.

3.7.1 Formulation of the model

One way to model paper is to describe the configuration by a function that gives the angle of the tangent at each point on the curve

$$\theta(s) : [0, 1] \mapsto [0, 2\pi) \tag{3.15}$$

and an initial condition (x_0, y_0) which describes the location of a particle on an edge of the paper. Figure 3.14 shows an example. The location of each

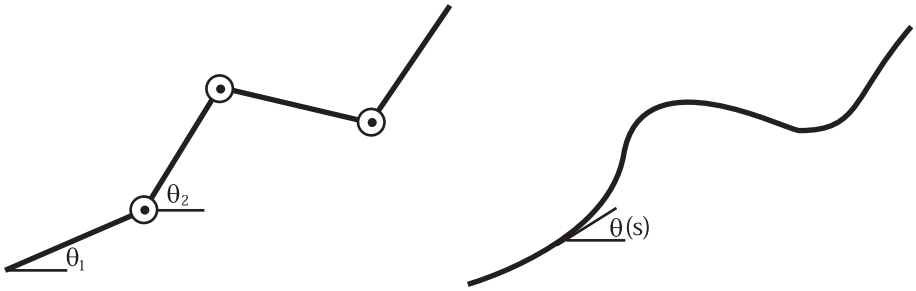


Figure 3.14: One-dimensional paper as the limit of a planar revolute arm.

particle is then given by

$$x(x_0, \theta(\cdot), s) = x_0 + \int_0^s \cos\theta(\tau)\, d\tau \tag{3.16}$$

$$y(y_0, \theta(\cdot), s) = y_0 + \int_0^s \sin\theta(\tau)\, d\tau \tag{3.17}$$

The tangent vector to the parameterized curve described by equations 3.16 and 3.17 is always of unit length – the paper is assumed not to stretch. Since θ is a continuous function of one variable, it can be approximated by a finite Fourier series. The Fourier coefficients then give a finite-dimensional representation for the configuration of the paper. Hirai used this method to find equilibrium points of model of paper held at two opposite edges.

If the Fourier coefficients are collected in a vector p then the state of the paper q is given by

$$q = \begin{pmatrix} x_0 \\ y_0 \\ p \end{pmatrix} \tag{3.18}$$

If we define the vector $b(s)$ containing two leading zeros followed by the Fourier basis functions,

$$b(s) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \sin 2\pi s \\ \cos 2\pi s \\ \sin 4\pi s \\ \cos 4\pi s \\ \vdots \end{pmatrix} \tag{3.19}$$

then we may write θ, x, and y as functions of q and s:

$$\theta(q,s) = q^T b(s) \tag{3.20}$$

$$x(q,s) = x_0 + \int_0^s \cos q^T b(\tau)\, d\tau \tag{3.21}$$

$$y(q,s) = y_0 + \int_0^s \sin q^T b(\tau)\, d\tau \tag{3.22}$$

We may describe trajectories of the paper by allowing q to be a function of time.

Hirai *et al.* [27] used this formulation together with non-linear programming software to find configurations of the paper that minimize potential energy due to some constraints. I implemented this algorithm. One difficulty with the algorithm is that there may be multiple local minima of the potential energy function. In order to better understand continuous motions of the paper, I used the Fourier basis representation to find the Lagrangian dynamics of the system. I will also briefly discuss the problem of controlling some control points or a subset of the degrees of freedom of the paper by applying forces at specified points along the paper. Although the dynamics of this model were also considered by [70], their algorithm involved numerically minimizing the difference between kinetic and potential energy at each time step; my derivation is analytical.

3.7.2 Potential energy functions

Hirai *et al.*used the following formula to calculate the potential energy due to gravity

$$V_g(q) = \rho \int_0^1 y\, ds = \rho y_0 + \rho \int_0^1 \int_0^s \sin q^T b(\tau)\, d\tau\, ds \tag{3.23}$$

where ρ is the mass of the paper per unit length, assumed constant over the paper for simplicity. To derive the dynamics of the paper analytically, we simplify to a single integral:

$$V_g(q) = \rho y_0 + \rho \int_0^1 (1-\tau) \sin q^T b(\tau)\, d\tau \tag{3.24}$$

Hirai used a constitutive law to model the spring energy of the paper

$$V_s = k \int_0^1 \theta'^2\, ds, \tag{3.25}$$

where k is a spring constant determined experimentally. Hirai calculated the energy numerically for a given configuration using the Fourier basis representation:

$$V_s(q) \quad = \quad k \int_0^1 (q^T b')^2 \, ds \tag{3.26}$$

In fact, the integral can be found analytically. If we define a diagonal matrix K with k_i as the ith diagonal element of K

$$k_i = \begin{cases} 0 & \text{if } i = 1 \\ k(i-2)^2\pi^3 & \text{if } i \text{ is even, } i \geq 2 \\ k(i-3)^2\pi^3 & \text{if } i \text{ is odd, } i \geq 3 \end{cases} \tag{3.27}$$

$$\tag{3.28}$$

It turns out that the spring energy can be written as a simple weighted sum of squares:

$$V_s(q) = q^T K q \tag{3.29}$$

3.7.3 Differential kinematics

In a typical manipulation task, we might want to be able to determine the motion of a number of points of interest on the paper, due to a change in the configuration of the paper as a whole. For example, the points of interest might be the set of points of contact between the paper and a manipulator. We can describe the points of interest by a vector of parameters:

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \end{pmatrix} \tag{3.30}$$

The location of the points is given by the vector

$$\mathbf{x}(q, \mathbf{s}) = \begin{pmatrix} x(q, s_1) \\ x(q, s_2) \\ \vdots \\ y(q, s_1) \\ y(q, s_2) \\ \vdots \end{pmatrix} \tag{3.31}$$

We first consider how the x coordinate of a single point changes as a single basis function coefficient is varied.

$$\frac{\partial x(q,s)}{\partial q_1} = \frac{\partial x(q,s)}{\partial x_0} = 1 \tag{3.32}$$

$$\frac{\partial x(q,s)}{\partial q_2} = \frac{\partial x(q,s)}{\partial y_0} = 0 \tag{3.33}$$

For $i \geq 3$,

$$\frac{\partial x(q,s)}{\partial q_i} = \frac{\partial}{\partial q_i} \int_0^s \cos\left(q^T b(\tau)\right) d\tau \tag{3.34}$$

Since all terms are continuous, we can exchange differentiation and integration:

$$\frac{\partial x(q,s)}{\partial q_i} = \int_0^s \frac{\partial}{\partial q_i} \cos\left(q^T b(\tau)\right) d\tau = -\int_0^s b_i(\tau) \sin\left(q^T b(\tau)\right) d\tau \tag{3.35}$$

We may calculate the partials of $y(s)$ similarly. For $i \geq 3$,

$$\frac{\partial y(q,s)}{\partial q_i} = \int_0^s \frac{\partial}{\partial q_i} \sin\left(q^T b(\tau)\right) d\tau = \int_0^s b_i(\tau) \cos\left(q^T b(\tau)\right) d\tau \tag{3.36}$$

Define the vector functions $x_q(q,s)$ and $y_q(q,s)$ that contain the partial derivatives of x and y with respect to q. Then define the $2m \times n$ *Jacobian* matrix:

$$J(q,\mathbf{s}) = \begin{bmatrix} x_q(q,s_1)^T \\ x_q(q,s_2)^T \\ \vdots \\ y_q(q,s_1)^T \\ y_q(q,s_2)^T \\ \vdots \end{bmatrix} \tag{3.37}$$

Then

$$\dot{\mathbf{x}}(q,\mathbf{s}) = J(q,\mathbf{s})\dot{q} \tag{3.38}$$

3.7.4 Kinetic energy and dynamics

In order to find the kinetic energy of the system, we first consider the velocity of each particle. By the multivariable chain rule:

$$\dot{x} = x_q^T \dot{q} \tag{3.39}$$

$$\dot{y} = y_q^T \dot{q} \tag{3.40}$$

The square of the velocity of the particle at position s is

$$\begin{aligned} v^2(q, \dot{q}, s) &= \dot{x}^2 + \dot{y}^2 && (3.41) \\ &= \dot{q}^T (x_q x_q^T + y_q y_q^T) \dot{q} && (3.42) \end{aligned}$$

We define ρ to be the mass of the paper per unit length; we assume ρ is constant over the paper. Then the kinetic energy is

$$\begin{aligned} T &= \frac{\rho}{2} \int_0^1 v^2(q, \dot{q}, s)\, ds && (3.43) \\ T &= \frac{\rho}{2} \int_0^1 \dot{q}^T (x_q x_q^T + y_q y_q^T) \dot{q}\, ds && (3.44) \\ T &= \frac{1}{2} \dot{q}^T \left(\rho \int_0^1 x_q x_q^T + y_q y_q^T\, ds \right) \dot{q} && (3.45) \end{aligned}$$

We define the mass matrix

$$M(q) = \rho \int_0^1 y_q y_q^T + x_q x_q^T\, ds \tag{3.46}$$

The kinetic energy is

$$T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \tag{3.47}$$

Once the kinetic energy and mass matrix have been defined, we can derive the dynamical equations. The details of the derivation are omitted, but it turns out that the equations can be written in a familiar form:

$$M\ddot{q} + \dot{M}\dot{q} = T_q - V_q + J^T f, \tag{3.48}$$

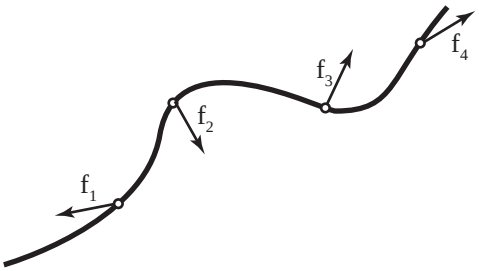
where V_q is the vector of partial derivatives of $V_s + V_g$ with respect to state.

One use of the dynamic equations is simulation. Truncate the Fourier series, invert M to solve for $\ddot{q}$, and numerically integrate $\ddot{q}$ to find the trajectory $q(t)$. Since the mass matrix is symmetric, the numerical stability of the simulation can be improved by using a Cholesky decomposition at each time step, rather than inverting M directly.

The dynamic equations also contain the familiar formulas for the static case. If $\dot{q} = \ddot{q} = 0$, then

$$V_q = J^T f \tag{3.49}$$

3.7.5 Force control



What if we can control forces at some specified points? Let J_1 be the Jacobian that relates the motion of the points where forces are applied to the generalized velocity $\dot{q}$. Then write the dynamic equations:

$$\ddot{q} = M^{-1}(T_q - V_q - \dot{M}\dot{q}) + M^{-1}J_1^T f \tag{3.50}$$

Let $\mathbf{s}$ be the vector of parameters describing the points we want to control. Then $\mathbf{x}(\mathbf{s})$ will be the Cartesian location of these points. Let J_2 be the Jacobian relating $\dot{q}$ and $\mathbf{x}$.

$$\dot{\mathbf{x}}(q, \mathbf{s}) = J_2(q, \mathbf{s})\dot{q} \tag{3.51}$$

Differentiate with respect to time.

$$\ddot{\mathbf{x}} = J_2\ddot{q} + \dot{J}_2\dot{q} \tag{3.52}$$

$$J_2\ddot{q} = \ddot{\mathbf{x}} - \dot{J}_2\dot{q} \tag{3.53}$$

Premultiply equation 3.50 by J_2:

$$J_2\ddot{q} = J_2M^{-1}(T_q - V_q + -\dot{M}\dot{q}) + J_2M^{-1}J_1^T f \tag{3.54}$$

Substitute:

$$\ddot{\mathbf{x}} = J_2M^{-1}(T_q - V_q - \dot{M}\dot{q}) + \dot{J}_2\dot{q} + J_2M^{-1}J_1^T f \tag{3.55}$$

which has the form

$$\ddot{\mathbf{x}} = d(q, \dot{q}) + C(q)f \tag{3.56}$$

C is square, and non-singular as long as both J_1 and J_2 have full row rank. (The number of Fourier coefficients chosen to represent the system should be large compared to the number of control points.) We can choose $\ddot{\mathbf{x}}$ as we like, and solve for f.

$$f = C^{-1}(\ddot{\mathbf{x}} - d) \tag{3.57}$$

3.7.6 Evaluation

What have we learned? One observation is that directly simulating the system using the dynamics equations derived requires the mass matrix to be computed and factored at each time step. Since the mass matrix is dense, this is computationally expensive. The inverse dynamics method of force control described suffers from the same problem. The result is that only a few Fourier coefficients can be used if numerical simulation or control is the goal. Much better results could probably be achieved by modelling the paper as a serial chain, and using efficient simulation techniques such as those described in [7].

On the other hand, the model derived may be useful from the perspective of analysis. It is particularly interesting that under the Fourier basis representation, the spring energy of the system is a simple weighted sum of squares. Another possible benefit is the smoothness of the model; derivatives are defined everywhere along the curve. (This would not be the case if the paper were modelled as a serial chain.)

One primary limitation of the model is that it is least applicable during creasing, since the Fourier representation needs many terms to describe sharp bends. Chapter 4 describes some experimental observations about the creasing process, but understanding the behavior of paper just as it creases is a primary direction of future work.

Chapter 4

Simple origami folding

Much can be learned from human origami experts. One observation is that most origami folds are quite simple. Folds are introduced sequentially, and previously folded facets are 'locked' down, allowing the origami to be treated as a serial chain. According to origami folder Luis Pena (discussion July l8, 2004), procedures for complicated 3D models usually have two phases; the first precreasing phase primarily involves simple straight line folds, and 3D manipulation only happens once all creases have been made.

We define two basic folding techniques, *simple folding* and *book folding*, and present a complete planner for origami that can be folded using these techniques. We also present a prototype of a machine that can fold simply-foldable origami – movies are available from `http://www.cs.cmu.edu/~devin`.

As discussed in chapter 2, "Pureland" origami is a human class of origami that only permits valley and mountain folds. Up to flap insertion and some three-dimensional manipulation permitted in Pureland, Pureland origami (like the samurai hat) is book-foldable. Simply-foldable origami is a special case of book-foldable origami.

Our folds are similar to those used by the virtual origami software described in Miyazaki *et al.* [50]; our contribution relative to this work is an analysis of the conditions under which these folds can be made, a requirement for automatic planning. We present a paper airplane and a hat that are simply foldable, and show that the traditional 'samurai hat' design can be folded using eight simple folds and a single book fold, without flipping the paper.

The structure of the chapter is as follows. After a discussion of related work, a high-level model of simple folds and book folds is presented. The

complete automatic planner and a folding machine will be described. During folding, the high-level rigid-body model is not applicable, and finding a suitable model for low-level folding is an open challenge. The final section describes some alternative techniques for making folds.

4.1 Related work

Box folding and sheet-metal bending are perhaps the two applications in robotic manipulation closest to origami folding. Lu and Akella ([43],[44]) describe a planner (modelled on traditional planners for robot arms) that enumerates all collision-free sequences to fold a carton blank into a carton. Possibly the most complete and practical system for sheet-metal bending is that described by Gupta *et al* [22], consisting of a high level planner that determines the sequence of bends, and a number of low level planners for individual actions. Unlike sheet metal and cardboard cartons, origami is often flat when folded, and typically contains closed chains in its kinematic structure. These characteristics have implications for collision detection, state representation, and available actions.

4.1.1 Another origami-folding robot

I have recently been made aware of another origami-folding robot, developed by Nigel Hinson as part of a master's thesis at Brunel University, shortly after our robot was developed. The mechanical design is very impressive, involving two arms, two suction cups, and rollers and blades to fold the paper. A picture is available at `http://www.interactiveorigami.com/`. Apparently (based on e-mail discussions with the builder), due to time constraints, the robot was not programmed to make more than the simplest folds. At the time of writing, more information on this very interesting robot is not available, but movies can be requested directly from Nigel Hinson.

4.1.2 Sheet-metal bending

The problem of sheet-metal bending has been studied by a number of authors. Possibly the most complete and practical system is that described by Gupta *et al* [22]. Their system consists of a high level planner that determines the sequence of bends, together with a number of low level planners. The low level planners provide 'primitive' actions, including:

- selecting appropriate bending tools and configuration
- inserting the part into the station
- removing positional error by gaging
- creating a bend
- removing the part from the station
- using a second gripper to allow re-grasping of the part

The hardware consisted of a robot arm (used to position the sheet metal), a gripper that was used for re-grasping, and a press brake that used a die and punch to create the bends. In fact, there are a large number of specialized tools for sheet-metal bending. The punch and die selection planner attempted to recognize features from the intermediate part shape, and used these features to select appropriate tools from a large database. A wide variety of grippers is also available; the system chose from a library of fifty different designs.

Each of the planners relied on kinematic models of the sheet metal and the press brake system. Most of the planners considered the sheet metal to be a single rigid body; the exception is the grip location planner, which also considered the fact that the gripped portion of the sheet metal may move during bending in the brake. Since the angle of the bends were chosen by a human designer, and all bends occured atomically, the top level planner needed to determine only the order of the bends, a discrete planning problem.

4.1.3 Box folding

Lu and Akella ([43],[44]) describe a planner that enumerates all collision free sequences to fold a carton blank into a carton. The carton blank is a flat piece of cardboard with creases separating the panels. The carton was modelled as a collection of joints (the creases) and links (the panels). The possibility that bends are not atomic was considered, so the configuration space was continuous rather than discrete.

The model of the cartons was kinematic, and the number of degrees of freedom was equal to the number of creases; typically between five and nine for the problems considered. The model included both explicit constraints (self-intersection collisions) and implicit constraints (the configuration of the carton was described in joint space.)

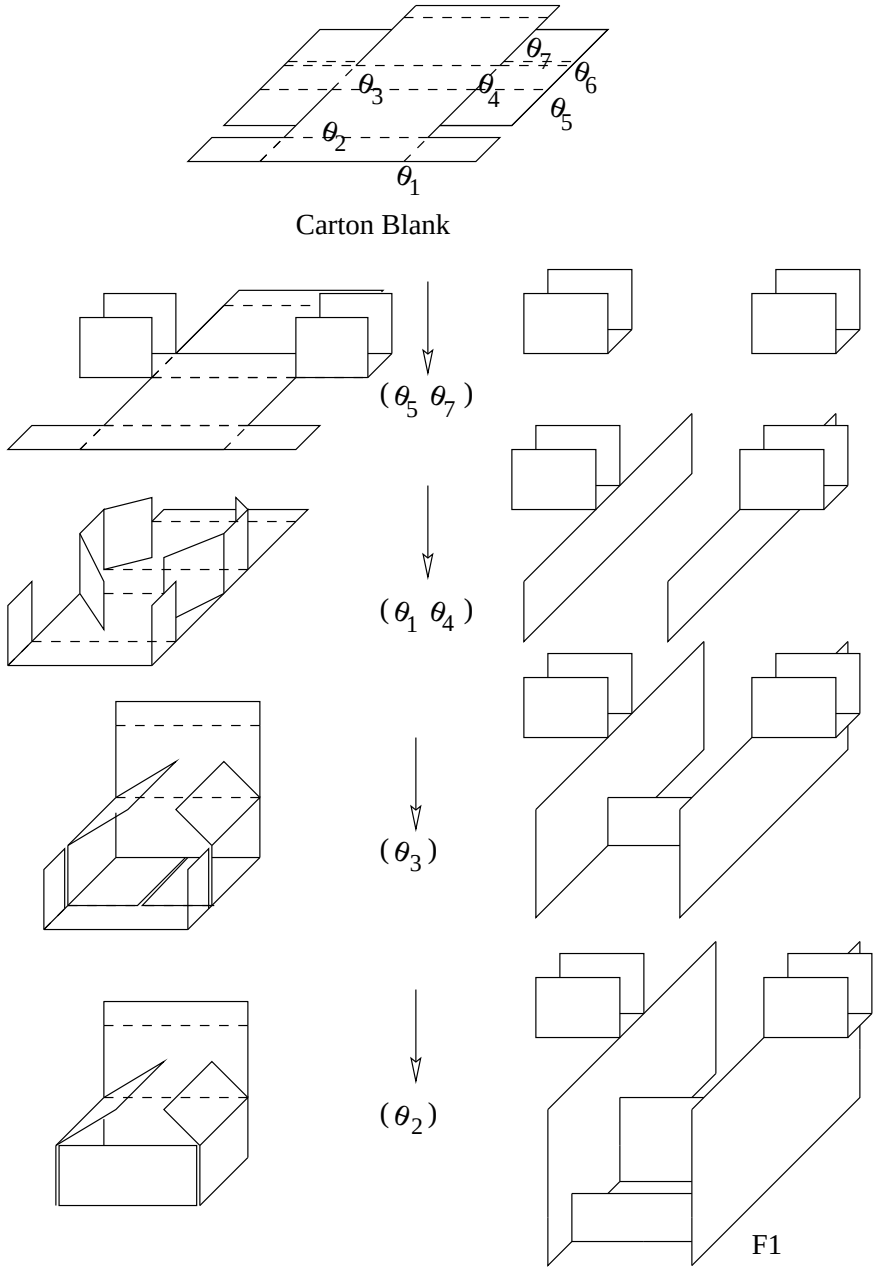


Figure 4.1: Folding a box using a fixture. Reprinted from [43] by permission.

The configuration space was represented by a recursive tree representation based on that used by Lozano-Perez [42] to model serial arms; some modification was necessary to take into the account the possibly branching sequences of links. Since creases in the cartons do not typically meet, the authors do not consider the possibility that the structure is a closed chain.

The intended use of the planner was to enumerate possible fold sequences to help a human design a carton-folding fixture. The authors applied the planner to a number of types of carton blanks, and designed a single fixture to fold two similar types of blank. They used a five-joint Adept arm to move the carton blanks through the fixture, and achieved good results – success rates were nine out of ten and nine out of sixteen for the two types. Figure 4.1 shows an example.

A similar problem was also explored by Song and Amato [57]. Because cartons may have many flaps, the dimensionality of the configuration space may be very high. Fully exploring the configuration space therefore may be prohibitively expensive. The authors designed and implemented a probabalistic roadmap planner, and applied it to the problem of folding a carton with twelve creases. The planner was also successfully applied to the problem of folding protein molecules with about one hundred degrees of freedom, and to a paper folding problem with eleven degrees of freedom. The problems considered involved only open chains.

4.1.4 Rope handling

Inoue and Inaba's paper, *Hand-Eye Coordination in Rope Handling* [30], describes an experiment in robotic manipulation of rope. Figure 4.2 shows the task. The rope was fed from a horizontally positioned hollow tube. The robot grasped the rope, and pulled a length of it from the tube. The robot re-grasped the rope closer to the tube, pulled out more rope, and dropped the dangling endpoint through a wire loop. The robot then released the rope, and regrasped a point on the rope below the wire loop. It then pulled more of the rope through the loop, and laid the endpoint over the feeder tube.

Inoue and Inaba's system comprised a six degree of freedom arm, a stereo vision system, tactile sensors in the fingers, an arm controller, and a micro-computer running custom software. Although this paper appears to be the first published work describing a robot manipulating a flexible object, the focus is actually the design and architecture of this system – the rope handling was seen as a virtuoso demonstration of capabilities of the robot.

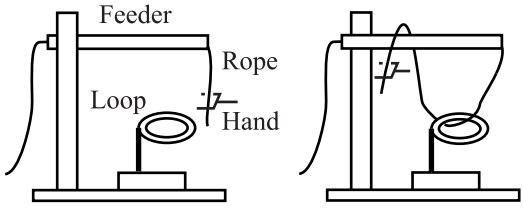


Figure 4.2: One of the first examples of robotic manipulation of a flexible object, re-drawn from [30].

This focus on the system architecture defines the perspective with which the authors view the manipulation task. Surprisingly, visual servoing was seen as the only feasible approach.

> The task of rope handling seems simple. But it inherently requires intensive machine vision. So far, robots have handled only solid objects...a flexible object like a rope does not keep its initial shape and it changes during motion. Therefore, visual feedback is essential in order to manipulate flexible objects successfully.

How should we classify the model of flexible objects used in this work? Based on the visual servoing approach, the model of the system appears to have a dimension of nine: three numbers for the position of the rope endpoint, and six numbers for the configuration of the gripper. After each move, the robot waited for the dangling rope to reach equilibrium; thus the model is essentially quasistatic. Constraints were not considered in the paper; the plan was hand-crafted, and reasoning about the motion of the rope through the ring was made by the researchers, not the robot.

4.1.5 Planning for flexible objects

Most work involving flexible objects has focused on dynamic simulation and control of flexible objects; there is relatively little work on the problem of manipulation planning for flexible objects among obstacles. The lack of work in this area may be due to the difficulties inherent in applying traditional planning techniques to flexible systems. Some of the problems involve:

1. **Dimensionality:** Models of flexible and continuously deformable objects typically use a large number of variables to approximate the pos-

sible configurations of the system. The combinatorics of search algorithms makes planning in high-dimensional spaces infeasible without strong heuristics.

2. **Predictability and Repeatability:** Modelling flexible objects is difficult in itself, and most models do not adequately capture all aspects of the system. Over short time intervals, flaws in the model may not become apparent, and closed-loop control can ensure that the model is synchronized with the world. However, planning is intrinsically an open-loop process.

3. **Underactuation:** Ultimately, only a few controls will be available to manipulate a flexible object. How should a flexible object be grasped? Should the planner operate in the lower-dimensional control space, or in the higher-dimensional configuration space? If the planner operates in the control space, there is an additional difficulty: the configuration may be dependent not only on the current grasp configuration, but on the entire history of grasp configurations.

Kavraki *et al* [34] explored the problem of planning the motions of a thin, rectangular, elastic plate among obstacles. The plate was modelled as a Bézier surface with control points. A potential energy function was defined over the location of the control points. The plate was assumed to be grasped by two opposite edges. Once the configurations of the edges had been determined, a conjugate gradient method was used to find a minimum of the energy function. A table of minimum-energy configurations was precomputed.

The planner was a probabalistic roadmap planner. Each random configuration was generated by fixing an edge, randomly selecting the configuration for the other grasped edge, finding a minimum-energy configuration from the pre-computed table, and randomly selecting a rigid transformation to apply to the body.

The approach was tested using two hundred pre-computed minimum-energy configurations. A problem involving moving the plate through a triangular hole was solved with an average running time of about three minutes.

An alternate method was also explored, in which it was assumed that the Bézier control points could be directly manipulated to determine a configuration; the potential energy function was used only to exclude 'unreasonable' configurations. In this case, the problem was of much higher dimensionality (26 DOF). The planner took about five hours to explore the

space of solutions for a problem involving bending the plate and moving it through a U-shaped hole.

4.1.6 Wire bending and insertion

Nakagaki *et al* [51] considered the task of inserting a flexible wire into a hole. When force acts on the tip of the wire, some deformation occurs; if the deformation is entirely elastic, the wire returns to its original shape when the force is removed. There may also be some *plastic* deformation; once the wire is released, it may not return to its original shape. These authors use a Fourier basis model similar that is described in chapter 3, together with a vision system and a force sensor at the wire tip to determine how much of the deformation is plastic, and how much is elastic. A gripper is used to remove the plastic deformation and straighten the wire. Visual servoing is then used to insert the wire into a hole.

4.1.7 Manipulation of fabric

If we grab the edges of a piece of paper and move them around, we expect to be able to indirectly control other points on the paper. This is an implicit assumption in most visual servoing approaches to the manipulation of flexible objects. Wada *et al.* [67, 66] explores this approach for the problem of manipulating a piece of fabric during automatic stitching. The authors model the cloth using a simple mass-spring system, and show that a small number of interior points can be controlled by moving a few corners of the cloth. They also show that if deformations are small, their visual servoing controller will still converge even without a model of the cloth. Hirai *et al.* [26] describes a similar technique for manipulating a grasped sponge.

4.1.8 Grasping of flexible objects

Wakamatsu *et al.* [69] considers the problem of grasping deformable objects. The authors point out that the principle of force closure is not really approprate for flexible objects. They propose that grasps of flexible objects can be evaluated using the idea of *bounded force closure* – can the current grasp resist a force of bounded magnitude in an arbitrary direction? The authors conducted some simulations involving the grasping of a thin rod.

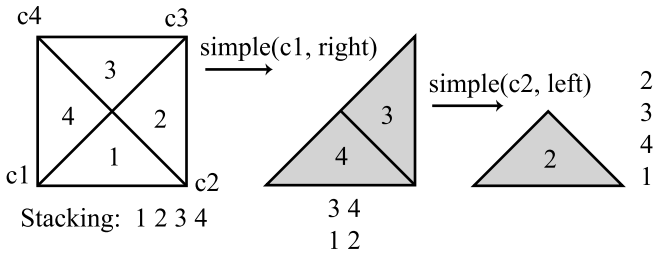


Figure 4.3: A sequence of two simple folds.

4.2 Simple folds

The most basic origami fold takes all paper on one side of a crease line and folds it to the other side; the final pose of facets can be determined by reflection across the crease line. Figure 4.3 shows a sequence of two simple folds.

Given a flat origami state, what simple folds are possible? First, find the minimal set of lines that contains all creases. Discard any crease lines that cross a facet. Each remaining crease line divides the facets into two compound facets; we arbitrarily assign one to be the 'base' and one the 'flap'. During folding, the base will not move. The flap can folded either up (a *valley simple fold*) or down (a *mountain simple fold*).

To execute the fold, all creases colinear with the crease line are folded simultaneously. During folding, the heights of facets in the flap are reversed, and then either stacked above or below the base, forming a single new compound facet.

There are many interesting origami designs that can be simply folded. Figure 4.4 shows two of these: a cup, and a paper airplane.

Given an origami state, it is possible to find all possible simple folds. First, find the set of all lines that contain creases; each crease line gives a candidate simple fold. Check each candidate crease line; discard any that intersect a facet in the interior. If the desired goal state is known and each crease has been assigned as either a mountain or valley, check that each crease in the crease line can fold in a consistent direction.

4.3 Book folds

Figure 4.5 shows two examples of origami that cannot be simply folded. Origami folders often refer to the type of fold shown in figure 4.5b as a

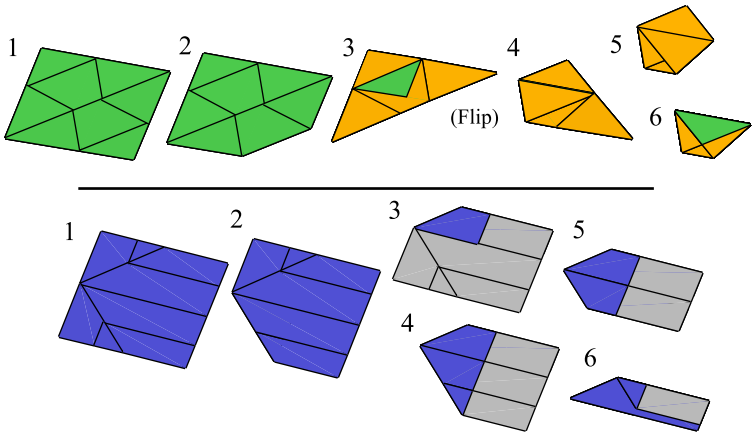


Figure 4.4: Simple foldings of two traditional origami designs.

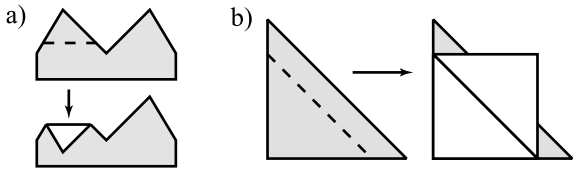


Figure 4.5: Two designs that can be book folded but not simply folded.

'book fold' – the moving facet separates from the fixed facets, just as a page of a book is separated from the pages below it. To formally define a *book fold*, we remove the restriction on simple folds that the base and the flap lie on opposite sides of the crease line. Specifically, we will say that any fold for which all the active creases are colinear such that the continuous rigid-body rotation of the moving facets does not cause self-intersection of the origami or tear any creases is a book fold. We will call the moving facets the flap, and the fixed facets the base. All facets lie in a plane both before and after the fold.

The simple folds from an origami state are easy to enumerate. Finding the possible book folds is somewhat more complicated. The following observation is useful to limit the number of possibilities that must be considered.

Theorem 7 *The set of active creases in any fold cuts the facet graph, separating each pair of relatively moving compound facets.*

Proof: Choose two compound facets that move relative to one another; call one the base, and one the flap. Any crease that connects the base and flap and is colinear with the crease line must be active; the crease angle will be the angle between the base and flap, up to sign. Any crease that connects the base and flap but is not colinear with the crease line will be torn by any rigid body rotation of the flap around the crease line. ■

We can determine the set of all book folds from an origami state, using the following algorithm. First, enumerate all crease lines. Sort the creases by height. Consider all sequential combinations of creases that contain either the minimum- or maximum-height crease. Test each of these crease sets to determine if it cuts the facet graph into at least two pieces. All pieces strictly to the left of the current crease line that do not include the root node of the facet tree (which is always fixed) are candidate flaps, as are all pieces strictly to the right. (For simplicity, we do not consider combinations of pieces of the graph as candidate flaps, since combinations can be folded by a sequence of book folds.) Test each candidate flap to see if and in which direction(s) folding is possible without self-intersection of the origami; this can be accomplished by polygon intersections in the plane of the compound facet.

The state of the origami after a book fold can be predicted: reflect the flap across the crease line, flip flap stacking, and stack the flap either above or below the base, depending on the direction of the fold.

4.3.1 Necessary conditions

Is it possible to determine from an origami state whether book folding is possible? One method would be to run a complete planner; such a planner will be described in the next section. However, this can be very time-consuming. This section will show that there are also necessary conditions on the vertices of the state that can quickly show that it will be impossible to complete the design using only simple folding.

Since book-foldable origami is a subset of flat origami, the conditions we present are stronger and more specific than Kawasaki and Maekawa's theorems about for origami to be flat-foldable. Unfortunately, they are also difficult to derive for vertices of degree 12 or higher. (Vertices of higher degree must have a subset of creases of lower degree that satisfy the conditions; this does serve as a weaker necessary condition.)

The necessary conditions are derived by considering the contribution of successive book folds to the crease pattern. The construction used in the proofs of Kawasaki's and Maekawa's theorems is also useful; we cut the folded vertex, and consider the flat polygon that is the cross-section of the paper.

After the first fold, the paper has two layers around the vertex; by induction, no later crease can ever add an odd number of layers. (For simplicity, we consider only vertices interior to the paper.) Also, notice that the vertex degree is equal to the maximum number of layers in the folded origami; with a single simple fold, it is not possible to add more layers than the vertex degree. Finally, if n folds are made to a vertex, then the vertex must have degree at least $2n$.

Figure 4.6 shows the family tree of vertices with up to four folds. The necessary condition for degree-four vertices is that one fold must be a straight line, and the next be a reflection across that line. Formally, if we label the creases as $(1_a, 1_b, 2_a, 2_b)$ as in the figure, where the leading number indicates the fold that created the crease, then 1_a and 1_b are colinear, and 2_a and 2_b make a constant angle with 1_a; as expected, this satisfies Kawasaki's theorem that odd sector angles sum to 360°.

There is also a condition on the mountain-valley assignment of creases; choose an orientation of the paper so that 1_a and 1_b are both valley. Then 2_a can be either mountain or valley, and 2_b must be the opposite; this is a stronger condition than Maekawa's theorem.

Degree-four and degree-six vertices can only be folded one way, as shown in figure 4.6; degree-eight vertices can be folded with either three or four folds, yielding two distinct classes of vertex. The following table

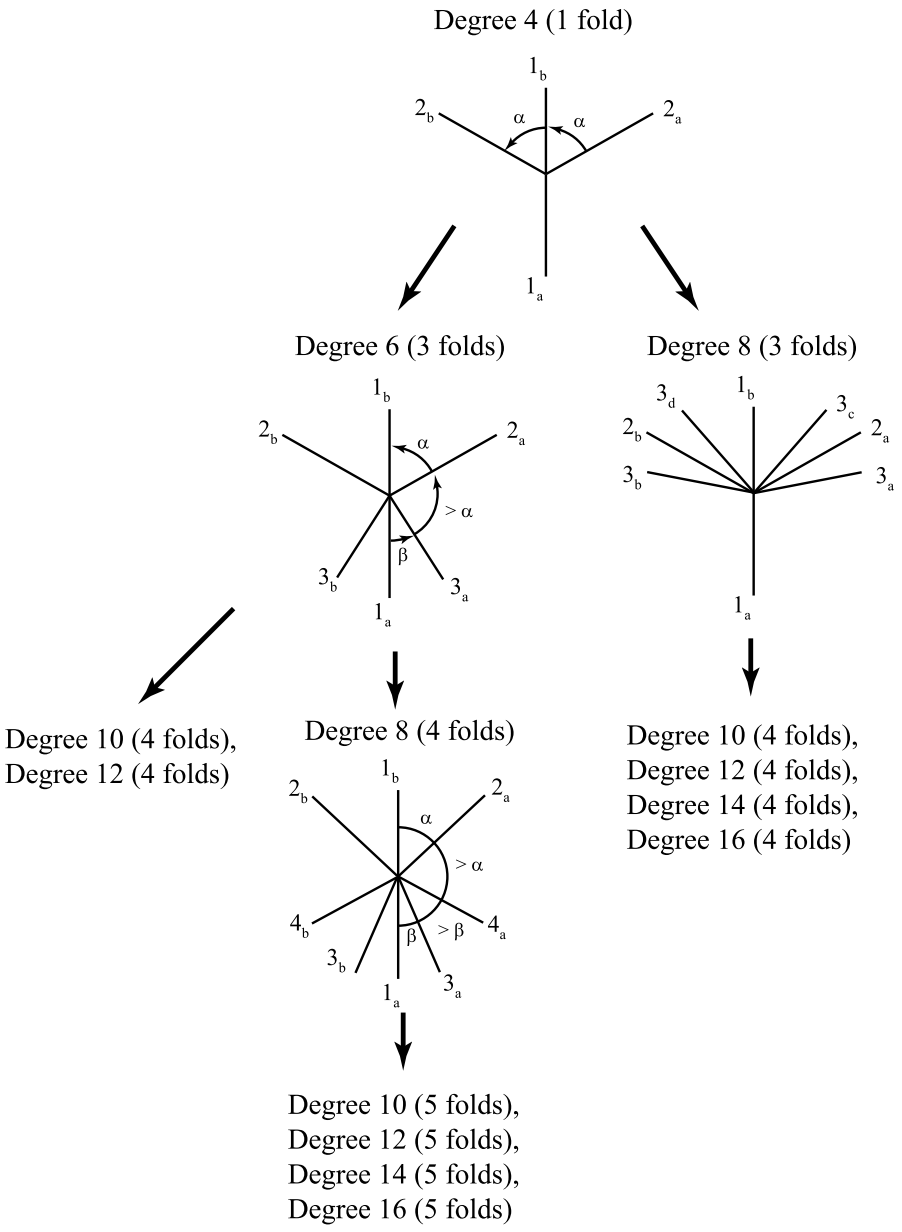


Figure 4.6: A 'family tree' of book-foldable vertices.

enumerates the classes of book-foldable vertices.

Vertex deg.	min. # folds	max. # folds	predec. degree	# classes
0	0	0	-	1
2	1	1	0	1
4	2	2	2	1
6	3	3	4	1
8	3	4	4, 6	2
10	4	5	6, 8	3
12	4	6	6, 8, 10	6
14	4	6	8, 10, 12	11
16	4	6	8, 10, 12, 14	22

For each class, there is a condition on the sector angles and on the mountain-valley assignment of creases. Figure 4.6 shows the condition on sector angles for degree less than or equal to eight; similar conditions could be derived for vertices of higher degree, although the number of classes appears to grow exponentially. The following table shows the necessary conditions on mountain-valley assignment; it is assumed that the paper is oriented so that creases 1_a and 1_b are valleys. 'v' means a valley fold, 'e' can be either, and a 'not' symbol in front of a crease index means that the two creases cannot be of the same type.

	1_a	1_b	2_a	2_b	3_a	3_b	3_c	3_d	4_a	4_b
Degree 4	v	v	e	$\neg 2_a$						
Degree 6	v	v	e	$\neg 2_a$	e	$\neg 3_a$				
Deg. 8 (3 folds)	v	v	e	$\neg 2_a$	e	$\neg 3_a$	$\neg 3_a$	3_a		
Deg. 8 (4 folds)	v	v	e	$\neg 2_a$	e	$\neg 3_a$			e	$\neg 4_a$

4.4 A planner for book-foldable origami

Origami that can be folded using book folds is flat after each fold. Since the motion of the flap occurs out of the plane of the base, collision detection is only necessary at the beginnings and ends of folds, and only requires simple polygon intersection tests. Futhermore, the origami state after each fold is just the stacking of the facets, together with the set of creases that have been folded, and is thus discrete. (Note that crease angles can be determined from the stacking, as long as we know which creases have been folded.) We have implemented a complete graph search planner for book-foldable origami; the nodes of the graph are flat origami states. Figure 4.7

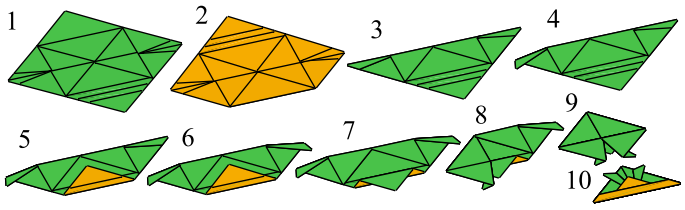


Figure 4.7: Automatically planned folding of the samurai hat. With the exception of step 7 (a book fold), all folds are 'simple'.

shows an automatically planned folding of the samurai hat comprised of eight simple folds and one book fold.

The input to the planner is the pattern and the desired stacking of the facets. Use the goal stacking and the pattern to determine signs on the crease angles. Insert the pattern into the search queue as the initial state. While the search queue has elements, pop, test for goal state, and if goal, backchain to find the plan. Otherwise, determine the book folds from the state and generate successor states. Cull any states that have crease angles that do not agree with those of the goal state. Also cull any states that have been previously visited. Insert remaining states into the visited list and into the search queue.

The visited list is implemented as a hash table that hashes based on the integer heights of facets in the stacking. Before testing against the visited list, the compound facet of the state is collapsed to determine a *minimal stacking*. The algorithm to find the minimal stacking is essentially a bubble sort – each facet is allowed to bubble downwards in the stacking as long as it does not intersect with any facets in the level underneath it.

The planner implementation is about 5000 lines of C++ code, and was run on a 500 mhz Pentium III. The table below shows results for four traditional origami designs.

Origami	creases	nodes	CPU time (sec)	folds
Cup	9	30	.1	5
Airplane	9	24	.1	5
Hat	14	75	.5	5
Samurai hat	20	4250	110	9

For the samurai hat, more than 99% of the CPU time was spent in polygon intersections to determine minimal stackings and to find book folds.

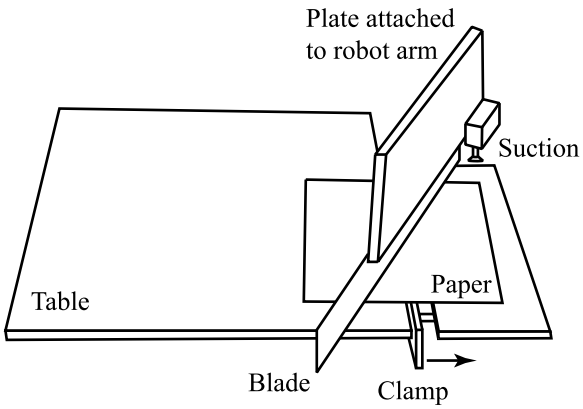


Figure 4.8: A machine that can fold simple origami.

4.5 An origami folding machine

Figures 4.8 and 4.9 show a machine designed to allow a 4DOF Adept SCARA robot arm to make simple folds. The arm was used to position the paper, which was gripped by a vacuum pad. The design is based on the following observation:

Theorem 8 *Any origami piece that can be folded by a sequence of flips and simple folds can be folded by a single initial flip and a sequence of valley simple folds.*

Proof: A mountain simple fold is equivalent to a flip, valley simple fold, flip sequence; write $m = fvf$. Also, either the facets on the left or on the right of the crease line can be chosen as the base. Using similar notation, $v_l f = v_r$, and $v_r f = v_l$. Finally, ff is the identity. These substitution rules imply that any fold sequence of the form $(f^* v^* m^*)^*$ can be replaced by a sequence of the form $f^* v^*$. ■

The folding procedure is outlined in figure 4.10. The paper is first positioned on the table above a slot; the position of the paper determines the location of the crease. A blade presses the paper into the slot (step 2); friction holds the paper in the slot as the blade is removed. The slot clamps shut, forming the crease (step 3). Steps 4 and 5 show a method for removing the paper from the slot and placing it flat on the table; this is required since the arm only provides one rotational degree of freedom at the wrist. First the blade sweeps across the paper, forcing it to lie flat. The clamp is released while the blade holds the paper against the table; the springiness of the paper allows it to swing free of the slot.

Figure 4.9: A machine that can fold simple origami.

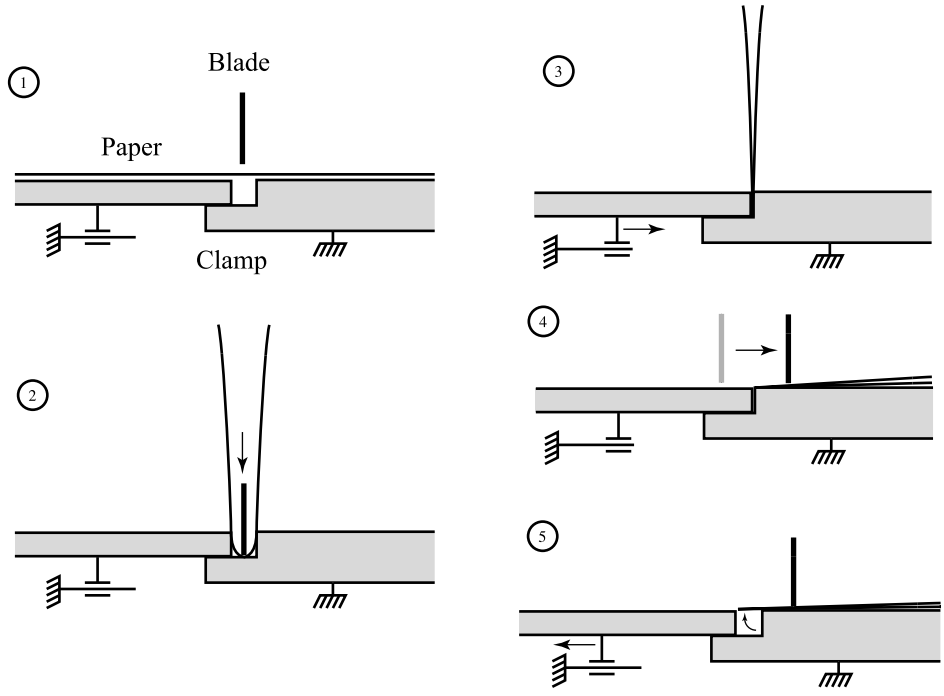


Figure 4.10: Making a simple fold with the machine; side view.

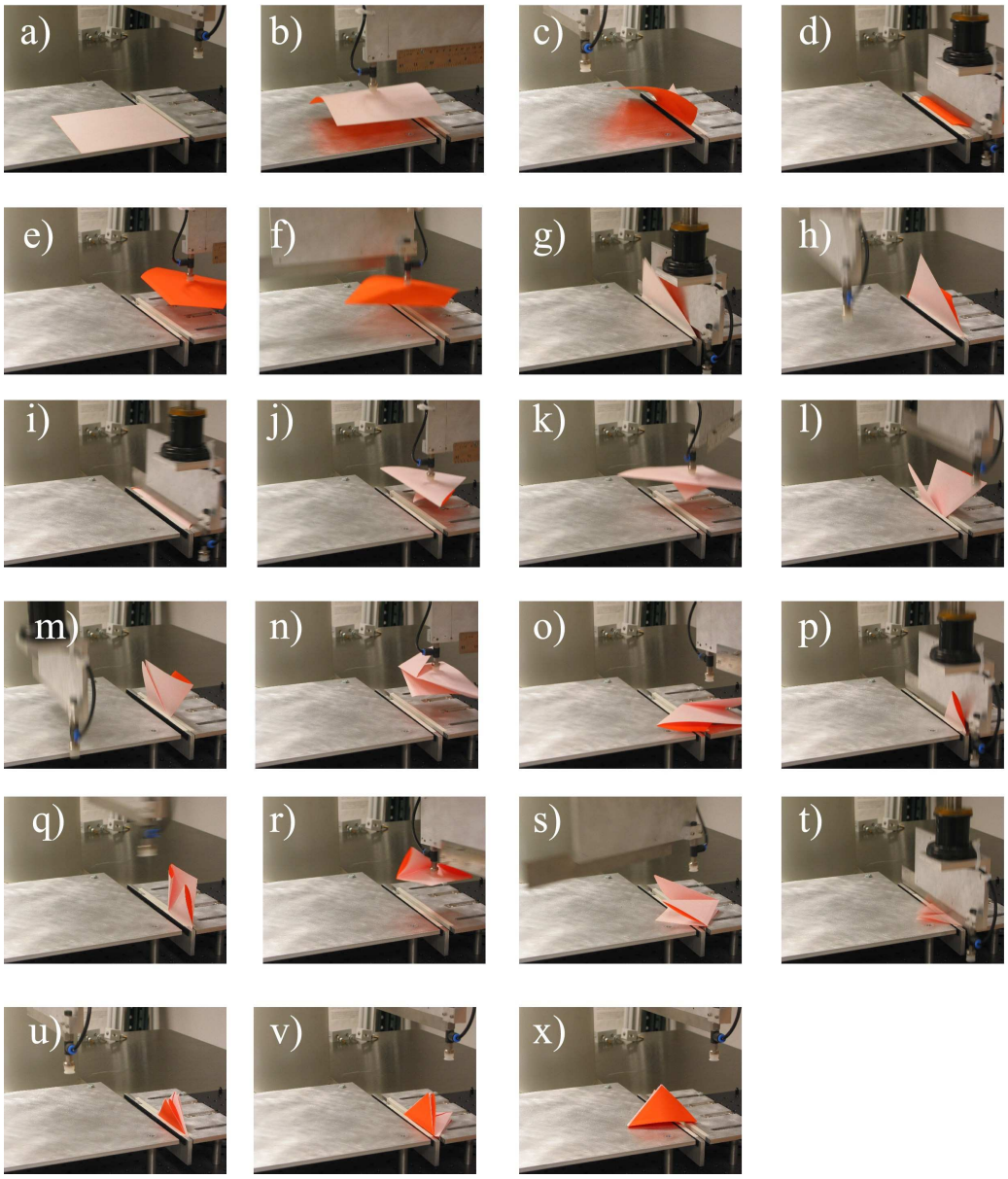


Figure 4.11: Robotic folding of a paper hat. Subfigures a) through d) show the first crease, e) through i) the second, j) through m) the third, n) through q) the fourth, r) through u) the fifth, and v) and x) show the final product – a folded hat.

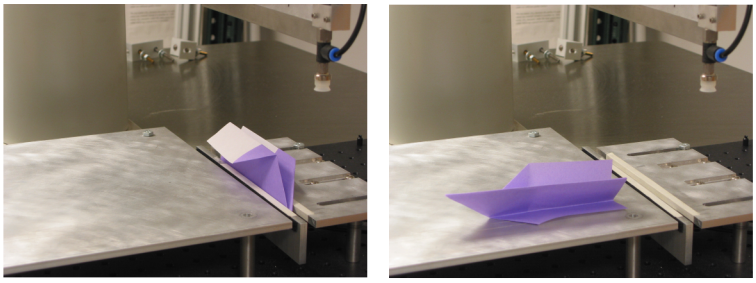


Figure 4.12: A completed paper airplane, folded automatically by the robot.

The machine is able to completely fold both the simple airplane design shown in figures 4.4 and 4.12 and a simplified 14-crease version of the samurai hat – each design takes about two minutes to fold. Figure 4.11 shows the folding sequence for the hat in detail. The machine has also folded a few other designs, including an envelope and a paper cup.

4.6 Machine evaluation; future directions

This section discusses each of the folding steps in figure 4.10 in more detail, and describes some of the the limitations and failure modes of the machine. Under ideal conditions and with the right paper selection, the folding of the paper hat is very reliable, and the robot is successful more than nine out of ten attempts. The paper-airplane folding is more difficult, and under ideal conditions, succeeds about half the time. The paper-cup folding is only occasionally succesful.

4.6.1 Step 1 – Paper positioning

In step 1, the paper is grabbed by the vacuum pad, and positioned over the creasing slot by the 4-DOF arm. The positioning itself is quite accurate, but picking up and releasing the paper can be challenges for folding some origami; these substeps are described in more detail below.

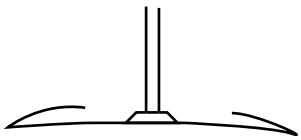


Figure 4.13: Lifting the paper from the bottom layer allows gravity to assist in keeping creases connected to upper facets from unfolding. If the vacuum pad grasped the paper on either of the other facets, we would expect the paper to sag and unfold.

Grasping the paper

Where and how should the paper be grasped? One design choice that affects paper positioning is the vacuum pad. Smaller vacuum pads of more rigid material deform the paper less, but larger flexible pads are more reliable in grabbing the paper. There are also more choices of where to grab the paper using a small pad, which turns out to be an important consideration during later folds.

Uncreased paper tends to droop; too much droop, and the paper can peel off the pad. If the paper has already been creased, the creases unfold somewhat when the paper is lifted; if they unfold too much, it may be difficult to return the paper to a flat configuration when placing it on the table. For this reason, it seems to be better to grasp lower layers of the paper, so that gravity assists in holding creases shut; see figure 4.13. But this can be difficult, since the vacuum pad approaches from the top. The folding of the hat in figure 4.11 shows a case where the fact that creases tend to spring open to allow the paper to be grasped at a lower layer (subfigure r), but if the creases do not spring open enough, this can be a source of error.

Grasping the paper near an edge typically causes more paper droop, but grasping the paper near the center can be a problem as well. If the paper is flat, suction between the paper and table can develop; this will be discussed more fully below.

The automatic planner used to generate the folding plans for the airplane, hat, cup, and envelope uses a very simple heuristic to grasp the paper, and simply selects the centroid of the vertices as the grasping point; this point tends to be near the center of the folded paper, and in a place where there are many thicknesses of paper, which minimizes the amount of crease unfolding during lifting. This heuristic was sufficient for the specific designs folded, but probably would not be effective for many origami

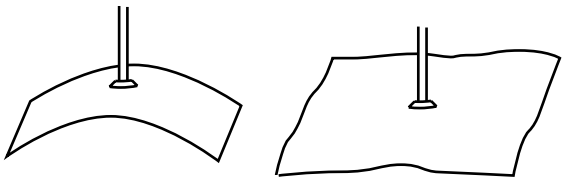


Figure 4.14: Two modes of lifting unfolded paper. On the left, the paper bends with rulings parallel to one edge of the paper. On the right, the paper crinkles slightly, and the crinkles prevent large-scale deformation.

designs.

Paper deformation caused by suction cup

Grabbing the paper and lifting it off the table causes deformation of the paper. There seem to be two modes, shown in figure 4.14. With a small suction cup, the paper tends to bend smoothly, with all rulings parallel to two of the sides of the square paper. (Which sides? Origami paper does seem to be much more flexible in one direction than in the other, presumably because of the direction of the grain of the paper.)

With a larger, flexible pad, and high suction, the paper tends to crinkle slightly as part of the paper is sucked inside the pad. Because of the non-stretchiness of paper, the result is interesting: the paper behaves relatively rigidly; 15 centimeter/side origami paper can be held essentially parallel to the table. This behavior can cause a problem if the goal is to bend the paper while maintaining a suction grip; this difficulty was encountered during previous folding experiments, described in section 4.8.3.

Vacuum between table and paper during lifting

If the paper is completely flat against the table, the vacuum between the table and the paper can be greater than between the paper and the vacuum pad. (From Nigel Hinson's movies of his origami-folding machine, which also uses a vacuum pad to grasp the paper, this problem seems to be a very common failure mode.) Large, flexible pieces of paper are particularly difficult to grasp.

There are some ways to avoid this problem. A large vacuum pad provides better suction, and also tends to crinkle the paper during grasping; the crinkling breaks the vacuum between the rigid table and the paper. Another approach is to grasp the paper from near an edge, rather than near the

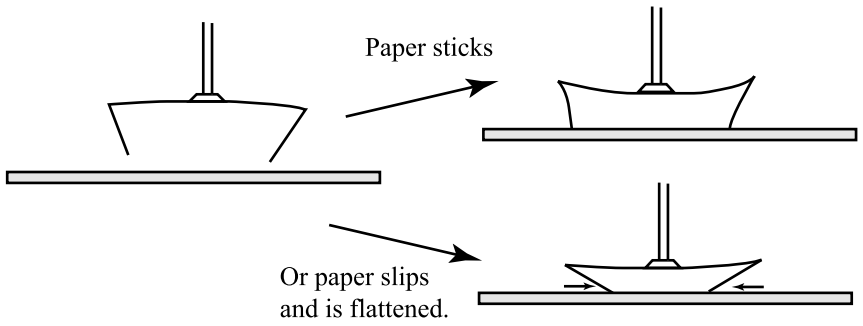


Figure 4.15: Pressing the paper flat against the table.

center; this allows the paper to be 'peeled' off the table, but the cost is that there will be more paper droop, which can cause difficulties in later stages of the folding process. Finally, careful selection of paper is important, and was vital to the successful folding of the paper hat; this will be discussed below.

Release of paper

As discussed, 'flat' origami designs become very non-planar when lifted off of the table. After the paper has been positioned, it needs to be pressed flat against the table and released. This can be difficult; figure 4.15 shows the problem. In the top right subfigure, the contacts between the paper and table stick, and the paper is not pressed flat; in the bottom right subfigure, the contacts slip, and the paper is flattened.

The case where the contacts stick and the paper is not pressed flat is the most common failure mode for the folding of the paper airplane. Some possible solutions (not implemented) might include either not lifting the paper completely off the table, or a more complicated motion to press the paper flat.

4.6.2 Step 2 – Paper bending and friction grasp

During step 2 of placing a crease, the paper is pressed into the slot by a ruler; friction between the paper and the sides of the slot holds the paper in place while the ruler is removed. One question is how the paper behaves as it is inserted into the crease. If the paper is thin and flexible, the contact between the paper and the ruler does not visibly slip, and a smooth bend is formed as the paper slides into the crease. If there are multiple thicknesses

of paper, or the paper is very stiff, the behavior is less predictable, and sharp crinkles can form as the paper is inserted.

If thick paper is used for the folding of the samurai hat, friction between the paper and the walls is insufficent to hold the paper stably, and the paper pops out of the slot when the ruler is removed. A thinner or taller slot would probably solve the problem, but the slot must be wide for the sweep-flattening of the paper in step 4 to work. A future machine design would remove step 4 by rotating the paper and slot as a unit, and would have a much thinner slot.

4.6.3 Step 3 – Dynamic crease formation

What happens when the slot clamps shut, creasing the paper? It is difficult to observe the process because of the speed of the pneumatic actuator that closes the gap. Simulation and analysis are also beyond the scope of the tools and models presented in this thesis, and better models for crease formation are a major goal of future work. In this section, we make a few simple observations, and pose some a few questions.

One observation is that the paper *must* slip as the slot closes; the paper is taller after it has been flattened. Which side slips first? The action of the pneumatic clamp is not symmetric for the current machine design; only one wall of the slot moves. Nonetheless, it does appear that the crease forms approximately in the center of the gap; if a diagonal crease is placed by placing two opposite corners above the center of the gap, the crease runs approximately from corner to corner; the error (on the order of a few millimeters) does not appear to be biased towards one side of the gap or the other.

Since the paper must slip, we might expect it to escape the closing slot by popping out the top. This does sometimes occur, typically when there are many thicknesses of paper. To correct this problem, rough masking tape was applied to the gap to increase friction with the paper. Although the paper must still eventually slip, it seems possible that the masking tape delays the slipping, allowing the sides of the paper time to comply with the parallel walls of the slot.

A future design would have a thinner, taller slot; we would expect this to increase the accuracy of the fold, assuming that the insertion of the paper into the thinner slot could be done without the paper slipping. Also, if the paper did not touch the bottom of the slot, it would not be forced to slip; this might increase reliability and accuracy.

There are some limitations of the method used to form the crease. One

problem is sharpness of the crease. When a human being uses a fingernail to create or sharpen a crease, all force is applied over a small area. Although the pneumatic actuator provides more force than a human being can comfortably apply, there is relatively low pressure applied at any point along the crease, since the entire crease is formed simultaneously. The fact that the creases are not sharp causes additional problems during manipulation of the paper, since they tend to open when the paper is lifted or released; facets on the bottom shift, moving the paper, and facets on the top get in the way of picking the paper up with the suction cup. This effect can be seen in figures 4.11n, 4.11r, 4.11s, and 4.11v.

Another limitation of the folding method is that there is no landmarking. Human beings line up points far away from the crease, where two sides of the paper can be treated as rigid bodies. The constraint that those two points on the pattern be a reflection across the crease constrains the location of the crease, allowing very precise positioning of the crease. The method used by the machine, however, is very local. The paper is grasped very near where the crease will be formed; in this region, the paper must deform greatly, and must slip with the current machine design. Section 4.8 discusses some initial experiments in the direction of human-inspired folding.

4.6.4 Step 4 – Sweep-flattening of paper

In step four, the ruler sweeps across the paper, flattening it against table, while the paper is still gripped by the crease-forming clamp. This step is not vital to crease formation, and is only used to rotate the paper in a direction in which the robot does not have any degree of freedom. Although interesting because it makes use of the flexibility of the paper, the method has a disadvantage; a shallow crease is formed on the sharp right edge of the slot as the paper is pressed against it! A future version of the machine would have an additional degree of freedom, and would not use this method to return the paper to the horizontal plane after creasing.

4.6.5 Step 5 – Paper release

Once the paper has been flattened, it is still clamped in the slot. How should it be removed? The current machine design uses the springiness and flexibility of the paper. The slot is designed to have equal width and height; when the slot opens, the shallow crease placed by the right wall of

the slot unfolds, rotating the part of the paper in the slot out. This step would probably be unneccessary in a future machine design.

4.6.6 Paper selection; effect of humidity

There are a number of steps in the folding process that depend on the characteristics of the paper. Thin paper creases more easily, and can be more easily folded into multiple layers. Step two, the friction grasp of the paper, depends on the roughness of the surface of the paper. Step four, the flattening of the paper with the ruler requires that the paper be somewhat stiff; if the paper droops too much, the ruler hits the wrong side of the paper.

In fact, the machine reliably folds the paper hat only with standard origami paper; even origami paper that is colored on both sides is too thick, and the paper pops out of the slot during step two of the fourth fold.

Weather conditions also affect the performance of the machine. Humidity seems to make the paper much less stiff, and step four fails on the first step of the hat folding on a wet day when the paper droops too much.

Future machine designs would take into account a wider range of paper behaviors. A narrower creasing slot would require less friction to hold the paper, and actuating the slot to allow the paper to be rotated into the flat position after folding would alleviate the problem of paper droop during sweep-flattening.

4.6.7 Sensing

The current design does not sense the location of the paper at all – it is somewhat surprising that the sequences of five folds needed to make the hat and paper airplane can be completed so precisely open-loop. The addition of a vision system might help later folds to be made more precisely, and make the folding process more forgiving of slight errors in early folds.

4.7 Comparison to other folding methods

The primary goal was to design a machine that could fold a class of origami, not just one design. Although the final machine is quite simple, there were many choices made during the design process. Some of these choices were made based on the limitations of the machine, and other choices were made based on the specific nature of origami folding. This section discusses some of the design choices, particularly with regard to other folding processes, including sheet-metal bending, carton folding, and human origami folding.

4.7.1 Comparison to sheet-metal bending

Sheet-metal is thicker than paper, and the types of shapes sheet-metal is bent into are different from most origami. (There are some exceptions in the art world; for example Lane Allen's origami-inspired sheet-metal scultures, which can be seen at `http://my.inil.com/~origami/`.) Although sheet metal does have some flexibility, bent sheet metal tends to remain bent; unlike origami creases, sheet-metal bends do not act like joints after the bending tools are removed. This means that it is easy to bend sheet metal into three-dimensional forms, whereas three-dimensional origami seems to require much care to lock facets into position.

The design of sheet-metal-bending tools and machines depends on the fact that bends are less than 180°. The punch presses the metal into the die. Friction between the punch, the metal, and the die holds the metal in place firmly. The unilateral bending forces are created by the walls of the die, and the punch prevents over-bending. If the metal is thick or the bending angle is very acute, then the metal may be prepared ahead of time by milling out a wedge of metal along the intended bend.

There seem to be three important requirements for a sheet-metal bending machine: holding the metal in place, applying sufficient bending forces from the outside, and ensuring the correct bending angle. Although the origami-folding machine described above at first seems very similar to a sheet-metal press, there are some important key differences. Since the desired fold is approximately 180°, either the punch must be very thin, or it must be removed before folding. Even a thin punch would have to be removed after folding, and it is not clear at all how this could be done reliably.

The solution used by our origami-folding machine to the problem of 180° folds is to bend the paper into a configuration that can be held in place by friction applied to the outside. There are some disadvantages to this approach. One problem is that the frictional forces are primarily created by the springiness of the paper pressing the sides of the paper against the walls of the creasing slot. These forces are likely to be small, since the contact between the paper and the walls is several millimeters away from the central bend in the paper. It also difficult to predict the exact shape of the paper when pressed into the slot, and as a result, it is difficult to estimate the forces.

It is also possible to locate a sheet-metal punch fairly precisely, and there is a line contact along the desired bending location. With the friction grip used by the origami-folding machine, the two lines of contact between the paper and machine can move as the paper flexes during folding (in fact,

the paper *must* slip, as described above, since some of the paper must be pushed out as the slot slams shut). Symmetry of the paper may help to cause the crease to form in the center of the slot during early folds, but in later stages, there may be more layers of paper on the left or right, biasing the location of the new fold.

A narrower folding slot would probably reduce the error, but there is a lower limit on how small the slot can be before it become difficult to press in multiple layers of paper using the blade. A narrower slot would also increase the difficulty of removing the blade without interfering with the rest of the paper.

4.7.2 Comparison to carton folding

As discussed in the related work section above, carton and origami folding are similar in some ways, but it is difficult to see how carton-folding fixtures like the one described in *e.g.* Lu and Akella [43] could be immediately applied to origami folding of the type for which our machine is suitable. One primary difference between the problems is the fact that origami paper is not precreased. Another is that origami is folded many times, creating multiple thicknesses of paper.

On the other hand, three-dimensional origami models seem to have more in common with cartons. According to origami folder Luis Pena, most complicated 3D models are almost entirely pre-creased during the first step of folding, since creases are much more difficult to place once the model has become three-dimensional. Folding 3D origami with fixtures would be very challenging, however, since the facets are flexible and springy, and it seems difficult to predict precisely how the model will behave when external forces are applied to it. 3D origami also tends to have many creases, and they may be arranged in complicated networks. Portions of the model may also be 2D, requiring creases to be folded 180°.

4.7.3 Comparison to human folding

Humans are far better at origami than robots. The industrial arm that provides the primary programmable motion for the origami-folding machine has only four degrees of freedom; humans have twenty four-degree-of-freedom fingers attached to two seven-degree-of-freedom arms. The folding done by the machine is open-loop; humans have vision and touch. Humans also have much more adaptable reasoning capabilities, and can anticipate, avoid, or correct problems that arise during folding.

The machine's folding technique is limited by its basic manipulation capabilities. Relative to human techniques, the machine does all manipulation of the paper very near the desired crease location. This approach has some advantages and disadvantages. An advantage is that in a smaller area near the crease, there may be fewer complicating features like layers, flaps, and prexisting creases. There is also less bending of the paper, and the paper is relatively stiffer near the crease. The large, springy loop of paper the humans typically create before flattening the paper to create a crease would be difficult for a robot to reliably deal with.

The primary disadvantage of local creasing is that it is more difficult to reliably grasp the paper near the crease, since the spring forces are larger, and there are less edges of the paper. Another disadvantage is that landmarking is impossible; there are no obvious features to line up, and the increased stiffness of the paper makes it difficult to bring the paper on each side of the desired crease together without actually creating the crease.

4.8 Experiments motivated by human folding techniques

I conducted a series of experiments with a four-joint Adept arm to gain a better understanding of how humans fold origami. The folds described in this section do not make use of the special purpose creasing-tool discussed above; a simple roller was used to make the creases. One of the limitations with this method was the lack of a second hand to hold the paper in place while creating a crease. A second limitation was the lack of degrees-of-freedom; it was very difficult to figure out how to bend the paper 180° to create a crease, or even how to flip the paper over. Although the origami folded during these experiments is much simpler than the origami folded by the robot described above (the most complicated piece is an envelope with four creases), the experiments do give some perspective on minimalist human folding.

4.8.1 Tool design

The design of the tool used is shown in figure 4.16. A vacuum pad (suction cup) was used to pick up the paper; a pneumatic vacuum generator supplied the vaccum. *Ink brayers* are usually used to roll ink onto rubber stamps. They are available in sizes from about 8 cm in length to about 15 cm in length, and are made from a variety of materials including foam, soft rubber, hard rubber, and acryllic. A 15 cm long soft rubber ink brayer was

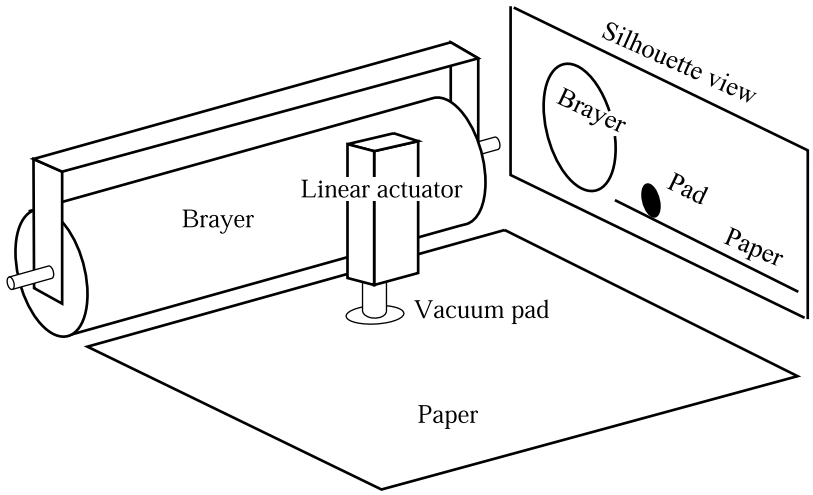


Figure 4.16: Tool design for making simple folds.

attached to the tool to allow the robot to 'roll out' creases in the paper. A pneumatically powered linear actuator moved the vacuum pad out of the way while the brayer was being used. I also built a wooden table to act as a workspace; for many of the experiments, the table played the role of a second hand.

4.8.2 Folding the paper

The first experiments I conducted were motivated by the observation that humans often place precise creases by aligning two edges or corners, and then 'rolling out' the curvature until the paper is creased. The alignment of the edges or corners determines the location of the crease – a person who has difficulty drawing a straight line free-hand can still fold a straight crease.

Figure 4.17 shows the experiment I designed to explore this technique. The operation requires two hands. As a substitute for the second hand, I taped one edge of the paper to the table. The robot grasped the other end of the paper with the vacuum pad, and pressed it down into contact with the table (1, 2). Once the brayer was firmly holding the paper against the table, the vacuum pad was removed (3). The robot then rolled the brayer forwards (4), rolling out the curvature and eventually creasing the paper (5).

One interesting observation is that not all paper configurations can be

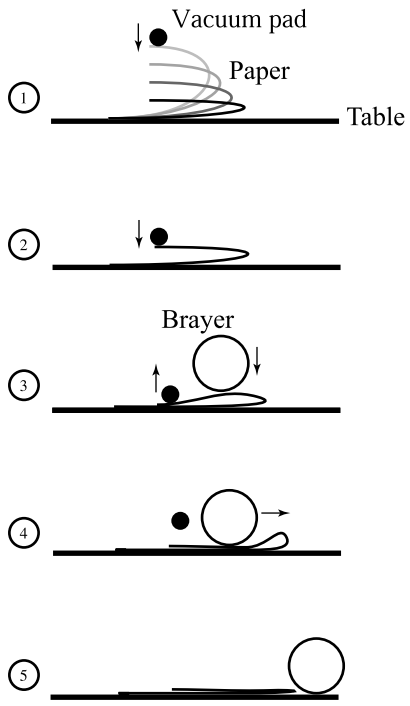


Figure 4.17: Placing a simple crease.

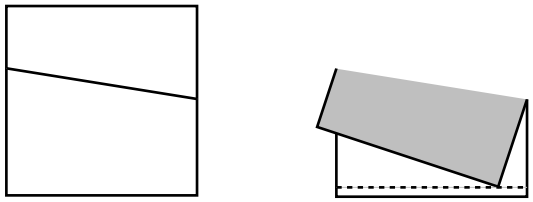


Figure 4.18: Creasing as a reflection.

'rolled out' in this way. When the taped edge and the grasped edge were precisely aligned, this simple operation was fairly repeatable, and the accuracy of the crease placement appeared to be within a few millimeters. Occasionally, the paper tore along endpoints of the crease, typically when I had not placed the paper carefully before the gripper grasped the top edge. When I intentionally rotated the gripper ten degrees before placing the top edge, the paper crumpled badly as the brayer rolled across it.

If there is only a single crease in a piece of paper, it acts as a line of reflection. If there is no line of reflection consistent with the placement of the two edges, then we should not expect the formation of a single crease. Figure 4.18 shows an example. If the upper right corner of the paper is dragged in either direction along the dotted line without rotation, there will be no single crease consistent with the relative position of the upper and lower facets of the paper.

Although it seems likely that most relative placements of the edges can be achieved with three creases, the 'rolling out' action does not appear to permit the formation of more than one crease due to the serial way in which curvature is removed from the paper. On the other hand, pressing the paper flat simultaneously (for example, by squashing it with a book) does seem to create three creases that are consistent with the relative placement of the two landmarks.

4.8.3 Bending the paper

The Adept arm used has only four degrees of freedom; we can describe the configuration of the tool by the coordinates x, y, z, and θ, the rotation about the z axis. This means that there is no way to flip a rigid body while grasping it from the top. If the paper is grasped, the grasped point cannot be flipped over to create a 180 degree bend. In the second experiment, I investigated the possibility of using the flexiblity of the paper to create a bend. Figure 4.19 shows the motion of the gripper and paper during the

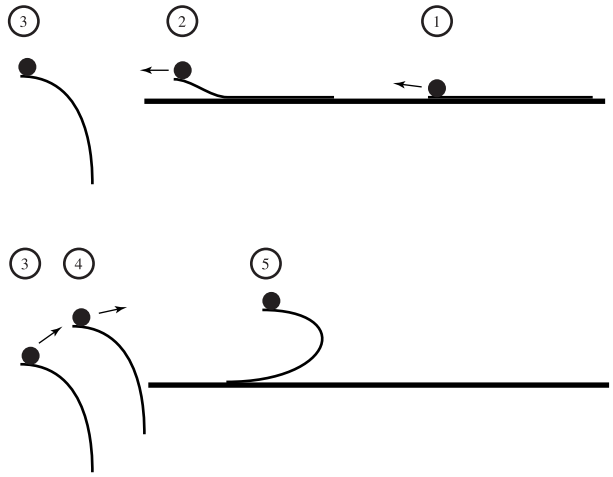


Figure 4.19: Bending the paper.

experiment.

The robot first gripped the paper, and dragged it off of the table. It allowed the paper to droop, and then swept the paper back onto the table. During this phase, the edge of the table caused the paper to bend.

I was surprised by the fact that the paper *did not* always droop once it was dragged off the edge of the table. This may have been because the paper crinkled slightly when gripped by the vacuum pad. When support was removed from the paper it remained in an essentially horizontal configuration. This is an interesting example of paper behaving like a rigid body.

When people hold a newspaper up to read it, it is important that the top edge not droop down. However, for this experiment, some droop was necessary. Crinkles or bends in the paper not parallel to the intending folding direction give the paper some rigidity, and picking the paper up with a suction cup can create such crinkles, as discussed above and shown in figure 4.14. I implemented a strategy to ensure that the rigidity was broken. After moving the paper off of the edge of the table, the robot moved the (ungrasped) edge of the paper underneath the table edge. The arm then lifted the paper, using the table edge to break the rigidity. The arm then swept the paper across the table edge and back onto the table to place the desired bend.

The flexibility of the paper also allows it to be flipped with one hand. If the vaccuum pad releases the paper once the bend has been placed,

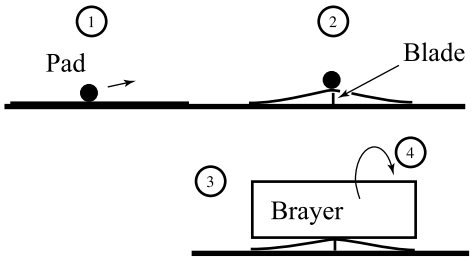


Figure 4.20: Placing a shallow crease using a blade.

the paper slides off, and usually ends up upside down. Unfortunately, the springiness of the paper makes this motion dynamic and very unpredictable!

Although the dynamic aspect could be ameliorated by grasping the bottom edge with a simple gripper before releasing the top edge, a larger problem is that the motion of the bottom edge and shape of the bend are difficult to predict. Typically, some twist develops between the top and bottom edges. Also, if the robot drags the paper from left to right starting in the final position shown in figure 4.19, the bottom edge will eventually move, trailing behind the robot. However, if the robot moves the gripper from right to left, the bottom edge does not move until it eventually flips over. This would probably make it hard to control the position of the bottom edge even with a feedback control law.

4.9 Experiments in folding pre-creased paper

Although bending and flipping the paper using the edge of the table turned out to be unpredictable, the method is more successful if there is already a crease in the paper. If there is already a crease, the deformation of the paper tends to occur along that crease during the bending phase. I explored pre-creasing the paper in the experiment outlined by figure 4.20.

I attached the blade of a long paint scraper to the table. The robot placed the paper above the blade. The brayer then rolled across the paper along the blade, forming a crease along the line of the blade. The process is analogous to bending sheet metal in a brake – the blade acted as the punch and the soft rubber brayer acted as the die.

The accuracy and repeatability of this procedure were not very good. As the brayer moved across the blade, the paper tended to be dragged along a few millimeters by the brayer. Increasing the accuracy is a prob-

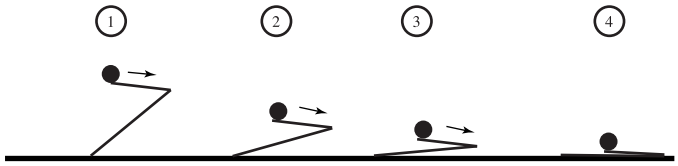


Figure 4.21: Folding pre-creased paper.

lem for a future mechanical design.

4.9.1 Folding creased paper

The method just described places a very shallow crease in the paper. However, even a shallow crease may be useful, since creased paper behaves differently from uncreased paper. Once there is a crease, applying forces to the edges of the paper tends to cause more bending along the crease than elsewhere in the paper. I used this observation to fold the paper once a shallow crease had been placed. Figure 4.21 outlines the experiment.

The motions of the paper for this experiment were the same as the motions shown in figures 4.17 and 4.19. The paper was grasped from above, near the crease. The paper was then oriented so that the crease was parallel to the edge of the table. The robot dragged the paper off the table, and used the edge of the table to break any rigidity and bend the paper. The robot then dragged the paper along the table while squeezing downwards; most of the bending of the paper occurred along the crease. Once the paper was essentially flat on the table, the robot released the vacuum grip and rolled the brayer across the paper to sharpen the crease. It is interesting that in addition to allowing the fold to be made precisely, pre-creasing also allows the paper to be flipped: part of the paper is upside down after the motion, and positioned in a known location.

4.9.2 Folding an envelope

The experiments discussed to this point developed a series of primitives. These primitives can be combined to fold the simple origami shown in figure 4.22. Figure 4.23 shows the strategy. First, one crease was placed in the paper using the method described in section 4.9. The robot then dragged the paper off the table, and used the table edge to 'fold under' the flap formed by the crease. The fold was made along the crease by squeezing the paper against the table while dragging the paper to the right. The brayer

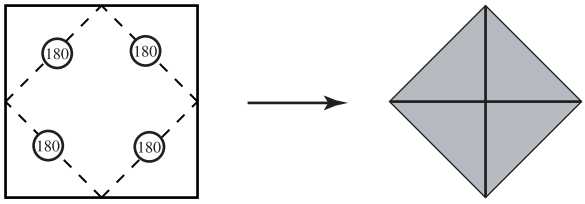


Figure 4.22: An 'envelope' origami shape. The four corner flaps are folded into the center of the square.

rolled over the paper to sharpen the crease. At the end of the procedure, one corner of the paper was folded under. The process was repeated for each of the three remaining flaps, folding the paper into the desired envelope shape.

4.9.3 Evaluation

How well did it work? The robot was able to repeatably place all four creases, and fold the flaps to form the envelope. The error in the pose of the first crease was small – on the order of a few milimeters and a few degrees. However, error accumulated, and the pose error of the last crease was typically nearly a centimeter, and about ten degrees. The primary source of error seemed to be the slip of the paper on the table as the brayer rolled over it to place a crease. Reducing this error will require that the paper be grasped during creasing, probably on both sides of the crease. Some simple sensing of paper edges could be used to reduce the problem of error accumulation.

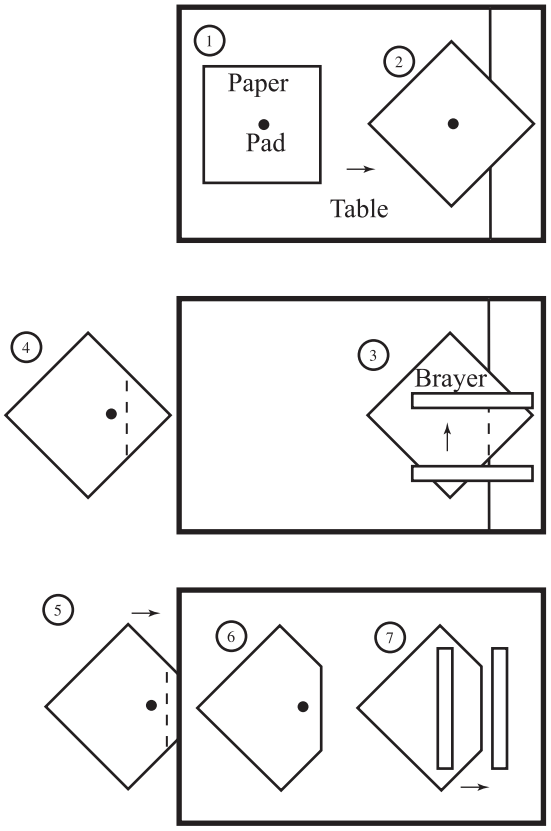


Figure 4.23: Folding the envelope.

Chapter 5

Vertex folding

Many origami designs cannot be book folded; figure 5.1 shows some examples. Inside and outside reverse folds, squash folds, and petal folds all manipulate four creases simultaneously; prayer folds manipulate six. How should we analyse these folds? This chapter defines and analyses two types of folding, *vertex folding* and *network folding*. Both fold types assume the articulated rigid-body model of paper, with revolute joints at creases.

Vertex folds are continuous trajectories of an articulated rigid body with a single crease vertex that do not cause intersection between any facets. (As usual, facets are allowed to become co-planar, but not to pass through one another.) The key results of this chapter relative to vertex folding are analytical local parameterizations of the configuration space of single-vertex origami, a graphical method for determining the global topology and connectedness of the c-space for four- and five-bar spherical closed chains, and some initial results describing the topology of the configuration space for n-bar spherical closed chains.

Path planning for vertex folds also requires some method of avoiding

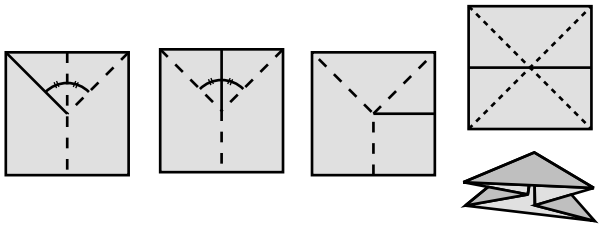


Figure 5.1: Only the first design is book-foldable. Dashed lines represent valley creases, and solid lines, mountain creases.

self-intersection. Local discretized planning only requires checking for for self-intersections at specific sampled states and avoiding those states; efficient and complete global path planning requires some better method of avoiding self-intersection. Results presented in the chapter show that in fact, flat-foldable four-bar mechanisms with planar facets can only self-intersect when flat. This means that complete planning like that applied to book folds in the last chapter could be applied to degree-four folds like squash folds and reverse folds; implementation of such a planner is a detail left for future work.

Network folds are continuous trajectories of an articulated rigid body with multiple crease vertices that do not cause intersection between any facets. The primary contribution of the thesis is a method of locally parameterizing the c-space of crease networks using local parameterizations of the individual vertices. These parameterizations allow local planning and simulation; some examples of trajectories generated using these parameterizations are also presented.

Sometimes, a network of creases cannot be network-folded, either because the network does not have enough degrees of freedom, or because of intersections between the facets for all trajectories of the articulated rigid body. Possibly the most common everyday of an origami-like structure, the paper shopping bag, falls into the first class. A proof that the configuration space includes only the folded and fully open states is presented. Observation of a real shopping bag does seem to confirm that some crinkling occurs along the sides as the bag is folded. But could the bag be folded if we added a finite number of creases? It turns out that it can; however, the flat state is not the same as the expected folded state of the shopping bag. A further question is therefore posed: can the bag be *unfolded* from its usual folded state by the addition of a finite number of creases. We conjecture that it can, and propose a possible solution.

For all figures in this section, we will choose coordinates so that the x axis points along the first (horizontal) axis, the y axis points up, and the z axis points out of the page. For single vertex patterns, this coordinate system will be anchored at the vertex.

5.1 Related work

5.1.1 Parameterization of closed-chain mechanism c-spaces

Four-bar spherical mechanisms have been well-studied; McCarthy [45] is a standard reference. Parameterizations of four-bar mechanisms appear in McCarthy; Huffman gives an alternate formula in Huffman [28]. Parameterizations of closed-chain mechanism c-spaces using an inverse kinematics approach like that described below are also quite common; Han and Amato [23] describe just such an approach to allow efficient path planning for spatial closed chains of line-segment links connected by revolute joints.

The primary use of the parameterization of spherical closed chains with non-sequential dependent joint angles developed in this chapter is to allow Huffman's work on degree-four crease networks to be extended to crease networks with vertices of arbitrary degree.

5.1.2 Topology of configuration spaces

Our approach to analysis of the topology of single-vertex origami configuration spaces is inspired by the work of Milgram and Trinkle [46]. The key results of the paper are the derivation of the topological structure of the configuration spaces of single-loop k-bar mechanisms with revolute joints, in unobstructed R^n. Trinkle and Milgram [63] describes how the analysis can be applied to motion planning problems.

The configuration spaces of 4-bar spherical linkages are well-understood, and a classification of mechanisms by configuration space appears in [45]; our contribution is a geometric method of analysis based on Milgram and Trinkle [46].

The analysis of configuration spaces for high-degree closed chains typically makes use of Morse theory; it should be pointed out that the work we present in this chapter presents only preliminary results in this direction. Milnor's books on Morse theory [47] and differential topology [48] seem to be standard references in the field. Allen Hatcher's book on algebraic topology [24] is freely available on the web, and is also a good reference. Kapovich and Millson have written papers on the c-spaces of k-bar mechanisms [32], and also on spherical mechanisms [33] that use a Morse approach to prove strong results about the global topology of the configuration spaces, but their work seems difficult to directly apply to robot motion planning problems.

Most of the topological analyses of closed-chain c-spaces, including our

work, that of Milgram and Trinkle [46], and Kapovich and Millson [32] ignores the possibility of self-intersections. The problem of determining whether a trajectory free of self-intersections exists without explicitly constructing one seems difficult; some preliminary work by Liu *et al.*does suggest an interesting avenue of approach. [40] presents an analysis of configuration spaces of a 2R manipulator with point obstacles; [39] shows how the results can be applied to efficiently solve the path existence problem, and presents a complete path planner.

5.1.3 Foldability of 3D structures

The fact that the articulated-rigid body model of a shopping bag with creases in the usual places does not collapse was proven independently by myself and Erik and Martin Demaine. The solution to the problem of adding finite creases to the shopping bag to allow it to be collapsed was joint work between myself and Erik and Martin Demaine, as is the problem of unfolding the shopping bag, together with the conjectured solution. The first published proofs of the results regarding shopping bags that we are aware of appear in [5].

5.2 Local parameterization

Reverse folding, prayer folding and more advanced origami skills require the simultaneous manipulation of four or more creases. We now present the relationship between crease angles for vertices of arbitrary degree; our result is also applicable to the case where there is missing or excess sector angle around the vertex.

The mobility of a vertex of degree n is $n-3$. We will therefore choose $n-3$ independent crease angles as input, and solve for the remaining crease angles.

5.2.1 Sequential crease angles

In the special case where the dependent crease angles are sequential, a simple solution is possible using an inverse kinematics approach.

Use forward kinematics to solve for the location of point (x, y, z) given

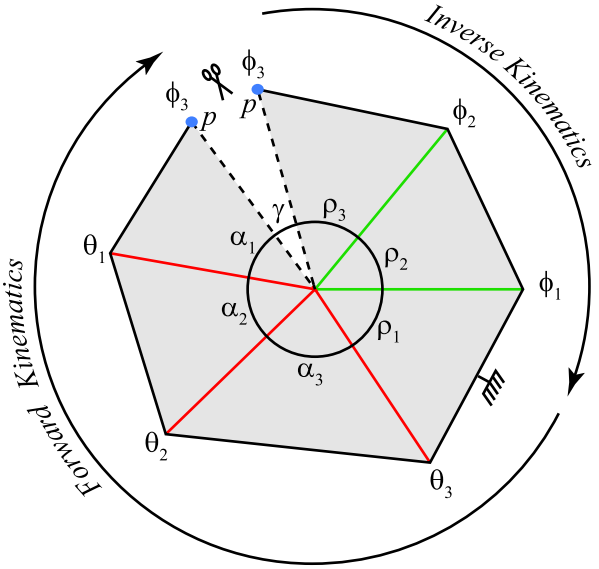


Figure 5.2:

values for the independent crease angles:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_{am}(-\theta_m)\ldots R_{a2}(-\theta_2)R_{a1}(-\theta_1)\begin{pmatrix} \cos(\rho_2+\rho_3+\gamma) \\ \sin(\rho_2+\rho_3+\gamma) \\ 0 \end{pmatrix} \quad (5.1)$$

The location of (x, y, z) can be used to determine the configurations of the dependent crease angles:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_x(\phi_1)R_{p2}(\phi_2)\begin{pmatrix} \cos(\rho_2+\rho_3) \\ \sin(\rho_2+\rho_3) \\ 0 \end{pmatrix} \quad (5.2)$$

$$= R_x(\phi_1)R_z(\rho_2)R_x(\phi_2)\begin{pmatrix} \cos(\rho_3) \\ \sin(\rho_3) \\ 0 \end{pmatrix} \quad (5.3)$$

$$= R_x(\phi_1)\begin{pmatrix} \cos\rho_2\cos\rho_3 - \sin\rho_2\cos\phi_2\sin\rho_3 \\ \sin\rho_2\cos\rho_3 + \cos\rho_2\cos\phi_2\sin\rho_3 \\ \sin\phi_2\sin\rho_3 \end{pmatrix} \quad (5.4)$$

The first row of equation 5.4 can be used to solve for values of ϕ_2, under

the assumption that all sector angles are less than π:

$$\cos\phi_2 = \frac{\cos\rho_2\cos\rho_3 - x}{\sin\rho_2\sin\rho_3} \tag{5.5}$$

There are a few cases. If the magnitude of the right-hand side is less than one, the inverse kinematics problem has two solutions – one 'elbow-up' and one 'elbow-down'. If the magnitude is one, there is a single solution: the second passive crease is either unfolded or folded completely flat. If the magnitude is greater than one, there is no solution.

For each of the solutions for ϕ_2, the possible solutions for ϕ_1 can be determined geometrically. From the last two rows of equation 5.4,

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 \\ \sin\phi_1 & \cos\phi_1 \end{bmatrix} \begin{pmatrix} \sin\rho_2\cos\rho_3 + \cos\rho_2\cos\phi_2\sin\rho_3 \\ \sin\phi_2\sin\rho_3 \end{pmatrix} \tag{5.6}$$

If the vector on the right-hand side of equation 5.6 is non-zero, ϕ_1 is the angle between the two vectors:

$$\phi_1 = \text{atan}\begin{pmatrix} z \\ y \end{pmatrix} - \text{atan}\begin{pmatrix} \sin\phi_2\sin\rho_3 \\ \sin\rho_2\cos\rho_3 + \cos\rho_2\cos\phi_2\sin\rho_3 \end{pmatrix} \tag{5.7}$$

If the right-hand side vector of equation 5.6 is zero, the joint corresponding to ϕ_1 is unconstrained. Geometrically, this occurs when the point (x, y, z) lies on the x axis. The following cases are possible:

$\cos\phi_2$	sector angles	ϕ_2	ϕ_1
-1	$\rho_2 = \rho_3$	one solution	$(-\pi, \pi]$
-1	$\rho_2 \neq \rho_3$	one solution	one solution
(-1, 1)	any	two solutions	one solution each
1	$\rho_2 \neq -\rho_3$	one solution	one solution
1	$\rho_2 = -\rho_3$	one solution	$(-\pi, \pi]$

The parameterization of crease angles allows for some simple local simulation and planning of motions of an origami mechanism. Figure 5.3 shows and animation of the prayer fold used to fold a crane base. The mechanism has six creases, and in general position, three degrees of freedom, assuming the base has been fixed to ground.

5.2.2 Non-sequential crease angles

The procedure for parameterizing vertex configuration spaces above relies on the dependent or 'output' crease angles being sequential around the

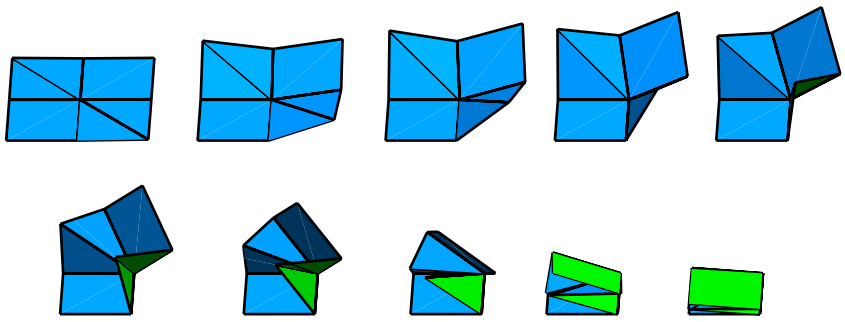


Figure 5.3: Frames from an animation of the initial 'prayer fold' of a crane.

vertex. If there is only one vertex, and the configuration is non-singular, this is not a problem. However, as will be discussed below, networks of high-degree-vertices are more easily parameterized if we allow an arbitary selection of dependent creases at each vertex.

Figure 5.4 shows the procedure; ϕ_1, ϕ_2, and ϕ_3 are the crease angles to be solved for. First cut the crease corresponding to ϕ_3. Anchor the next (counter-clockwise) facet, and choose a coordinate system so that the ϕ_1 crease falls on the x axis. The closure constraint can be written using a product of rotation matrices,

$$R_1 R_x(\phi_1) R_2 R_z(\alpha) R_x(\phi_2) R_z(-\alpha) R_3 p_l = p_r, \tag{5.8}$$

where R_i are rotation matrices corresponding to independent crease angles, and p_l and p_r are points that must touch to close the loop. Rewrite equation 5.8:

$$R_x(\phi_1) Z R_x(\phi_2) a = b, \tag{5.9}$$

where

$$\begin{aligned} Z &= R_2 R_z(\alpha) && (5.10)\\ a &= R_z(-\alpha) R_3 p_l && (5.11)\\ b &= R_1^T p_r && (5.12) \end{aligned}$$

are known. Multiplying out equation 5.9 gives three equations, the first of which is

$$k_3 = k_1 \cos\phi_2 + k_2 \sin\phi_2, \tag{5.13}$$

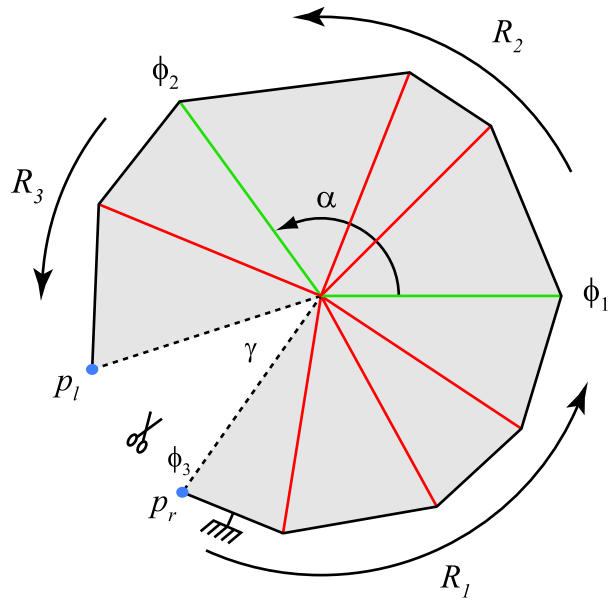


Figure 5.4: Solving for three dependent crease angles.

with known variables

$$k_1 = z_{12}a_2 + z_{13}a_3 \tag{5.14}$$
$$k_2 = z_{13}a_2 - z_{12}a_3 \tag{5.15}$$
$$k_3 = b_1 - z_{11}a_1. \tag{5.16}$$

If $k_1 = k_2 = 0$, then equation 5.13 implies that ϕ_2 can take on any value. Otherwise, equation 5.13 has the solution(s)

$$\phi_2 = \text{atan}(k_2, k_1) \pm \text{acos}\left(\frac{k_3}{\sqrt{k_1^2 + k_2^2}}\right). \tag{5.17}$$

There may be zero, one, two, or infinitely many solutions for ϕ_2. For each value of ϕ_2, the remaining two rows of equation 5.9 can be used to solve for the consistent values of ϕ_1. Specifically,

$$R_x(\phi_1)c = b \tag{5.18}$$

where

$$c = ZR_x(\phi_2)a \tag{5.19}$$

If c and b both fall on the x axis, then any value for phi_1 is a consistent solution. Otherwise,

$$\phi_1 = \text{atan}(b_3, b_2) - \text{atan}(c_3, c_2) \tag{5.20}$$

Once ϕ_1 and ϕ_2 are known, ϕ_3 is determined by the angle between the normals to the facets at either end of the cut chain.

5.3 The c-space topology of spherical n-bar linkages

The parameterizations described in section 5.2 allow simulation and local planning for origami and other spherical n-bar linkages. However, they have some disadvantages:

1. The domain of the parameterization is only a subset of the torus of input joint angles. Determining what input joint angles are valid is a difficult inverse kinematics problem.

2. The parameterizations are not global: the mapping from certain input joint angles to output joint angles may be one-to-many.

3. Parameterizations give no information about the connectedness of configuration space. Configuration space may have multiple components, or there may be low-dimensional subsets that connect components.

This section describes the connectedness and topology of configuration spaces of n-bar spherical closed chains. The analysis uses techniques described in Milgram and Trinkle [46].

5.3.1 Four- and five-bar mechanisms

We first consider four-bar closed chain mechanisms, and present a graphical method for determining the configuration space.

Following Milgram and Trinkle [46], we first remove one of the links, and consider the workspace of the resulting open three-link spherical linkage. Figure 5.5 shows an example. If we anchor one end, the configuration space is a torus, $S \times S$.

Consider a point one unit of distance from the vertex along the cut crease, at the terminal end of the linkage. What is the workspace of this point? Since the distance of the point from the vertex is fixed, the workspace

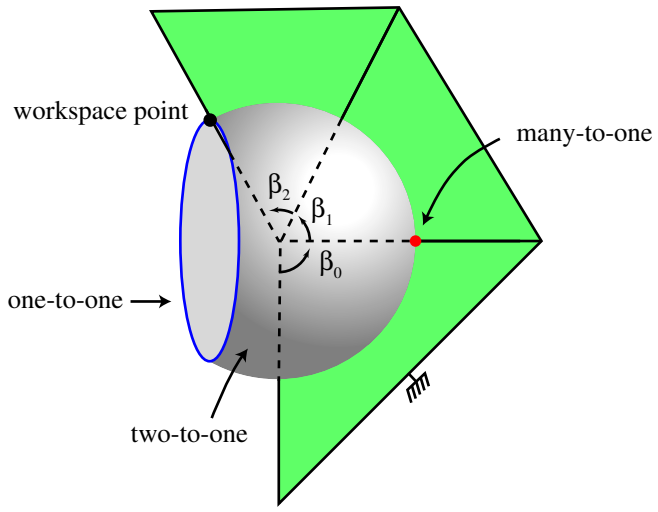


Figure 5.5: An open spherical three-bar linkage, and workspace. The linkage shown has $\beta_1 = \beta_2$, so the forward kinematics map is many-to-one at the point where the horizontal axis intersects the workspace. Since $\beta_1 + \beta_2 < \pi$, the map is one-to-one along the circular boundary of the workspace.

is a subset of the unit sphere. We construct the workspace by first spinning the point around the second axis, and then spinning the resulting circle around the first axis. Since the first joint axis is along the x axis, the workspace is a portion of the sphere bounded by two planes normal to the x axis. The x coordinates of the planes are determined by the maximum and minimum x coordinates for the point, which occur when the second crease angle is an integer multiple of π.

The forward kinematics mapping from joint angles to the location of the workspace point is two-to-one on the interior of the workspace. If a bounding plane is less than one unit of distance from the vertex, the mapping is one-to-one along that boundary. If a bounding plane is one unit from the vertex, the mapping is many-to-one at the intersection of the plane, the x axis, and the sphere.

Now that the workspace of the open chain has been determined, we replace the removed link, and make a cut along the joint that contains the workspace point. The closure constraint requires that the two copies of the workspace point (one on each side of the cut) touch. The workspace of the second copy of the point point is a circle. The configuration space of

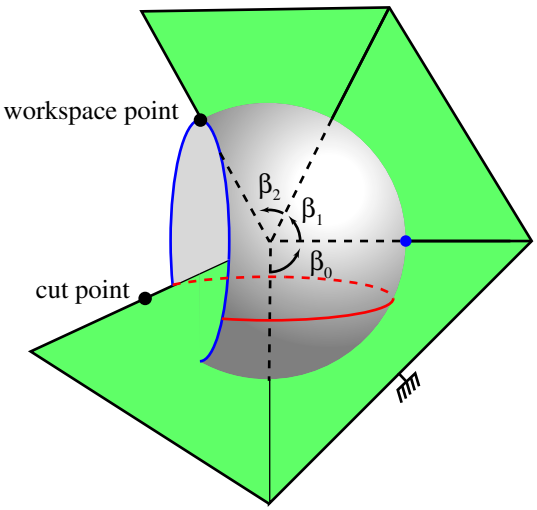


Figure 5.6: A closed spherical four-bar linkage, and workspace. The workspace is of class 'transverse-null', and the configuration space of the mechanism is topologically a circle.

the closed chain is the pre-image of the intersection of this circle with the workspace previously constructed.

Figure 5.6 shows an example. The workspace of the reference point on the closed chain (the intersection of the portion of the sphere and the circle) is an arc of a circle. On the interior of the arc, the forward kinematics map is two-to-one, and the pre-image of the interior of the arc is two arcs. At each endpoint of the arc, the map is one-to-one, and the pre-image of each endpoint is a point. The pre-images of the endpoints (two points) connect the pre-image of the arc (two arcs), and the pre-image of the entire arc (and thus the configuration space of the mechanism) is topologically a circle.

The configuration spaces of planar four-bar mechanisms can be classified by considering the possible intersections of circles on the sphere with workspaces of open three-link mechanisms.

- *Null intersection*. One side of the circle may be completely contained in the workspace. The pre-image of an arc completely contained within the workspace is two arcs.

- *Transverse intersection*. One side of the circle may be cut by the bounding plane at two points. The pre-image of an arc touching the bounding plane is an arc.

- *Tangent intersection*. The circle just touches a bounding circle of non-zero radius. The pre-image of an arc tangent to the bounding circle is a pair of arcs touching at a single interior point.

- *Radius-zero intersection*. The circle touches the bounding plane at one of the poles of the sphere on the x axis. The pre-image of this point is a circle of configurations corresponding to spinning links about the x axis; the pre-image of an arc through this point is two arcs connected by a circle.

- We ignore the case where the circle is completely contained within the boundary of the open workspace.

It is possible to determine algebraic conditions for each case.

Case	Condition
Null	$-\lvert\cos(\alpha+\beta_0)\rvert > \max(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$ or $\lvert\cos(\alpha+\beta_0)\rvert < \min(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$
Transverse	$-\lvert\cos(\alpha+\beta_0)\rvert < \max(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$ or $\lvert\cos(\alpha+\beta_0)\rvert > \min(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$
Tangent	$-\lvert\cos(\alpha+\beta_0)\rvert = \max(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$ or $\lvert\cos(\alpha+\beta_0)\rvert = \min(\cos(\beta_2-\beta_1), \cos(\beta_2+\beta_1))$
Radius-zero	$\beta_0 \pm \alpha = \beta_2 \pm \beta_1 = n\pi$

If the 'tangency' condition occurs in the $z = 0$ plane, then the paper is flat. Kawasaki's theorem gives a simpler algebraic condition, for the cases that all creases are folded and the sum of the sector angles is 2π – alternate sector angles must add to $\pi/2$.

Since there are four different ways the circle can intersect each of the two workspace boundaries, there are sixteen cases to consider. Thirteen of these are shown in figure 5.7; the remaining three are symmetric, as are a few of the cases shown. We label each case by the type of intersection the circle makes with the workspace boundary; for example, 'transverse-null' (see figure 5.6) describes the case where the circle intersects the left (minimum x coordinate) bounding plane in two places, and does not intersect the right bounding plane at all.

There are not sixteen different topological classes of configuration space, but only seven. There are two types of symmetry. First, there is a reflection symmetry; for example, 'transverse-null' and 'null-transverse' have the same configuration space. Second, by choosing different base links, it may be possible to change the way in which the circle intersects the

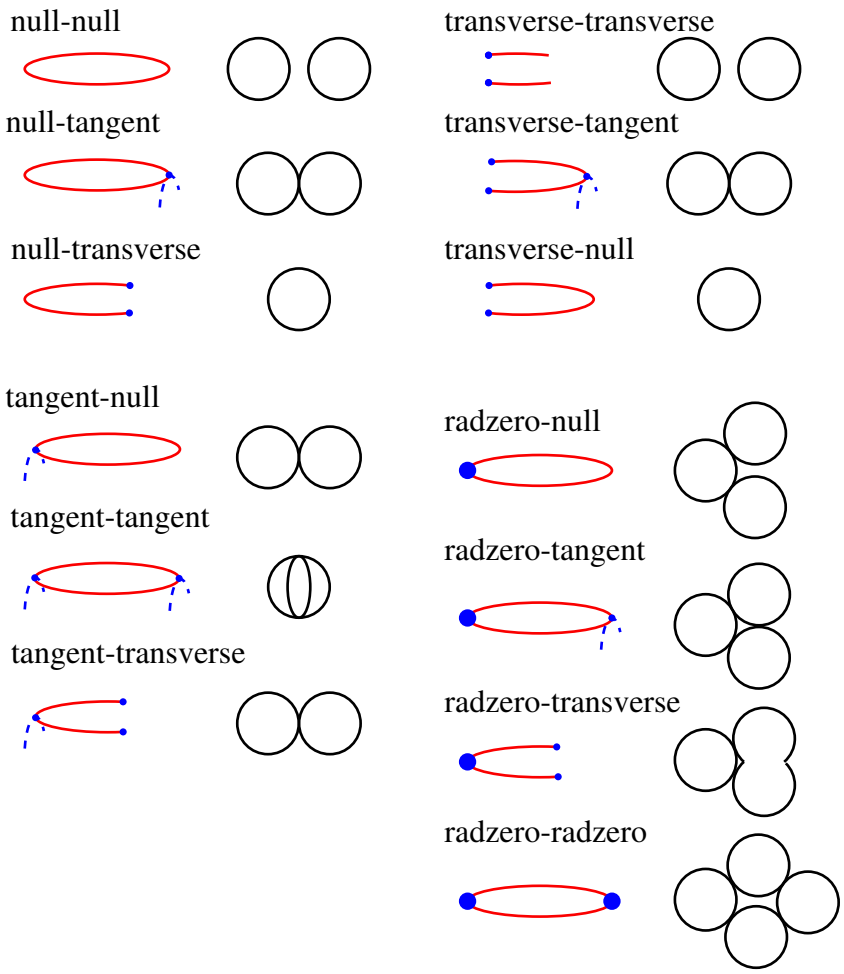


Figure 5.7: Thirteen of the sixteen possible ways a circle can intersect the workspace of an open three-bar spherical chain. For each class, the ellipses on the left show the workspace; the circles on the right show the configuration space (the pre-image of the workspace). There are seven distinct topological classes of configuration space.

workspace of the open linkage, but changing base links does not change the topology of the configuration space. For example, it may be possible to choose a different base link, and change the intersection class from 'null-null' to 'transverse-transverse'; the c-space will be a pair of circles in either case.

Two different mechanism may have topologically equivalent configuration spaces even if there is no apparent symmetry between the mechanisms. The 'radzero-transverse' and 'tangent-tangent' classes are an example; the configuration space of each class is a pair of circles connected at two points. (In figure 5.7 these are drawn slightly differently, to more clearly show the relationship between adjacent classes in the table.)

Five-bar-mechanism configuration spaces can be seen as a parameterized family of four-bar c-spaces; we move one link an infinitesimal amount, fix it in place, and analyse the topology of the four-bar mechanism. Informally, slices of the five-bar c-space are four-bar c-spaces. As the parameterizing link moves through critical points, the topology of four-bar configuration-space slices change.

5.3.2 Many-link mechanisms

In order to analyze the configuration space of the closed spherical chain, we will analyze the singularities and the workspace of an open spherical chain. In this analysis, we ignore joint limits and self-intersections.

The following definitions are taken from Milnor [48]. We say that a map between two manifolds is *smooth* if all of the partial derivatives exist and are continuous. Consider a smooth map $f : M \mapsto N$, from a manifold of dimension m to a manifold of dimension n. Let C be the set of all $x \in M$ such that

$$df_x : TM_x \mapsto TN_{f(x)}$$

has rank less than n (is not onto). Then C will be called the set of *critical points*, $f(C)$ the set of *critical values*, and the complement $N - f(C)$ the set of *regular values* of f.

Consider the *forwards kinematics map* $f : M \mapsto N$ from the torus $M = S^1 \times S^1 \times \ldots \times S^1$ of dihedral angles to the workspace of an endpoint on the last facet. The map can be written as a product of rotation matrices applied to the initial location of the endpoint, and is smooth.

The workspace N may be constructed iteratively. Spin the endpoint around the $m - 1$ axis, creating a circle with radius dependent on sector

angle β_{m-1}. Call this circle N_{m-1}. Spin the circle around the $m-2$ axis; call the result N_{m-2}. Since all axes intersect at the origin, N_{m-2} is a section of a sphere, bounded by two half-planes perpendicular to the $m-2$ axis. Spin each resulting workspace around the preceding axis. The workspace N is equal to N_1, and is the intersection of the unit sphere S^2 with two halfspaces with normals pointing along the first axis N is therefore either a two-manifold or a two-manifold with boundary.

Lemma 1 *Consider an open spherical chain with all sector angles $B_0 \ldots B_{n-1}$ less than π. The critical points of the forwards kinematics map which sends a configuration of the chain to its endpoint are the configurations for which the first dihedral angle ranges over $[0, 2\pi)$, and each of the remaining dihedral angles is one of $\{0, \pi\}$.*

Proof: The configurations of the system can be described by a list of vectors corresponding to the current location of the endpoint X_n and each axis X_i in the workspace,

$$q = (X_1, X_2, \ldots, X_n), \tag{5.21}$$

with the constraints

$$||X_i|| = 1 \tag{5.22}$$

$$\angle X_i X_{i+1} = \beta_i. \tag{5.23}$$

The linear map df_θ between the tangent spaces of the torus and the workspace can be described by the Jacobian of f. Use the cross product method to write the Jacobian:

$$J_f = [\ X_1 \times X_n \ \mid \ X_2 \times X_n \ \mid \ \ldots \ \mid \ X_{n-1} \times X_n\ ] \tag{5.24}$$

If $X_1 \ldots X_n$ lie in a plane, then the Jacobian has rank less than two, and the configuration is a critical point. Proof of the converse: X_1 and X_2 are linearly independent, so at least one column of the Jacobian (either $X_1 \times X_n$ or $X_2 \times X_n$) is non-null. All axes and the endpoint must lie in a plane perpendicular to this column. Since $X_1 \ldots X_n$ lie in a plane iff all of the dihedral angles except the first are one of $\{0, \pi\}$, this completes the proof. ■

Corollary 2 *The critical values of f are the circles formed by rotating the points $(\cos\rho_k, \sin\rho_k)$ around the X_1 axis, where k ranges over $0 \ldots 2^{n-1}$, and*

$$\rho_k = \sum_{i=1}^{n-1} (-1)^{e_{i-1}(k)} \beta_i, \tag{5.25}$$

with $e_i(k)$ denoting the ith bit of k.

Lemma 1 from Milnor [48] states that if $f : M \mapsto N$ is a smooth map between manifolds of dimension $m \geq n$, and if $y \in N$ is a regular value, then the set $f^{-1}(y) \subset M$ is a smooth manifold of dimension $m - n$.

Corollary 3 *Let γ be a curve on the unit sphere that intersects critical circles only at discrete points, and that does not contain any critical circles of radius 0, and let $W = f^{-1}(\gamma)$. W is a differentiable manifold if and only if γ intersects each critical circle transversally.*

Proof: Assume the curve γ is described by a pair of constraints of the form

$$p(x, y, z) = 0 \tag{5.26}$$
$$x^2 + y^2 + z^2 = 1, \tag{5.27}$$

where p has the property that its gradient ∇p is tangent to the unit sphere.

The algebraic variety W is a subset of the torus of dihedral angles, and can described by the composition of the constraints described by equation 5.26 with the forwards kinematic map f. Since the forwards kinematic map already constrains the endpoint to lie on the unit sphere, along any path $\theta(t)$ contained in the variety W,

$$\frac{d}{dt} p(f(\theta(t))) = \nabla p^T \dot{f} = \nabla p^T J_f \dot{\theta} = 0. \tag{5.28}$$

So the Jacobian of the variety W is

$$J_C = \nabla p^T J_f \tag{5.29}$$
$$= \begin{bmatrix} \nabla p^T (X_1 \times X_n) & \dots & \nabla p^T (X_{n-1} \times X_n) \end{bmatrix}. \tag{5.30}$$

The Jacobian has only one row, and describes the normal to W. At any regular point of f, at least two of the cross products will be linearly independent, and the rank of J_W is therefore one. At a critical point of f where γ is transverse to the critical circle, ∇p makes a non-zero dot product with $X_1 \times X_n$, and the rank of J_W is one. At a critical point of f where γ is tangent to the critical circle, every dot product is zero, and J_W is degenerate. ■

Corollary 4 *If W is a differentiable manifold, then the set of critical points of the composition of the length function on γ with f on W are exactly the critical points of f restricted to W.*

Proof: Consider a curve $\theta(t)$ on M, the torus of dihedral angles. The length of the corresponding curve $X(n)$ on the workspace N is

$$s(t) = \int_0^t ||\dot{X}_n(\tau)||\,\delta\tau \tag{5.31}$$

The derivative of s is

$$\frac{d}{dt}s = ||J_f\dot{\theta}|| \tag{5.32}$$

At a regular point of f, the image of the tangent space of M under f has full rank, and we can choose a vector $\dot{\theta}$ from W such that

$$||J_f\dot{\theta}|| \neq 0 \tag{5.33}$$

$$\nabla p^T J_f\dot{\theta} = 0. \tag{5.34}$$

At a critical point x of f, the image of TM_x under df_x is a line through the origin parallel to the tangent to the critical circle through $f(x)$. Since γ is transverse to each critical circle, the intersection of this image with the tangent vector to γ is the origin. Therefore for $\dot{\theta}$ in W, $J_f\dot{\theta} = 0$, and critical points of f restricted to W are critical points of s. ■

5.4 Self-intersection

The planner for book folding described in chapter 4 relies on a key observation – self-intersection can occur only when the origami is flat. It turns out that a similar result holds for origami where four creases that intersect at a vertex are manipulated simultaneously:

Theorem 9 *Rigid flat-foldable degree four origami can only self-intersect when flat.*

Proof: First show that continuous motions of the origami cannot cause self-intersection without at least one crease angle reaching either zero or 2π. This will be proven in lemma 2. If no adjacent sector angles sum to 0 or 2π, then if one crease angle is zero or π, the origami is flat; see lemma 2. If adjacent sector angles sum to 0 or 2π, then the inverse kinematics approach described in lemma 3 suggests that book folding may be possible, but book folds also only self-intersect when flat. ■

Lemma 2 *Continuous motions of planar four-bar-linkages with links that lie in the same plane as their joint axes cannot cause self-intersection without at least one joint angle reaching either* 0 *or* 2π.

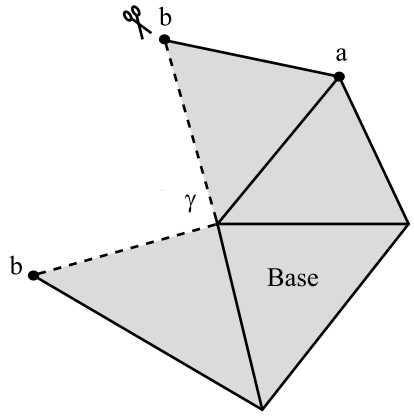


Figure 5.8: A 'cut' spherical four-bar.

Proof: Adjacent links cannot intersect without the angle between them reaching 0 or 2π. Choose any link; call it the base. Call the endpoints of the opposite link p_1 and p_2. Since both the base and the opposite link are line segments, intersection between them must first occur when an endpoint of one link is colinear with the other link; *i.e.* when at least one of the following sets of points is co-linear: (b_1, b_2, p_1), (b_2, p_1, p_2), (p_1, p_2, b_1), (p_2, b_1, b_2). Each of these sets of three points defines a joint angle; when any set contains colinear points, the corresponding joint angle is zero or 2π. ■

Lemma 3 *Continuous motions of a spherical four-bar linkage with links that are coplanar with the joint axes cannot cause self-intersection without at least one joint angle reaching either zero or* 2π.

Proof: Adjacent links cannot intersect without the angle between them reaching 0 or 2π. Pick a link; call it the base. (See figure 5.8.) Call the endpoints of the opposite link a and b. Intersection between the links must first occur when a or b is coplanar with the base. If a is coplanar with the base, then the joint angle between the base and the adjacent link containing a must be one of 0 or 2π; if b, then the joint angle between the base and the adjacent link containing b must be one of 0 or 2π. ■

Notice that lemma 3 applied even for origami that is not flat-foldable, or has has total sector angle that is less than or greater than 2π. A stronger condition is true for flat origami folded out of a flat piece of paper:

Lemma 4 *For flat-foldable degree-four origami with total sector angle of* 2π*, if no two adjacent sector angles are equal or sum to* π*, and if one crease angle is* 0 *or* π*,*

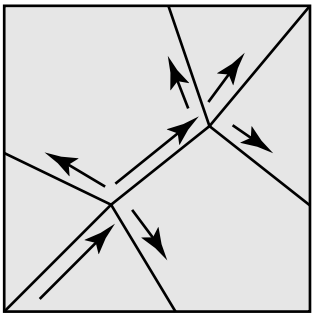


Figure 5.9: A multi-vertex pattern with a mobility of one.

the origami is flat.

Proof: Huffman [28] derives a relationship between opposite crease angles m and n for degree-four origami,

$$1 - \cos n = \frac{\sin A \sin B}{\sin C \sin D}(1 - \cos m), \tag{5.35}$$

where A, B, C, and D are sector angles counter-clockwise from the crease corresponding to m. For flat-foldable origami, Kawasaki's theorem constrains $A + C = \pi$ and $B + D = \pi$; thus for flat-foldable origami, there exists an integer i s.t. $n = m + i\pi$. Choose n to be the crease angle that is 0 or π. If no two sector angles are equal or sum to π, inverse kinematics implies that the other angles are flat as well. ■

5.5 Multi-vertex patterns

The results discussed so far in this chapter deal only with single-vertex origami. Simple folding and book folding apply to multi-vertex origami, but folds are along colinear crease lines, and must be sequential. This section discusses crease networks of the type shown in figures 5.9 and 5.10, and considers the case where multiple non-colinear creases are folded simultaneously. Networks of degree-four vertices have been studied by Huffman [28], the extension of this work to networks with higher-degree vertices is possible using the parameterization of vertices with non-sequential creases.

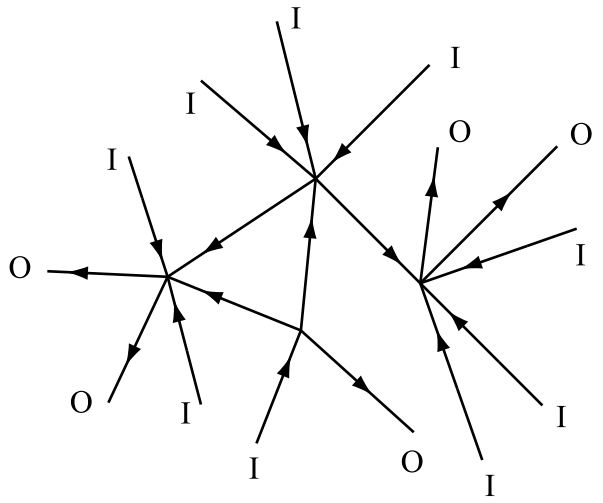


Figure 5.10: A multi-vertex pattern with a mobility of six.

5.6 3D foldability

So far, we have considered only a rigid-body model of origami, with a finite number of creases whose locations in the pattern are fixed. When is this model useful? What things can be folded using this model?

In this section, we consider what is probably the most commonplace origami-like structure: the paper shopping bag. It turns out that a shopping bag with rigid facets, and creases in the usual places, has a configuration space that is just isolated points corresponding to the flat and fully open states. This might be considered a design feature; since the facets resist bending and crinkling, the bag tends to stay in its current configuration, either open or closed.

5.6.1 The bellows theorem

It was conjectured in [13] and proven in [14] that "any continuous flex that preserves the edge-lenghts of a closed triangulated surface of any genus in three-space must flex in such a way that the volume it bounds stays constant." This is an extremely important theorem. It says, for example, that no polyhedron with a fixed, finite number of creases is sufficient to model the deflation of a closed airbag, or the inflation of a teabag or origami waterbomb. But where are the boundaries? What origami structures can or cannot be satisfactorily modelled with fixed crease patterns?

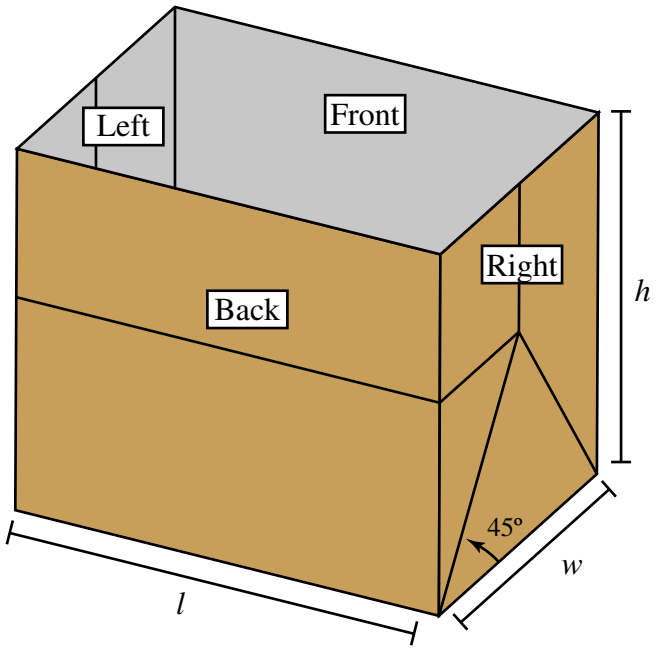


Figure 5.11: A tall box, with paper bag crease pattern. The configuration space is isolated points, and the box cannot be collapsed.

5.6.2 Can a shopping bag be collapsed?

We will consider a rectangular box with one facet (the top) missing. The box has a bottom, a front, a back, a left side, and a right side, as shown in figure 5.11. The right and left sides have dimensions $w \times h$; the front and back have dimensions $l \times h$, where h is the height of the box.

The shopping bag has a pattern of creases. On each of the left and right sides of the box, there are two creases that make a 45° angle with the bottom edge. If $h \leq w/2$, then the creases on each side do not intersect on the interior of the facet, and there are no other creases on the box; we say the bag is *short*. (This situation is shown in 5.12.)

If $h > w/2$, then the two creases on the right side of the box intersect, as do the creases on the left side. (See figure 5.11.) In this case, the shopping bag is *tall*, and there are five additional creases: one vertical crease on each side from the intersection point to the top, one horizontal crease on each side from the intersection point to the back, and on horizontal crease on the back connecting the horizontal creases on the sides. We say that the box has been *collapsed* if all facets lie in a plane.

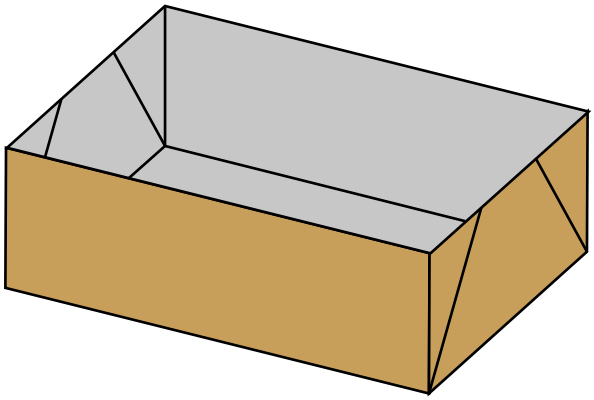


Figure 5.12: A 'short' box with paper bag crease pattern, similar to a collapsible garment box.

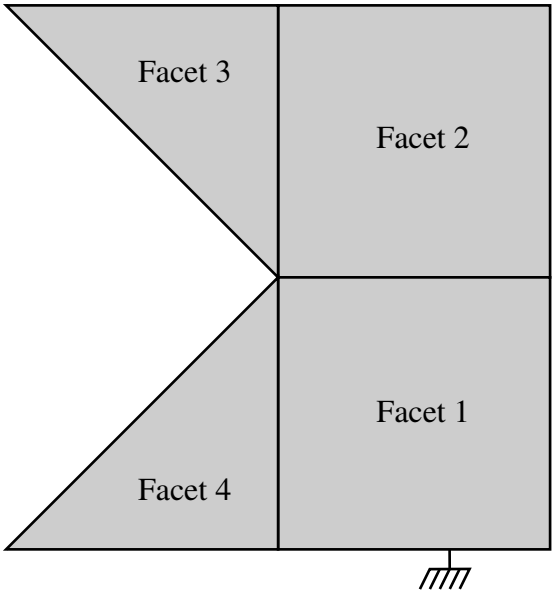


Figure 5.13: A corner of the short box (garment box), cut between two facets, and anchored to the ground.

Theorem 10 *For every 'short' open box with a shopping-bag crease pattern and $h \leq l/2$, there exists a continuous folding that does not cause self-intersection, and takes the bag from the fully open to the collapsed configuration.*

Proof: Consider a single corner of the box, with associated crease pattern. If we ignore self-intersections, the configuration space is a single connected componenent. Let the links be numbered as shown in figure 5.13, and anchor link 1. The workspace of the endpoint of link 3 is the portion of the sphere bounded by two halfplanes; $|x| < \sqrt{(}2)/2$. The workspace of the endpoint of link 4 is a circle of radius $\sqrt{(}2)/2$, centered at the point $(0, -1)$. The pre-image of the intersection of these two workspaces is a pair of circles connected at two points. (These points correspond to two collapsed configurations.)

We choose the collapsed configuration in which all joint angles are π, and choose a trajectory that moves to this configuration directly (*i.e.*, without passing through the other collapsed trajectory). We let θ_1 increase monotonically from $\pi/2$ to π. From 5.5), $\phi_2 = \pm\theta_1$; for the trajectory we have chosen, $\phi_2 = \theta_1$. ϕ_1 also increases monotonically from $\pi/2$ to π for this trajectory.

Adjacent links can only collide if the angle between them is zero or *pi*; no joint angles are 0 or *pi* except at the start and end of the trajectory. Intersections between links must first occur when the ϕ_2 or ϕ_3 axis touches the $z = 0$ plane; since ϕ_1 and θ_1 are positive except at the end of the trajectory, there are no intersections. The case of intersection between link 2 and link 4 is symmetric.

The four corners of the box are connected by creases along the bottom of the box, and all four corners fold occur simultaneously and symmetrically; the condition that $h \leq l/2$ is sufficient to ensure that no facets that do not share a vertex can intersect.

■

Theorem 11 *An 'tall' open box with a shopping-bag crease pattern has a configuration space that is isolated points corresponding to the fully open and fully closed configurations.*

Proof: Consider the vertex on the back right edge of the box that has all sector angles equal to $\pi/2$. The two vertical creases that meet at this vertex have crease angles that are equal in magnitude; if the magnitude is not 0 or π, then the two horizontal creases from this vertex must be one of $\{0, \pi\}$. Choose a crease that is 0 or π, and connected to another interior

vertex. Walk the crease network; each of the left and right sides is flat (open or folded), and each of the corners is either fully open or collapsed. ■

The following observations can be made:

1. A shopping bag with creases in the 'usual' places is rigid.

2. Two shopping bags taped together at their tops cannot be flattened with a finite number of fixed creases. (Thanks to Robert Lang for this example of the bellows theorem.)

3. A shopping bag cannot be turned inside-out with a finite number of fixed creases. (According to Erik Demaine, is has been proven by Connelly that a convex vertex cannot be turned inside out using a finite number of creases, but this work may not yet have been published.) Robert Lang points out that this work implies that the 'closed sink' origami move that inverts a convex vertex cannot be modelled with a finite number of creases.

But in spite of all these results, a shopping bag can be collapsed with the addition of a finite number of fixed creases. Theorem 12 will show the procedure. In order to verify the procedure, it is necessary to show that facets do not collide during folding; the primary method for showing this will be to consider the workspaces of pairs of facets, and show that the workspaces do not intersect on the interior. We will need the following simple lemma:

Lemma 5 *Two right circular cones that share a vertex and with axes separated by at least* 90° *do not intersect on their interior.*

Theorem 12 *A tall box can be collapsed with a finite number of creases.*

Proof: The approach is to shorten the box, by adding creases that allow the top to be rolled inside the box. Once the box is short, theorem 10 allows the box to be collapsed. We consider a single edge of the box, with crease pattern shown in figure 5.14.

The crease between facets 1 and 2 is fixed at 90°. The fold takes place in three steps. During step 1, we fix crease 5, and drive crease 3 from 180° to 0°, choosing the solution such that crease angle 1 is positive, crease angle 2 is negative, and crease angle 4 is positive. During step 2, we fix crease 3 at 0°, and drive crease 1 to 180°. Crease angles 2 and 4 do not change sign, and crease angle 5 becomes positive. During step 3, we fix crease 1 at 180°,

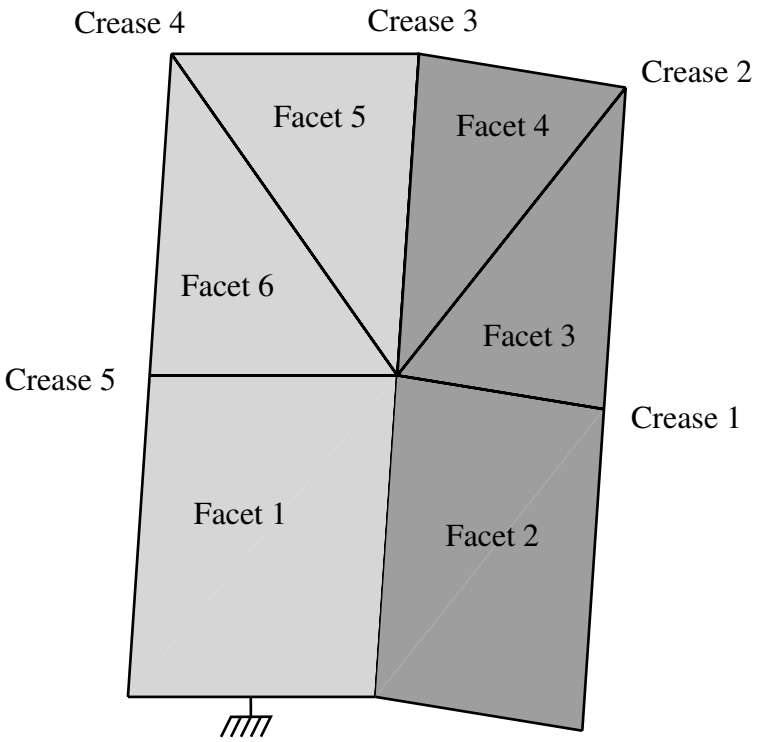


Figure 5.14: An edge of the tall box (shopping bag) with creases added to allow folding. Facets 1 and 2 are rigidly connected; there are joints (creases) between all other pairs of adjacent facets.

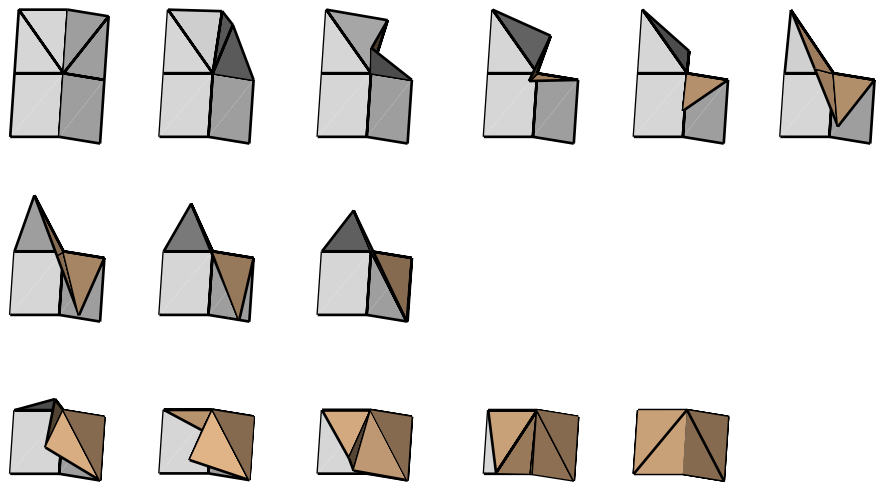


Figure 5.15: Shortening an edge of a paper bag. The rows correspond to steps 1, 2, and 3 of the folding process described in the proof of theorem 12.

and drive crease 3 to -180. Crease angles 2 and 4 return to $0°$, and crease angle 5 reaches $180°$.

	θ_1	θ_2	θ_3	θ_4	θ_5
Start	$0°$	$0°$	$90°$	$0°$	$0°$
After step 1	$+, < 180°$	$-, > -180°$	$0°$	$+, < 180°$	$0°$
After step 2	$180°$	$-, > -180°$	$0°$	$+, < 180°$	$+, < 180°$
After step 3	$180°$	$0°$	$-180°$	$+, < 180°$	$0°$

Ignoring self-intersection, the existence of a trajectory of this form can be verified using the graphical method for determining the topology of of a degree-four spherical linkage, since each of the steps is a fixes two of the six creases (the crease between facets 1 and 2, and one other.)

To prove that self-intersection does not occur, consider pairwise intersections of facets. No two adjacent facets can collide unless the angle of the crease between them crosses $180°$; this never happens for our choice of trajectory. The following table summarizes the analysis of collision possibilities for non-adjacent facets.

	facet(s)	vs. facet(s)	Don't intersect because:
Step 1	3	1, 6	Workspace of 3 is a right circular cone that intersects plane of facets 1, 6 only at origin.
	3	5	Workspaces of 3, 5 right circular cones; lemma 5.
	4	1, 2, 6	Facet 4 is bounded by two creases. Crease 2 is inside the box for $\theta_1 \geq 0$, and crease 3 is as well for $\theta_5 = 0, \theta_4 \geq 0$.
	5	2	Workspace of 5 is a right circular cone that intersects plane of facet 2 only at origin.
	5	6	Facet 6 is co-planar with facet 1.
Step 2	4, 5	1, 2	Crease 2 and crease 4 are inside the box for $\theta_1 \geq 0$ and $\theta_5 \geq 0$.
	3	1	Cone workspace vs. plane; intersection is origin.
	3	6	Cone workspaces; lemma 5.
	6	2	Cone workspace vs. plane; intersection is origin.
Step 3	6	2, 3	Cone workspace vs. plane.
	6	4	Two cones; lemma 5.
	5	1, 2, 3	Creases 3, 4 inside the box for range of θ_2, θ_5.
	4	1	Cone workspace vs. plane.
	4	6	Cone vs. cone, lemma 5.

A single edge can be rolled inside the box using the procedure above; to shorten the box, place symmetric crease patterns at each edge. Figure 5.16 shows an animation. For a tall box, or a box with dissimilar length and width, it may be necessary to perform a number of shortenings before collapsing the box. Note that the height removed during a shortening can be as small as desired, so it is possible to shorten the box to any desired height. ■

5.6.3 Unfolding the shopping bag

The solution to folding the bag proposed by theorem 12 is surprising in that the the collapsed state of the shopping bag is not the same as the folded state that uses the original creases. The obvious question then is, can the bag be *unfolded* from the original folded state by adding a finite number of creases. We conjecture that it can; a possible solution is shown in 5.17.

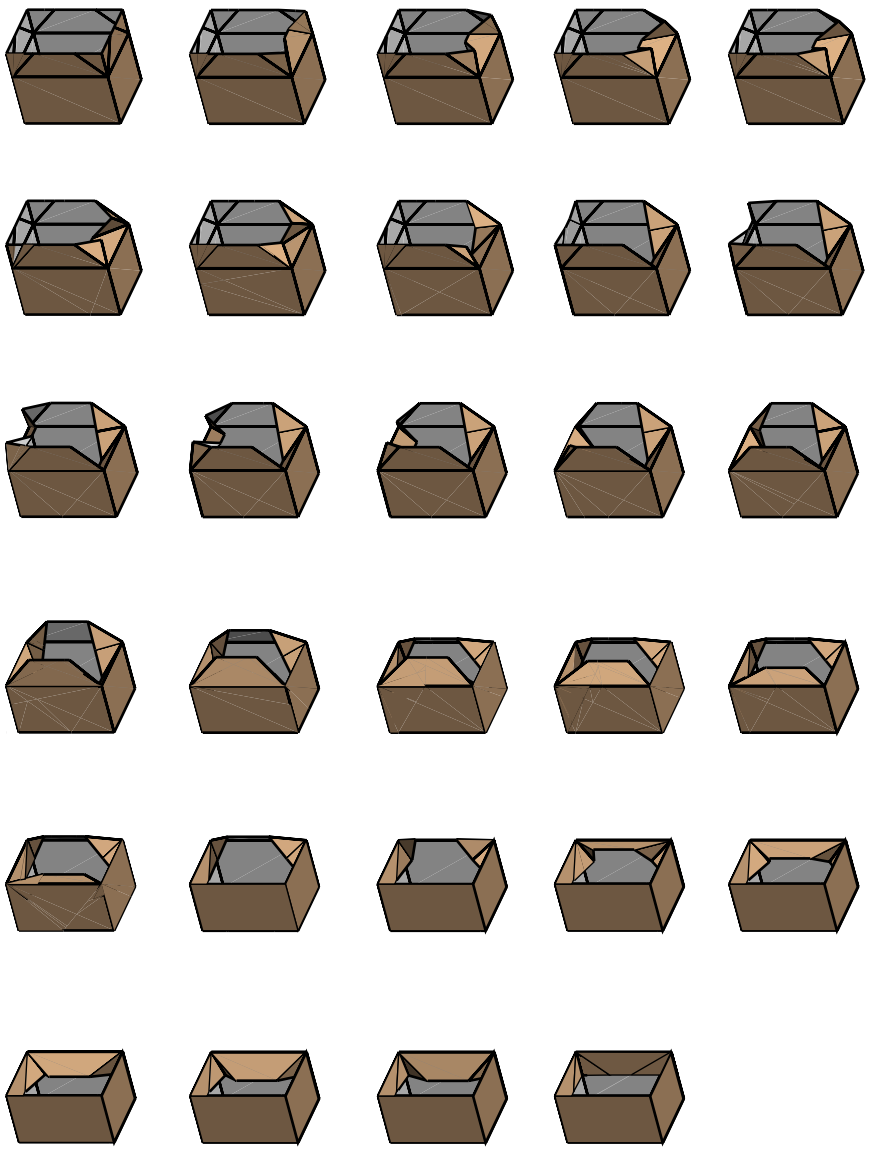


Figure 5.16: Procedure for shortening a rectangular tube.

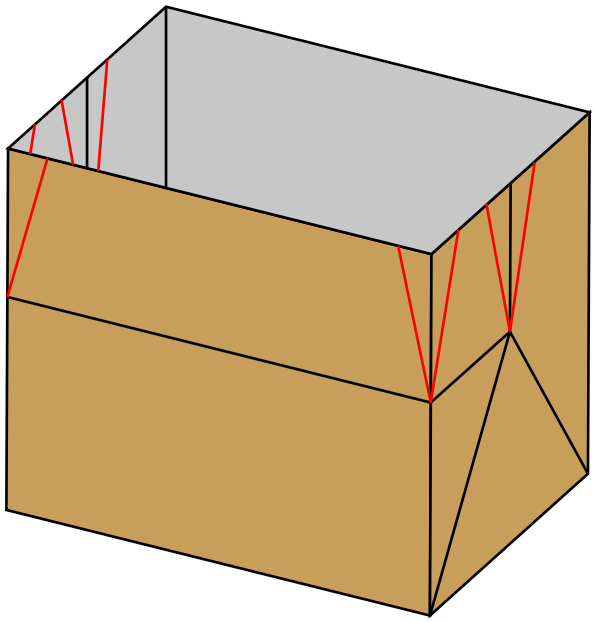


Figure 5.17: Conjectured solution to the problem of unfolding a paper bag by adding a finite number of creases. The creases to be added are shown in red.

The idea behind the conjectured solution is this. Recall that a short bag can be collapsed. One difference between a short bag and a tall bag is the addition of four additional interior vertices as the bag is lengthened and creases intersect on sides and at the intersection between the front and the sides. These additional vertices prevent the bag from being folded, as the proof of theorem 11 demonstrates. If these vertices could be 'cut out', the tall box could be folded. One way to do this would be to cut out four wedges, from the top of the box to each new vertex, removing the paper around the vertical crease in each case.

Although we can't cut the paper, we can add additional creases that will allow each vertex to act as though the vertical crease has been cut, at least within some range of motion. Adding just one crease connecting each vertex would seem to be enough, but the range of motion appears too limiting to collapse the bag; adding two creases around each vertex as shown in 5.17 is a more conservative solution.

We have built a plastic model of the conjectured solution, and it does seem to fold and unfold smoothly; in fact, it is hard to keep the bag unfolded, since gravity tends to pull it into the collapsed state!

Proving that this bag is truly collapsible would seem to require somewhat more detailed work than was done in theorem 12, since although the vertices are of equal degree (six), no degree-four motion that locks some of the creases will be sufficient to fold or unfold the bag from its initial open or collapsed state. The primary question is, "what happens near the collapsed state?" It seems obvious that the bag has some configurations that are neither open nor closed, and degrees of freedom at those points (the shopping bag with no additional creases does not), but are they lost as the crease vertices near singular configurations? Are there enough degrees of freedom to avoid intersections between the facets in the nearly flat configuration? A detailed proof that the new bag is foldable is left for future work.

Chapter 6

Conclusion

Where next? Current work has posed a lifetime of questions, with answers to only the most basic of them. How should flexible paper be manipulated and creased? We don't know. What is the configuration space of the simplest rigid-body model of a multi-vertex origami pattern? Even ignoring self-intersections, we don't know. What shapes can be collapsed into a flat state, and how? We don't know that, either.

Still, this thesis has presented some results. Primary among these is a more clear understanding of the origami domain, and some concrete problems on which to focus future efforts. There are also more practical results: what seems to be the world's first origami-folding robot, and what may be the world's first rigidly-collapsible shopping bag. Complete fold-sequence planning for a large subclass of human Pureland origami. New necessary conditions for flat-foldability and book-foldability. Proofs that simple, book, and degree-four vertex folding of flat origami are atomic, and self-intersection can occur only when the origami is flat. Formulas for the kinematics of single- and multi-vertex patterns. Geometric analysis of the topology of the configuration spaces of degree-four and degree-five single-vertex patterns, and a few results about the topology for degree-n patterns.

Immediate future work includes applying the results of the thesis to other domains where foldable structures are important. Further analysis of the configuration spaces of foldable mechanisms will also be an important next step. And of course, there will always be one more robot to build, and one more design to fold!

Appendix A

Some notes on Morse theory

This appendix collects some definitions and theorems from Morse theory that are useful for analysis of the topology of surfaces. A simple example is also worked out.

A.1 Definitions

The definitions in this section are based on Milnor [48], except where noted.

Smooth mappings between open sets: Let $U \subset R^k$ and $V \subset R^l$ be open sets. A mapping f from U to V is *smooth* if all of the partial derivatives exist and are continuous.

Diffeomorphism: A map $f : X \mapsto Y$ is called a *diffeomorphism* if f carries X homeomorphically onto Y and if both f and f^{-1} are smooth.

Manifolds: A set $M \subset R^k$ is called a smooth manifold of dimension m if each $x \in M$ has a neighborhood $W \cap M$ that is diffeomorphic to an open subset U of the Euclidean space R^m. Any particular diffeomorphism $g : U \mapsto W \cap M$ is called a parameterization of the region $W \cap M$. (The inverse diffeomorphism $W \cap M \mapsto U$ is called a system of coordinates on $W \cap M$.)

The derivative of a mapping between manifolds; tangent space: To define the derivative df_x for a smooth map $f : M \mapsto N$ between smooth manifolds, we associate with each $x \in M \subset R^k$ a linear subspace $TM_x \subset R^k$ of dimension m called the tangent space of M at x. Then df_x will be a linear mapping from TM_x to $TN_{f(x)}$. We write $df_x : TM_x \mapsto TN_{f(x)}$.

Concretely, df_x is represented by the Jacobian matrix of partial derivatives of f, evaluated at x. If $x \in M$ is a function of a parameter $t \in R$, then

$x(t)$ is a trajectory, and $\dot{x}$ evaluated at some specific time is a vector in the tangent space TM_x. If $y = f(x)$, then $J(x)\dot{x} = \dot{y}$, and $\dot{y}$ is in the tangent space TN_y.

Critical points, critical values, and regular values: We say that a map between two manifolds is *smooth* if all of the partial derivatives exist and are continuous. Consider a smooth map $f : M \mapsto N$, from a manifold of dimension m to a manifold of dimension n. Let C be the set of all $x \in M$ such that

$$df_x : TM_x \mapsto TN_{f(x)}$$

has rank less than n (is not onto). Then C will be called the set of *critical points*, $f(C)$ the set of *critical values*, and the complement $N - f(C)$ the set of *regular values* of f.

Index of a map: Definition based on Milnor [47]. The index of a bilinear functional H, on a vector space V is the maximal dimension of a subspace of V on which H is negative definite.

A.1.1 Theorems

Lemma 6 (Lemma 1 from Milnor [48]) *If $f : M \mapsto N$ is a smooth map between manifolds of dimension $m \geq n$, and if $y \in N$ is a regular value, then the set $f^{-1}(y) \subset M$ is a smooth manifold of dimension $m - n$.*

Lemma 7 (Morse) *Let p be a non-degenerate critical point for f. Then there is a local coordinate system $(y_1, \ldots, y_n)$ in a neighborhood U of p with $y_i(p) = 0$ for all i and such that the identity*

$$f = f(p) - y_1^2 - \ldots - y_\lambda^2 + y_{\lambda+1}^2 + \ldots + y_n^2 \tag{A.1}$$

holds throughout U, where λ is the index of f at p.

If f is a real-valued function on a manifold M, let

$$M^a = f^{-1}(-\inf, a] = p \in M : f(p) \leq a. \tag{A.2}$$

Theorem 13 (Theorem 3.1 from [47]) *Let f be a smooth real-valued function on a manifold M. Let $a < b$ and suppose that the set $f^{-1}[a, b]$, consisting of all $p \in M$ with $a \leq f(p) \leq b$, is compact, and contains no critical points of f. Then M^a is diffeomorphic to M^b. Furthermore, M^a is a deformation retract of M^b, so that the inclusion map $M^a \mapsto M^b$ is a homotopy equivalence.*

Theorem 14 (Theorem 3.5 from [47]) *If f is a differentiable function on a manifold M with no degenerate critical points and if each M^a is compact, then M has the homotopy type ofa CW-complex, with one cell of dimension λ for each critical point of index λ.*

A.1.2 A simple example: Morse on a sphere

A surface $M \subset R^n$ can be represented implicitly by a constraint function

$$g(x_1, \ldots, x_n) = 0. \tag{A.3}$$

Take as an example the sphere:

$$g(x_1, \ldots, x_n) = x_1^2 + x_2^2 + x_3^2 - 1 = 0. \tag{A.4}$$

We will apply some of the theorems above to determine the topological structure of the sphere. First, we want a mappping $f : M \mapsto R$. Pre-images of this function of the form $f^{-1}(c)$ slice the manifold. A height function is a good choice. Let's take

$$f(x_1, x_2, x_3) = x_3. \tag{A.5}$$

The slices of M generated by the pre-images of constants under f are circles in planes parallel to the $x - y$ plane. We want to find the critical points of the function f on M, since these are points where the slices can change topology (pretend we don't know what the slices look like).

We could find the critical points of f by finding coordinates for the implicit surface M, but parameterizing implicit surfaces is usually hard. So, we'll find critical points using the implicit representation. Geometrically, a critical point is any point $x \in M$ where for all vectors $v \in TM_x$ the directional derivative of f in the direction v is zero:

$$\nabla_v f = 0. \tag{A.6}$$

We can describe the tangent space to the sphere, TM, by taking a partials of the constraint function g. Specifically, $v \in TM_x$ implies that

$$\nabla g^T v = 0 \tag{A.7}$$

$$(x_1\ x_2\ x_3)v = 0. \tag{A.8}$$

The directional derivative, $\nabla_v f$, can be written using the gradient of f:

$$\nabla_v f = \nabla f^T v = (0\ 0\ 1)v = 0 \tag{A.9}$$

So critical points may occur where:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{A.10}$$

So at any critical point, $v_3 = 0$, and $x_1 v_1 + x_2 v_2 = 0$. There are critical points at $x_1 = x_2 = 0$; we can plug into the constraint function g to find that $x_3 = \pm 1$. (Technical note – there are no other critical points. If x_1 or x_2 is non-zero, then there exists a vector $v \in TM_x$ with $v_3 \neq 0$. Choose any non-zero v_3, set the element of v not associated with the non-zero element of x to zero, and solve for the final element of v.)

Now that we have the critical points, we want to describe the topology of the sphere. One way to do this is to find a special coordinate system (y_1, y_2) on the sphere near the critical points such that $f(y_1, y_2)$ has the special form described in the Lemma of Morse, above. Although possible for this special case, this typically requires inspired guess-work. Another way to find the topology is to choose some arbitrary coordinate system and evaluate the signature of the Hessian at the critical point. Choose

$$y_1(x_1, x_2, x_3) = x_1 \tag{A.11}$$
$$y_2(x_1, x_2, x_3) = x_2 \tag{A.12}$$

for $x_3 < 0$, and the same thing for $x_3 > 0$. Then

$$f(y_1, y_2) = \text{sign}(x_3)\sqrt{1 - y_1^2 - y_2^2}. \tag{A.13}$$

We want to evaluate the Hessian at critical points. The first partials of f are

$$\frac{\partial f}{\partial y_1} = -\text{sign}(x_3) y_1 (1 - y_1^2 - y_2^2)^{-1/2} \tag{A.14}$$
$$\frac{\partial f}{\partial y_2} = -\text{sign}(x_3) y_2 (1 - y_1^2 - y_2^2)^{-1/2}. \tag{A.15}$$
$$\tag{A.16}$$

As expected, the partials are zero when evaluated at the critical points. We

need three second partials to evaluate the (symmetric) Hessian:

$$\frac{\partial^2 f}{\partial y_1^2} = -\text{sign}(x_3)(1-y_1^2-y_2^2)^{-1/2} - \text{sign}(x_3)y_1^2(1-y_1^2-y_2^2)^{-3/2} \quad \text{(A.17)}$$

$$\frac{\partial^2 f}{\partial y_2^2} = -\text{sign}(x_3)(1-y_1^2-y_2^2)^{-1/2} - \text{sign}(x_3)y_2^2(1-y_1^2-y_2^2)^{-3/2} \quad \text{(A.18)}$$

$$\frac{\partial^2 f}{\partial y_1 \partial y_2} = -\text{sign}(x_3)y_1 y_2(1-y_1^2-y_2^2)^{-3/2} \quad \text{(A.19)}$$

At the critical points, $y_1 = y_2 = 0$, and the Hessian is:

$$\begin{bmatrix} -\text{sign}(x_3) & 0 \\ 0 & -\text{sign}(x_3) \end{bmatrix} \quad \text{(A.20)}$$

Since the Hessian is non-degenerate, our slice function f is Morse. For the first critical point, with $x_3 < 0$, the signature (number of positive eigenvalues, number of negative eigenvalues) of the Hessian is $(2,0)$. For the second critical point, with $x_3 < 0$, the signature is $(0,2)$.

Bibliography

[1] M. Anitescu and F. Potra. Formulating multi-rigid-body contact problems with friction as solvable linear complementarity problems. *ASME Journal of Nonlinear Dynamics*, 14:231–247, 1997.

[2] S. S. Antman. *Nonlinear Problems of Elasticity*. Number 107 in Applied Mathematical Sciences. Springer-Verlag, 1995.

[3] U. Ascher and P. Lin. Sequential regularization methods for simulating mechanical systems with many closed loops. *SIAM Journal on Scientific Computing*, 21(4):1244–1262, 1999.

[4] G. Aumann. Interpolation with developable Bézier patches. *Computer Aided Geometric Design*, 8:409–420, 1991.

[5] D. J. Balkcom, E. Demain, and M. Demaine. Finite-crease models for folding shopping bags, Sept. 2004. To be published as a Carnegie Mellon University technical report.

[6] D. J. Balkcom, E. Gottlieb, and J. Trinkle. A sensorless insertion strategy for rigid planar parts. In *Proceedings, IEEE International Conference on Robotics and Automation*, 2002.

[7] D. Baraff. Linear-time dynamics using Lagrange multipliers. In *SIGGRAPH*, pages 137–146, Aug. 1996.

[8] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH*, pages 43–54, July 1998.

[9] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact, and friction for cloth animation. In *SIGGRAPH*, 2002.

[10] Casady and Greene, Inc. *Origami: the secret life of paper*. Casady and Greene, 1997. Computer software.

[11] G. Chirikjian. Inverse kinematics of binary manipulators using a continuum model. *Journal of Intelligent and Robotic Systems*, 19:5–22, 1997.

[12] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. In *SIGGRAPH*, 2002.

[13] R. Connelly. Conjectures and open questions in rigidity. In *Proceedings of the international congress of mathematicians*, pages 407–414, 1978.

[14] R. Connelly, I. Sabitov, and A. Walz. The bellows conjecture. *Contributions to Algebra and Geometry*, 38(1):1–10, 1997.

[15] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, second edition, 1989.

[16] E. D. Demaine and M. L. Demaine. Recent results in computational origami. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, 2001.

[17] M. A. Erdmann. On motion planning with uncertainty. Master's thesis, Massachusetts Institute of Technology, Aug. 1984.

[18] C. Fernandes, L. Gurvits, and Z. Li. A variational approach to optimal nonholonomic motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 680–685, Apr. 1991.

[19] C. Gauss. *Disquisitiones generales circa superficies curvas*. Göttingen: Dieterich, 1828. (English translation by A. Hiltebietel and J. Morehead: Hewlett, NY, Raven Press, 1965.).

[20] M. Gleicher. *A Differential Approach to Graphical Manipulation*. PhD thesis, Carnegie Mellon University, 1994.

[21] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*. Addison-Wesley, third edition, 2002.

[22] S. Gupta, D. Bourne, K. K. Kim, and S. S. Krishnan. Automated process planning for robotic sheet metal bending operations. *Journal of Manufacturing Systems*, September 1998.

[23] L. Han and N. M. Amato. A kinematics-based probabalistic roadmap method for closed chain systems. In *Workshop on the Algorithmic Foundations of Robotics*, pages 233–246, May 2000.

[24] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2003. Available online from `http://www.math.cornell.edu/~hatcher`.

[25] D. Hilbert and S. Cohn-vossen. *Geometry and the Imagination*. Chelsea, 1952.

[26] S. Hirai, T. Tsuboi, and T. Wada. Robust grasping manipulation of deformable objects. In *Proceedings of the IEEE Symposium on Assembly and Task Planning*, pages 411–416, May 2001.

[27] S. Hirai, H. Wakamatsu, and K. Iwata. Modeling of deformable thin parts for their manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2955–2960, 1994.

[28] D. Huffman. Curvature and creases: A primer on paper. *IEEE Transactions on Computers*, C-25(10):1010–1019, Oct. 1976.

[29] T. C. Hull. On the mathematics of flat origamis. In *Congressus Numerantium*, volume 100, pages 215–224, 1994.

[30] H. Inoue and M. Inaba. Hand-eye coordination in rope handling. In *Robotics Research: The First International Symposium*, pages 163–174, 1985.

[31] D. L. James and D. K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e, the Electronic Journal of Haptics Research*, 2(1), September 2001. `http://www.haptics-e.org`.

[32] M. Kapovich and J. Millson. On the moduli space of polygons in the euclidean plane. *Journal of Differential Geometry*, 42(1):133–164, 1995.

[33] M. Kapovich and J. Millson. Moduli spaces of linkages and arrangements. *Canadian Mathematics Bulletin*, 42(3):307 – 320, 1999.

[34] L. E. Kavraki, F. Lamiraux, and C. Holleman. Towards planning for elastic objects. In *Workshop on the Algorithmic Foundations of Robotics*, pages 313–326, 1998.

[35] Y. L. Kergosien, H. Gotoda, and T. L. Kunii. Bending and creasing virtual paper. In *IEEE Computer Graphics and Applications*, pages 40–48, Jan. 1994.

[36] R. Lang. Trees and circles: an efficient algorithm for origami design. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, 2001.

[37] R. Lang. Papers and folding techniques for 3-d shaping, Dec. 2003. A three-part article posted to the Origami List (for more information on the origami list, see `http://www.origami-usa.org/internet.htm`.).

[38] S. Leopoldseder and H. Pottmann. Approximation of developable surfaces with cone spline surfaces. *Computer Aided Design*, 30:571–582, 1998.

[39] G. Liu and J. Trinkle. The motion planning existence problem for a planar 2-R manipulator with point obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004. submitted.

[40] G. Liu, J. Trinkle, and R. Milgram. Complete path planning for a planar 2-R manipulator with point obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004. submitted.

[41] P. Lötstedt. Coulomb friction in two-dimensional rigid-body systems. *Zeitschrift für Angewandte Mathematik und Mechanik*, 61:605–615, 1981.

[42] T. Lozano-Perez. A simple motion-planning algorithm for general robot manipulators. *IEEE Journal of Robotics and Automation*, RA-3(3):224–238, June 1987.

[43] L. Lu and S. Akella. Folding cartons with fixtures: A motion planning approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1999.

[44] L. Lu and S. Akella. Folding cartons with fixtures: A motion planning approach. *IEEE Transactions on Robotics and Automation*, 16(4):346–356, Aug. 2000.

[45] J. M. McCarthy. *Geometric Design of Linkages*, volume 11 of *Interdisciplinary Applied Mathematics (Systems and Control)*. Springer-Verlag, 1995.

[46] R. J. Milgram and J. Trinkle. The geometry of configurations spaces for closed chains in two and three dimensions. *Homology, Homotopy, and Applications*, forthcoming.

[47] J. Milnor. *Morse Theory*. Number 51 in Annals of Mathematics Studies. Princeton University Press, 1973.

[48] J. Milnor. *Topology from the Differentiable Viewpoint*. Princeton Landmarks in Mathematics. Princeton University Press, 1997.

[49] M. Minsky. Manipulator design vignettes. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1972.

[50] S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaki. An interactive simulation system of origami based on virtual space manipulation. In *Proceedings of the IEEE International Workshop on Robot and Human Communication*, pages 210–215, 1992.

[51] H. Nakagaki, K. Kitagaki, T. Ogasawara, and H. Tsukune. Study of deformation and insertion tasks of a flexible wire. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2397–2402, Apr. 1997.

[52] D. K. Pai. STRANDS: Interactive simulation of thin solids using cosserat models. In *Eurographics*, 2002.

[53] P. Painlevé. Sur les lois du frottemont de glissement. *C. R. Acad. Sci.*, 121:112–115, 1895.

[54] H. Pottmann and J. Wallner. Approximation algorithms for developable surfaces. *Computer Aided Geometric Design*, 16:539–556, 1999.

[55] P. Redont. Representation and deformation of developable surfaces. *Computer Aided Design*, 21(1):13–20, 1989.

[56] M. Rubin. *Cosserat Theories: Shells, Rods, and Points*. Kluwer Academic Publishers, 2000.

[57] G. Song and N. M. Amato. A motion planning approach to folding: From paper craft to protein folding. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 948–953, May 2001.

[58] D. Stewart and J. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39:2673–2691, 1996.

[59] D. E. Stewart. Existence of solutions to rigid body dynamics and the paradoxes of Painlevé. *Comptes Rendus de l'Acad. Sci., Sér. I*, 1997. To appear.

[60] M. Sun and E. Fiume. A technique for constructing developable surfaces. In *Graphics Interface*, pages 176–185, May 1996.

[61] K. R. Symon. *Mechanics*. Addison-Wesley, third edition, 1971.

[62] J. Thorpe. *Elementary Topics in Differential Geometry*. Springer-Verlag, 1979.

[63] J. Trinkle and R. Milgram. Motion planning for planar *n*-bar mechanisms with revolute joints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1602–1608, 2001.

[64] J. Trinkle, J. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with coulomb friction. *Zeitschrift für Angewandte Mathematik und Mechanik*, 77(4):267–279, 1997.

[65] T. Wada, S. Hirai, T. Hirano, and S. Kawamura. Modeling of plain knitted fabrics for their deformation control. In *Graphics Interface*, pages 176–185, May 1996.

[66] T. Wada, S. Hirai, S. Kawamura, N. Kamiji, and K. Tsukamoto. Robust manipulation of deformable objects by a simple pid feedback. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 85–90, May 2001.

[67] T. Wada, S. Hirai, S. Kawamura, N. Kamiji, and K. Tsukamoto. Robust manipulation of deformable objects by a simple position feedback. In *3rd Asia-Europe Congress on Mechatronics*, pages 112–117, Oct. 2001.

[68] T. Wada, B. J. McCarragher, H. Wakamatsu, and S. Hirai. Modeling of shape bifurcation phenomena in manipulations of deformable string objects. *International Journal of Robotics Research*, 15(8):833–846, 2001.

[69] H. Wakamatsu, S. Hirai, and K. Iwata. Static analysis of deformable object grasping based on bounded force closure. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3324–3329, 1996.

[70] H. Wakamatsu, T. Matsumura, S. Hirai, and E. Arai. Dynamic analysis of rodlike object deformation towards their dynamic manipulation. In

Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 196–201, 1997.

[71] G. Weiss and P. Furtner. Computer-aided treatment of developable surfaces. *Computers and Graphics*, 12(1):39–51, 1988.