AUTOMATED PAPER POP-UP DESIGN: APPROXIMATING SHAPE AND MOTION

CONRADO DEL ROSARIO RUIZ JR

(B.S. (cum laude), DLSU, M.Sc., NUS)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Conrado del Rosario Ruiz Jr

July 2015

Acknowledgments

I would like to thank the rest of the pop-up research team, Ngoc Sang Le, Vu Le and Su-Jun Leow. I would also like to thank Armandarius Darmadji and Jinze Yu for their help in the implementation. I am also grateful to my supervisor Dr. LOW Kok-Lim for his guidance and support throughout my PhD candidature. I would also like to thank all the members of the G3 Lab for their help and encouragement. I also want to express my deepest gratitude to my friends and family for all their support. Finally, I offer the completion of this dissertation to our Lord.

This work was funded by the Singapore MOE Academic Research Fund (Project No. T1-251RES1104). The PhD candidate was supported by the President's Graduate Fellowship. The 3D models are from Google 3D Warehouse and Blender Swap. All trademarks, brands and photos of books are property of their respective owners.

Contents

Summary

Paper pop-ups are interesting three-dimensional books that fascinate people of all ages. The design and construction of these pop-up books however are generally done by hand and given the lack of expertise in this area has necessitated the need for computer-automated or -assisted tools in designing paper pop-ups. Pop-up design is usually centered on two qualities, namely three-dimensionality and movement. In this thesis, we consider both aspects in our automated design. Previous computational methods have only focused on single-style pop-ups, where each is made of one type of pop-up mechanism. This dissertation explores the facets of the problem for the automated design of multi-style paper pop-ups. In addition, we also consider movement, which has not been the focus of any previous work.

First, we conduct a geometric study of the valid configurations of the paper patches to obtain the conditions for the foldability and stability of pop-up structures. Second, we study the motion of the patches during the folding process, which artist take advantage of to create pop-ups with some form of animation. We then propose a method for approximating the shape of an input mesh using paper pop-ups. Our method abstracts a 3D model by fitting primitive shapes that both closely approximate the input model and facilitate the formation of the pop-up mechanisms. Each shape is then abstracted using a set of 2D patches that combine to form a valid pop-up that is supported by our formulations.

We also propose an approach to reproduce the motion of 3D articulated characters. We map each linkage chain of an articulated figure to a specific pop-up mechanism based on the type of motion it can produce. We then obtain the initial values of the parameters of the mechanisms, based on our formulations and parameter estimation. Subsequently, we utilize simulated annealing to search for a plausible layout from a valid configuration space. Our main goal is to propose a framework to support the automated design of multi-style animated paper pop-ups.

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Paper pop-ups are fascinating three-dimensional books containing paper pieces that rise up or move when the book is opened and folded completely flat when the book is closed. Although, now popularly used for children's books, it was not until the 18th century when pop-up books were used for children's literature. Historically, it was also used for a wider range of topics like philosophy, astronomy, geometry and medicine. One of the first movable books was recorded in Spain during the 13th century that was made by Ramon Llull for mystical philosophy. Today's pop-up books still continue to fascinate readers of all ages and cultures, some of the more notable titles are made by artists like Robert Crowther, Robert Sabuda, David Carter and Matthew Reinhart (see Figure 1.1).

Recently, there has been much interest in the physical fabrication of 3D models. Paper pop-ups are a practical candidate for this task since they do not require specialized hardware and they can be folded flat for easy storage. Just as algorithms in origami have found applications in protein folding and deploying instruments in space, pop-up algorithms could be potentially used for other applications. Examples include 3D micro-fabrication from 2D patterns and collapsible objects such as foldable furniture.

Pop-up design is challenging because it requires both artistic skill and technical expertise. It requires an artistic sense of what the message the author wishes to convey through the use of colors, shapes and images. At the same time, it also requires some technical knowledge of the proper configuration of the pieces

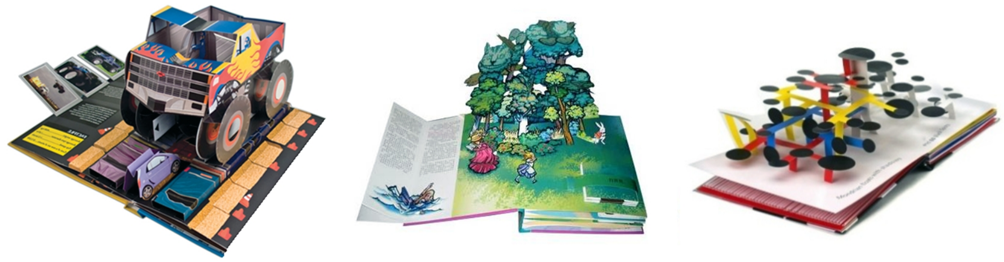


Figure 1.1: Sample pop-up books (left to right): Amazing Pop-up Trucks [Cro11], Alice's Adventures in Wonderland [CS03] and Yellow Squares [Car08].

to make it a valid paper pop-up. For this reason it is also known as paper engineering and pop-up designers are also known as paper engineers.

Creating a pop-up can be a tedious task even for an experienced designer. It usually entails a trial-and-error approach to find configurations of the pieces that would work. A pop-up prototype usually takes weeks to complete. An entire pop-up book can take up to a year to finish. Furthermore, paper engineers are scarce and there are no formal venues to acquire the necessary skills to make paper pop-ups. Computer aided-design has found numerous applications in industrial and architectural design and now shows great potential in pop-up design. Coupled with the proliferation of 3D models on the web and the easy accessibility to 3D authoring software, we propose an automated approach for converting 3D models into valid paper pop-ups.

A pop-up is considered valid when it is both *foldable* and *stable*. A pop-up is said to be *foldable* if the structure can fold completely flat when the ground and backdrop patches are fully closed. Note that during the folding process, the rigidity and connectivity of the patches need to be maintained at all times and it should not introduce new fold lines. On the other hand, a pop-up is said to be *stable* if all its patches are stationary when the ground and backdrop patches are held still at any fold angle. In other words, the closing and opening of a pop-up do not need any extra external force besides holding the two primary patches.

Most of the work in computation pop-up design focuses on a small set of mechanisms and on developing interactive design tools. These tools are meant to replace the actual cutting, gluing and folding paper during the design process with virtual simulations. Nonetheless, some have also explored the geometric properties of pieces of paper and the conditions that make it a valid pop-up. Investigating these conditions can lead to systems that can guarantee the validity of a design just by considering the opened state of the pop-up. It can also give better feedback on the design of the pop-up. Research on computation pop-up design is still at its early stages and numerous directions have not yet been explored.

The only methods that are able to automatically generate pop-up designs are [LSH$^+$10], [LJGH11], [LNLRL13] and [LLLN$^+$14]. In these works, a pop-up is made of only a single type of pop-up *mechanisms* (i.e. single-*style* pop-up), and a very specialized method is used to generate the pop-up design. The v-style pop-ups addressed by [LJGH11] seem to be the most versatile in terms of geometry. However, the main focus of [LJGH11] was on the geometric study, and its automatic method can only generate pop-up patches restricted to three perpendicular orientations. As such, it is not able to demonstrate the full potential of the v-style mechanism. [LNLRL13] focuses on sliceforms or lattice-style pop-ups and [LLLN$^+$14] focuses on Origamic Architectures. These are our previous publications leading to this work.

In actual pop-up books created by artists, numerous styles are used to suitably represent different parts of the objects. Our objective is thus to combine multiple styles in a pop-up, and use the most suitable mechanism for each part of the object. Combining multiple styles presents new challenges in the validation of its stability and foldability. In this thesis, we aim to provide new geometric conditions for the validity of multi-style pop-ups.

In our work, we consider several types of mechanisms, the *step-fold*, *tent-fold*, *v-fold* and *box-fold* (refer to Figure 1.2). Of these mechanisms, the box-fold has not yet been studied in any previous work. As such we formally include a

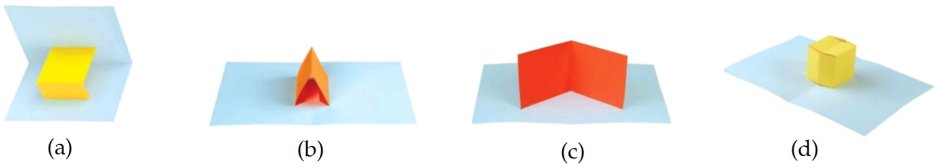


Figure 1.2: Pop-up mechanisms: (a) step-fold, (b) tent-fold, (c) v-fold and (d) box-fold.

description of the box-fold and outline the conditions for its validity considering its foldability and stability.

Our current method abstracts the input 3D model using suitable primitive shapes that both facilitate the formation of the considered pop-up mechanisms and closely approximate the input model. Each shape is then abstracted using a set of 2D patches that combine to form a valid pop-up.

In the automated approaches of [LSH$^+$10] and [LJGH11], voxelization is used to approximate the input 3D model, which leads to the possible loss of important features. In our work, the shape abstraction allows us to fit a minimal number of 3D primitives to approximate the model, resulting in fewer patches. The final patches are produced using an image-based approach to preserve the textures, finer details and important contours of the input model.

Most pop-up artists are concerned with representing the shape of 3D objects, however some have also used the movement of the paper pieces during the opening process to reproduce motion. This technique has been used to produce animations of persons swimming, running and objects peeking out. For example, in his pop-up book [CS03], Robert Sabuda has used sophisticated mechanisms to produce the movement of characters in “Alice’s Adventures in Wonderland” (refer to Figure 1.3). In this thesis, we also study the movement of the paper pieces of the different mechanisms used in pop-up structures in order to use this knowledge to automatically design pop-ups from the motion of articulated 3D characters.

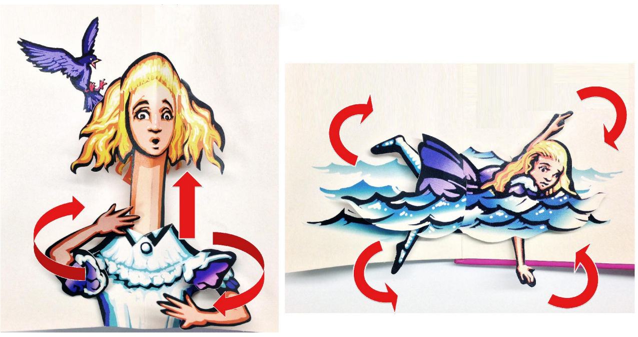


Figure 1.3: Examples of movement in the "Alice's Adventures in Wonderland" pop-up book by Robert Sabuda [CS03].

Our input is an animation file, containing the mesh and armature information. The armature is divided into single linkage chains and their end effectors are matched to the motion of a pop-up mechanism. After the mechanism mapping and parameter estimation, we have an initial configuration of the combination of mechanisms and their respective parameters. We continuously modify the parameters and mappings to find an optimal layout that best approximates the input motion while avoiding intersections. This leads to a huge configuration search space, which we explore using simulated annealing, keeping non-collision as a hard constraint. We show the feasibility of our approach by presenting the actual paper pop-ups constructed using the generated design layout.

1.1 Contributions

This thesis provides a framework to support the automated design of multi-style animated paper pop-ups. These pop-ups combine multiple mechanisms and incorporate motion that is more representative of actual artist's creations, which has not been extensively studied. The specific contributions of this thesis are as follows:

1. A formal study the craft of designing paper pop-ups. In order to automate the process of designing pop-up books, it also requires the examination of

the manual process of designing pop-up books. This provides the necessary knowledge and foundation for work in this area.

2. A geometric study of the valid configurations of pop-up structure. We determine the constraints on the positions, orientations and linkages of the paper patches of a pop-up structure that lead to a valid paper pop-up. Specifically, we focus on the complications when combining multiple types of mechanisms together.

3. A study of the motion of the patches during the folding process, which artists use to create pop-ups with some forms of animation. We parameterize each pop-up mechanism and describe the motion produced in relation to these parameters.

4. Implementations of the framework to convert 3D models into valid paper pop-up designs. We present implementations for representing the 3D volume of the input mesh and for reproducing the motion of its parts during the folding process.

1.2 Methodology and Scope

Our methodology involves a geometric study to determine the constraints of valid pop-up structures. These serve as the foundation of our automated algorithms. The volume and shape representation algorithm is based on the work on shape abstraction. The reproduction of motion approach is inspired by kinematic synthesis of mechanical assemblies. We implement these techniques and verify the validity and realizability of our pop-up designs.

Paper pop-up books are part of a general class of movable books including those that use strings, rotating disks and other mechanisms, however for the purposes of this thesis we mainly refer to those pop-up books made of only paper. Even with cutting pieces of paper and gluing alone, elaborate and complex pop-up books can be created and are already difficult to model computationally. In addition, we will not consider the mechanisms that require addition force from the user other than holding the base patches or cover of the book. Examples of mechanisms that require additional user intervention are pulling tabs or flaps, sliding or dissolving scenes, etc.

Specifically we will consider the following mechanisms and the combination of the mechanisms: parallel folds (e.g. tent-fold, box-fold step-fold) and angled folds (e.g. v-fold). We assume that we can use multiple sheets of paper and this is not a constraint like in origamic architectures.

In addition, our formulations will only be sufficient conditions for validity and not necessary conditions. We also consider paper as a rigid material, which is the assumption held by all of the current research in the area. Although several pop-up mechanisms rely on bending the paper, without this assumption it will significantly change the definitions for stability and foldability. Furthermore, the presented geometric formulations here do not take into account the physical characteristics of paper. In actual pop-up design, the thickness, mass, strength and elasticity of paper are important considerations.

Lastly, we do not have quantitative assessment of the aesthetic quality of our pop-ups. Such measurements can be very beneficial in creating more visually appealing paper pop-up designs but is beyond the scope of this dissertation. We can however quantitatively measure the volume difference with the input 3D mesh and mathematically check the validity of our pop-up structures.

1.3 Organization

This dissertation starts by providing the necessary background and related research on paper pop-ups. We present a survey of work on computational pop-up designs and other related areas. Then, we present current formulations for paper pop-ups and present our work in this area. After that, we discuss the algorithms and present the results. The contents of the organization of the chapters are as follows:

Chapter 2 provides a background of paper pop-ups focusing on basic terminologies and types of pop-up mechanisms, as well as the taxonomy of these mechanisms.

Chapter 3 describes the related work on computational paper pop-up designs. We also discuss work in other forms of papercrafts like origami and kirigami. In addition, we also review work on shape abstraction and mechanism synthesis of mechanical assemblies.

Chapter 4 presents the formal definitions of pop-up mechanisms and the geometric conditions for the validity of pop-up structures. We also describe the output motion of a set of pop-up mechanisms.

Chapter 5 describes the details of our work on converting 3D models into valid paper pop-up designs, focusing on reproducing its 3D shape and volume.

Chapter 6 explains our algorithm to recreate the motion of an articulated figure using an animated pop-up structure.

Chapter 7 describes the technical design of our system using UML 2.0 and implementation details.

Chapter 8 presents our results for automatically converting 3D models into pop-up designs, both considering the volume and shape of the input 3D model as well as the motion of its parts.

Chapter 9 concludes our work and presents possible future work.

Chapter 2

Background

This chapter aims to provide the necessary foundation in order to understand pop-up design and construction. First, we explain common terms and mechanisms used in paper pop-up design. Note that there is no standardized nomenclature for pop-ups, different books and artists use different terminologies. Here we consolidate some of the more respected books on pop-up design [Hin86, Jac93, Bir11, CD99] and use these terms throughout the dissertation.

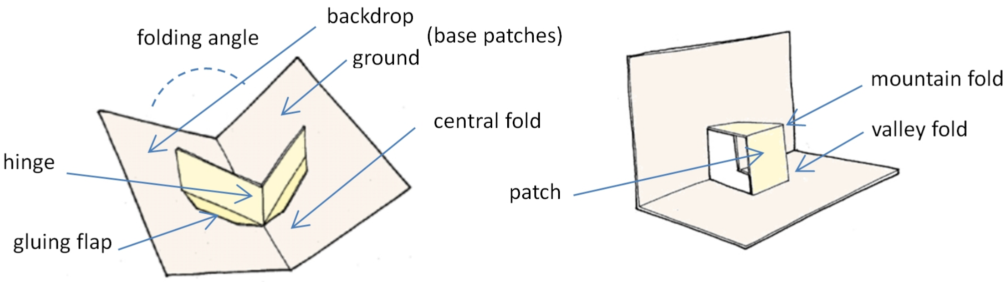


Figure 2.1: Parts of a Paper Pop-up.

2.1 Terms and Definitions

1. Pop-up book. Pop-up books refer to a variety of movable books that employ numerous mechanisms. For the purposes of this thesis we define it as a book composed of pieces of paper that *"pop out"* when the book is opened and is completely folded when it is closed. It is made up of paper pieces that are glued to other pop-up pieces.

2. Paper engineering and engineers. Also known as pop-up art/craft and pop-up artist. The art and craft of creating a pop-up is also called paper engineering because of the technical skills also required to make a pop-up foldable. Pop-up artist are also therefore called paper engineers.

3. Base patches (ground/backdrop). Also known as base pages/cover/card, backing sheet and primary patches. This serves as the base of the pop-up; these are the two main pages on which the pop-ups are built on.

4. Central fold. Also known as spine-fold, central crease or hinge. The main crease that is co-planar with both pages of the backing sheet.

5. Folding angle. The angle between the two base patches.

6. Hinge. The line segment where two patches meet. This may be a fold or gluing tab.

7. Folds. Mountain folds are creases that move towards the viewer, while valley folds are those that move away from the viewer. Crease or seam is a line segment made by folding or scoring.

8. Slits and slots. A slit is a simple cut on the piece of paper; slots are wider and may allow other paper pieces to pass through.

9. Patch. A plane whose boundary is a cut, fold or hinge.

10. Scaffold. It is a collection of patches that are connected using hinges.

11. Mechanism. The basic element of a pop-up structure. A minimal set of paper patches that form a valid pop-up scaffold.

12. Style. A class of mechanisms that share topological or geometric attributes.

2.2 History and Evolution of Pop-ups

In this section we examine the history of paper pop-ups focusing on the evolution of its mechanisms. This is necessary in order to appreciate and understand the structure, craft and uses of pop-ups in contemporary times. Pop-ups are part of a more general class of books, called movable books. Movable books generally focus on the three-dimensionality and movement of its paper pieces. Most of the information from this section are collated from the works of [Hen08, Hin02, Rub13].

The first movable books were intended for much more serious subject matters like medicine, mathematics and astronomy. Before the 18th century, books in general were not intended for children. The first books that were eventually made for children were usually of a religious nature. Children's books filled with stories or intended to teach a subject matter is a relatively modern idea.

The first known movable books before the 1700s used two main mechanisms, volvelles and flaps. Volvelles or wheels are made by attaching its center to the page with a knotted linen string or a rivet, the volvelle could rotate independent of the page or used in conjunction with other volvelles. An example of a volvelle as used by Ramon Llull is shown in Figure 2.2.

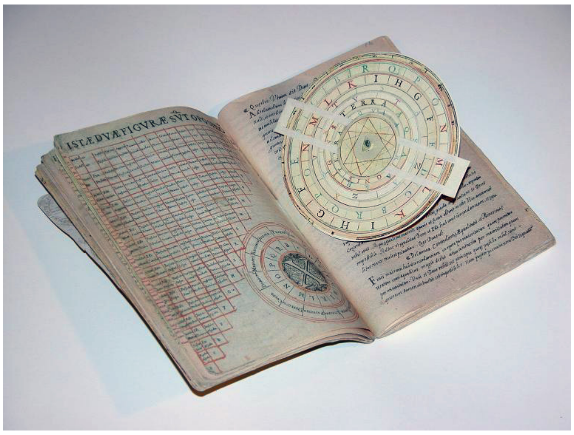


Figure 2.2: Volvelle in Ramon Llull's *Ars Magna*.

Flaps are additional pieces of paper added to a book, either by gluing or folding. This provides extra layers and cutting away the upper layers to allow the reader to open the flap to expose the lower layers. The flap can be lifted to see what is underneath it. The flap was the first attempt to incorporate 3D into books and was the beginning of attempts in the next centuries to give book illustrations more depth. Examples of flaps are shown in Figure 2.3.

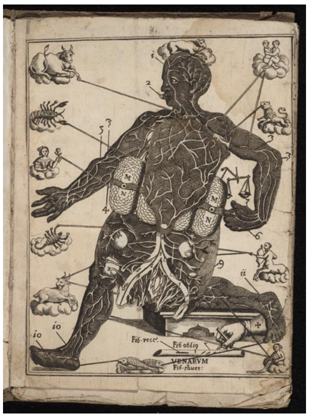


Figure 2.3: Flaps in Daniel Ricco's *Ristretto Anotomico.* Photo from [oC15].

In the 18th century, the idea of books specifically for children was introduced. In terms of mechanisms, there was no major change and most of them still used wheels and flaps. By the 19th century, we have seen the advent of a new generation of movable books namely, panoramas and tunnel books.

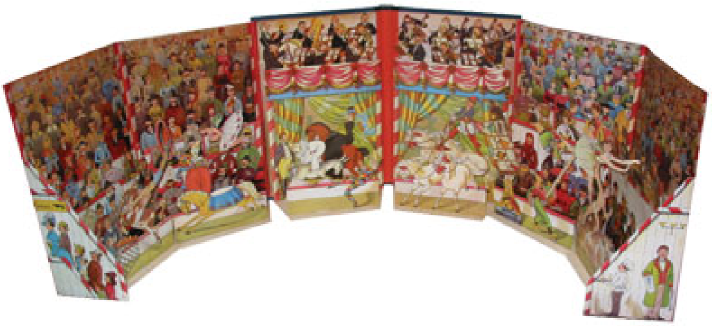


Figure 2.4: Panorama of Lothar Meggendorfer's *International Circus* [Meg79].

A panorama is a type of book that does not have the usual cover and pages, instead it can be unfolded into a long zig-zag image. Figure 2.4 shows the use of a panorama in Lothar Meggendorfer's *International Circus.* A carousel book is a variation, where the book is folded into itself forming a star shape.

A tunnel book is an accordion-type mechanism. It is also known as a peep show since you have to peep through a hole to see an image inside the mechanism. It usually requires the reader to pull the mechanism out. Figure 2.5 shows an example of a peep show from Crystal Palace Tunnel Book printed in 1851.

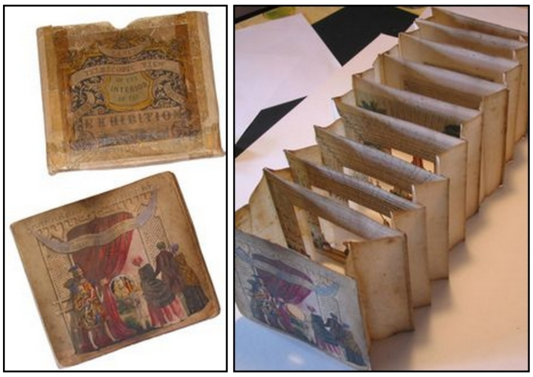


Figure 2.5: Crystal Palace Peep Show Tunnel Book

Towards the early 20th century, a few more mechanisms were introduced. These were the pull-out tabs, scenes and transformations. This was considered as the golden age of movable books. *Scenes* are pieces of paper that form overlapping layers that can be pulled out to create a 3D effect. If we look closely at *International Circus* each page actually uses this mechanism (see Figure 2.6).

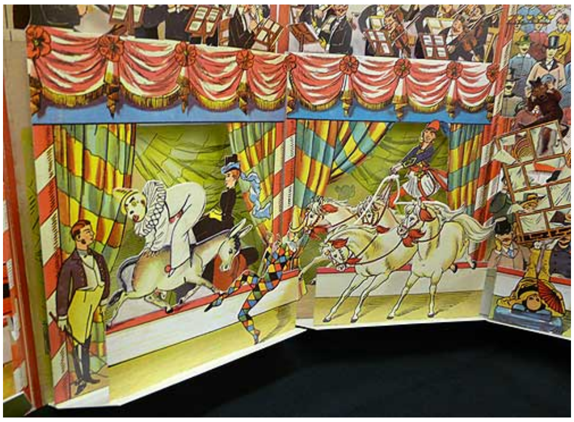


Figure 2.6: Pull-out scene from Lothar Meggendorfer's *International Circus* [Meg79].

The transformation mechanism in pop-up books is used to transition from one scene to another picture by pulling a tab. Usually the images are fitted together so that when they are slided it covers one part and reveals another. Figure 2.7 shows a transformation scene from J.F. Schreiber's *Schoolboy Pranks* [Sch97].

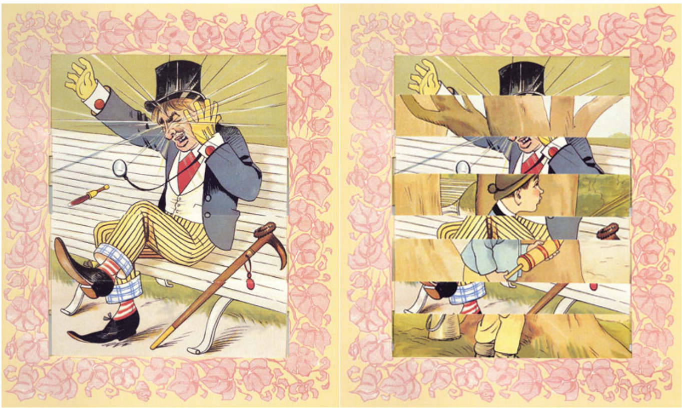


Figure 2.7: Transformation scene from J.F. Schreiber's *Schoolboy Pranks* [Sch97].

Most of these mechanisms however have to be manually operated. It was only in the later part of the 20th century when the pop-up books that we know today were introduced. Most of the modern pop-up books only require the reader to open the cover to initiate the movement of the paper pieces inside.

One of the first publishers of these types of modern pop-up books was S. Louis Giraud. He also called these books, *Bookano*. However, it is said that the idea originally came from Theodore Brown. Figure 2.8 shows some of the pop-up books the Giraud published. The mechanisms used here, i.e. v-fold and parallel folds, are still used in modern paper pop-ups.

Recently, paper pop-up books have become more elaborate and intricate. It has now come full circle and is not only intended for children but for adults as well. The most famous work are made by pop-up artists like Robert Crowther, Robert Sabuda, David Carter and Matthew Reinhart. Figure 2.9 shows some of the recent titles published by Reinhart [Rei07, Rei13, Rei14].

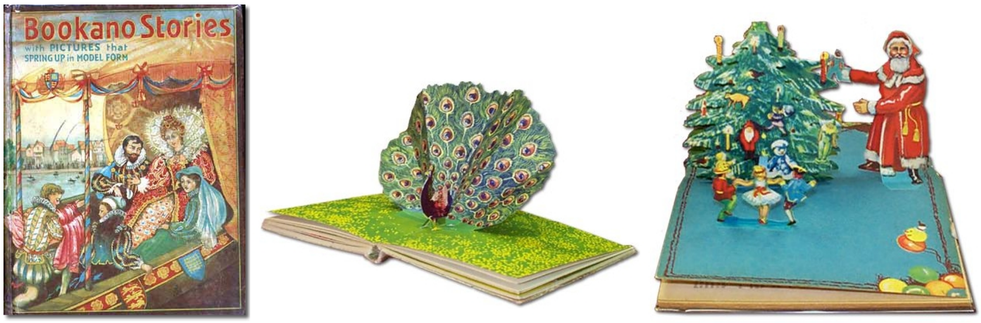


Figure 2.8: Pop-up books or Bookano made by S. Louis Giraud.

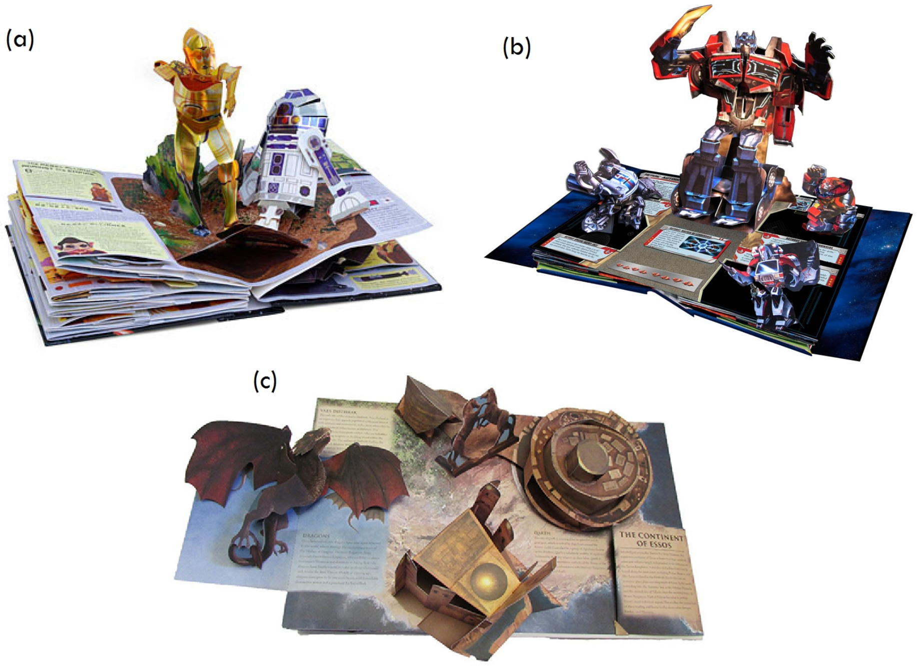


Figure 2.9: Pop-up books by M. Reinhart: (a) Star Wars: A Pop-Up Guide to the Galaxy [Rei07], (b) Transformers: The Ultimate Pop-Up Universe [Rei13], and (c) Game of Thrones: A Pop-Up Guide to Westeros [Rei14].

2.3 Pop-up Mechanisms

Modern paper engineers employ numerous pop-up mechanisms to create pop-up books. However, these techniques are usually based on only a handful of basic mechanisms. We discuss these basic mechanisms consolidated from the prominent design books of [Hin86, Jac93, Bir11, CD99, Wen10, Hen08]. As with the terminologies used for pop-ups, there is no standardized names for these mechanisms. However, we will use the terms listed here consistently throughout this thesis.

1. Single-slit angle fold. The most elementary mechanism used for a single piece of paper is a slit. This is made by simply cutting and folding at certain angles so that portions of the paper will pop out when the folding angle is 90°. Since it is made of a single sheet of paper without gluing, it will not pop-out when the backing sheets are completely opened. This is usually used for depicting mouths, beaks and other such openings. See Figure 2.10 for an example.

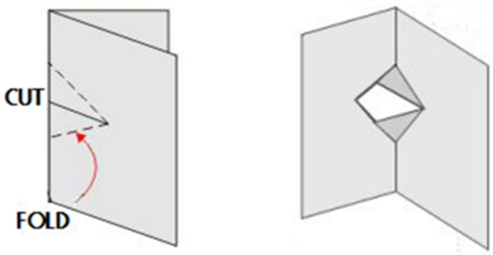


Figure 2.10: Single-slit Angle Fold Mechanism.

2. Double-slit folds. This mechanism also uses only a single sheet of paper but with two slits and a hinge (fold line). If the fold line is always parallel to the central fold it is called a *parallel-fold*; otherwise it is called a *non-parallel fold*. It erects when the base patches are at 90°. Refer to Figure 2.11 for some examples.

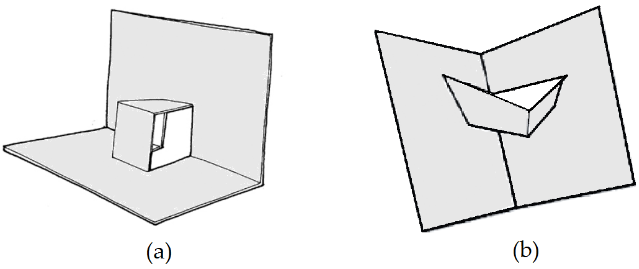


Figure 2.11: Double-slit Fold Mechanism. (a) Parallel (b) Non-Parallel.

More complicated pop-ups can be created by simply using slits and a single sheet of paper. One very interesting pop-up style using this mechanism was created by Masahiro Chatani called the *Origamic Architecture (OA)* or *Paper Architecture.* As the paper is opened to a 90 degree angle the structure *stands-up* or *pops-up.* A parallel OA is where all the patches remain parallel to one of the backing sheets. This is a common mechanism for pop-up cards and usually depicts buildings that requires no gluing or additional sheets. See Figure 2.12 for some examples designed by [GS09].

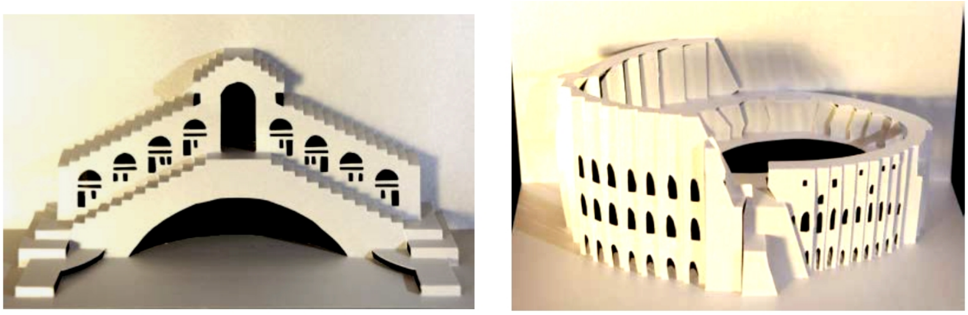


Figure 2.12: Origamic Architecture Examples.

3. Step Folds. These folds share similar attributes with the double-slit folds, with the exception that it is made up of more than one paper patch. Figure 2.13 shows how the step fold differs from the double-slit mechanism due to the fact that another paper patch was added using gluing flaps to the pop-structure. Similarly, it erects at 90° and its fold may be parallel or non-parallel to the central fold.

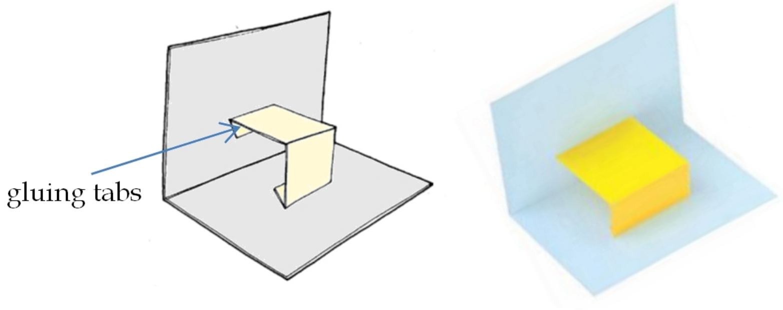


Figure 2.13: Step Fold.

4. V-Fold. Also known as 180° angle fold. It is one of the most common mechanisms used in pop-up books. Some pop-up books use only this mechanism or use this as the main backbone structure of the entire pop-up. The v-fold is a pair of stable patches that stands up when the base patches are opened; it collapses into itself when the book is closed. The fold is aligned or converges the central fold. This versatile form is what most people think of when they hear the term "pop-up." An example of a v-fold is shown in Figure 2.14 .

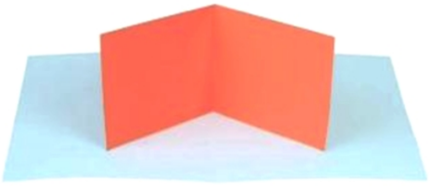


Figure 2.14: Simple V-fold

Changing the angle of patches allows the designer to create other shapes; however the main mechanism of the pop-up still remains the same. Figure 2.15 shows some of the variations of the v-fold.

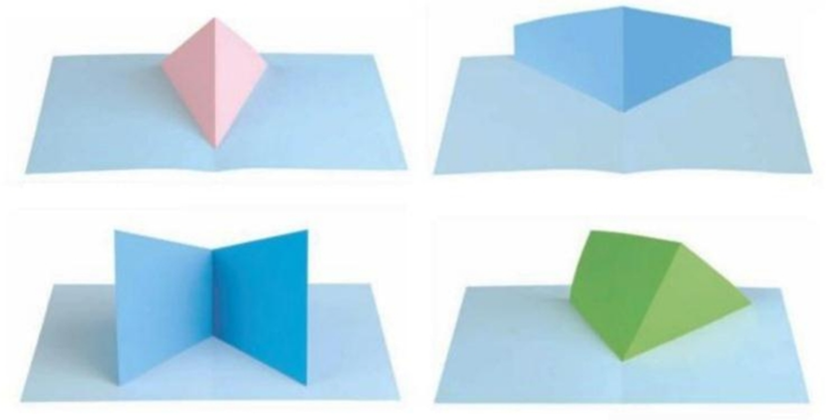


Figure 2.15: Variations of the v-fold.

5. Tent Folds. This mechanism is similar to the V-fold, but its fold never converges with the central fold (converges at infinity) or is parallel with the central fold. It also has similar properties with the step fold, except that it erects at 180°. An example of a tent fold is shown in Figure 2.16. The two patches maybe symmetric or asymmetric.

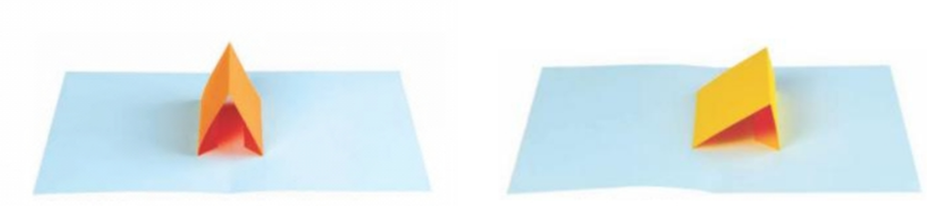


Figure 2.16: Tent Folds. Symmetric and Asymmetric Folds.

6. Parallel folds. Also known as floating layers or platforms, see Figure 2.17. It can be considered as two step folds put together, with a patch or tent fold in the center on top of the central fold. It erects at 180°. Hinged multi-tier paper supports lift pieces off the page, creating the illusion that it is floating over the surface.

7. Box folds. Box folds are more complicated structures, usually built on top of simpler mechanisms. For example the v-box fold, is a v-fold with additional patches to create a closed polyhedron, see Figure 2.18. A parallel box fold

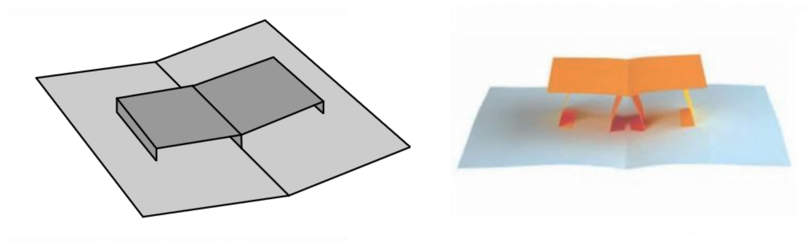


Figure 2.17: Parallel Fold.

on the other hand is built by closing off a parallel fold.

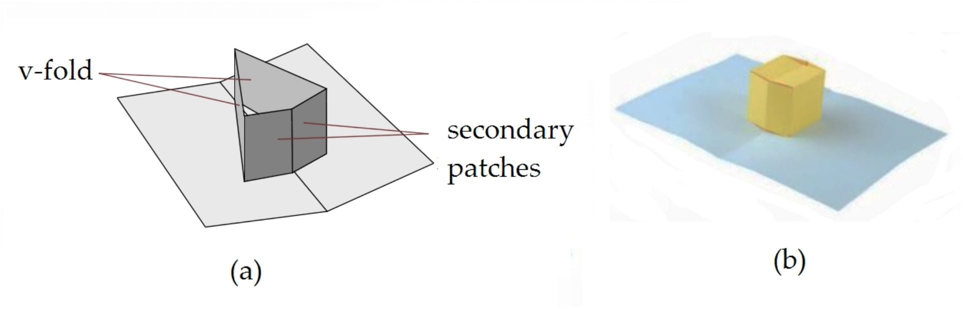


Figure 2.18: Box Fold. (a) v-box fold (b) parallel box fold.

8. Curved mechanisms. Curved shapes, also called boats, can also be created by warping the paper and gluing small flaps at certain points. Examples of curved pop-up shapes are shown in Figure 2.19. Note that most work on the geometric properties of pop-ups assume that the paper is rigid like metal and cannot be warped to simplify the formulations.

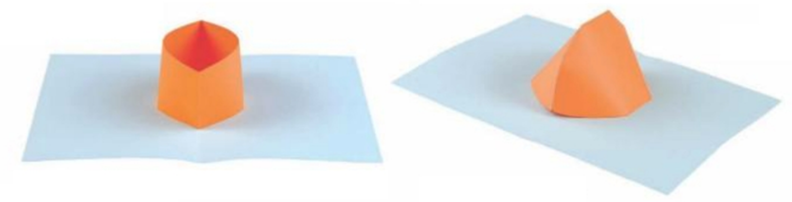


Figure 2.19: Curved shaped pop-ups.

9. Lattice-style Pop-ups. Also known as *sliceforms* or *trellises.* It uses two sets of parallel paper patches slotted together to make a foldable structure.

These mechanisms can stand on its own without a backing sheet, or it can also be attached to a backing sheet using strings or another mechanism like the v-fold. Examples of this mechanism are shown in Figure 2.20.

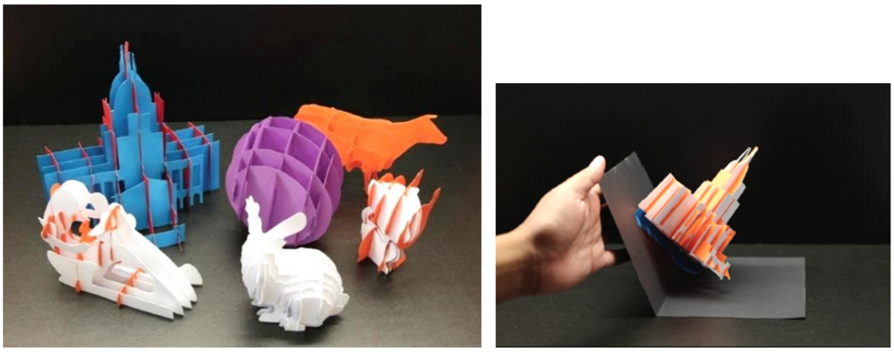


Figure 2.20: Examples of Sliceforms or Lattice-type pop-ups.

10. Moving Arm. The moving arm mechanism is a combination of the step-fold and an angled fold used to create a circular movement. An example of this mechanism is shown in Figure 2.21

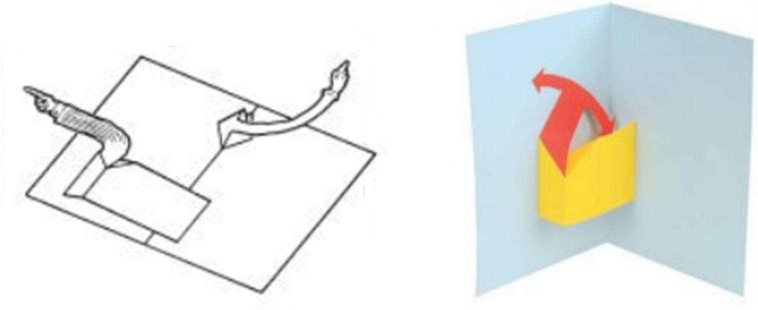


Figure 2.21: Moving Arm Mechanism.

11. Mechanical Devices. Other pop-up mechanisms require additional interaction from the reader, such as pulling a tab or turning a rotating disk. With the exception of [Gla02b], very few have tried to model geometric properties for these types of mechanisms in pop-ups.

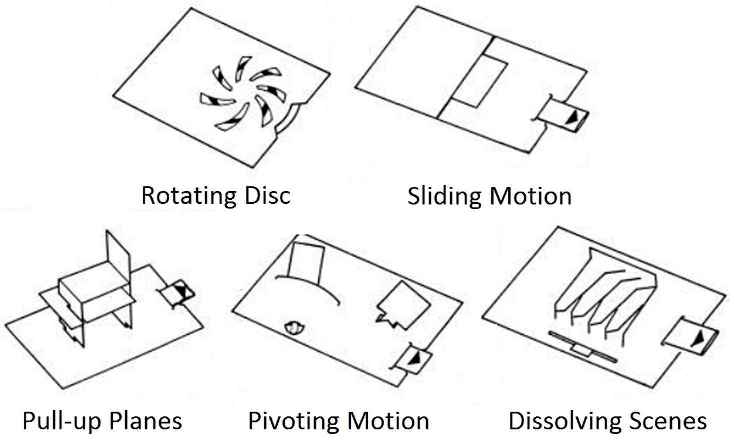


Figure 2.22: Other pop-up mechanisms requiring more user intervention.

2.4 Taxonomy of Pop-up Mechanisms

Pop-up mechanisms are actually a subset of the mechanisms used in movable books. [Hen08] presented a partial taxonomy of movable mechanisms, shown in Figure 2.23. Here we see that pop-ups are a specific type of device that does not require any other external force other than opening and closing the covers.

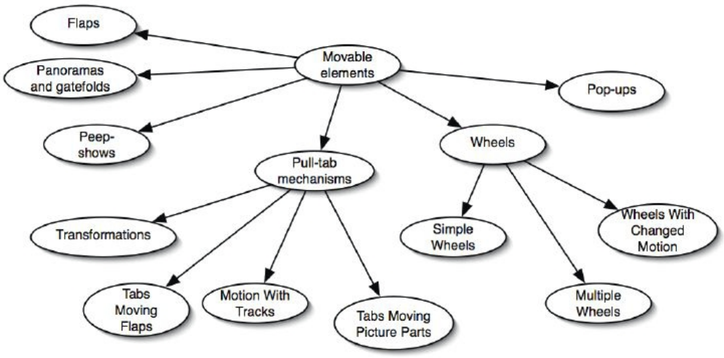
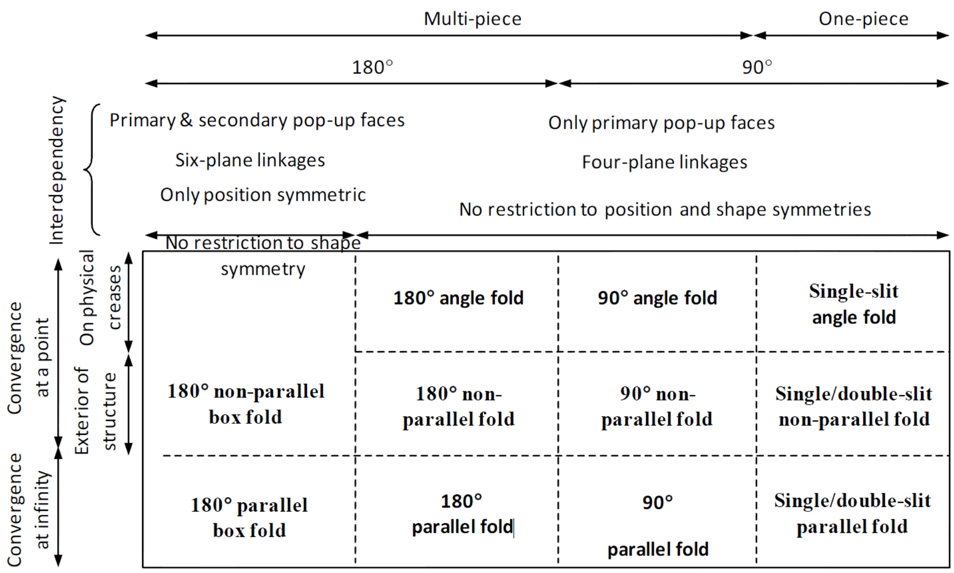


Figure 2.23: Partial Taxonomy of Movable Devices [Hen08].

For pop-up devices in particular, [Wen10] has come up with the most complete classification of pop-up mechanisms. His classification is based on six features of topological and geometrical characteristics of pop-up structures, see Figure 2.24. The topological features include the number of paper pieces needed for a structure, the type of pop-up faces required for construction and the basis of linkages in the structures. The geometrical attributes are the fold angle for a full erection of the structure, the symmetry about the gutter crease and the convergence of creases.



symmetry	180° angle fold	90° angle fold	Single-slit angle fold
180° non-parallel box fold	180° non-parallel fold	90° non-parallel fold	Single/double-slit non-parallel fold
180° parallel box fold	180° parallel fold	90° parallel fold	Single/double-slit parallel fold

Figure 2.24: Feature categorization of pop-up structures according to [Wen10].

First, let us discuss the topological features of pop-ups of [Wen10]. These are:

1. Number of Paper Pieces. Most pop-up instructions books classify mechanisms based on the number of pieces of paper used. A single-piece pop-up structure is created by creasing and cutting one piece of paper, and can fully erect at 90°. A multi-piece pop-up mechanism has two or more paper pieces and requires tabbing and gluing. Multi-piece pop-up can erect at both 90° or 180°. Figure 2.25 shows an example of the difference of the two categories.

2. Essential Patches. Here we group mechanisms based on the essential patches, the minimum set of patches for it to be a valid pop-up structure. [Wen10] defines two types of patches, the primary patches and the secondary patches. Primary patches are directly connected by a hinge to the base patches, secondary patches are connected to the primary patches. Tent fold for example uses only primary faces, while box folds need secondary patches to become a valid pop-up structure, see Figure 2.26.

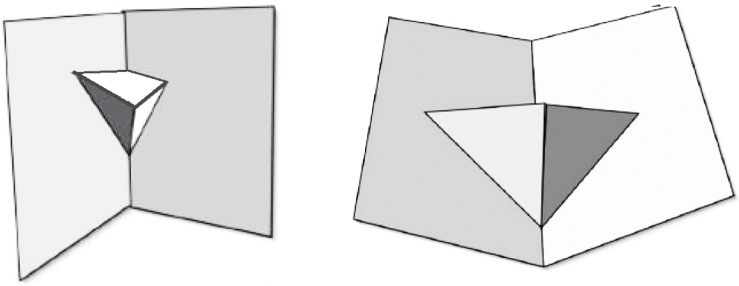


Figure 2.25: Single-piece (single-slit angle fold) and a multi-piece (v-fold) mechanisms.

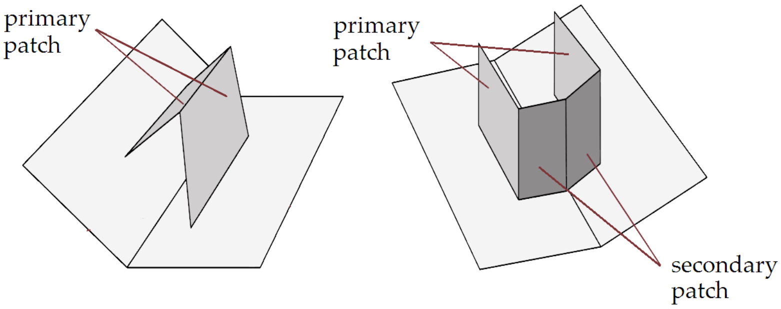


Figure 2.26: Mechanisms that use only primary patches (tent fold) and those that use secondary patches (box fold).

3. Basis of Linkages. Pop-up structures in [Wen10] are considered as bar linkages or plane linkages . The *basis of linkages* refers to the minimum number of linkages sufficient to make a valid pop-up structure. For example, the base patches forms a two-plane linkage and a tent-fold has the basis of a four-plane closed loop linkage. We can also use this property to classify mechanisms.

Next, we consider the geometric features, which are more interesting for our case since we do not have any constraints on the topology. These features are:

1. Folding Angle. One of the easiest ways to classify mechanisms is based on the fold angle when they are fully erected. For example, v-folds, box folds, parallel folds erect at 180° while step folds and slit mechanisms erect at 90°, see Figure 2.27.

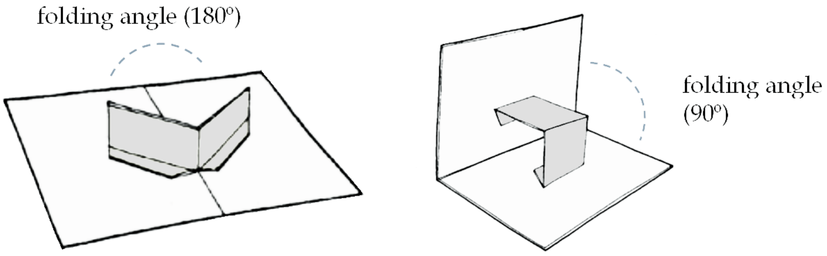


Figure 2.27: Mechanisms that erect at 180°(v-fold) and 90°(step fold).

2. Symmetry. Another way to classify mechanisms is by considering symmetry. We consider if the position of the patches is symmetric with respect to the central fold. For example, take the symmetric and asymmetric tent folds in Figure 2.28.

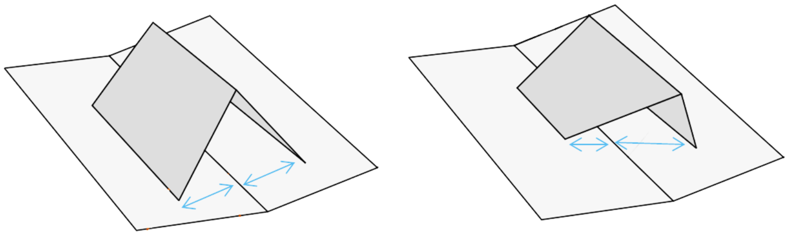


Figure 2.28: Symmetric and Asymmetric Tent Folds.

3. Convergence. Mechanisms have a primary hinge that is either parallel or non-parallel to the central fold. Take the example of tent folds and v-folds, see Figure 2.29. Convergence of the primary fold and the central fold could either be on the base patches, outside the base patches, or at infinity (parallel).

A partial taxonomy of pop-up mechanisms we consider in this thesis is shown in Figure 2.30.

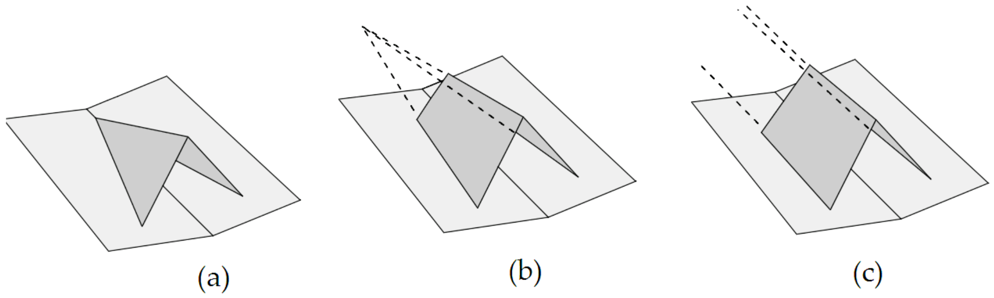


Figure 2.29: (a) Convergence inside the base patches along the central fold (v-fold), (b) outside the base patches (non-parallel 180° fold) or at infinity (tent fold).

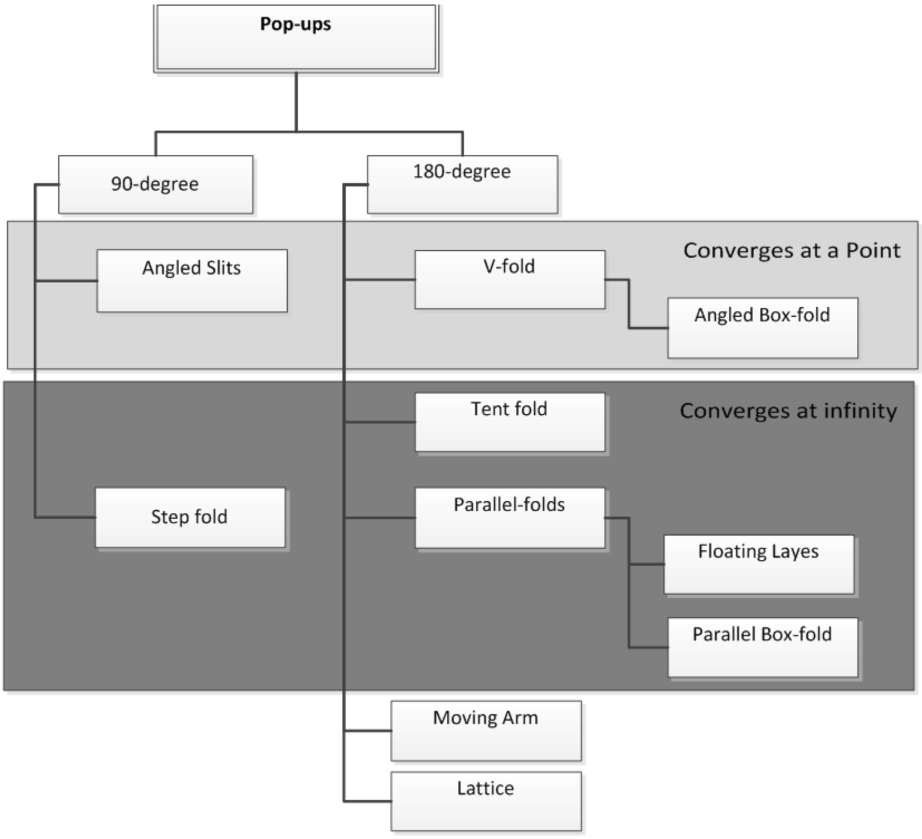


Figure 2.30: Partial Classification of a Paper Pop-up Mechanisms.

Chapter 3

Survey

This section details the current work done in the field of computation pop-up design, but given the scarcity of the work directly related to paper pop-ups we also explore other work in other forms of papercrafts. Related topics such as mesh simplification or abstraction into a set of 2D planes and the kinematic synthesis of mechanical toys are also discussed.

3.1 Papercrafts

Since the invention of paper itself, papercrafting has fascinated people of different cultures and ages. Consequently, researchers have studied numerous forms of papercraft in the mathematical and computational setting most especially origami. [ZZ13] presented a toolkit for designing automated movable papercraft.

Paper strip modeling aims to represent 3D models with paper strips, or piece-wise developable surfaces. It can be viewed as a form of Kirigami, the Japanese art of paper-cutting that may also involves folding and gluing. [MS04b] proposed a method by using mesh simplification to approximate general surfaces by paper-strip that is then used to create paper-craft toys from 3D meshes (see Figure 3.1). This type of mesh simplification is also studied by other works [GH97, COM98, WL10]. Alternative methods also have been proposed in [STL06] and [Massarwi et al. 2007]. Strip modeling can achieve more complicated geometry than paper pop-ups, but the final result is not foldable.

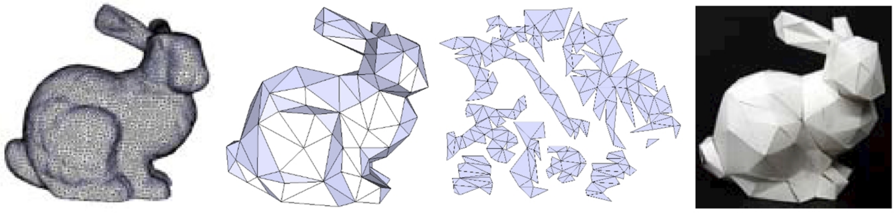


Figure 3.1: The paper strip modeling results of [MS04b].

Paper-cutting, originally a Chinese folk art, cuts out stylistic patterns and figures from a piece of paper. A simple and efficient algorithm for automatic paper-cutting given input images was proposed in [XKM07], see Figure 3.2. [LYMS07] considered extensions to 3D paper-cuts and interactive design of animations with paper-cuts.

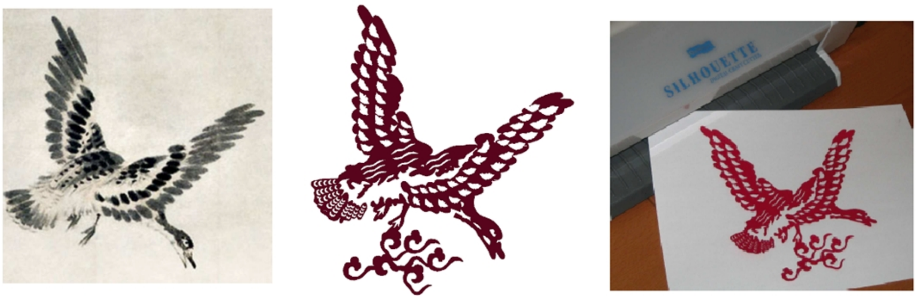


Figure 3.2: Paper cutting results of [XKM07].

Paper sculptures have very similar characteristics with paper pop-ups and even use some similar techniques. [Jac96] defines a paper sculpture as multiple cut-out pieces of paper, which are then rolled, creased, bent or otherwise mangled, and then glued together to form a picture, see Figure 3.3. Again, the main difference with pop-ups is that paper sculptures are not foldable.

Unlike paper sculptures that use multiple layers of paper pieces viewed in at a specific angle, 3D paper models are proportionally identical representations of real objects and can be viewed from any viewpoint. It usually entails cutting the outer body of the model and gluing them together. The Canon - Creative Park

Figure 3.3: Paper sculptures created by [Che05].

website [Can12] provides numerous layout designs that users can print and create on their own, see Figure 3.4. Touch-3D [Des] is a CAD software that can design such 3D objects that can be unfolded into 2D planar patches. This is problem is similar to unfolding 3D models or polyhedra.

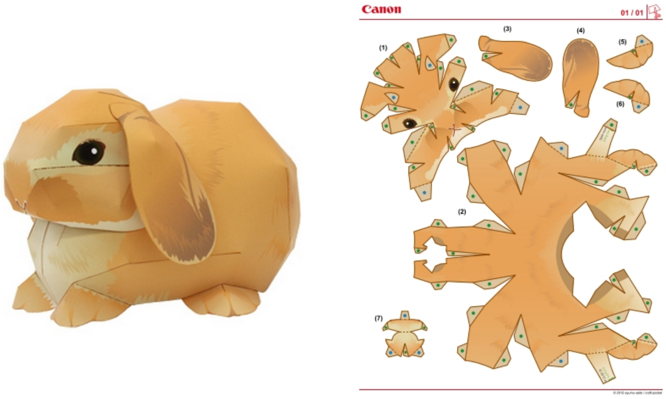


Figure 3.4: Bunny paper 3D model and layout design by [Can12].

Origami, the traditional Japanese art of paper folding, is one of the most popular types of papercraft. It has been extensively studied in literature [Hul06, DO07, O'R11], particularly folding algorithms and conditions for foldability. Recently, [Tac10] proposed an algorithm to automatically generate origami design for arbitrary polyhedral surfaces, see Figure 3.5. Curved folding has also been considered [Kilian et al. 2008] based on analysis of developable surfaces.

The central problem in origami is folding and foldability, which is also the main

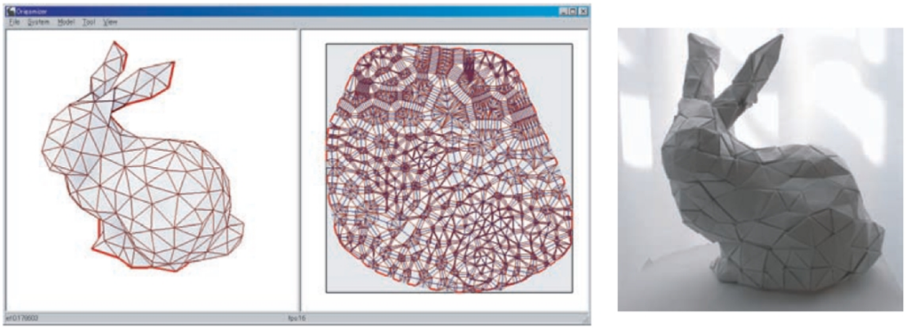


Figure 3.5: Origami results for Stanford bunny [Tac10].

concern of paper pop-ups. Pop-ups however can be made by cutting and gluing multiple pieces together, hence possessing a richer geometric structure. On the other hand, the closing process of a pop-up is more restricted than origami, as it requires all pieces to flatten simultaneously by only moving the two pages of the backing sheet. As a result, the formulations related to these two art forms are likely to differ significantly.

3.2 Mesh Simplification and Abstraction

Automatically generating paper pop-up designs from 3D models can also be viewed as a form of model simplification or abstraction of a 3D mesh into a set of 2D planes. Many existing works already provide techniques to simplify a model by approximating its surface with simpler representations. However, we focus our attention to those that simplify to a set of planes or cross sections rather than those that reduce the number of vertices or edges like mesh refinement or decimation. Such as the work of [EPD09] that developed a view-dependent method for converting 3D models into 2D layers.

Billboards and image impostors have long been used in 3D games or virtual walkthroughs to simplify scenes especially for far away objects. An impostor is a billboard that always faces the camera with an applied texture that visually

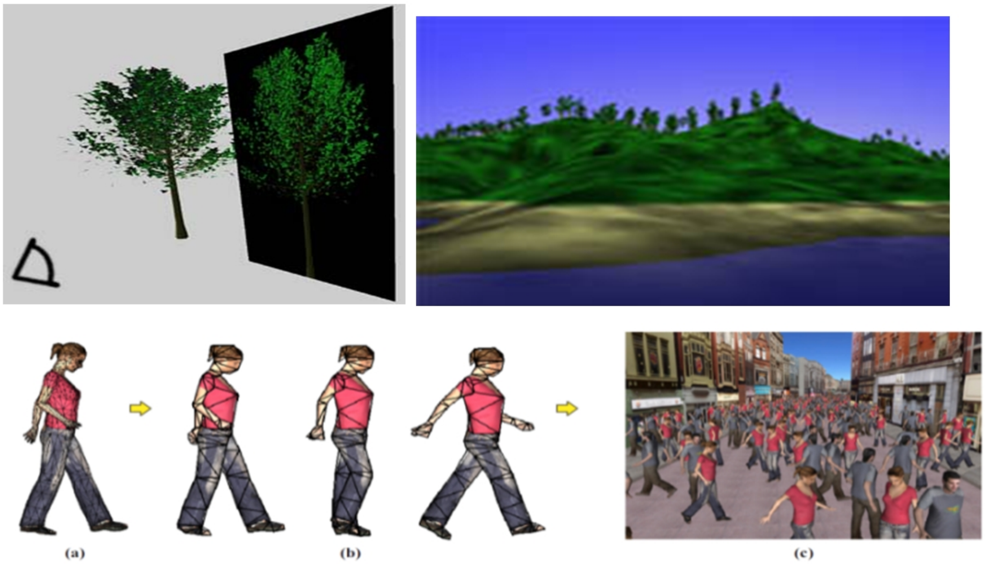


Figure 3.6: Billboards used in 3D Scenes [KGBS11].

represents the geometric object it replaces. Meshed imposter [SDB97, DSSD99] is another type of image-based simplification technique that uses a single image but maps it onto a depth mesh that roughly approximates one side of the object. Recently, [KGBS11] proposed another alternative method that significantly improves upon pre-computed impostors by automatically generating 2D polygonal characters or *polypostors* (Figure 3.6).

Another variation of the billboard is that of [DDSD03] that proposed an extreme form of simplification using billboard clouds. A 3D model is simplified into a set of planes with texture and transparency maps. Their optimization approach builds a billboard cloud given a geometric error threshold. After computing an appropriate density function in plane space, a greedy approach is used to select suitable representative planes. An example is shown in Figure 3.7. For our case however, this approach cannot be readily applied as its configuration of billboard planes does not guarantee that it can be converted to a valid pop-up.

Motivated by their popularity in art and engineering, [MSM11] used planar sections such as the contours of intersection of planes with a 3D object, for

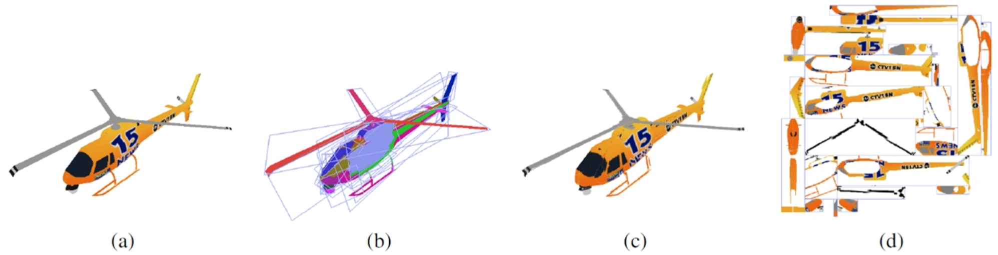


Figure 3.7: Billboard cloud results of [DDSD03]: (a) input model (b) one-color per billboard (c) output model (d) billboards side by side.

creating shape abstractions. They also conducted user studies to show that humans do define consistent and similar planar section proxies for common objects. Guided by the principles inferred from their user study, their algorithm progressively selects planes to maximize feature coverage, which in turn influence the selection of subsequent planes. An example of a shape proxy is shown in Figure 3.8. However, the structures produced again do not necessary conform to the strict properties of a valid paper pop-up.

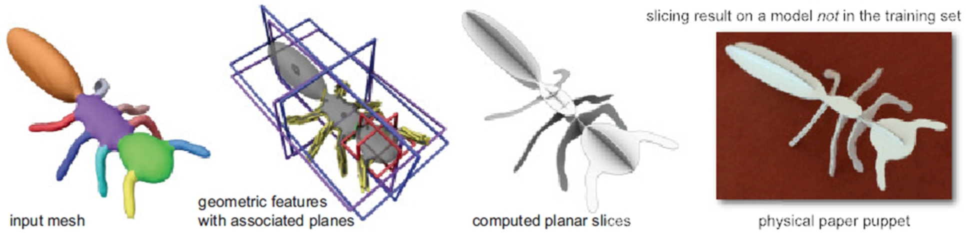


Figure 3.8: Shape proxy results of [MSM11].

Mesh simplification is a well-studied area in computer graphics and numerous researches have been conducted in this area. The works we have presented here only reflect a very small percentage of the published work in the field. The survey of [GH97] provides more information about some of the notable simplification algorithms like vertex clustering, incremental decimation and resampling. We have only focused on the approaches that could be potentially adapted for automated pop-up design.

3.3 Mechanical Toy Modelling

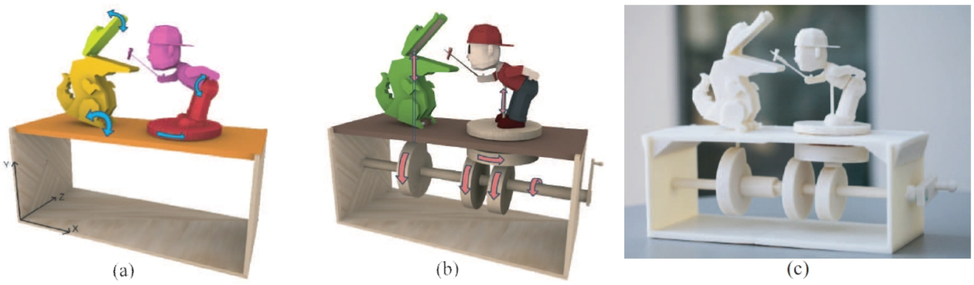


Figure 3.9: Results of [ZXS$^+$12]. (a) Input (b) Mechanical assembly synthesized by the system (c) Fabricated result.

Our approach for reproducing the motion of 3D articulated figures using animated multi-style pop-ups is inspired by research done in mechanical assemblies and kinematic synthesis. [ZXS$^+$12, CTN$^+$13, CLM$^+$13, TCG$^+$14] proposed methods for creating animated mechanical characters or toys. Using user-specified motion, their methods automatically generate the mechanisms need to reproduce the input. Figure 3.9 shows the results of [ZXS$^+$12].

[ZXS$^+$12] automatically generates a mechanism assembly located in a box below the feature base that produces the specified motion. Parts in the assembly are selected from a parameterized set including belt-pulleys, gears, crank-sliders, quick-returns, and various cams. The locations and parameters for these parts are optimized to generate the specified motion. Similarly, paper engineers combine pop-up mechanisms to create animations in pop-up books.

[CLM$^+$13] proposed an automated algorithm that takes a motion sequence of a humanoid character and generates the design for a mechanical figure that approximates the input motion. First, they compute a motion that approximates the input sequence as closely as possible at the same time being compatible with the geometric and motion constraints of the mechanical parts in the design. Then, they solve for the sizing parameters, interconnections, and spatial layout of all the elements, while considering the fabrication and assembly constraints.

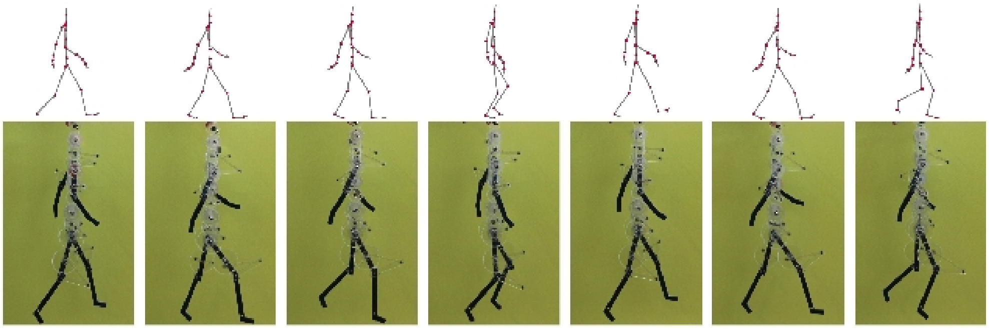


Figure 3.10: Results of [CLM$^+$13]. Input motion sequence (top) and approximated mechanical automaton (bottom).

3.4 Computation Pop-ups

For work on paper pop-ups in particular, Glassner [Gla02a] has described the use of simple geometry to create various pop-up features such as the single-slit and v-fold. Both have hinges that converge to a particular point. His technique involves finding the intersecting point of the three spheres, which locates a moving vertex on the pop-up. Figure 3.11 illustrates the geometry of the single-slit mechanism. He also described some mechanical mechanisms like pull-tabs, spinning wheels, etc. [Gla02b]. The author states that he has a complete design system, but no details are given, other than that elements are added by drag-and-drop.

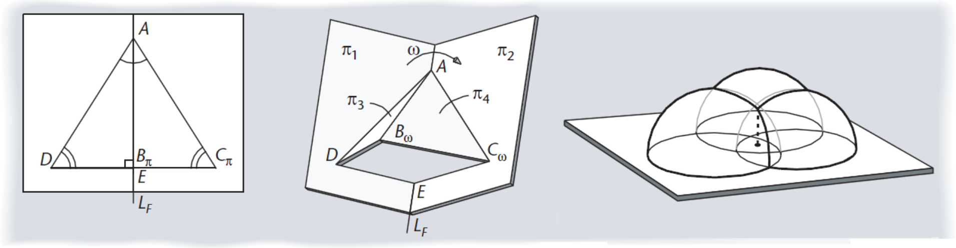


Figure 3.11: Single-slit geometry by [Gla02a].

[HE06] designed an application, the Popup Workshop, whose purpose is to introduce children to the craft and engineering discipline of paper pop-up design. The Popup Workshop uses several pop-up mechanisms and automatically enforces the geometric constraints necessary to keep the elements foldable. It also simulates the pop-up by using a constraint system to allow animation of the 3D representation of the pop-up. Figure 3.12 shows a screenshot of their system.

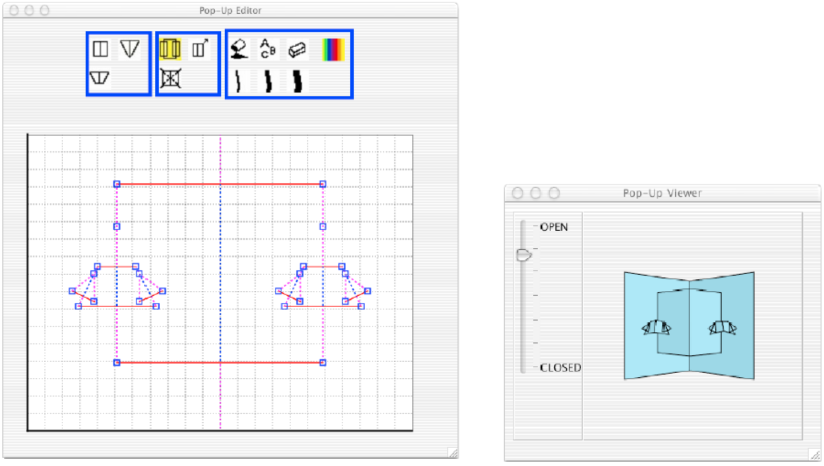


Figure 3.12: Pop-up Workshop by [HE06].

[IEM$^+$11] also presented an interactive system the supports v-folds and parallel folds. Their system simulates folding and opening of the pop-up card using a mass-spring model. The simulation detects collisions and protrusions, and animates the movement of the pop-up card. Figure 3.13 shows a screenshot of their system and sample results.

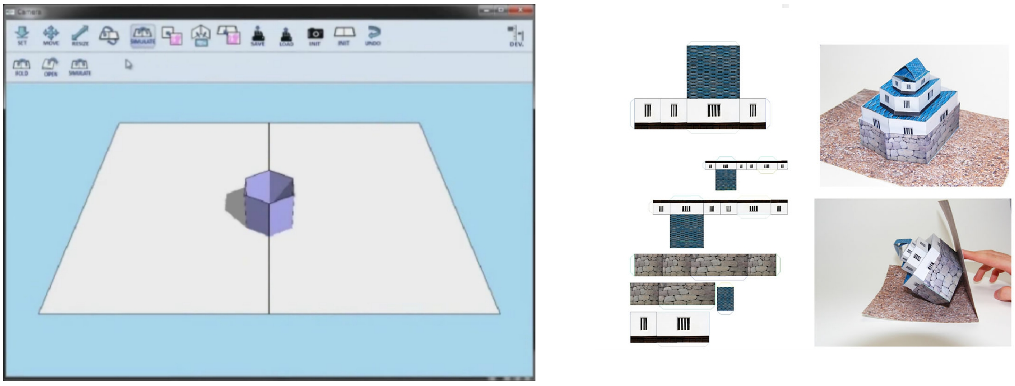


Figure 3.13: Interactive System and pop-ups generated by [IEM$^+$11].

[HEH05, HEH08] developed an application to create very simple pop-ups from photographs. Their algorithm labels regions of the input image into coarse categories: *ground*, *sky*, and *vertical*. These labels are then used to cut and fold the image into a pop-up model using a set of simple mechanisms (Figure 3.14). Similarly, [LYLC14a] proposed an image-based paper pop-up design.

[LTS96] developed a model for simulating the opening and closing of parallel v-folds. The authors used the trigonometric equations to model the geometric properties of this class of pop-ups. [ADD$^+$13] proposed a polynomial-time algorithm that creates pops-ups by subdividing the polygon into a single-degree-of-freedom linkage structure, such that closing the pop-up flattens the linkage without collision.

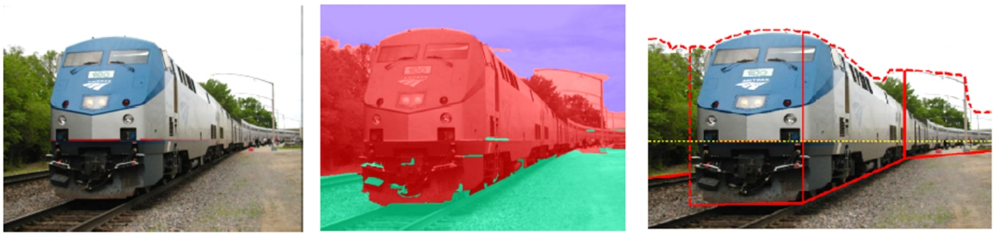


Figure 3.14: Sample pop-up from 2D image [HEH05].

A number of works focused on *Origamic Architecture* (OA) in particular. The pioneering work in OA came from Mitani and Suzuki, who created applications that allow users to design and construct OA models [MSU03, MS04a] . Their applications allowed the user specify and position each of the horizontal and vertical faces of the OA pop-up. The application also checks for validity and shows any invalid faces. Their simple algorithm caters for cases like pull-offs, but invalid features like dangling pieces are not detected. This work led to Tama Software's Pop-Up Card Designer [Tam07] (see Figure 3.15). Another attempt at assisted OA design was by [CZ06]. Their approach is similar to the previous method, but differ in validity checking. The approach automatically creates horizontal faces and the user only needs to input vertical faces. All the applications discussed for designing OA require heavy user interaction to create an OA, and the user must have an idea of how to make each part of the object into valid OA faces. Even simple shapes can be very difficult for the user to design. [Leo10] used depth maps from a 45-degree projection to create segmentation maps that are used to determine the patches of an OA. [LYLC14b] generated OAs from 2D images.

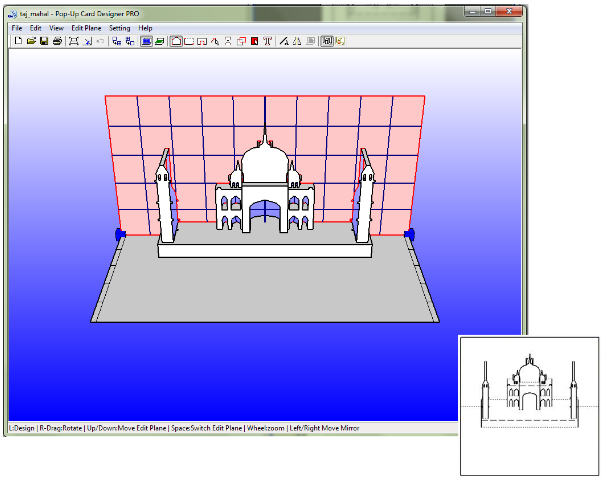
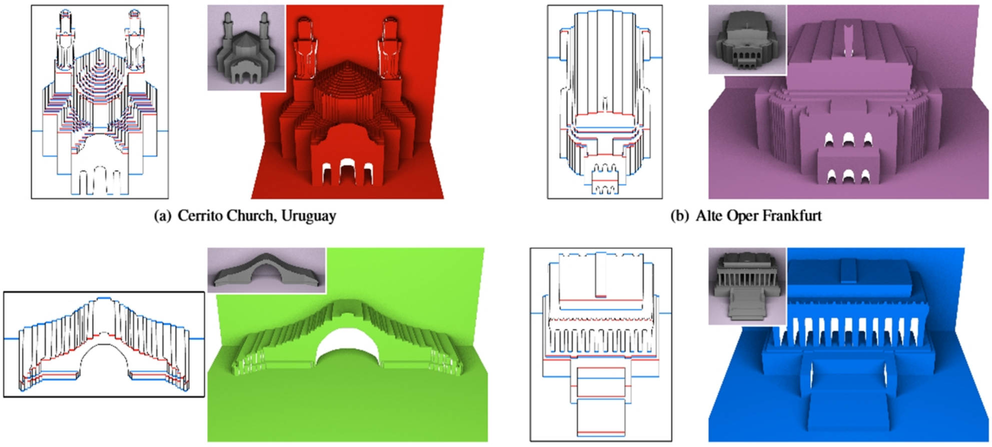

taj_mahal - Pop-Up Card Designer PRO
File Edit View Edit Plane Setting Help
L:Design | R-Drag:Rotate | Up/Down:Move Edit Plane | Space:Switch Edit Plane | Wheel:zoom | Left/Right Move Mirror

Figure 3.15: Tama Software's Pop-up Card Designer [Tam07].

The most notable recent work on paper pop-ups have been from Tsinghua University. The work of [LSH$^+$10] provided a theoretical foundation and an algorithm for their system that automatically converts 3D models into origamic architectures. The algorithm is grounded on geometric formulation of layout for paper architectures that can be popped-up in a stable manner, with sufficient conditions for a 3D surface to be popped up from such a planar layout (Figure 3.16). Based on these conditions, their algorithm generates paper architectures that have two sets of parallel patches, which approximate the input geometry while guaranteeing that it can be physically created. Figure 18 shows some examples of their result.

Subsequently, in [LJGH11], they extended the notions of validity to a more general class of v-style popups. They gave sufficient conditions for a v-style paper structure to be a valid pop-up. These conditions are: it can be closed flat while maintaining the rigidity of the patches; the closing and opening do not need extra force besides holding backing sheets; it does not contain intersections; and the closed paper is enclosed within the backing sheet's borders. These conditions lead them to identify other mechanisms for making pop-ups. Based on the theory and derived mechanisms, they also developed an interactive tool for designing v-style



(a) Cerrito Church, Uruguay

(b) Alte Oper Frankfurt

Figure 3.16: OA pop-ups generated by the system of [LSH+10].

pop-ups (Figure 3.17) and an automated construction algorithm from an input 3D model, both of which guarantee that the result is a valid pop-up. Figure 3.18 shows the results of their interactive tool and automated construction.

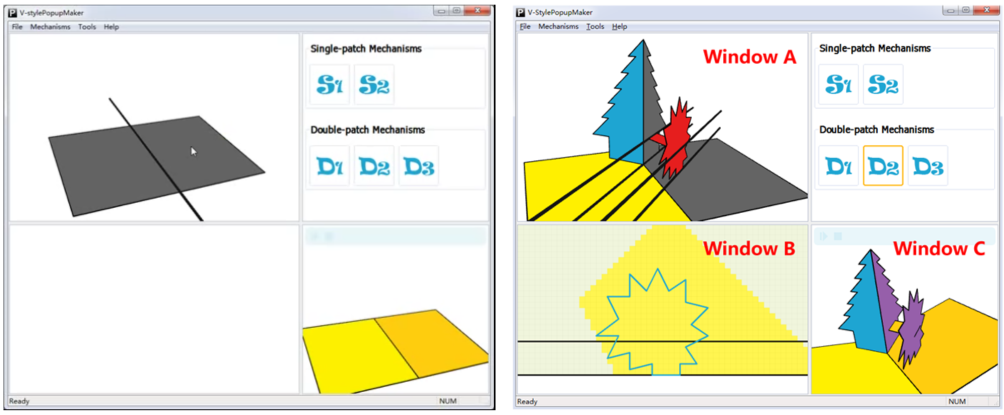


Figure 3.17: V-style Pop-up Maker Tool by [LJGH11].

Table 1 shows a summary of the computational pop-up literature we have surveyed. Note that most of the systems before [LSH+10] are limited to a few known mechanisms and do not offer validity guarantees on the designs. They are also mostly CAD software and check validity by simulating the opening and closing of the pop-up. There are only a few studies on the geometric properties and constraints of a general paper pop-up, especially those that are made up of

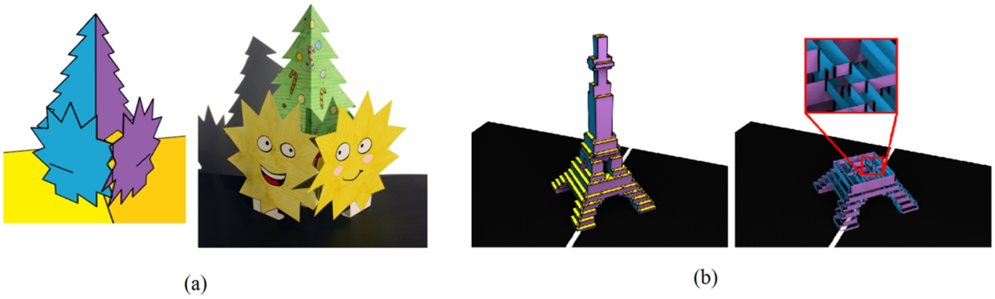
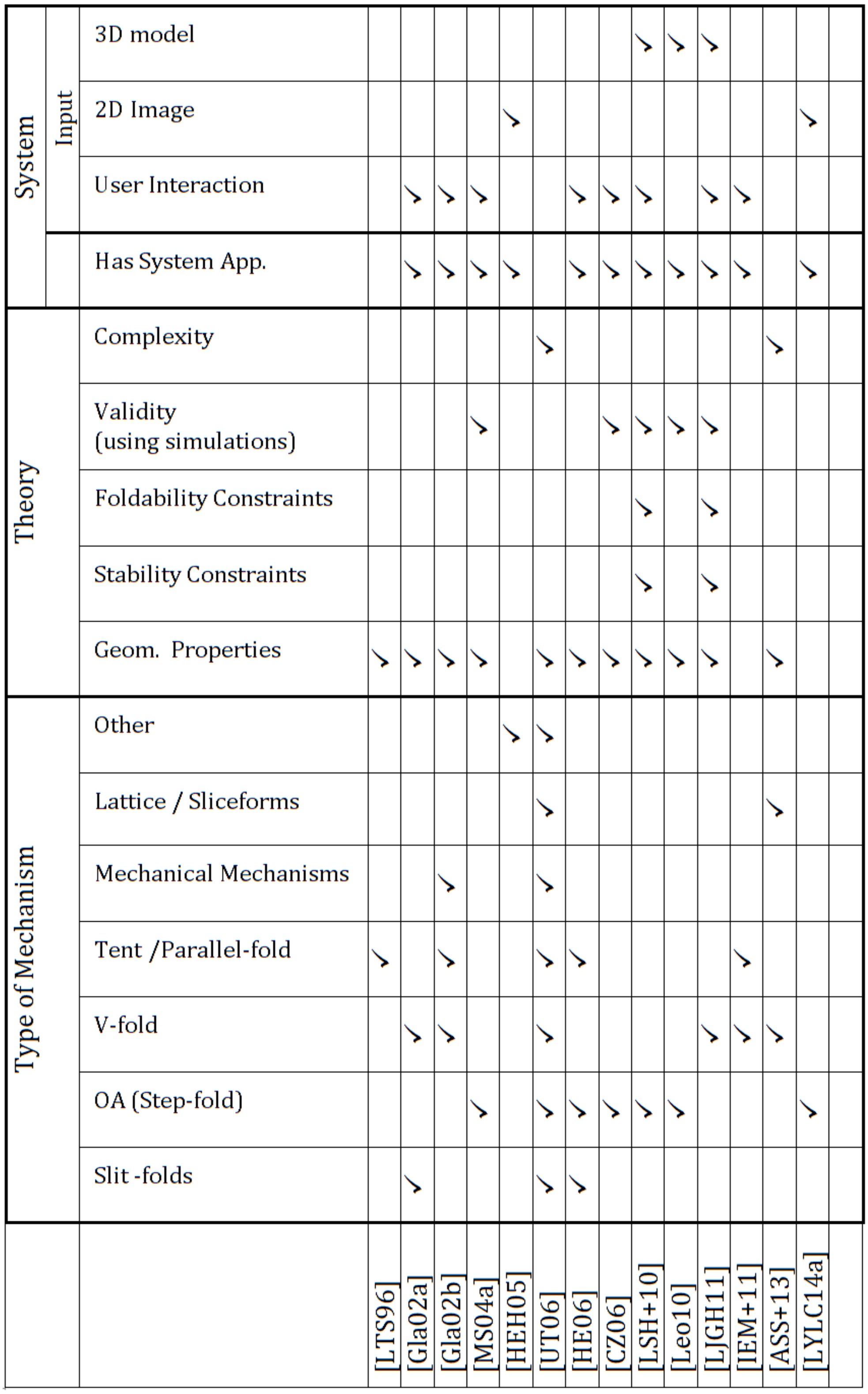


Figure 3.18: Results of [LJGH11]: (a) using the interactive tool, (b) automated construction.

multiple pop-up styles. This is primarily because determining the foldability of a general pop-up is NP hard, as it has been shown by the work of [UT06]. Notice that no other work has has considered motion in their designs.



System	Input	3D model									✓	✓	✓				
		2D Image					✓									✓	
		User Interaction		✓	✓	✓			✓	✓	✓		✓	✓			
		Has System App.		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	
Theory		Complexity						✓							✓		
		Validity (using simulations)				✓				✓	✓	✓	✓				
		Foldability Constraints									✓		✓				
		Stability Constraints									✓		✓				
		Geom. Properties	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓		
Type of Mechanism		Other					✓	✓									
		Lattice / Sliceforms						✓							✓		
		Mechanical Mechanisms			✓			✓									
		Tent /Parallel-fold	✓		✓			✓	✓					✓			
		V-fold		✓	✓			✓					✓	✓	✓		
		OA (Step-fold)				✓		✓	✓	✓	✓	✓				✓	
		Slit -folds		✓				✓	✓								
			[LTS96]	[Gla02a]	[Gla02b]	[MS04a]	[HEH05]	[UT06]	[HE06]	[CZ06]	[LSH+10]	[Leo10]	[LJGH11]	[IEM+11]	[ASS+13]	[LYLC14a]	

Table 3.1: Summary of work on computation pop-up design.

Chapter 4

Geometric Study

In this chapter, we formally define a pop-up structure and a pop-up mechanism, and define the geometric conditions to ensure its validity. These formulations serve as the foundation of our algorithm for approximating shape. We also present a study of the motion of pop-up mechanisms in relation to the lengths, orientations and locations of its patches. These parameterized mechanisms are used in our algorithm for approximating motion.

First, we formally define a *valid* pop-up. We use the formulations of [BH02, LSH$^+$10, LJGH11]. We introduce geometric formulations of a paper structure, called a *scaffold*, and the properties of this structure that would make it a valid pop-up. For simplicity we make the assumption that the paper has zero thickness, zero weight, and is rigid, and therefore cannot be warped. Although in practice many pop-ups can be bent, this is the current assumption of most the work that describes the geometric properties of a pop-up.

Formally, a *scaffold* is a collection of patches or planar polygons that are connected at straight line segments or *hinges*. The *hinges* may lie either on a patch or on its border. A *scaffold* always contains two patches, known as the *base patches*, which are two identical rectangles connected on an edge that is called a *central fold*. The angle between these two patches is called the *fold angle*.

We also follow the same formal definitions of validity as used in [LSH$^+$10, LJGH11]. A valid pop-up should have the following properties:

1. Foldable. The pop-up can be closed down to a flat state and can be re-opened without tearing the paper or creating new creases other than those in the design. Note that during the folding process, the rigidity and connectivity of the patches need to be maintained at all times and it should not introduce new fold lines.

2. Stable. The closing and opening of the pop-up do not need extra forces other than holding and turning the two backing sheets. In other words, it is stable if all its patches are stationary when the ground and backdrop patches are held still at any fold angle.

3. Intersection-free. The paper pieces or patches do not intersect during closing or opening.

4. Enclosing. When closed, all pieces of the pop-up are enclosed within the base patches.

The designer usually develops a pop-up design in the opened state. Thus, we would like to determine if a scaffold representing the open-state pop-up can be closed while maintaining the four properties mentioned. [LJGH11] formulates these properties as conditions of a transformation called the fold transform on a scaffold. These transforms are:

Definition 1 . A fold transform $f(S, t)$ on a scaffold S is a continuous deformation of S, where the deformation is identity when $t = 0$ and is a combination of translations and rotations on each patch of S for any $t \in [0, 1]$. A fold transform necessarily maintains the rigidity and hinge connectivity of the patches.

Definition 2. A flattening transform $f(S, t)$ on a scaffold S is a fold transform with two additional properties:

1. The fold angle decreases monotonically to 0 as t increases from 0 to 1.

2. All patches in $f(S, 1)$ are co-planar.

The properties of a valid pop-up are formally defined as follows:

Definition 3. A fold transform $f(S, t)$ on a scaffold S is said to be *stable*, if for any $t \in (0, 1)$, there does not exist any non-identity fold transform on $f(S, t)$ that keeps the backing sheets in $f(S, t)$ still.

Definition 4. A fold transform $f(S, t)$ on a scaffold S is said to be *intersection-free*, if for any two topologically distinct points p, q on S, their deformed locations on $f(S, t)$ are spatially distinct for any $t \in (0, 1)$.

Definition 5. A flattening transform $f(S, t)$ on a scaffold S is said to be *enclosing*, if all patches in $f(S, 1)$ lie interior to the backing sheets.

If a *scaffold S* has a flattening transform that is stable, intersection-free and enclosing then it can be closed down as well as opened up. We call such scaffold a valid pop-up.

We formally define a *mechanism* as the most basic geometric structure that, together with other structures of the same type, gives a pop-up a unique style. Pop-up artists use several types of mechanisms to make their works come to life. In this work, we focus on four mechanisms: *step-fold*, *tent-fold*, *v-fold*, and *box-fold*. These mechanisms were selected because of their ability to represent volumetric objects. Among these mechanisms, step-fold structure pops up at 90° fold angle, while the others pop up at 180°.

4.1 Pop-up Mechanisms

In this section, we formally define the pop-up mechanisms used in our automated design. The craft of paper pop-ups has not been extensively studied in the mathematical setting. Although some pop-up manuals explain rudimentary mechanisms to produce basic motion, the relationship between the geometry of the paper pieces and the output motion is not thoroughly explored. Some of the mechanisms described in this thesis are based on our own observations and study of pop-up books by renowned artists.

Step-fold

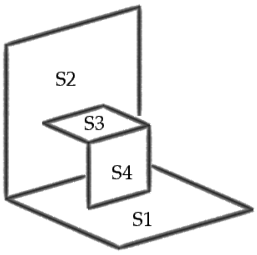


Figure 4.1: Step-fold mechanism and its patches.

Step-fold, which is referred to as mechanism D2 in [LJGH11], is a scaffold comprised of 4 patches, S_1, S_2, S_3, and S_4, folded together such that S_1 is parallel to S_3 and S_2 is parallel to S_4 (Figure 4.1). Patches S_3 and S_4 are bounded by their cut lines and a common hinge.

Tent-fold

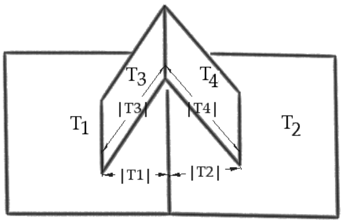


Figure 4.2: Tent-fold mechanism and its patches.

Tent-fold, as described in [HS09], is more general than step-fold in that it does not require its patches T_1, T_2, T_3 and T_4 to form parallel pairs, although the fold lines need to be parallel. We construct tent-fold so that $|T_1| = |T_2|, |T_3| = |T_4|$ and $|T_3| > |T_1|$, where $|T_i|$ is the distance between two hinges on T_i (Figure 4.2). Note that tent-folds can be asymmetric. However, for our initial work in combining multiple pop-up styles together we first consider the symmetric case as asymmetric structures may lead to more complicated intersection detection. The asymmetric case can considered in future work.

V-fold

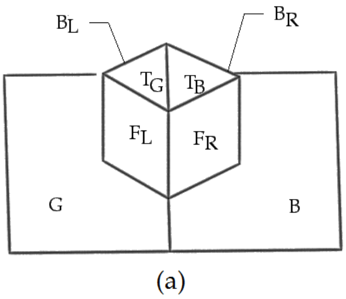


Figure 4.3: V-fold mechanism and its patches. (a) Type-1 and (b) Type-2.

Our *v-fold* is also known as mechanism D1 in [LJGH11]. However, it may comprise more patches to depict more shapes. In this work, we use two types of v-fold.

In the first type (v-box fold), when the primary patches are opened at 180°, the v-fold structure forms a complete box as shown in Figure 4.3a. More specifically, our *v-fold* structure consists of 8 patches, G, B, F_L, F_R, B_L, B_R, T_G, T_B, in which we glue $\{B, F_R\}$, $\{G, F_L\}$, $\{F_L, T_G\}$, $\{F_R, T_B\}$, and $\{F_L, F_R, B_R, B_L\}$. When B and G are opened at 180°, T_G and T_B are parallel to B and G, while F_L, F_R, B_L and B_R are all perpendicular to B and G, and form 45° with the central fold. In addition, the hinge between T_G and T_B lies in the bisecting plane of B and G. Note that, in order for the top patches to fold up, we do not glue $\{T_B, B_R\}$ or $\{T_G, B_L\}$.

The second type of v-fold (Figure 4.3b) is similar to the first type, except that it forms only a triangular half of a box when B and G are fully opened at 180°, and it does not have the back patches B_L and B_R.

Box-fold

Unlike the other pop-up mechanisms, *parallel box-fold* has yet to be formulated in detail in previous work. The parallel box-fold is the most complex of the four mechanisms considered. We choose to include it because of its ability to capture rectangular objects aligned with the central fold line and its common usage in many artworks, such as [Cro11].

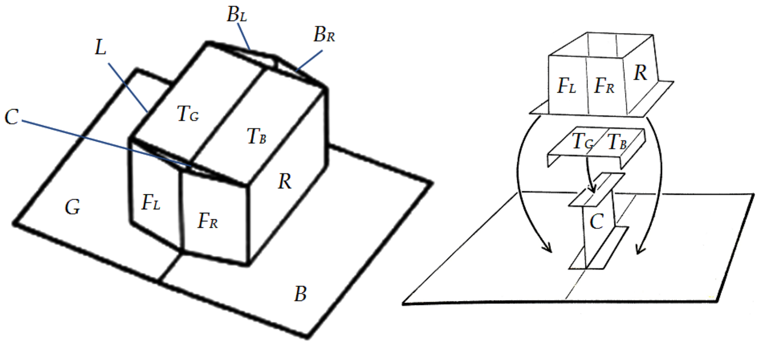


Figure 4.4: The box-fold mechanism.

Consider a pop-up opened at 180°. A box-fold structure comprises 11 patches labeled as G, B, L, R, F_L, F_R, B_L, B_R, T_G, T_B and C (see Figure 4.4). Patches G and B are parallel to the ground and backdrop. Patches L and R are the left and right sides of the box structure, forming equal angles with G and B, and are equidistant to the central fold. Patch C is a special backbone glued perpendicularly to G and B at their common fold line. On top of the box, patches T_G and T_B are connected to L and R, respectively, and are glued to C at their common fold line. In our work, all the fold lines connecting B, R, T_B, T_G, L and G are made parallel to the central fold.

On the front side of the box, patches F_L and F_R share a fold and are connected to L and R, respectively. Similarly, B_L and B_R are equivalent patches on the back of the structure. The folds between F_L, F_R and between B_L, B_R must be coplanar with patch C. Note that the front and back patches are not glued to C. In principle, only one of the two pairs, either (F_L, F_R) or (B_L, B_R), is needed. However, in practice, box-style pop-ups normally contain both sides for better symmetry and sturdiness. In addition, each side is allowed to have multiple pairs of (F_L, F_R) or (B_L, B_R), as long as their fold lines do not intersect.

4.2 Pop-up Validity

4.2.1 Validity of Individual Pop-up Mechanism

The validity of the step-fold, tent-fold and v-fold pop-ups have been studied in earlier work [LJGH11, HS09, LTS96, Wen10], here we present an aggregation of their findings. Given a *step fold* with patches S_3 and S_4, connected to the base patches, S_1 and S_2, with a fold angle θ.

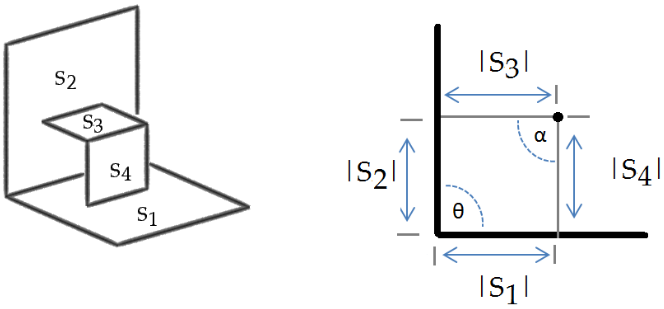


Figure 4.5: Step-fold mechanism.

It is foldable when the sum of the cross-sectional distances, perpendicular to the central fold, between the central fold and the fold on the primary pop-up mechanism should be equal on both sides. That is,

$$|S_1| + |S_4| = |S_2| + |S_3|.$$

Similarly, a *tent fold* with patches T_3 and T_4, connected to the base patches, T_1 and T_2, with a fold angle θ.

Is foldable when,

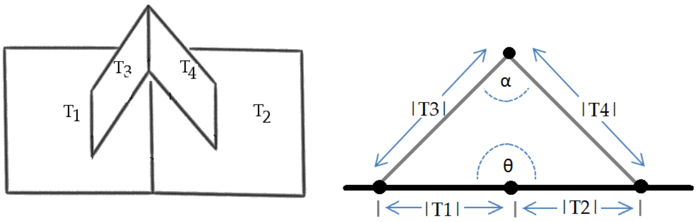


Figure 4.6: Tent-fold mechanism.

$$|T_1| + |T_3| = |T_2| + |T_4|.$$

Note the difference between the two mechanisms is that the *step fold* fully erects at 90° while the *tent fold* fully erects at 180°. We can also say that for the tent fold,

$$|T_1| < |T_3|, |T_2| < |T_4|, \theta > \alpha,$$

and

$$\frac{|T1||T2|}{|T3||T4|} < 1.$$

If the step-fold is completely flat at 180°. Then,

$$|S_1| = |S_3|, |S_2| = |S_4|, \theta = \alpha,$$

and

$$\frac{|S_1||S_2|}{|S_3||S_4|} = \frac{(\cos\alpha + 1)}{(\cos\theta + 1)} = 1.$$

This is also the geometric constraints for the *parallel double-slit fold.* Now, we present our own study of the box-fold.

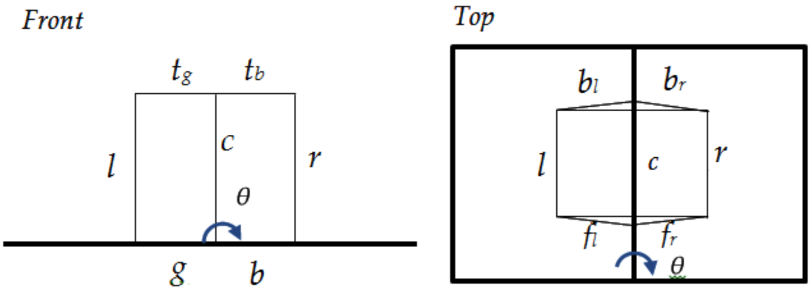


Figure 4.7: Two cross sections of a box-fold scaffold.

Foldability of a box-fold. According to [HS09], if a 2D quadrilateral with 4 sides S_1, S_4, S_3 and S_2, in that order, has $S_1+S_4 = S_3+S_2$, then it can fold completely flat when S_1 and S_2 are fully closed. Hence, a box with patches L, R, T_G, T_B, and C (refer to Figure 4.7) is foldable if

$$b + r = c + t_b \text{ and } g + l = c + t_g. \tag{4.1}$$

In addition, because the folds between F_L, F_R and between B_L, B_R are coplanar with the central fold and bisect the angle between L and R, the front and back patches, F_L, F_R, B_L, and B_R, can also be folded completely flat following the motions of L and R.

Stability of a box-fold. We show that a box-fold as constructed in Section 4.1 is stable. Assume that the structure is opened at an arbitrary angle in $(0, 180°]$ or t in $(0, 1]$. If the angle between C and G may change while G and B are held stationary, then the box will undergo a shearing effect. However, it also contains the front and back patches, F_L, F_R, B_L, and B_R, which keep L and R from shearing. In other words, C cannot rotate while G and B are opened at an arbitrary angle. As a result, a box-fold structure is stable.

4.2.2 Validity of Multi-style Pop-ups

Although each mechanism is foldable, the combined multi-style pop-up may not be foldable, due to the possible intersections between the mechanisms. In particular, during the closing of a pop-up, patches F_L, F_R, B_L, B_R of a box-fold or F_L, F_R of a v-fold emerge and may intersect with the corresponding patches in another box-fold, v-fold, or tent-fold. To make the whole pop-up foldable, it is important to position the mechanisms, so that such intersections do not occur.

First, we define a right-handed Cartesian coordinate system, in which the z-axis is perpendicular to the central fold and bisecting the interior fold angle between the primary patches. The y-axis lies along the central fold and the x-axis points in the direction of the ground patch.

Based on the structures defined in Section 4.1, a tent-fold, v-fold and box-fold can only be positioned along the central fold or between T_G and T_B of a box-fold. Step-fold can only lie between a base patch and patch L or R of a box-fold. A base patch can be a primary patch, or T_B, T_G of a box-fold. For instance, we can position a box-fold on top of another box-fold, then a step-fold between patch R of the upper box and T_B of the lower box.

Note that a step-fold always forms parallel pairs of patches, a tent-fold always remains symmetric to the central hinge, and they do not move along the y-axis during folding. Hence, we only need to consider the range of movement of box-folds and v-folds for intersection checking.

Box-fold. We consider a box-fold lying on patches B and G, in which the hinge between F_L and F_R is h_F. Let d be the thickness of the box along the x-axis, and y_F be the y-coordinate of h_F when the box is opened at 180°. Then when the box is fully closed (see Figure 4.8), the y-coordinate of h_F becomes $y_F + d/2$. Similarly, the y-coordinate of the hinge h_B on the back is $y_B - d/2$ when the box is closed. Hence, to avoid intersection, no other mechanism should be placed on B and G within the range $[y_B - d/2, y_F + d/2]$ along the y-axis.

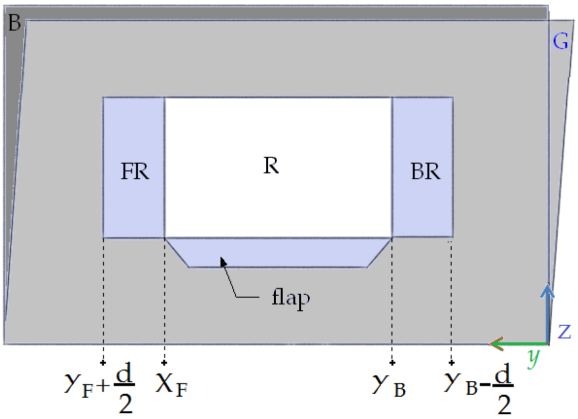


Figure 4.8: A fully-closed box-fold.

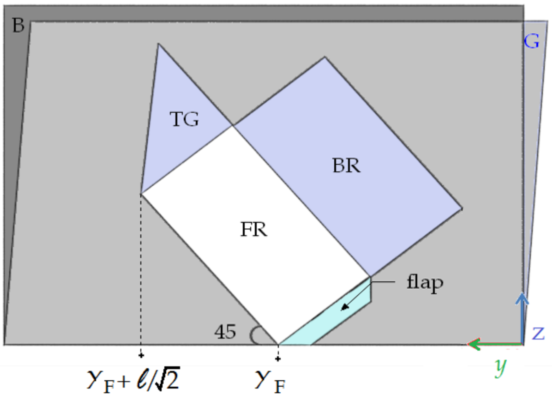


Figure 4.9: A fully-closed type-1 v-fold.

V-fold. In a type-1 v-fold, let y_F and y_B be the y-coordinates of the hinges between F_L, F_R, and between B_L, B_R respectively when the fold is opened at 180°. Note that, when B, G, and the two base patches of the v-fold are being closed, only F_L and F_R still touch G and B, and the v-fold leans toward the positive direction of the y-axis. Hence, the smallest y-coordinate of the v-fold is y_B. On the front of the v-fold, the intersecting point between F_L, F_R, T_B and T_G has the greatest y-coordinate, which becomes $y_F + l/\sqrt{2}$ at 0° fold angle, where l is the height of the v-fold along z-axis when the fold angle is 180°. To avoid intersection, no other mechanism should be placed on B and G within the range $[y_B, y_F + l/\sqrt{2}]$ along the y-axis. The range for a type-2 v-fold is similar, with

y_B being the y-coordinate of the right cut lines of F_L and F_R when the fold is fully opened.

By following the two sufficient conditions above, we avoid intersections between the considered mechanisms and guarantee that the combined pop-up is fully foldable. Furthermore, because all the patches of each mechanism are stable, by induction, their resulting multi-style structure is also stable.

4.3 Motion of Pop-up Mechanisms

In this section, we parameterize each pop-up mechanism based on the range of motion it can produce. (see Figure 4.10)

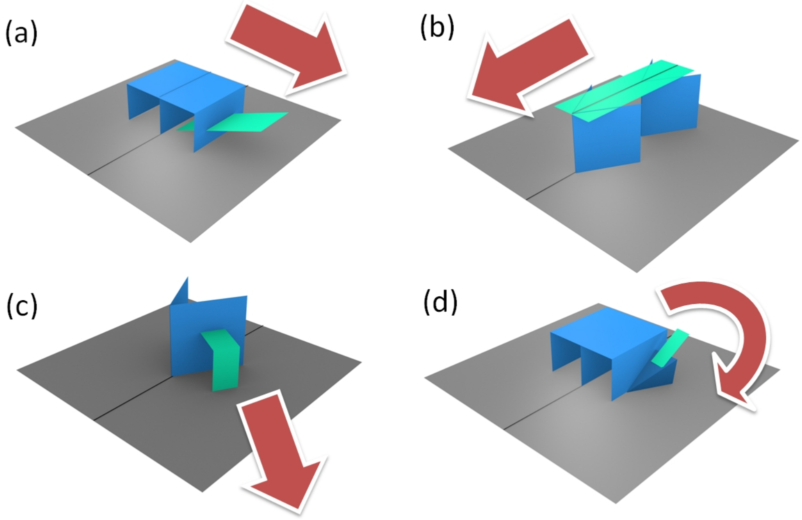


Figure 4.10: Pop-up mechanisms used to produce motion. (a) Floating layer and a single patch, (b) v-folds and a single patch, (c) v-fold and step-fold, (d) floating layer and an angled v-fold.

We first define a fixed coordinate system as shown in Figure 4.11. The y-axis lies along the central hinge, while the z-axis always bisects the fold angle θ. For convenience, we also use δ to denote $\theta/2$. The x-axis forms with the y- and z-axes a right-handed system. We aim to design a pop-up whose projected motion on the xy-plane resembles the input motion.

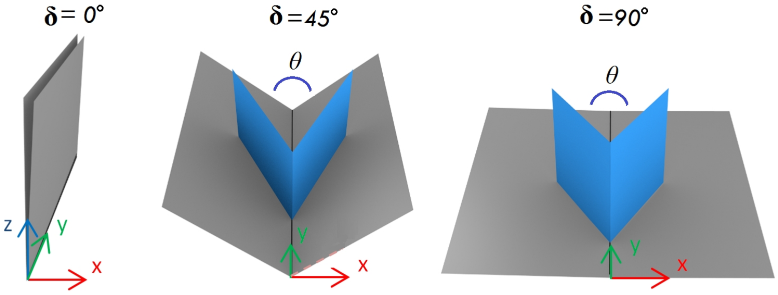


Figure 4.11: Pop-up showing the coordinate system and fold angles.

4.3.1 Horizontal Translation

In this section, we describe a pop-up mechanism to approximate an arm having horizontal or nearly horizontal translation (translation in the x-axis). An example of which is a snake coming out of a tree (Figure 8.9(e)) Although none of the patches in a pop up can be floating, we can still simulate this effect by combining a floating layer with a patch **M** glued to the central hinge and going through a slit S on the step-fold (Figure 4.12). To maintain symmetry while opening and closing the pop-up, we attach two step folds to the base patch. They form a rectangular structure when the pop-up is fully opened.

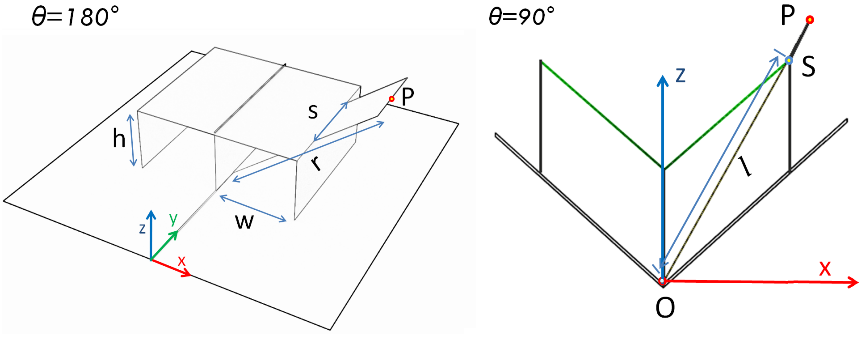


Figure 4.12: Mechanism for horizontal translation, using a step-fold and an extruding patch. Parameters: h, w and r.

We map the end effector of the translating arm to the endpoint P of patch **M**. The translation of the end effector can be simulated by the visible portion of **M** during the pop-up opening and closing. In particular, the visible portion of **M** becomes longer when the pop-up is being opened, and shorter when the pop-up is being closed. The structure is created so that at 180° fold angle, P matches the end effector's farthest position.

Let r be the actual length of $\mathbf{M}$, as measured from the central hinge to P, and w, h be the width and height of the step-fold patches along the x- and z-axes when the pop-up is fully opened. At each opening angle $\theta = 2\delta$, the visible portion of $\mathbf{M}$, with respect to the slit point S, is projected onto the xy-plane as

$$(P_x, P_y) = ((r - l)\frac{w \sin \delta}{l}, 0), \tag{4.2}$$

where l is the length of the portion of $\mathbf{M}$ hidden by the step-fold denoted by

$$l = w^2 + h^2 + 2wh \cos \delta. \tag{4.3}$$

In Equation 4.3, w and h are set based on the position of the root of the translating arm. The actual length of $\mathbf{M}$ (r) is obtained from the farthest position of the end effector using

$$P_{x\_farthest} = r\frac{w}{\sqrt{w^2 + h^2}} - w. \tag{4.4}$$

4.3.2 Vertical Translation

Besides horizontal translations, we observe various vertical and nearly vertical motions (translation in the y-axis). A simple example is a frog's tongue thrown out of its mouth (Figure 8.7). We utilize a special set of v-folds to simulate this effect. Two v-folds are glued along a hinge so that their patches form with the xy-plane a 90° angle when the pop-up is fully opened (Figure 4.13). Note that the translation may not occur along the central fold line. In such a case, we use a step-fold to form an additional hinge.

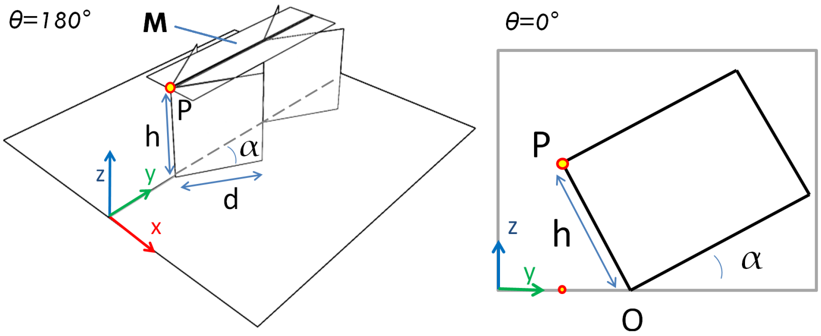


Figure 4.13: V-fold mechanisms for vertical translation. Parameters: α, h and d.

A patch **M** parallel to the xy-plane is attached on top of the two v-folds, such that when the pop-up is closed, **M** moves along the y-axis. In order to compute the amount of vertical movement of **M**, we consider P, the shared point between **M** and one of the v-folds. Let h be the z-coordinate of P and α be the angle the v-fold hinges form with the y-axis, as shown in Figure 4.13. Assume the y-coordinate of P at 180° fold angle is 0, then its y-coordinate at 0° fold angle is $-h \sin \alpha$. The projected motion of P is governed by

$$(P_x, P_y) = (0, -h \sin \alpha \cos \delta). \tag{4.5}$$

In equation 4.5, we obtain h from the depth of the considered body segment, and then α from the amount of vertical translation of the considered end effector. Note that when the target translation exceeds h, it is not possible to obtain α. To handle this, we allow the user to parametrically select between the desired translation and the desired depth. For instance, if the depth of the considered body part leads to $h = h_0$, while the target translation is $v_0 > h_0$, then the final h will be set to $ch_0 + (1 - c)v_0$, based on a user specified c. Usually, we set c to a high value such as $0.7 - 0.8$ because we wish to retain the original height or depth ratio of the patch in relation to the input model. In general, the direction of the motion is more important than the exact or actual translation distance in producing the perceived movement.

4.3.3 Diagonal Translation

To approximate translations that are not aligned closely to the x- or y-axis, we combine a v-fold, as used for vertical translation, with a step-fold (Figure 4.14). We use a point P on the step-fold hinge to simulate the translating end effector. When the pop-up is being closed, the movement of P along the x- and y-axes are

$$(P_x, P_y) = (h \sin\delta \sin\alpha, -h \sin\alpha \cos\delta). \tag{4.6}$$

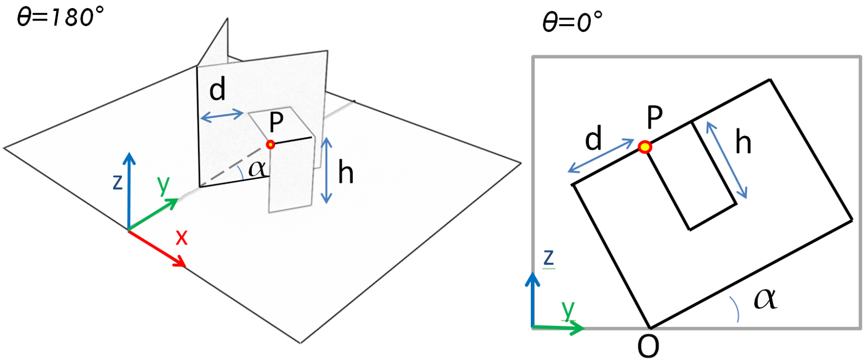


Figure 4.14: V-fold and step-fold mechanisms for a diagonal translation. Left: Opened pop-up from a perspective view. Right: Closed Pop-up from a side view. Parameters: α, h and d.

In equation 4.6, α is computed by matching $(h\cos\alpha,\ h\sin\alpha)$, the position of P when the pop-up is fully closed, with the farthest translation of the actual end effector. Similar to the mechanism for vertical translation, we allow for adjustable h.

4.3.4 Rotation

In our work, we consider 2D rotation as a precursor to more complex motions. To do so, we use a v-fold mechanism with an extended patch (Figure 4.15). The v-fold patches lie exactly on the base patches at 180° fold angle. Let O be the intersection between the v-fold patches and the base patches, which is also the origin of our coordinate system. Let Q and R be two endpoints on the central fold line and the popped-up fold line that coincide when the fold angle is 180°. Let S be the point on the shared hinge between a popped-up patch and the corresponding base patch, so that QS and RS are perpendicular to the y-axis.

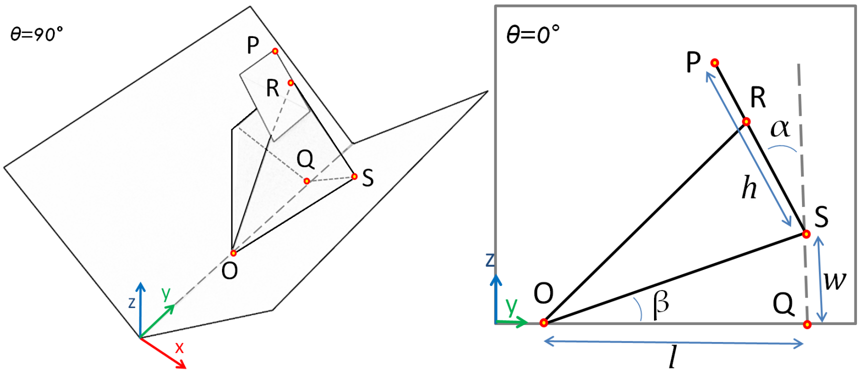


Figure 4.15: Rotation approximation using an extended v-fold mechanism. Parameters: h, w, and l.

The end effector of the rotating arm is approximated using a point P on the plane containing patch ORS (Figure 4.15). Since Q, R and S can be positioned arbitrarily along the y-axis, we can assume P, R and S are collinear without loss of generality.

We use l, w, h to denote $||OQ||$, $||QS||$ $||PR||$, and δ to denote half of the fold angle, which is the angle between the z-axis and the base patch that S lies on. The x- coordinate of P, the corresponding point to the rotating end effector, can be computed easily as

$$P_x = -h \sin \delta.$$

In Fig 4.15, the y-coordinate of P can be computed as follows,

$$P_y = l - (w + h) \cos \delta \sin \alpha, \tag{4.7}$$

where

$$\sin\alpha = 2\sin\beta\cos\beta = \frac{2lw\cos\delta}{l^2 + w^2\cos^2\delta}. \tag{4.8}$$

From Eq.4.7 and 4.8, we can formulate P_y as

$$P_y = l - \frac{2(w+h)lw\cos^2\delta}{l^2 + w^2\cos^2\delta} = \frac{l^3 - lw^2\cos^2\delta - 2lwh\cos^2\delta}{l^2 + w^2\cos^2\delta}, \tag{4.9}$$

thus,

$$(P_x, P_y) = (-h\sin\delta, \frac{l^3 - lw^2\cos^2\delta - 2lwh\cos^2\delta}{l^2 + w^2\cos^2\delta}). \tag{4.10}$$

When the pop-up is fully opened, $|P_x|$ is simply h. Hence, we can easily obtain h by considering the last position on the trajectory of the end effector. In addition, to simulate the depth of the input character, we need to constrain w so that when the pop-up is closed, R_y does not exceed a precomputed height d of the body part that the rotating arm originates from. Since $R_y < 2w < d$ and a greater w allows a larger range of rotation, we set w to $d/2$.

As a result, the v-fold generation becomes optimizing l so that the projected trajectory of P on the xy-plane matches the rotating end effector closely. The trajectory fitting is described in Section 6.3.

Note that our formulation still holds if the center of rotation does not lie close to the central hinge. To handle such case, we add a floating layer that forms a new hinge with a base patch, and create a v-fold along that hinge. The technique is used in most of our results (Figure 8.7, 8.8(a,b,c), 8.9(e)).

4.3.5 Stationary Mechanism

For linkage chains that do not have any motion or for parts of the mesh that are not assigned to any linkages, we make use of the floating layer mechanism (shown in Figure 4.16). Note that no pop-up mechanism can actually be completely stationary during the opening of the pop-up. However, patches on floating layers are perceived to be stationary because they are always parallel and have the same distance with respect to the base patches.

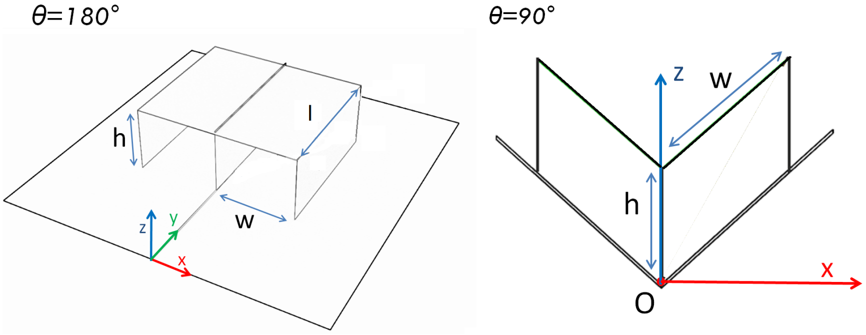


Figure 4.16: Stationary patches use the floating layer mechanism. Parameters: h, w, and l.

The h parameter is computed based on the average normalized depth of the vertices that are visible from the viewpoint. We have the parameter h_{max}, which represents the highest height any floating layer in the actual physical paper pop-up. It will be assigned to the shallowest depth value. We usually set h_{max} to 2.5 cm using an A4 base patch. These variations in the height allows us to have multiple layers in the final pop-up design. Note that modifications to l and w of this mechanism will not affect its foldability or stability, making it a good candidate for parameter modification when resolving intersections.

Chapter 5

Approximating 3D Shape

Paper engineers have two considerations when they want to represent real-world objects through paper pop-ups. First, how to closely approximate the volume and shape of the object. Second, how to take advantage of the movement of the paper piece during opening or closing to convey some forms of animation. Here we describe our automated algorithm to convert an input 3D mesh into a valid animated paper pop-up design specifically focusing on reproducing its shape.

5.1 3D Volume and Shape Representation

We use the geometric and texture information of a 3D mesh as input. The main idea is to abstract a mesh into sub-volumes by fitting 3D primitives, and then choose the best mechanism to represent each primitive. Each primitive-mechanism pair has its own set of steps to convert a shape into a set of valid pop-up patches that is guaranteed by our formulation. Once the patches have been generated and stabilized, we produce a design layout and determine an assembly order for the printable instructions.

5.1.1 3D Primitive Fitting

First, we align the input model using Normal Principal Component Analysis (NPCA). We obtain the principal axis of the 3D model using the surface normals weighted by the area of the faces. Our aim is to automatically align the model

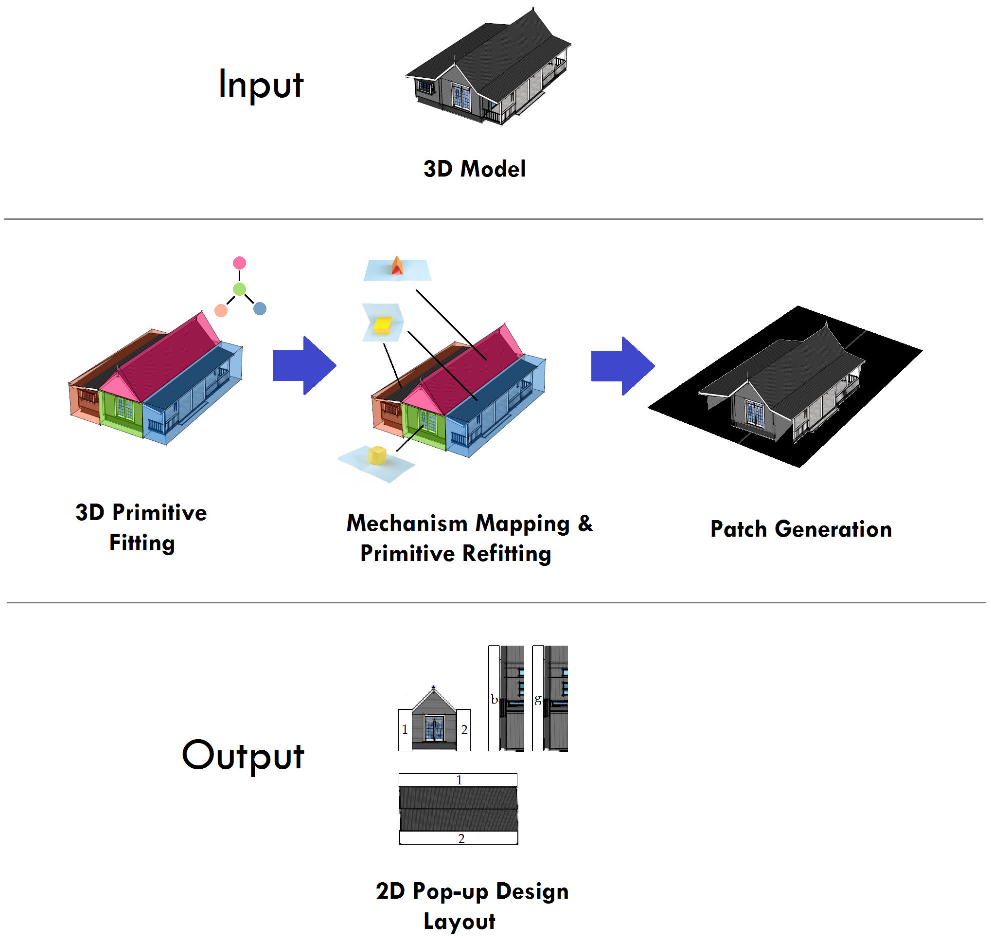


Figure 5.1: Overview of the automatic pop-up design algorithm.

such that the principal axis will point in the same direction as the normal of the faces with the most cumulative surface area. We set this principal axis as our y-axis, and the x- and z-axis to the vectors orthogonal to this axis. It is also possible to allow the user to align the model based on his own preference. Alternatively, we can also use the dominant long edges in the model to determine the alignment to facilitate foldability.

Then we obtain the symmetry plane and the tightest axis-aligned bounding box of the model. The base of the bounding box forms our initial backdrop and ground patches. They are connected to form the central fold line, where the

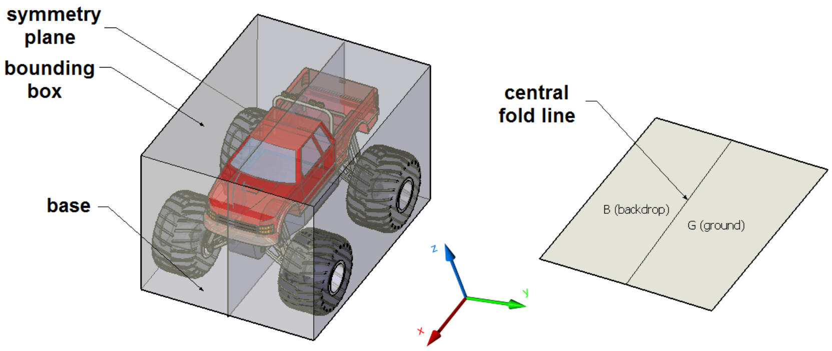


Figure 5.2: Model aligned with NPCA and the corresponding bounding box.

symmetry plane intersects the base.

We then perform a Random Sample Consensus (RANSAC) primitive fitting similar to the techniques used for point clouds [SWK07]. However, in our case we fit volumes rather than surfaces, similar to the work of [YK12]. Since some meshes may only have a few sparse vertices, we also employ a preprocessing step to add pseudo vertices to the input mesh in a technique similar to [SLCH11] for grid-based PCA. This ensures that our vertices are evenly distributed on the object's surface. Algorithm 5.1 shows how a single primitive is fit to a set of vertices and Algorithm 5.2 how the best set of primitives is obtained.

Our primitives also have some constraints on their orientation and connectivity. We primarily have two basic 3D primitives, which are the rectangular and triangular prisms. Because of the constraints on these primitives, we only need a minimum of two points to specify a rectangular prism and four for a triangular prism. In these prisms, the side's edges are always parallel to one another, while the top and bottom edges form a rectangle or a triangle.

The spatial constraints on our primitives are:

Constraint 1. Angles: Every angle on the rectangular prism must be 90°. Similarly, the angle between a top or bottom edge and a side edge in a triangular prism must also be 90°.

Algorithm 5.1: Random Sample Consensus (RANSAC)

```
$iterations := 0$
$best\_model := nil$
$best\_consensus\_set := nil$
$best\_error := infinity$
while $iterations < k$ do
    $maybe\_model := random(min\_no\_of\_points)$
    //these points must satisfy our spatial contraints
    $consensus\_set := nil$
    foreach point in input 3D model do
        if point is enclosed in maybe_model then
            add point to $consensus\_set$
        end
    end
    if $size_{consensus\_set} > d$ then
        $this\_model := maybe\_model$
        $this\_error := difference(this\_model, consensus\_set)$
        if $this\_error < best\_error$ then
            $best\_model := this\_model$
            $best\_consensus\_set := consensus\_set$
            $best\_error := this\_error$
        end
    end
    increment iterations
end
return $best\_model, best\_consensus\_set, best\_error$
```

Constraint 2. Orientation: The rectangular prism's four side edges should only be either parallel or orthogonal to the central fold line (x-axis). The triangular prism's three side edges, must similarly be either parallel or orthogonal to the central fold line (x-axis).

Constraint 3. Placement: A new primitive can only be placed on a *Base Patch*

Algorithm 5.2: Primitive fitting using RANSAC modified from [SWK07]

```
Data: Vertices $V$, $\tau$ is the minimum shape size
Result: A set $S$ of extracted primitive 3D shapes
// pre-process 3D mesh and generate psuedo vertices
$S \leftarrow \varnothing$
$C \leftarrow \varnothing$
repeat
    $C \leftarrow C \cup new\_candidates()$
    $m \leftarrow best\_candidiate(C)$
    //if the probability that no better candidate was overlooked during sampling
    if $P(|m|, |C|) > p_t$ then
        $V \leftarrow V \backslash V_m$ // remove points
        $S \leftarrow S \cup m$ // add the primitive
        $C \leftarrow C \backslash C_m$ // remove the invalid candidates
    end
until $P(\tau, |C|) > p_t$
```

Pair. A Base Patch Pair (BPP) is a set of two patches that share a common hinge and can act as intermediary backdrop and ground patches. For example, in Figure 5.6, we start with our first BPP, which are the primary patches $\{B, G\}$. Once we add the rectangular prism in the center, three more base pairs are generated: $\{F1, F2\}$, $\{F3, B\}$, $\{F4, G\}$. Adding another rectangular prism on $\{F4, G\}$ generates two more base pairs $\{F4, F5\}$ and $\{F6, G\}$. Finally, we add a triangular prism on $\{F1, F2\}$ pair, and this does not add any new BPP.

These constraints are based on the possible combinations of the different mechanisms, as described in Section 4.2.2. For example, a triangular prism is permitted on top of a rectangular prism, but not on its side, since such combination may not map to a foldable pop-up.

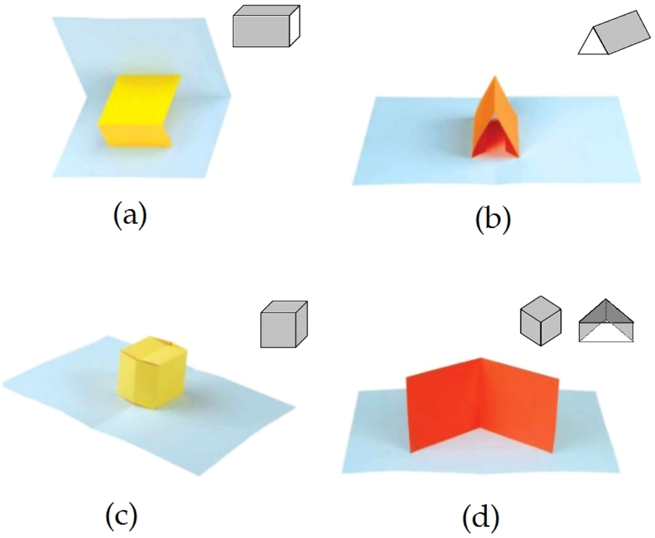


Figure 5.3: Pop-up mechanisms and the corresponding 3D primitives: (a) step-fold, (b) tent-fold, (c) box-fold and (d) v-fold. The shaded faces are the principal faces.

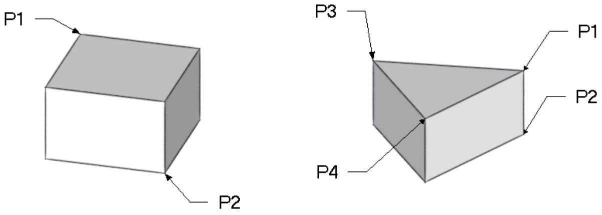


Figure 5.4: The minimum points to specify the RANSAC 3D volumetric primitives.

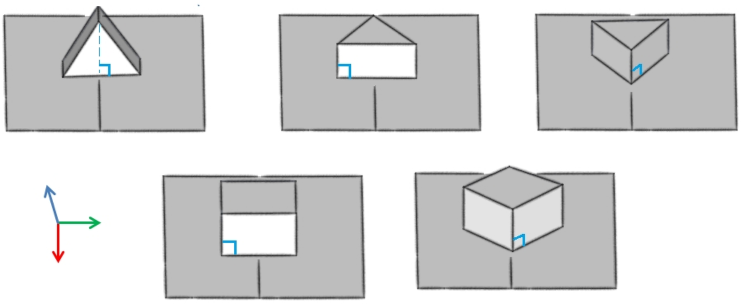
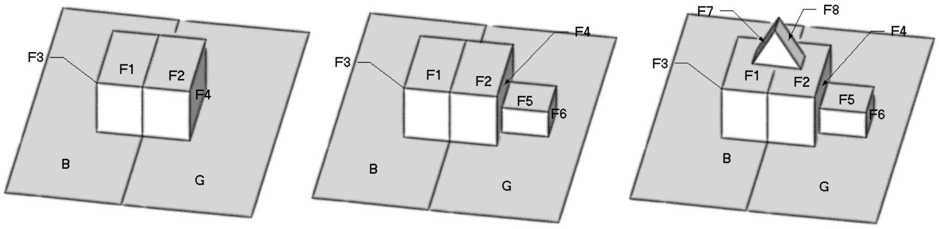


Figure 5.5: Valid orientations of the 3D primitives.



(a) BPP = { {B,G} } (b) BPP = { {B,G}, {F1,F2}, {F3,B}, {F4,G} } (c) BPP = {{B,G}, {F1,F2}, {F3,B}, {F4,G}, {F4,F5}, {F6,G}}

Figure 5.6: Base Patch Pairs.

5.1.2 Mechanism Mapping and Primitive Refitting

After the 3D primitive fitting step, we now have a 3D model that is abstracted using primitives that best approximate the original shape and at the same time can be mapped to pop-up mechanisms. Using the valid primitive-mechanism pairs shown in Figure 5.3, we derive Table 5.1.

Primitive	Step	V-fold	Box	Tent
Rectangular Prism	✓	✓	✓	
Triangular Prism		✓		✓

Table 5.1: Possible primitive-to-mechanism mappings.

Most of the abstraction and approximation is done in the previous step. Normally, there will only be a few possible mappings of the primitives and we can do an exhaustive search of all the possibilities. We select the combination that minimizes a certain error criterion. Our error criterion is based on the coverage of the mechanism or how well its patches can approximate the primitive, defined as

$$error_{cover} = 1 - (\sum Area_{principalface} / \sum Area_{face}). \tag{5.1}$$

Unlike [LJGH11] that uses all the faces of the voxel as patches, we only use some faces of the primitives called the *principal faces* (the shaded faces in Figure 5.3). This is due to the fact that some mechanisms do not cover all the sides of a volume like a tent-fold. The error is based on the surface area covered by only the principal faces over all the surface area of the faces of the primitives in the structure.

During the mechanism mapping, we also examine the validity of the generated structure based on the conditions in Sections 4.2.1 and 4.2.2. These formulations allow us to test the foldability and stability of the structure without the need for simulations.

The primitives can also be refitted as long as they satisfy the foldability of the mechanisms. For example, the side patches for box-folds do not necessarily need to be orthogonal to the ground or backdrop patches, as long as the lengths of the patches still satisfy $b + r = c + t_b$ and $g + l = c + t_g$ (Equation 4.1), such as the case of the ship in Figures 5.7 and 8.3, where the sides of the box are refitted to slant along the body of the ship. In addition, if sufficient gap between different box-folds along a common hinge is not achievable, the front and back patches of the box-folds can be partially trimmed to avoid the intersection.

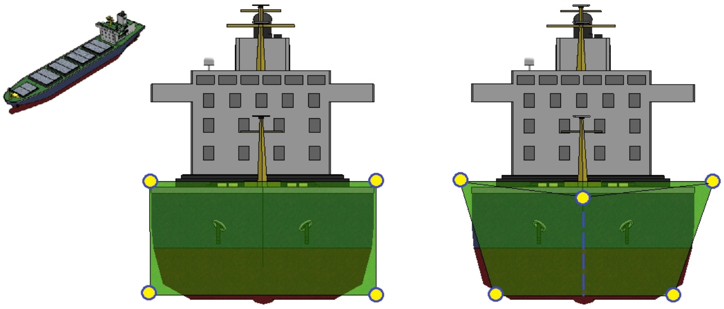


Figure 5.7: Refitting a rectangular prism.

5.1.3 Patch Generation

Once we know the mapping of the primitives, we now convert each principal face of the primitive into a pop-up patch. Unlike previous approaches that simply consider the entire face as a patch, we use an image-based approach to better approximate the shape and contours of the object. In general, we render an orthographic projection of the mesh unto a plane co-planar with the principal face. This produces an image that becomes the basis of the shape and texture of the patch.

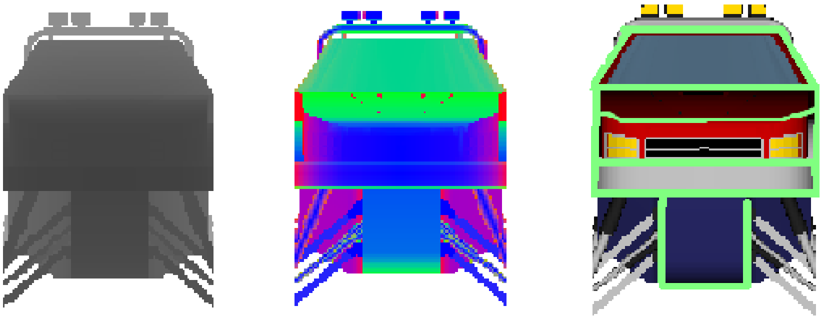


Figure 5.8: Depth map, normal map and image segments.

Box-folds are of particular interest because the front, back and top patches (F_L, F_R, B_L, B_R, T_G, T_B) can be composed of multiple layers of patches, which can better approximate the volume of the original mesh. For example, the box-folds used in the car and monster truck in Figure 8.2 are composed of several front, back and top patches at different depths or heights. These patches are obtained by performing image segmentation on the depth and normal maps taken from an orthographic projection of the front, back and top views.

Basically, the image segmentation works by locally fitting a quadratic surface on the segmented pixels in the neighborhood of a candidate pixel, similar to [LLLN+14]. It determines whether a pixel p should be part of the current segment by thresholding $f(p)-q(p)$, where $f(p)$ is the depth value and the x-, y-, z-components of the normal vector at p, and $q(p)$ is the quadratic approximation from the previously segmented pixel p_0,

$$q(p) = f(p_0) + f'(p_0)(p - p_0) + \frac{1}{2}f''(p_0)(p - p_0)^2. \tag{5.2}$$

5.1.4 Design Layout Generation

After we obtain the patches, we continue to generate the design layout for cutting, folding and gluing. We position the patches separately on a sheet of paper, adding a flap to each edge where it connects with another patch. When two adjacent patches are bounded by cut lines and their common hinge, we connect them in the layout so that the user will only need to create a fold without gluing. We finally assign IDs to the patches and flaps accordingly.

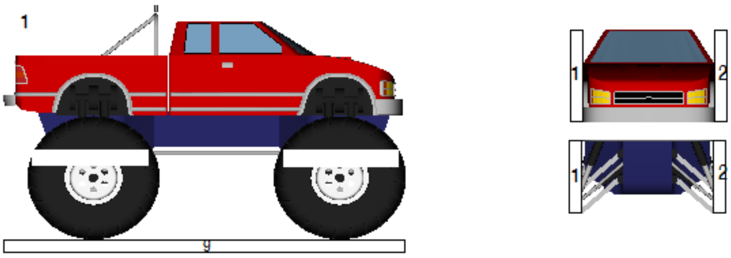


Figure 5.9: Sample 3D printable pop-up design layout.

The order of assembling the patches is important. Some orders are not feasible, while some are easier to construct in practice. We can employ a method similar to [APH+03], which produces assembly instructions for rigid components. While we achieve feasible layouts for our pop-ups, further study is needed to improve the ease of assembly.

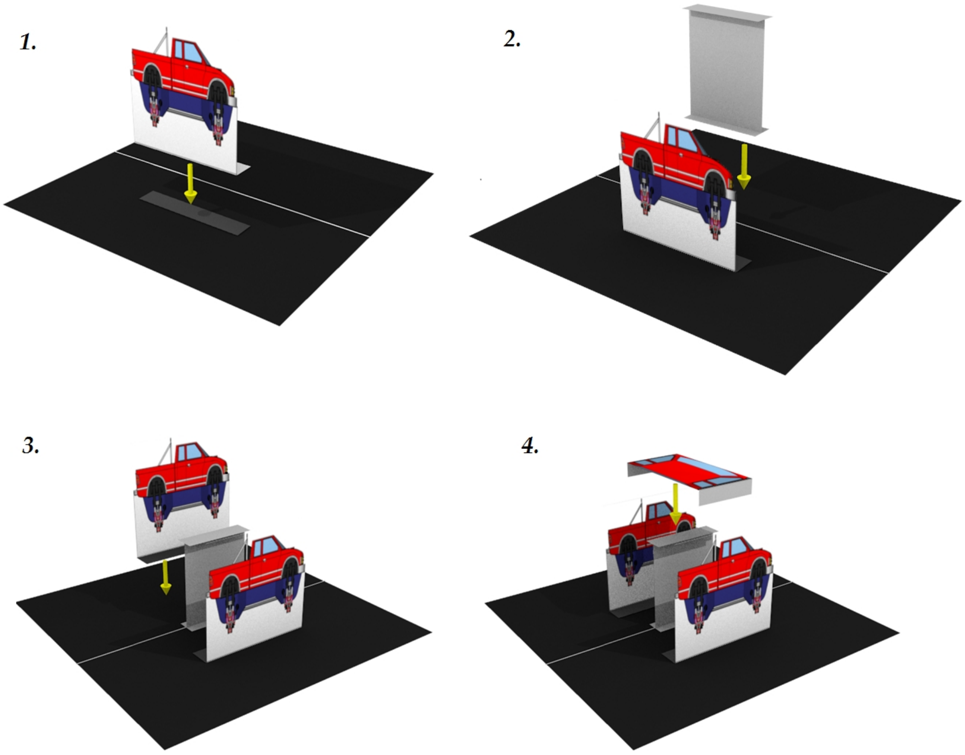


Figure 5.10: Sample instruction manual for paper pop-up construction.

Chapter 6

Approximating Motion

In this section, we discuss how we automatically generate a 2D design layout of an animated pop-up from the motion of a 3D articulated character. Our input is an animation file, containing a 3D mesh, armature (with skinning and rigging information) and 3D motion (using orientation keyframes of the joints). The user only specifies the location, viewpoint and viewport of the 3D character by positioning it on top of the base patches. We obtain the 2D motion by projecting the input onto the xy-plane. The mechanism mapping, parameter estimation and collision avoidance is done automatically. The process flow is shown in Figure 6.1.

6.1 Linkage Segmentation

We employ a straightforward method to group the linkages of the armature. We assume there are no cycles; therefore we deal with a tree-like structure that we divide into single-chain structures. We employed this approach and assumption based on our own observations of common armatures and skeletons used for 3D humanoid and animal characters. These armatures are often already grouped into IK chains by the animator.

We handle the problem of generating numerous small linkage chains by merging some of the leaf chains. This is done by performing simple thresholding using the skinning information of the linkages. We prune small linkage branches whose number of associated vertices to a specific linkage or bone falls below a particular

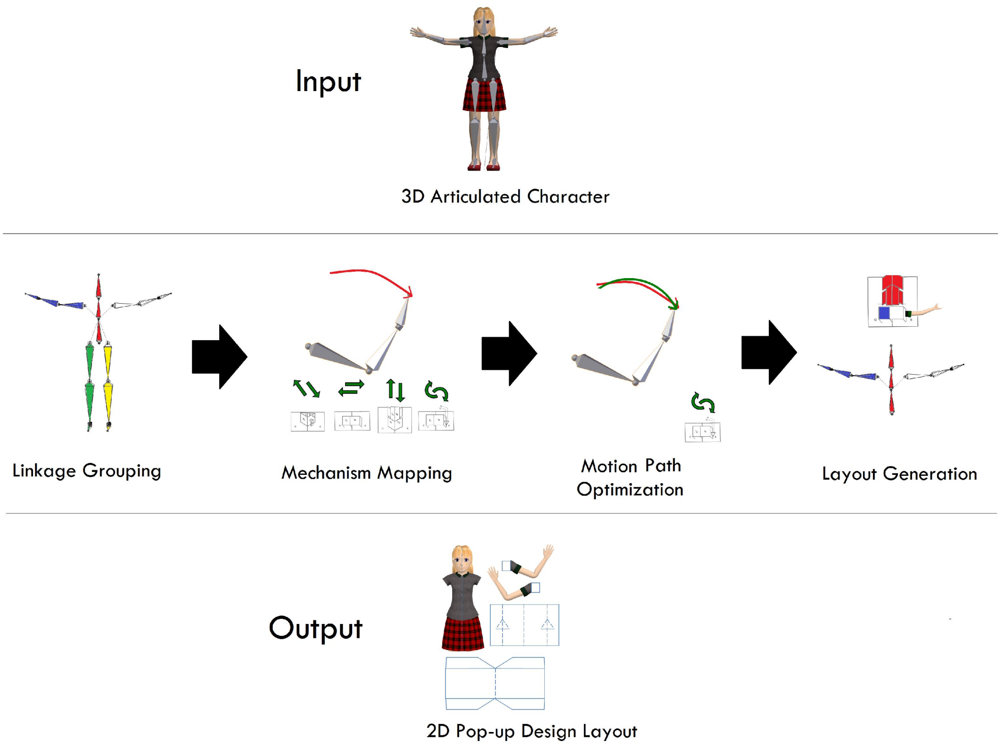


Figure 6.1: Overview of the automatic animated pop-up design algorithm.

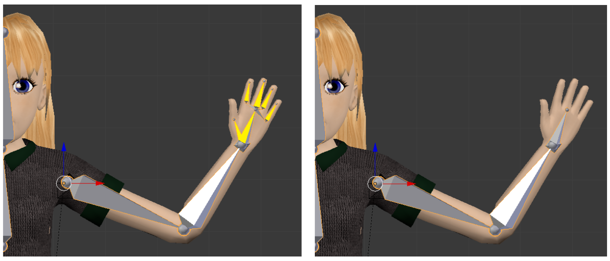


Figure 6.2: Skeleton pruning of the armature of the finger in the hand using $\tau_p = 0.09$.

threshold. For example, in Figure 6.2, we set $\tau_p = 0.09$, meaning that the links with less than 0.09× (ave. no. of vertices associated to a linkage or bone) will be pruned. If this happens, the vertices associated to a pruned linkage will be incorporated to its parent.

6.2 Pop-up Mechanism Matching

Each of the linkage chains now has to be mapped to a pop-up mechanism that best approximates its motion. First, we use forward kinematics to determine the motion path of each linkage chain. Then, we sample the motion, using the adaptive sampling technique described in [dF95]. Our goal is to have more samples in the regions of high curvature.

In their approach, they recursively sample a curve segment based on a specified criterion. In our case, we use the flatness of the sampled segment. The segment between A_a and A_b will have a new uniformly sampled point A_{new} if the angle $\delta > \tau_{angle}$ (see Figure 6.3). We set $\tau_{angle} = 2°$. This produces the set $\mathbf{T} = \{t_0, t_1, ..., t_{N_t}\}$ of N_t samples.

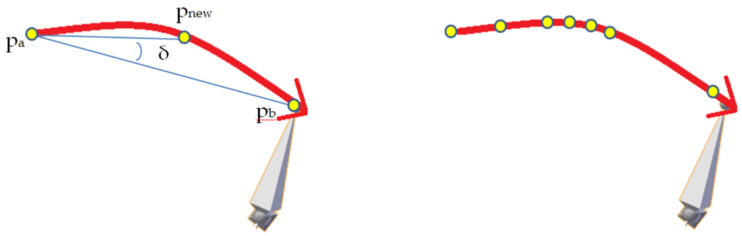


Figure 6.3: Adaptive sampling of the input motion.

Pop-up Mechanism	**Type of Motion**	**Direction**
V-folds & top patch	linear	vertical
V-fold & step-fold	linear	diagonal
Step-fold & extruding patch	linear	horizontal
Extended v-fold	circular	cw, ccw

Table 6.1: Possible output motion-to-mechanism mappings.

Currently, we only consider mechanisms that produce linear or circular motion (shown in Figure 4.10 and summarized in Table 6.1). As such, we can simply perform line or curve fitting on the samples in order to determine the appropriate mechanism. Note that non-moving linkages are automatically mapped to floating layers and the height is based on the average depth D of the pixels weighted to the linkage.

However, to consider more general cases and more complex pop-up mechanisms, we can make use of an error function (e.g. Equation 6.6). This error function measures the fidelity of the generated motion with the input motion. We use some estimated initial values for the parameters and select the mechanism with the lowest error.

6.3 Motion Parameter Estimation

After mapping the linkage chains to pop-up mechanisms, we determine the initial values of the parameters for each mechanism. The parameters for different types of translation can be computed directly from the key poses of the animation, as described in Section 4.3. However, there is no exact solution for parameter l of the rotating mechanisms. To estimate it, we define each rotation in the animation using center (x_0, y_0) and radius R, then minimize the following matching objective function,

$$F = \int_{t=0}^{1} ((P_x(t) - x_0)^2 + (P_y(t) - y_0)^2 - R^2)dt, \tag{6.1}$$

where P_x, P_y and t are as defined in section 4.3.4. The value of t goes from 0 to 1 when the pop-up is being closed from the fully opened state. In equation 6.1, P_y is the only function that contains the unknown l. Hence, the estimation of l becomes minimizing

$$F = \int_{t=0}^{1} ((P_y(t) - y_0)^2)dt. \tag{6.2}$$

Since there is no constraint on the largest value of l, we iteratively search for its value until the objective function is below our predefined threshold. The objective function is evaluated from a discrete number of t values. This simple approach has shown to work reasonably well in our domain of research.

6.4 Layout Generation & Refinement

After the previous steps, we now have an initial configuration of the pop-up structure. Combining mechanisms together usually results in a layout with intersections either in the opened state or when the pop-up is closed. We therefore progressively modify the parameters and mechanism mappings to find a valid configuration. Our main goal is to find a layout that best approximates the overall motion of the entire 3D character while avoiding any collisions.

We use simulated annealing (Alg. 6.1) to search in the configuration space $\mathbf{S}$ of the pop-up structure. The search minimizes a Boltzmann-like objective

$$f(s) = exp(-\frac{E(s)}{T}), \tag{6.3}$$

where $E(s)$ is a positive definite cost function of state $s \,\epsilon\, \mathbf{S}$. A new state s' is proposed at each iteration and accepted with probability

$$P(s'|s) = min(1, \frac{f(s'))}{f(s))}). \tag{6.4}$$

We start with $T = 10(t_0)$, and it is decreased by a factor $\mu = 0.8$ every 200 iterations.

6.4.1 Cost Function

The cost function we utilize measures the fidelity of the input motion with the output motion of the pop-up mechanisms, and the actual depth of the patches compared with the input mesh depth values. The cost function is

$$E(s) = w_m \sum_{j=1}^{N_l} (v_j \cdot E_m(j)) + w_d \sum_{j=1}^{N_l} (v_j \cdot E_d(j)), \tag{6.5}$$

where w_m and w_d are the weights for the contribution of the motion difference and depth disparity (we use 85 and 15, respectively), v_j is the weight contribution of a linkage to the overall motion based on the number of vertices assigned to j, N_l is the number of linkage chains, and $E_m(j)$ is the motion error of linkage chain j, we can use the L_2 norm

$$E_m = \sqrt{\frac{1}{N_t} \sum\nolimits_{t \epsilon \mathbf{T}} \|A(t) - P(t)\|^2}, \tag{6.6}$$

where $A(t)$ is a sample point from the input animation motion path and $P(t)$ is a sample from the output path of mechanism using the prescribed equations in Sec. 4 at time t, for all the t in set $\mathbf{T}$ (refer to Figure 6.4).

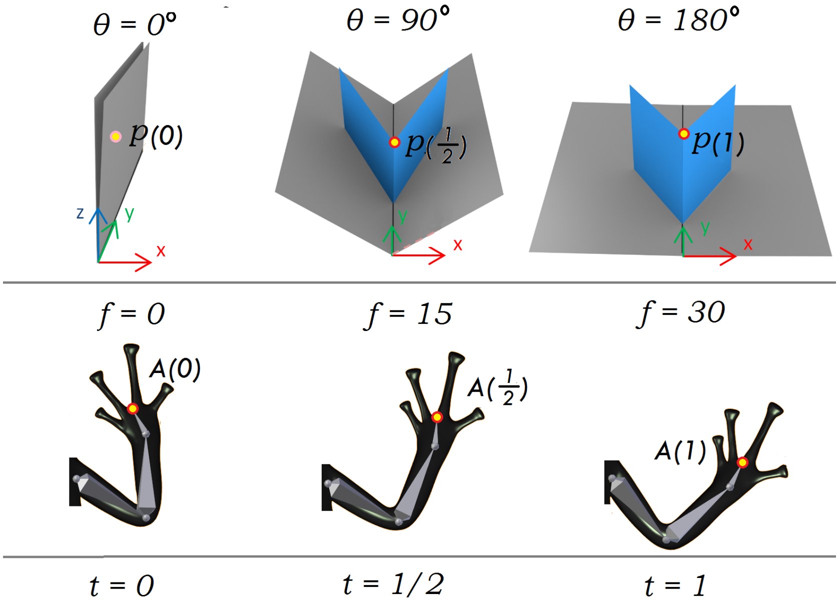


Figure 6.4: Pop-up at different fold angles θ (from left to right: 0°, 90°, and 180°), f is the frame number, and $t = \theta/180$ or f/no_of_frames. P is a sample point from the output paper pop-up and A is a sample point from the input animation.

$E_d(t)$ is the depth disparity of linkage j and is computed as

$$E_d = \|D - H\|, \tag{6.7}$$

where D is the normalized average depth of the vertices assigned to the linkage chain, and H is the actual height used in the pop-up for mechanism.

If we start with a state that has intersections, it is possible that even after using the possible moves to alter the current state that no new state will be intersection-free. We then employ a special procedure to avoid being stuck this initial configuration with intersections. We include an addition cost

$$E_c = exp(w_c \sum_{i=1}^{N_c} D_c), \tag{6.8}$$

where N_c is the number of collisions, D_c is the depth of the collision, and w_c is a relatively large value that we set to 10^5. This allows the algorithm to gradually reach a valid state. After which, we revert to the original cost function and enforce non-collision as a hard constraint.

6.4.2 Intersection Checking

Unlike in mechanical assemblies [ZXS+12, CTN+13], where the detection of the intersecting gears is straightforward, in our case, we must not only consider the opened state, but also check if there will be any collisions during the closing process of the pop-up. We employ a conservative approach to intersection checking by defining *Collision Bounding Volume* (CBV) for every pop-up mechanism.

The CBV is the volume that the mechanism's patches may occupy while the pop-up is being closed (see Figure 6.5). We determine the dimensions of CBV using the intersection checking formulations by [LSH+10, LJGH11] and our own geometric study. It is similar to merging the convex hulls of pop-up mechanism at different fold angles. Note that two CBV's may intersect but the two mechanisms

Algorithm 6.1: Simulated annealing algorithm for optimizing motion fidelity while avoiding intersections.

```
// initial mechanism mappings and parameters
$s := s_0$
// special procedure to check and fix initial collisions
$s := resolve\_intersections(s)$
// initial temperature and energy
$T := t_0$ ; $e := E(s)$
$s\_best := s$
$e\_best := e$
$k := 0$
while $k < kmax$ and $e > e\_max$ do
    $T := T \times \mu$
    repeat
        // change state by using possible moves
        $s\_new := move(s)$
    until $has\_collisions(s\_new) = false$
    $e\_new := E(s\_new)$
    if $P(s\_new, s, T) > random()$ then
        $s := s\_new$
        $e := e\_new$
    end
    if $e\_new < e\_best$ then
        $s\_best := s\_new$
        $e\_best := e\_new$;
    end
    increment $k$
end
return $s\_best$
```

involved may be non-intersecting if the patches occupy the same location but at different folding angles. However, we employ this technique to quickly search our search space for a plausible layout.

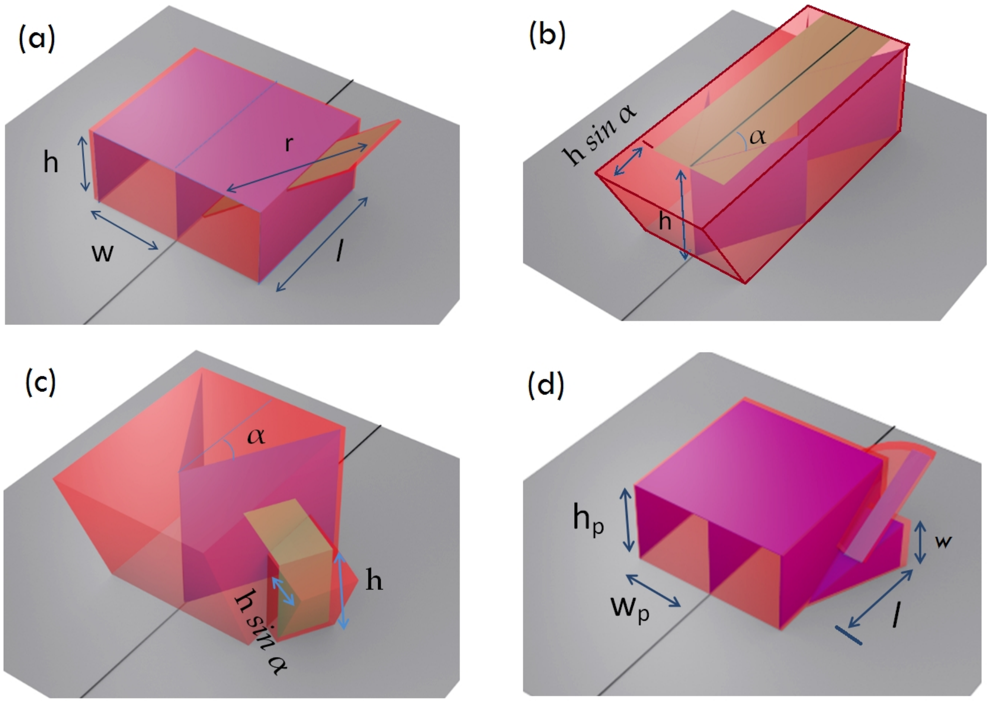


Figure 6.5: Example of a Collision Bounding Volume (pink region) for (a) horizontal, (b) vertical, (c) diagonal translation and (d) rotation mechanism.

Alternatively, we can also check for possible intersections by focusing on the end effectors since this will most likely cause the collisions. We can get the motion equations of two mechanisms and if there exist a t for which they are equal then there is an intersection. Another alternative is to use simulations. However, since it might take numerous iterations for the simulated annealing to find an optimal and valid state, we find that simulating the folding to detect intersections at every iteration is inefficient.

6.4.3 Possible Moves

The following are the possible moves to alter the current state:

1. Modifying the mechanism parameters. We randomly select one of the parameters and assign it a new value from a Gaussian distribution. The variance of the Gaussian is 10° for angle parameters and 1/20 of the diagonal

of the bounding box of the 3D mesh for length parameters.

Non-moving mechanisms such as floating layers can be cropped without theoretically affecting its stability. As such, this move has a higher probability of being selected. In Figure 6.6, part of the body the frog intersects with the mechanism that moves the tongue. It is possible to crop part of floating layer without affecting the overall structure and motion.

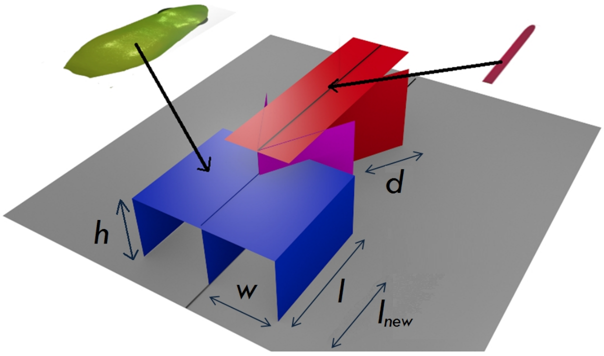


Figure 6.6: Intersecting floating layer and *v*-fold (magenta).

2. Merging two mechanisms together. In some cases the two intersecting mechanisms may share a *primary mechanism* (a basic mechanism directly attached to the base patches). An example is the case of Figure 8.8(a), in which the mechanisms for the left and right arms intersect. They share a floating layer mechanism and as such can be merged (see Figure 6.7). Note that the two mechanisms may have different parameter values. In this case, we compute a common set of parameters by weighting the values using the number of vertices assigned to the linkage. Furthermore, when computing the cost function, the mechanisms are treated separately, but sharing the same values for some of the parameters.

3. Changing the location of the mechanism. Another alternative is to move a mechanism in the $x-$ or $y-$axis, the new value is obtained from a Gaussian distribution with a variance of 1/50 of the length of the diagonal of the bounding box of the input 3D mesh.

4. Changing the mechanism mapping. It is possible to change the mapping

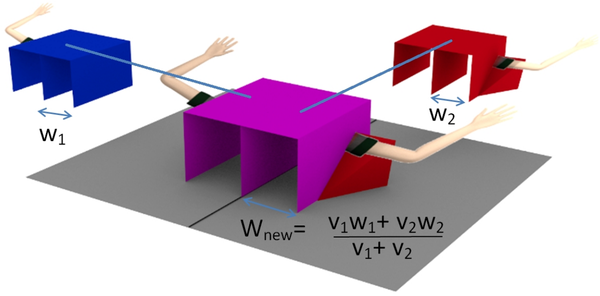


Figure 6.7: Two intersecting rotating arm mechanisms and merged primary mechanism (magenta).

> of a linkage to another type of pop-up mechanism. For example, if an intersection cannot be resolved, a linkage maybe reassigned to a stationary mechanism (i.e. floating layer).

The moves are selected based on a discrete distribution, with the probabilities shown in Table 6.2.

Possible Moves	**Probability**
Modifying the Mechanism's Parameters	0.6
Merging Mechanisms	0.2
Moving the Location (Translation)	0.1
Changing the Type of Mechanism	0.1

Table 6.2: Probability distribution of the possible moves.

6.5 Printable Pop-up Design

Finally, we are able to generate the 2D pop-up design layout for printing. Each mechanism has a specific set of rules in order to convert them from 3D patches to a 2D layout. Generally, it involves unfolding the pop-up mechanism and adding cut lines, fold lines and labeling flaps for gluing.

For the textured patches, we get an orthographic projection of the character unto a plane co-planar with the base patches. This produces an image that becomes the basis of the shape and texture of the patch. Note that this image may vary depending on the frame of animation we consider. For our case, we use the last frame since we want the opened state to reflect the end of the input animation.

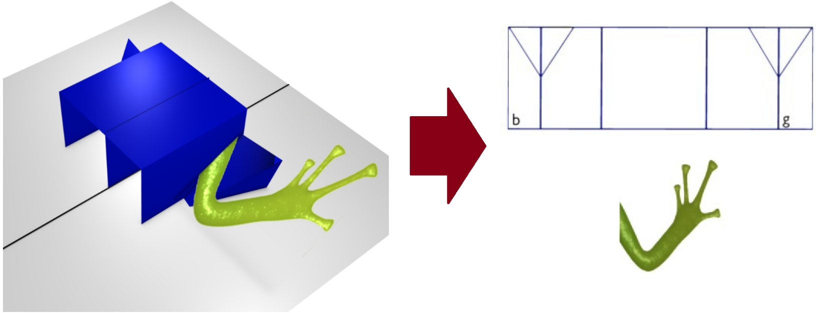


Figure 6.8: Generating 2D layout from a 3D patch structure of a pop-up mechanism.

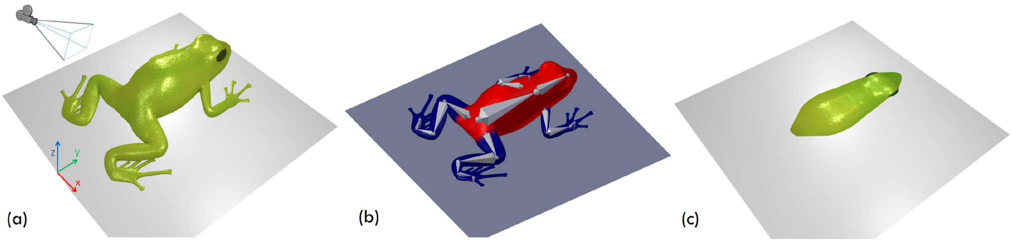


Figure 6.9: Textured patch generation. (a) Input mesh and viewpoint, using the inverted z-axis, (b) skinning information, red indicates the vertices assigned to the linkage, and (c) final output patch

From the generated 2D printable layout, the physical paper pop-up can be constructed. It usually entails, cutting and folding the patches at the indicated lines and gluing them together. Generally, we start with the mechanisms that lie directly on the base patches, specifically those that lie on the central fold and work outwards. Some mechanism maybe easier to construct first in their on their own before attaching them to the base patches or to other mechanisms.

Chapter 7

Technical Design & Implementation

In this chapter, we describe the technical design details for automated pop-up design. We use Unified Modeling Language (UML) 2.0 to describe our design. UML is a graphical language for visualizing, specifying, constructing, and documenting a software system. We present the class, use-case, activity and component diagrams for both approximating shape and motion using paper pop-ups. We also include a discussion of how we implemented our work.

7.1 Class Diagrams

Here, we present the class diagrams of the objects and their interrelationships, attributes and methods. The two main classes for both types of approximations are the pop-up and 3D mesh classes. Most of their methods implement the majority of approaches discussed in the previous chapters.

The 3D mesh class contains the data structures to store the mesh information, such as the vertex, face and material information. It also contains most of the mesh operators like methods for loading the mesh, setting the alignment and getting the projections for the patches.

The pop-up class is mostly made of methods that execute the automated pop-up design algorithms. It checks for validity and generates the 2D printable layout. Every pop-up has a set of base patches, the ground and backdrop. The fold angle is in radians and is also an attribute of the pop-up class.

Every pop-up is made up of mechanisms. Each mechanism is made up of a set of patches. Each patch is a plane that may have a texture. Each mechanism also has a corresponding primitive, which could either be a triangular or rectangular polyhedron.

A box-fold is a special kind of mechanism with some additional patches. It has a support patch in the center. The front, top and back patches may also be subdivided into multiple layers. The class diagram for approximating 3D shape through pop-ups is shown in Figure 7.1.

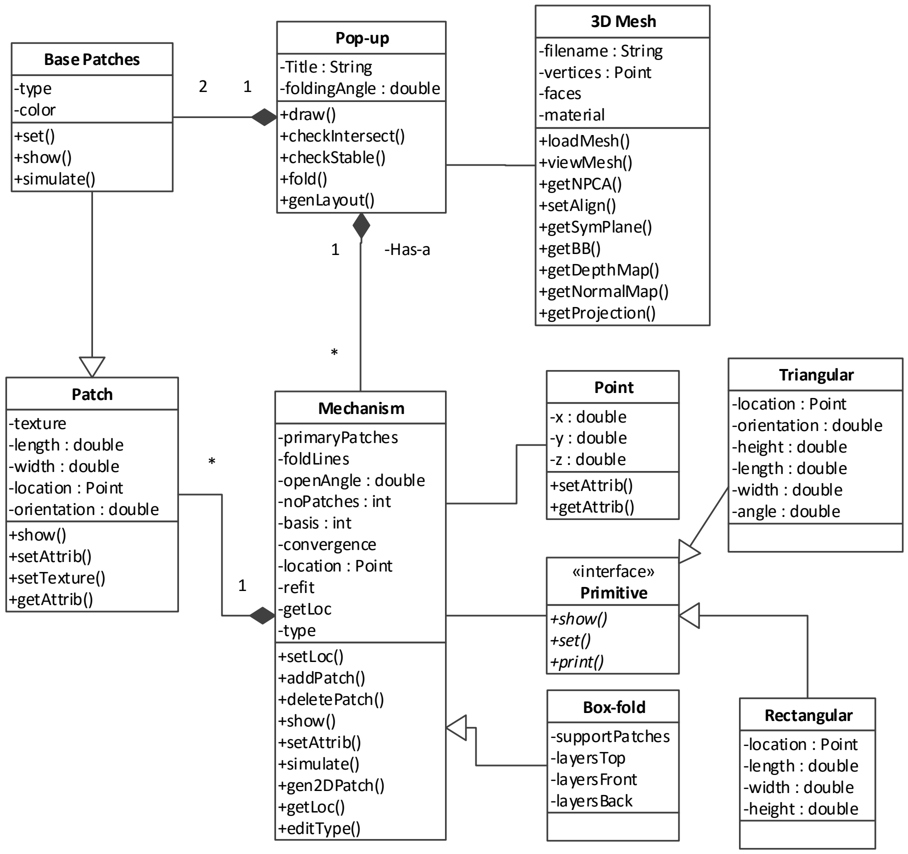


Figure 7.1: Class diagram of the Automated Paper Pop-up Design approximating 3D shape.

The class diagram for approximating motion is similar to the diagram for approximating shape. The main classes are likewise the pop-up and 3D mesh classes. However the 3D mesh class is now attached to an armature. The armature contains the bones and inverse kinematics (IK) grouping of the articulated figure. It also has the orientation keyframes that describe the motion of the character.

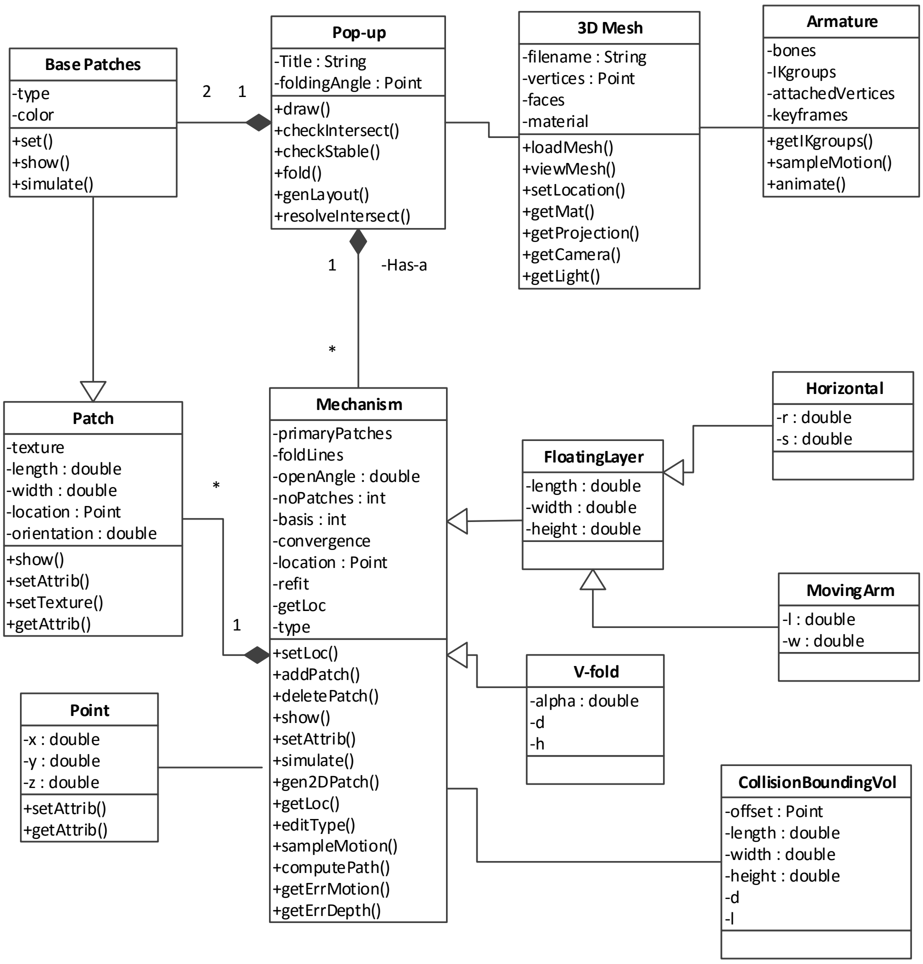


Figure 7.2: Class diagram for the Animated Paper Pop-up System.

Similarly, a pop-up is composed of mechanisms. There are four types of mechanism based on their motion. The horizontal and rotational motions (using the moving arm mechanism) use a floating layer as a primary mechanism and as such have some common attributes. The vertical and diagonal motion both use a v-fold and have the same attributes. The class diagram for approximating motion through pop-ups is shown in Figure 7.2.

7.2 Use-case Diagrams

Here, we present the functionalities of the system using a use-case diagram. The use-case is very straightforward because we developing an automated design system. Most of the other functionalities are for loading and visualizing the 3D mesh input. Other than tweaking the thresholds, the user only has to select the 3D mesh input, choose the alignment and the system will generate the 2D design automatically.

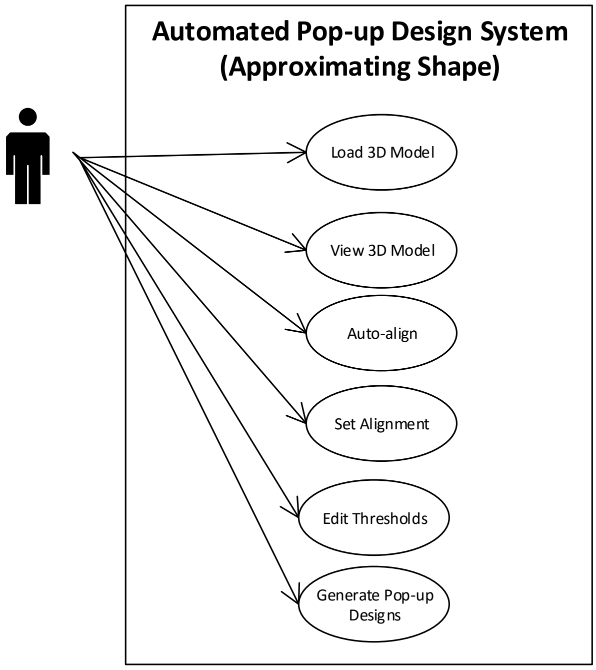


Figure 7.3: Use-case diagram of a Paper Pop-up System for approximating shape.

For the animated pop-up design system, we design it as a plug-in to *Blender*. This means that we take advantage of the functionalities that are already provided by *Blender*, such as loading and editing the 3D mesh, as well as functions for setting up the armature and animating the rig.

The use-case for the *Blender* plug-in itself is also very straightforward. The user simply has to position the mesh with respect to the base patches. The camera and lighting can be automatically obtained from *Blender*. The user can edit some of the parameters, although they will be set to default values that we have found to work for most models.

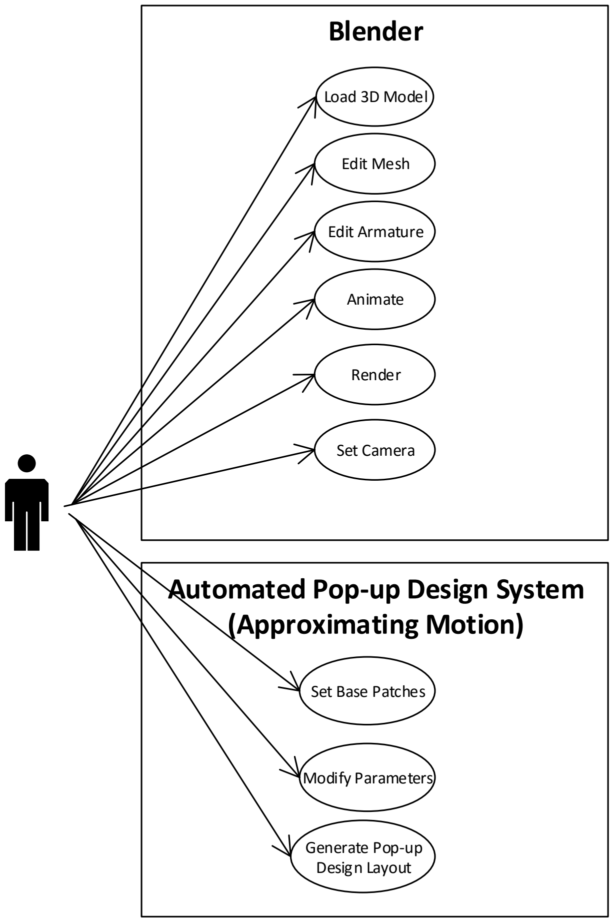


Figure 7.4: Use-case diagram of a Paper Pop-up System for approximating motion.

7.3 Activity Diagrams

Here, we present the activity diagrams for our automated algorithms. They describe in more detail the process flow of the system, which have been presented in the previous chapters.

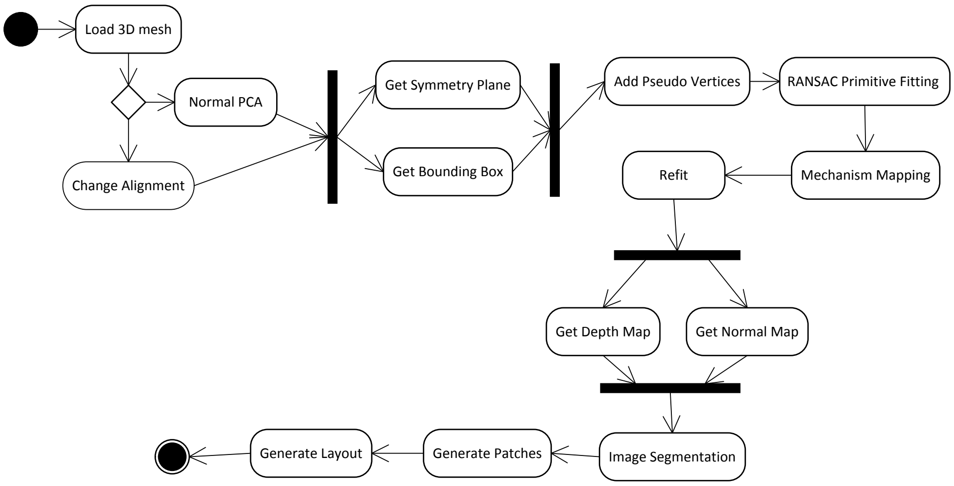


Figure 7.5: Activity diagram of a Paper Pop-up System.

The system starts by loading the 3D mesh, then the user decides if he wants to manually align the mesh or use NPCA to do the alignment automatically. After, the system gets the bounding box and symmetry plane as a guide for the position of the mesh on the base patches. Then, we add pseudo vertices to ensure that the mesh is uniformly sampled for the next step, which is the RANSAC primitive fitting. Once we have the set of primitives, we now map it to specific pop–up mechanisms. We then refit it based on the geometric constraints for foldability. We then obtain the depth and normal map for our image segmentation step. We then generate the patches of the mechanism, including the support patches. After that, we can place the patches in a 2D printable layout.

For the automated design algorithm approximating motion, we similarly start with loading the mesh. The user indicates the position on the base patches. The

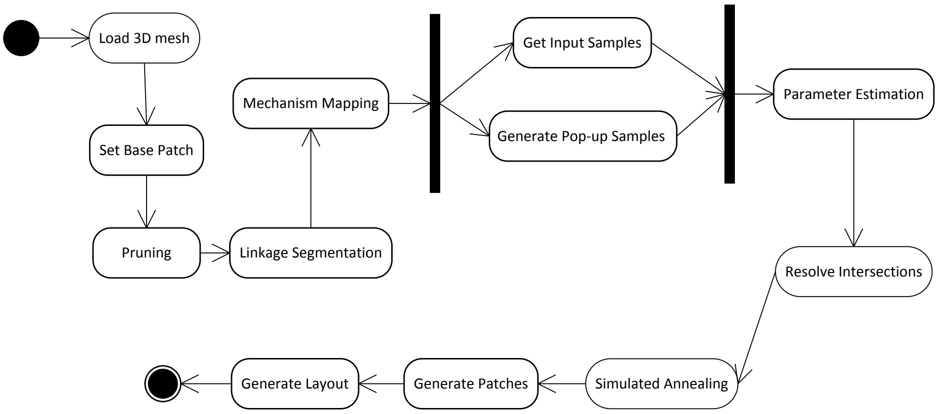


Figure 7.6: Activity diagram of the Animated Paper Pop-up System.

camera, lighting and material information can automatically be extracted from the file. We then obtain the armature information and prune out small leaf links/bones. We then get the IK groups and get the motion path. We map the linkage chain to a mechanism, based on the best fit between the mechanism's output motion and the input motion. We then sample the input motion path. Simultaneously, we get samples from the pop-up using the equations for the mechanism's output motion, which will be used for the motion error computation. We use simulated annealing to minimize the error while ensuring that the layout is intersection free. Once we have the 3D layout, we convert it to a 2D printable design.

7.4 Component Diagrams

Here we present the component diagrams for the automated system. For approximating shape, there are two main components, the 3D mesh component and the pop-up component. The 3D mesh component contains the 3D mesh class and all of its methods. As such, it handles the mesh operations. Its required interface passes the OBJ file to the component. Its provided interfaces connect to the pop-up component, which passes the geometric information of the mesh.

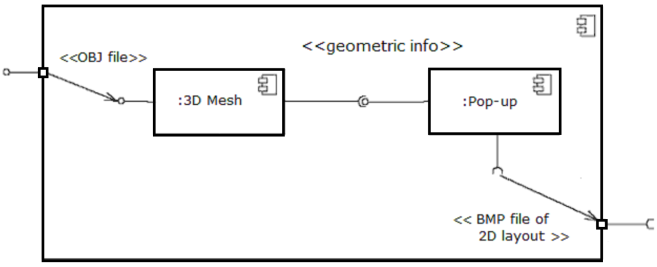


Figure 7.7: Component diagram of a Paper Pop-up System for approximating shape.

The pop-up component contains the pop-up class as well all of its aggregate classes. It does most of the work of our automated algorithm. Its provided interface is the image file of the printable 2D layout of the pop-up design.

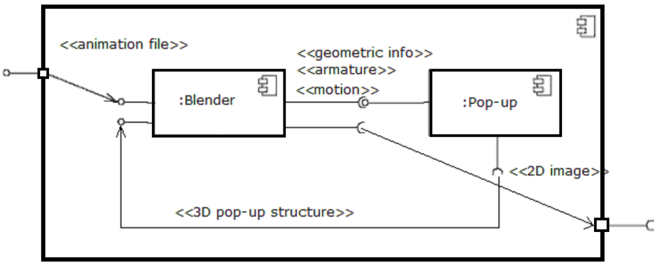


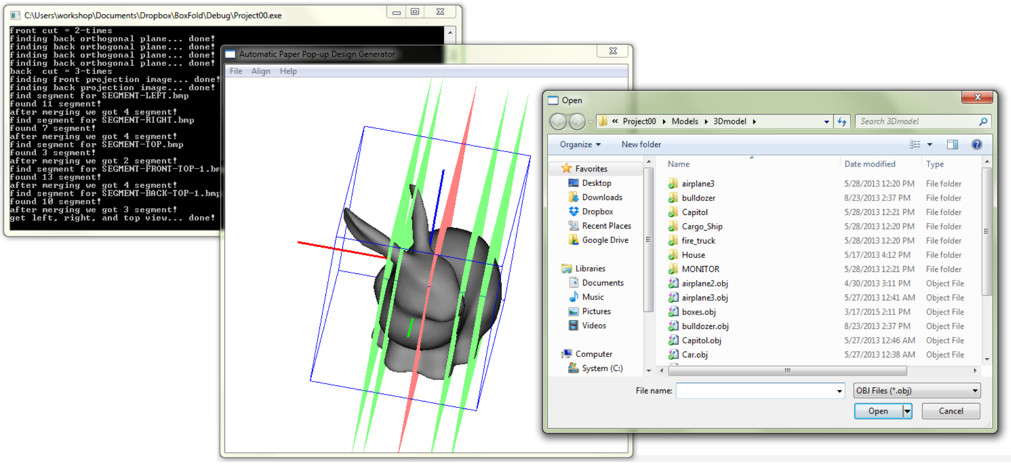
Figure 7.8: Component diagram of a Paper Pop-up System for approximating motion.

For the automated system for approximating motion using pop-up designs, we also have two components *Blender* and our own plug-in. We treat *Blender* as a separate component and describe how it interfaces with our plug-in. *Blender* does most of the mesh operations. It receives the animation file and sends to its provided interface the geometric, texture, armature and motion information.

The plug-in component creates a 3D mesh of the output pop-up structure. It unfolds this mesh and sends it back to the *Blender* component for rendering. *Blender* ultimately renders the 2D image of the design layout the animated pop-up.

7.5 Implementation

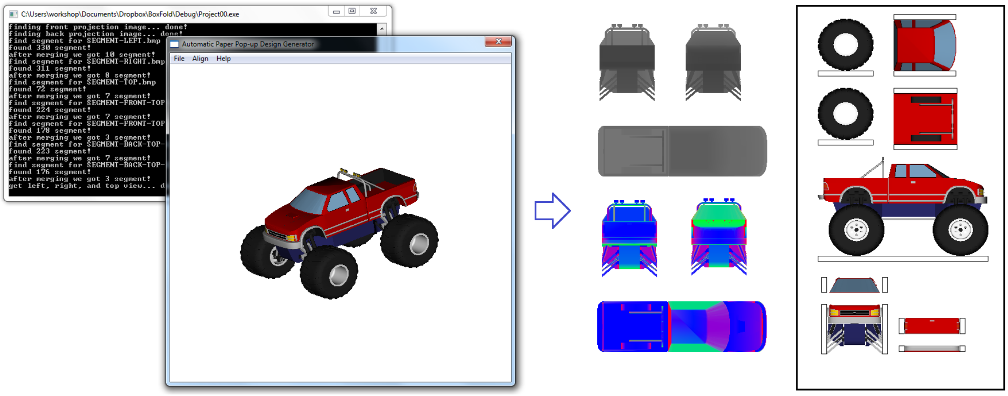
The automated pop-up design system for approximating 3D shape was developed using Visual Studio C++ 2010. Libraries used include OpenGL 4.5, Assim and Devil for OBJ mesh loading and visualization. We used the Globfit library as a basis for our primitive fitting. Figure 7.9 shows a screenshot of our system loading a mesh. Figure 7.10 shows the results of the system as a BMP file generated automatically from the 3D mesh after alignment.



```
front cut = 2-times
finding back orthogonal plane... done!
finding back orthogonal plane... done!
finding back orthogonal plane... done!
finding back orthogonal plane... done!
back  cut = 3-times
finding front projection image... done!
finding back projection image... done!
find segment for SEGMENT-LEFT.bmp
found 11 segment!
after merging we got 4 segment!
find segment for SEGMENT-RIGHT.bmp
found 7 segment!
after merging we got 4 segment!
find segment for SEGMENT-TOP.bmp
found 3 segment!
after merging we got 2 segment!
find segment for SEGMENT-FRONT-TOP-1.bm
found 13 segment!
after merging we got 4 segment!
find segment for SEGMENT-BACK-TOP-1.bmp
found 10 segment!
after merging we got 3 segment!
get left, right, and top view... done!
```

Name	Date modified	Type
airplane3	5/28/2013 12:20 PM	File folder
bulldozer	8/23/2013 2:37 PM	File folder
Capitol	5/28/2013 12:21 PM	File folder
Cargo_Ship	5/28/2013 12:20 PM	File folder
fire_truck	5/28/2013 12:20 PM	File folder
House	5/17/2013 4:12 PM	File folder
MONITOR	5/28/2013 12:21 PM	File folder
airplane2.obj	4/30/2013 3:11 PM	Object File
airplane3.obj	5/27/2013 12:41 AM	Object File
boxes.obj	3/17/2015 2:11 PM	Object File
bulldozer.obj	8/23/2013 2:37 PM	Object File
Capitol.obj	5/27/2013 12:46 AM	Object File
Car.obj	5/27/2013 12:38 AM	Object File

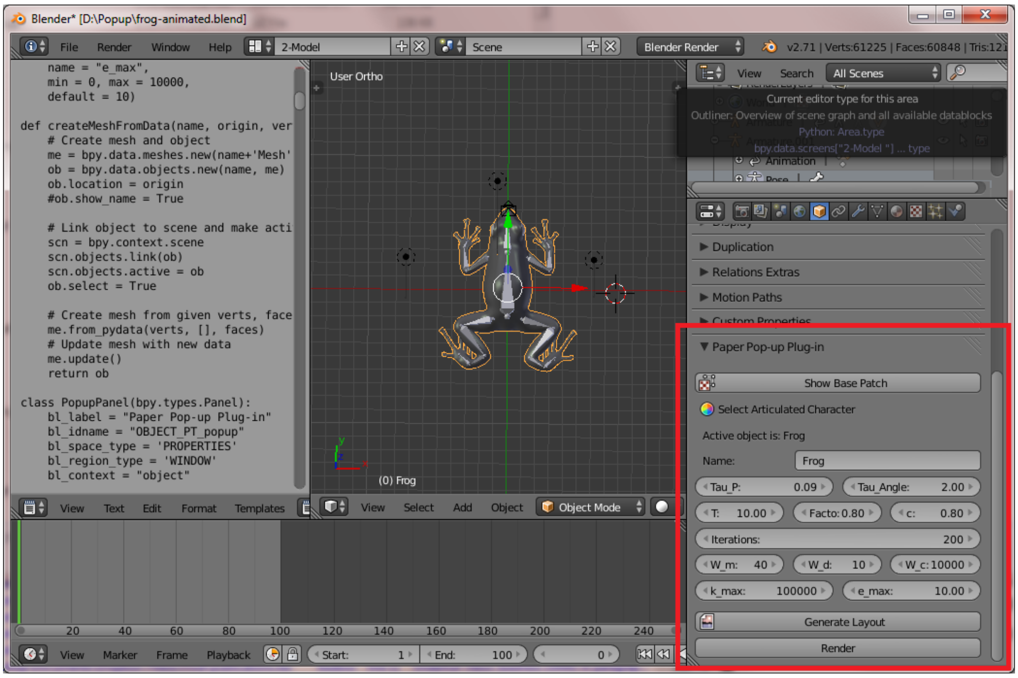
Figure 7.9: Screenshot of the system, loading 3D mesh.


C:\Users\workshop\Documents\Dropbox\BoxFold\Debug\Project00.exe
Automatic Paper Pop-up Design Generator
File
Align
Help

```
finding front projection image... done!
finding back projection image... done!
find segment for SEGMENT-LEFT.bmp
found 330 segment!
after merging we got 10 segment!
find segment for SEGMENT-RIGHT.bmp
found 311 segment!
after merging we got 8 segment!
find segment for SEGMENT-TOP.bmp
found 72 segment!
after merging we got 7 segment!
find segment for SEGMENT-FRONT-TOP
found 224 segment!
after merging we got 7 segment!
find segment for SEGMENT-FRONT-TOP
found 178 segment!
after merging we got 3 segment!
find segment for SEGMENT-BACK-TOP-
found 223 segment!
after merging we got 7 segment!
find segment for SEGMENT-BACK-TOP-
found 176 segment!
after merging we got 3 segment!
get left, right, and top view... d
```

Figure 7.10: Screenshot of the system, depth and normal maps, and output printable pop-up design layout.

We implemented our approach for approximating motion using Python scripts inside *Blender*, an open source 3D authoring and animation software. Our aim is to create a plug-in that users can use to create paper pop-ups. We use *Blender*'s built-in functions for rendering and simulating the kinematics of the armature. Consequently, we can also use any mesh that can be loaded by *Blender* (e.g. 3DS,obj, wrl, etc.) and use its interface to edit the mesh and animation. A screenshot of the plug-in is shown in Figure 7.11.



Current editor type for this area
Outliner: Overview of scene graph and all available datablocks
Python: Area.type
bpy.data.screens["2-Model"] ... type

```
    name = "e_max",
    min = 0, max = 10000,
    default = 10)

def createMeshFromData(name, origin, ver
    # Create mesh and object
    me = bpy.data.meshes.new(name+'Mesh'
    ob = bpy.data.objects.new(name, me)
    ob.location = origin
    #ob.show_name = True

    # Link object to scene and make acti
    scn = bpy.context.scene
    scn.objects.link(ob)
    scn.objects.active = ob
    ob.select = True

    # Create mesh from given verts, face
    me.from_pydata(verts, [], faces)
    # Update mesh with new data
    me.update()
    return ob

class PopupPanel(bpy.types.Panel):
    bl_label = "Paper Pop-up Plug-in"
    bl_idname = "OBJECT_PT_popup"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "object"
```

Active object is: Frog

Name: Frog

Tau_P: 0.09
Tau_Angle: 2.00
T: 10.00
Facto: 0.80
c: 0.80
Iterations: 200
W_m: 40
W_d: 10
W_c: 10000
k_max: 100000
e_max: 10.00

Figure 7.11: Screenshot of the Blender Paper Pop-up Plug-in.

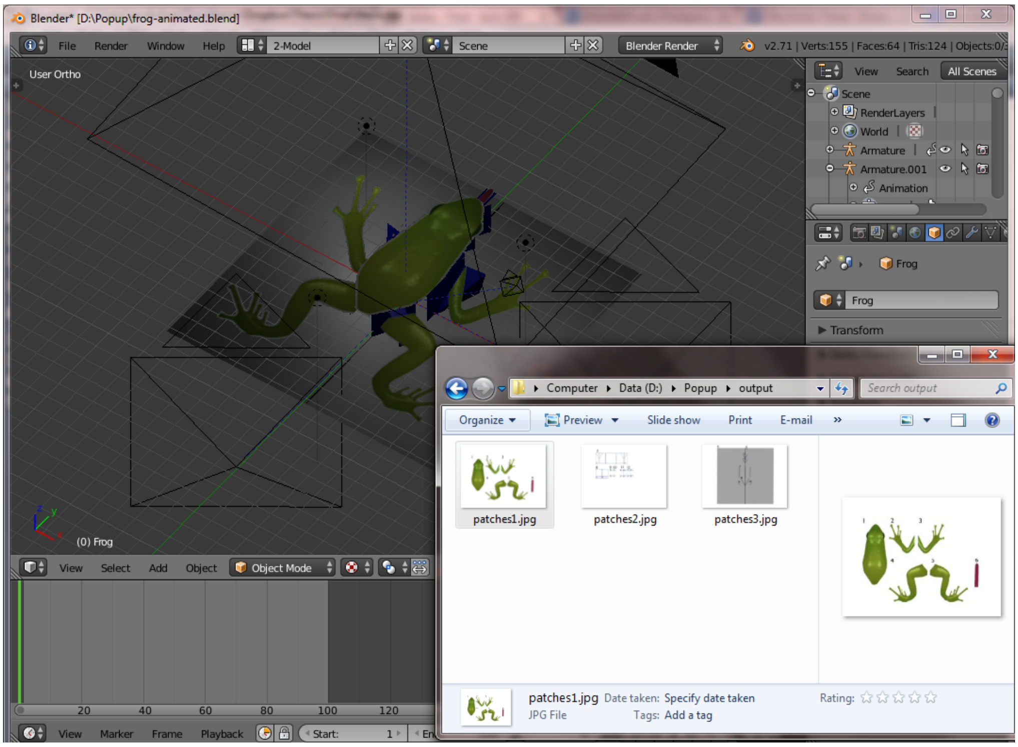


Figure 7.12: Results generated by the Blender Paper Pop-up Plug-in.

Chapter 8

Results

8.1 Approximating Shape

We run our algorithm on an Intel Core i7 PC with 8GB of RAM on a NVDIA GT 330 graphics card. The entire process completes in a few seconds to a few minutes with the primitive fitting accounting for most of the running time. We have tested our approach on several 3D models from the Google 3D Warehouse [Goo13], where some models contain up to 30000 vertices. Figures 8.2 and 8.3 show some of the models we have used and the sample printable pop-up design layouts can be found in Appendix A and some actual pop-ups in Appendix B.

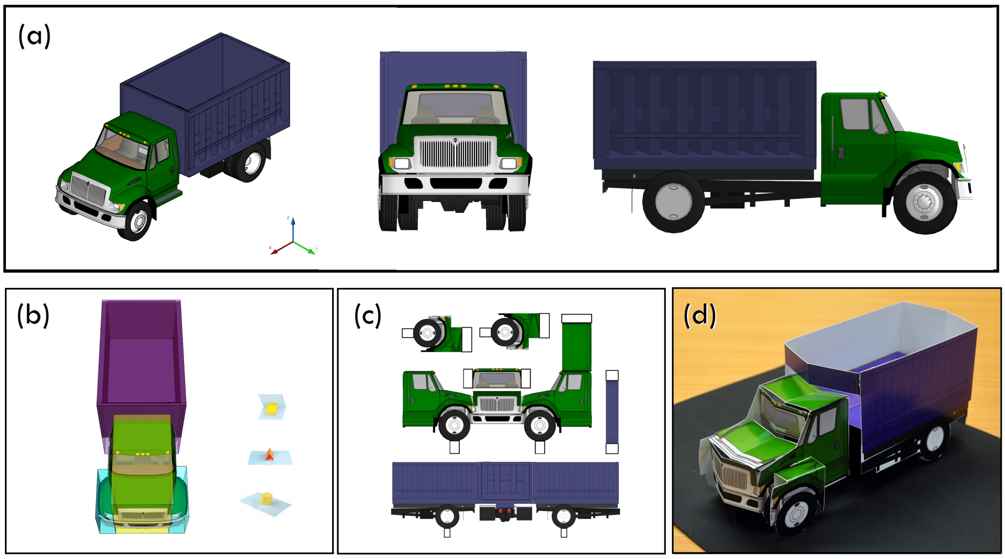


Figure 8.1: (a) Input 3D model - Truck, (b) 3D Primitive Fitting, (c) 2D Printable Pop-up Design Layout and (d) Actual Pop-up

Figure 8.1 shows our resulting pop-up for an input 3D mesh of a dump truck. The primitive fitting step fits four rectangular prisms to the mesh. The rectangular prisms that envelop the tractor and semi-trailer are mapped to box-fold. The primitives that encompass the fender and front wheels are later mapped to step-folds. Notice that although we use rectangular primitives, the wheels still maintain their round shape because of our image-based approach for generating the patches in the final pop-up layout design.

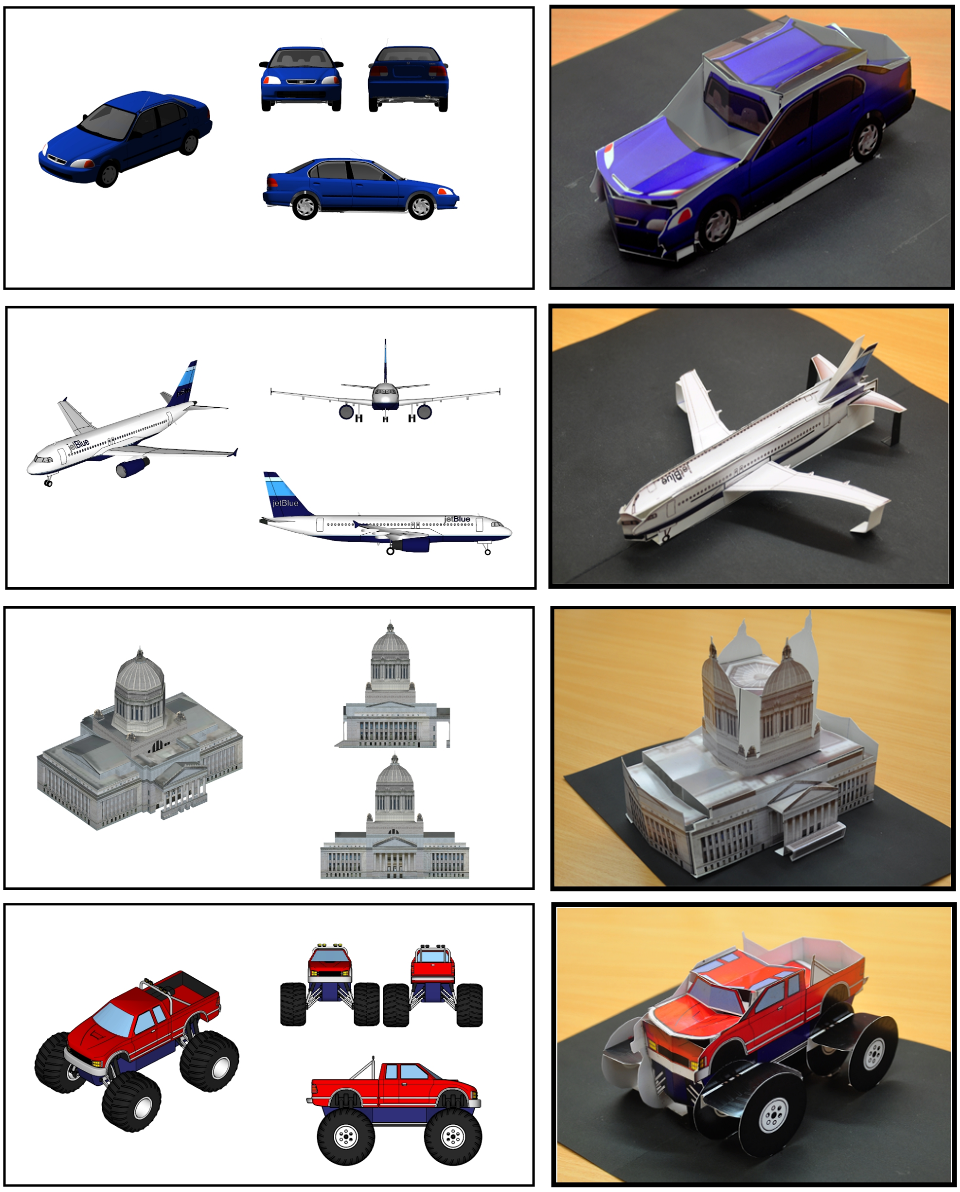


Figure 8.2: Approximating 3D shape results. Input models (left) and their corresponding actual pop-ups (right).

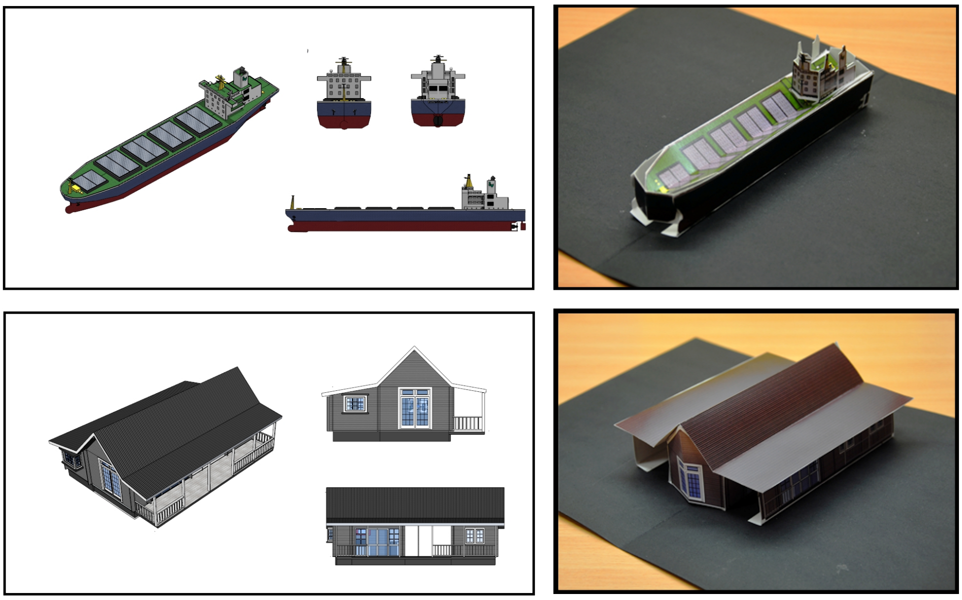


Figure 8.3: Approximating 3D shape results continued. Input models (left) and their corresponding actual pop-ups (right).

Our approach works well for block-like models, such as man-made objects (e.g. cars, trucks, buildings), since they can be closely approximated using our set of primitives. Although our primitives are symmetric, the combinations of these primitives can be asymmetric structures.

Table 8.1 shows the volume difference and Hausdorff distance of the input mesh and the fitted primitives used to generate the pop-up designs. Hausdorff distance is the maximum distance of a set to the nearest point in the other set [HK90]. Formally, Hausdorff distance from set A to set B is a maximin function, defined as

$$H(A,B) = \max(h(a,B), h(B,A)), \tag{8.1}$$

where

$$h(A,B) = \max_{a \in A} \min_{b \in B} \left\| a - b \right\|, \tag{8.2}$$

where a and b are points of sets A and B, respectively. We use Meshlab to sample the mesh using 53206 samples. Notice that blockier models, the cargo ship and the Capitol building, have the least volume difference. The airplane model and the house model have bigger volume disparity due to the wing and the veranda that are not enclosed portions of the mesh that eventually get fitted to rectangular primitives. The Hausdorff distance however is consistently low across the different type of models. This means that the surface of the input mesh is always near a face of our primitives. Note that all of the models reported a min Hausdorff distance of 0.0, meaning that at least one sample is at the same location on the mesh and on the primitive.

Input 3D Model	Hausdorff Distance				Vol. Diff.
	min	max	mean	RMS	
Car	0.0	0.082687	0.023096	0.030612	0.206570
Airplane	0.0	0.153234	0.022523	0.035811	0.388602
Capitol Building	0.0	0.149990	0.034826	0.047120	0.188612
Monster Truck	0.0	0.081734	0.018348	0.024189	0.2709481
Cargo Ship	0.0	0.085651	0.012802	0.020739	0.053275
House	0.0	0.067794	0.014370	0.020620	0.550768

Table 8.1: Deviations from the input surfaces. Smaller value means better approximation.

Our approach has difficulty in approximating curved and rounded parts, since we only have rectangular and triangular prisms as primitives. Professional artists are able to approximate these shapes in their work using very specialized rounded or curved mechanisms (Figure 8.4 shows a sphere-like structure used to represent the carriage). These specific mechanisms however are not included because their formulations would require a different definition of stability, under the assumption of non-rigid and bendable paper.

Organic and rounded models also may require higher levels of abstraction (see Figure 8.5). Objects that are difficult to align to the principal axes may also

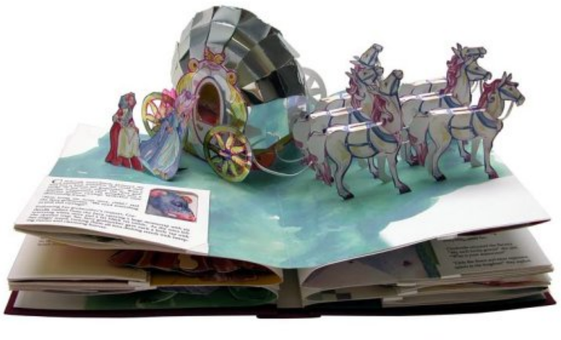


Figure 8.4: Cinderella: A Pop-Up Fairy Tale by Matthew Reinhart [Rei05].

be hard to approximate accurately. In some cases, like the T-shape, necessary additional supports may also distract from the original shape.

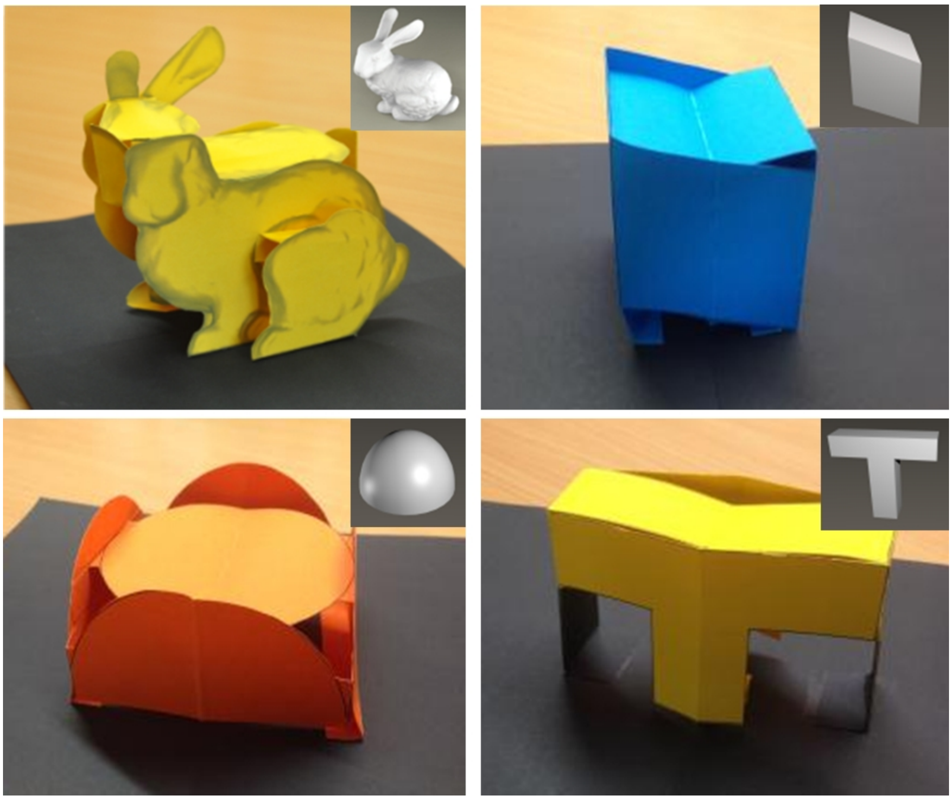


Figure 8.5: Our actual paper pop-ups for Stanford Bunny (textured using the rendered model), skewed cube, half-sphere and T-shape.

We compare our results with those of [LSH^{+}10] and [LJGH11] in Figure 8.6. Because of the voxelization, their results contain many small box structures, which are difficult to make in practice. Our pop-ups consist of only two box-folds for the Waldorf-Astoria Hotel and a single tent-fold for the Eiffel Tower. Although our geometry is not as detailed, we are able to use textures to capture the details and shape information. Projecting textures onto patches that have been generated using voxelization is not as straightforward since they introduce changes to the shape of projections from certain viewpoints.

Most of our pop-up designs employ only a few mechanisms and as such only a few patches. This is intentional since our main goal is to generate the simplest design that is still able to capture the general shape of the input 3D model. We have determined that in most cases capturing the side contours of a 3D model is enough to give us a good abstraction of the model.

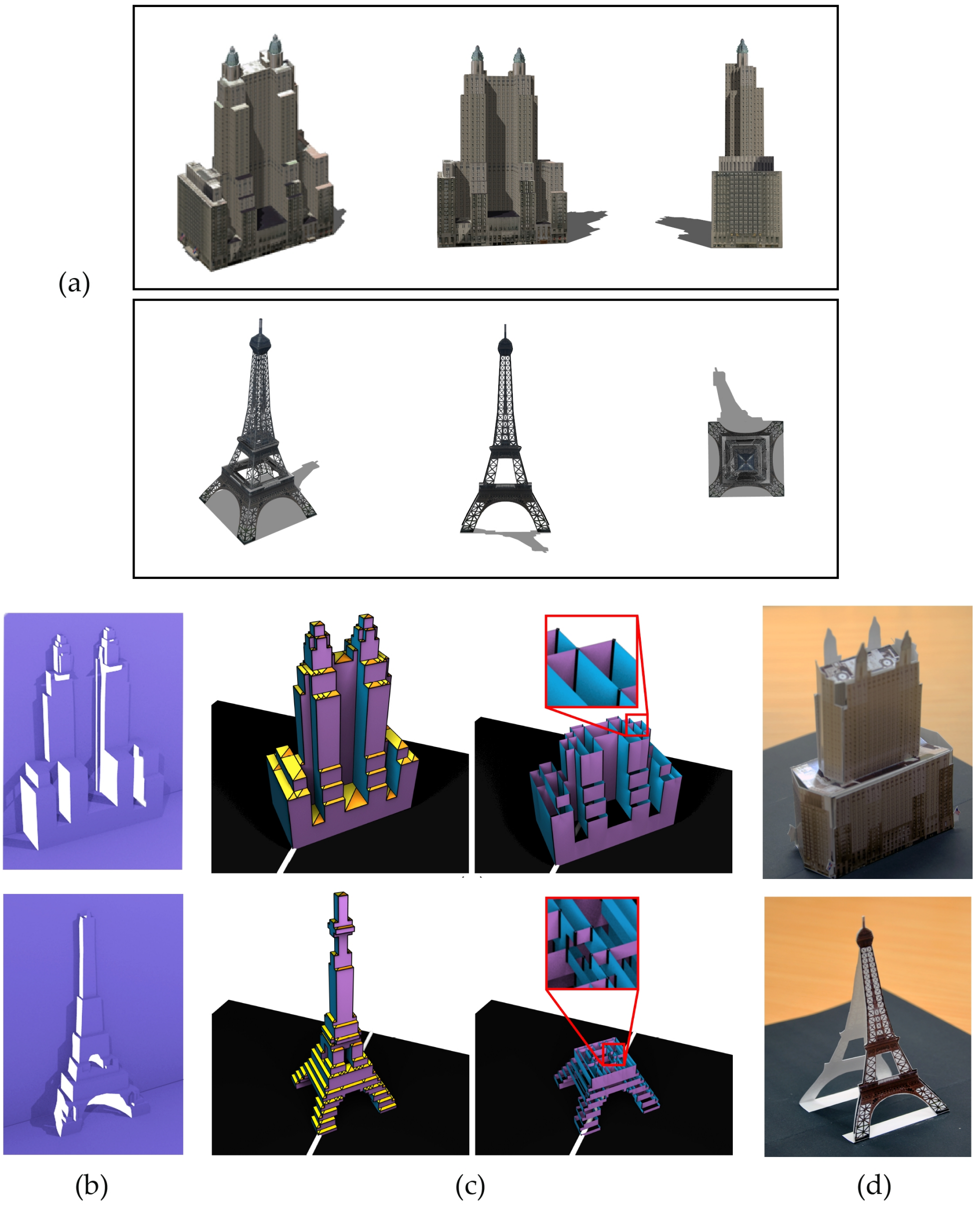


Figure 8.6: Waldorf-Astoria Hotel and Eiffel Tower models. (a) Input 3D model, (b) [LSH+10] results, (c) [LJGH11] results (from paper) and (d) our actual paper pop-ups.

8.2 Approximating Motion

We show some of the physical paper pop-ups created using our generated design in Figure 8.7, 8.8 and 8.9. The input 3D models are from Blender Swap - http://www.blendswap.com. The girl model modified from L. Kaplinski, the tree model is by E. James, and the monkey by J. Newnham.

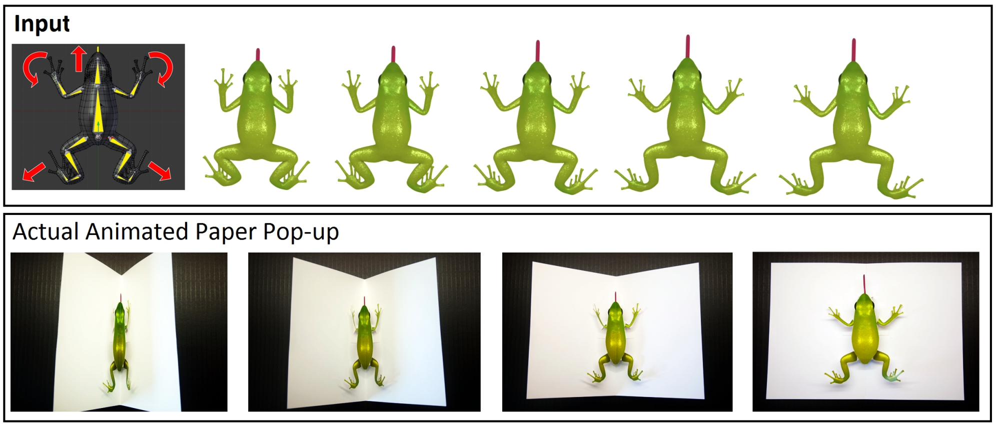


Figure 8.7: (Top) Input articulated 3D model of a frog with motion, rotating arms, moving legs and tongue (Bottom) Actual paper pop-up created using the layout design generated by the system.

In Figure 8.8(a), we illustrate how that system can generate a combination of two motions (rotation and vertical translation) in one character. It also shows that our motion does not need to be symmetric.

Figure 8.8(b) shows that our approach can reproduce common motion used for humanoids such as walking. In this example however, we notice that only one arm and one leg is animated in the pop-up. This is because the conflict between the left arm's and right arm's mechanisms cannot be resolved. As such the right arm's motion was disregarded (the chain was mapped to a non-moving mechanism). The weight of the contribution of the left arm to the overall cost is larger because more vertices are visible from the viewpoint. As such, its original mechanism mapping was retained.

Figure 8.8(c) shows the animation of a pony galloping. Notice that the two front

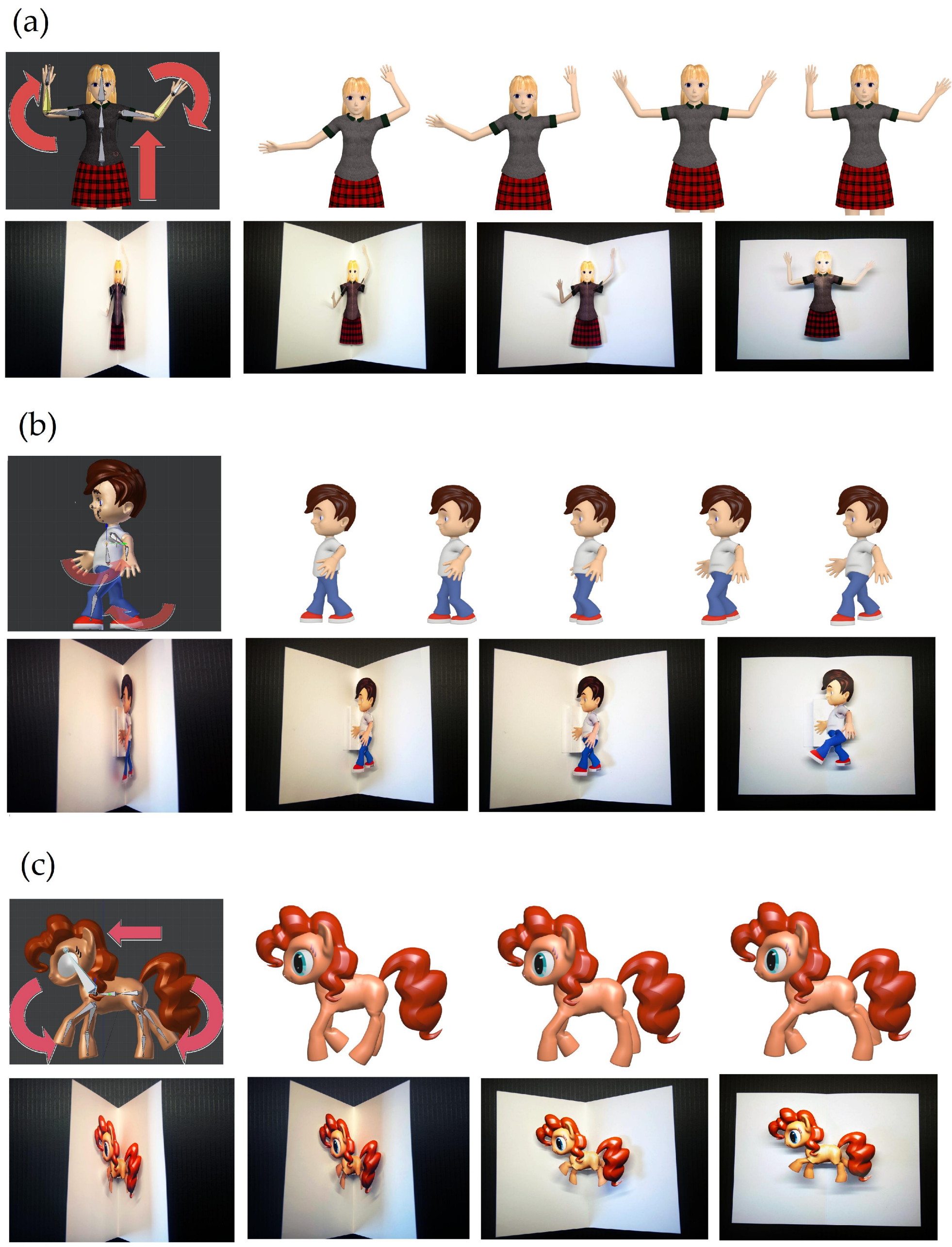


Figure 8.8: Approximating motion results (a) Girl with hands waving and torso moving up, (b) boy walking, (c) pony galloping.

legs are merged as well as the two hind legs. Since the front legs' mechanisms intersect, they were resolved by being merged into one patch, as they have relatively similar direction of movement. Figure 8.9(d) shows a shark opening

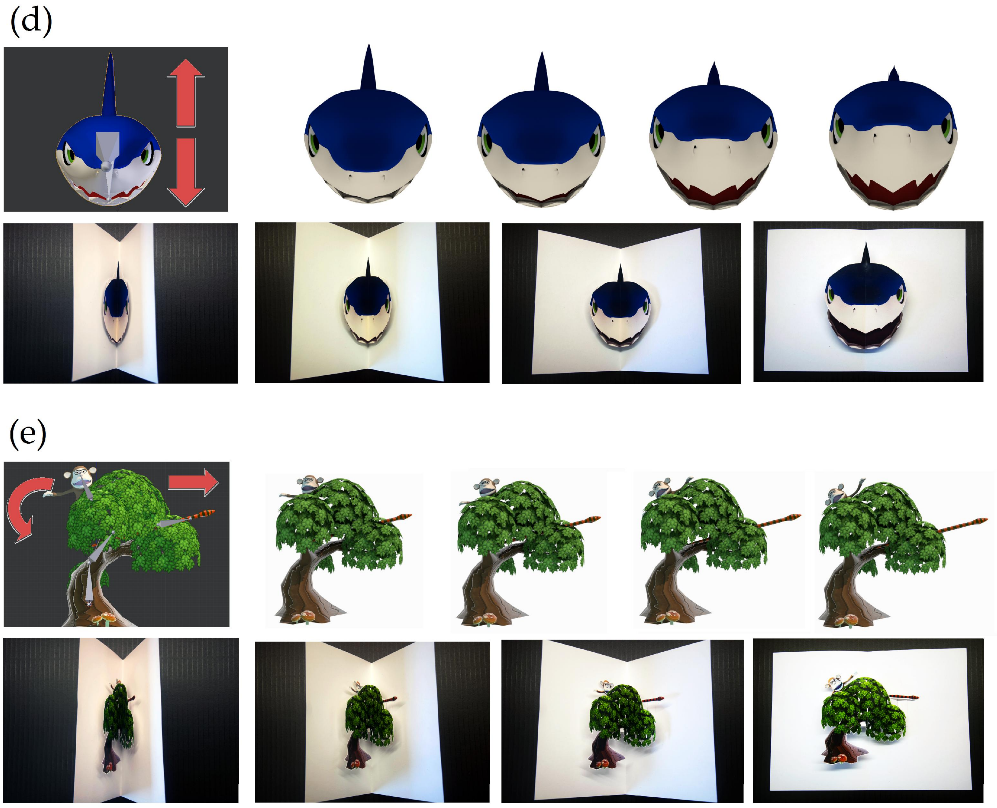


Figure 8.9: Approximating motion results continued. (d) shark opening its mouth and (e) a scene with monkey and snake in a tree.

its mouth using vertical movement for its jaws. We can also handle multiple characters, such as in Figure 8.9(e). Here, we use a linkage for every character, one for the monkey with rotating motion, one for the snake with horizontal motion and the tree is stationary.

Using the same machine described in the previous section, the plug-in runs for a few seconds depending if the collisions can be easily resolved. For example, the frog layout (Figure 8.7) takes a few seconds to generate, while the boy walking animation (Figure 8.8(b)) takes longer since the collisions cannot be resolved except by remapping some parts to floating layers. This produces a high discrepancy in the motion that leads to the algorithm reaching the maximum number of iterations. Note that the rendering time of mesh using the specified material and lighting in the file will also affect the overall running time.

Table 8.2 shows the motion discrepancy from sampled linkage motion paths and the corresponding mechanisms in the actual pop-up. It is based on the error metric in Equation 6.6. The animations of the girl waving her arms and the pony galloping have the lowest difference. This is due to the fact the rotary motion of the arms and legs can be adequately approximated by our rotation pop-up mechanism.

Input 3D Articulated Figure	Sampled Motion Difference (per linkage chain)				Links
	min	max	mean	weighted	
Girl	0.023600	0.054900	0.039331	0.039348	3
Boy	0.000098	1.000000	0.513749	0.221999	4
Pony	0.016900	0.051400	0.031100	0.032800	4
Shark	0.030287	0.139804	0.085046	0.095990	2
Tree	0.000200	0.140935	0.070568	0.063531	2
Frog	0.024400	0.173053	0.122021	0.113061	5

Table 8.2: Motion Fidelity of the input 3D articulated figure and output animated pop-up. Smaller value means better approximation.

The highest difference can be seen in the walking boy animation. This is because an arm and a leg are mapped to relatively stationary mechanisms due to unresolvable collisions. The shark animation only has linear motion but notice that it still incurs some error. This is due to the fact that pop-up motion can only have a constant speed, if there is any acceleration or deceleration in the input motion, it cannot be exactly reproduced. Nonetheless, the overall perception of the motion is not greatly influenced.

The range of motion allowed within the geometric constraints of paper pop-ups is limited. As such, we usually limit ourselves to a few frames of the animation. Furthermore, the possible types of motions produced by pop-ups are also limited. For example, in Figure 8.10(a), the girl that moves her hand in an S path might only be mapped to a simple rotation. In Figure 8.10(b), the boy that opens his arms may only be mapped to a horizontal motion, losing some of the intended movements. However, we believe that, as pop-up artists discover more elaborate mechanisms, these can be eventually added to the system to handle more complex animations.

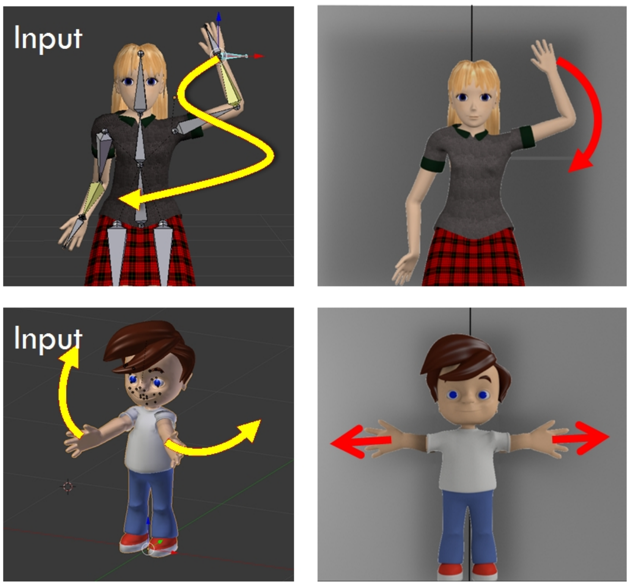


Figure 8.10: Examples of a complicated motion path and 3D motion and the rendered pop-up.

Chapter 9

Conclusion

In this dissertation, we have presented the craft and mathematics of paper pop-ups. We have discussed the current work on computational paper pop-ups and identified gaps in the research area. All of the previous automated approaches focused on a very specific style of paper pop-up design and three-dimensionality. We explored challenges in combining multiple styles and recreating motion.

We have presented our approach for generating multi-style pop-up designs automatically from 3D models. We have also investigated the new problem of automatically designing animated paper pop-ups from the motion of 3D articulated figures. We have studied the geometric properties of some pop-up mechanisms, specifically focusing on their output motions and geometric conditions for validity. We have also presented a pipeline to automate the entire process.

Although the methods for approximating 3D shape and motion differ in their specific implementations, they share a common framework. Figure 9.1 shows the unified framework for automated pop-up design. Both start with some form of segmentation or abstraction. After which in both approaches, the different parts are mapped to a pop-up mechanism.

Then some type of optimization is employed based on some error metric to gauge the quality of the approximation. The patch generation is done similarly for 3D shape and motion. Utilizing an image-based approach for generating the texture patches and unfolding the 3D structure of the pop-up mechanism to create the

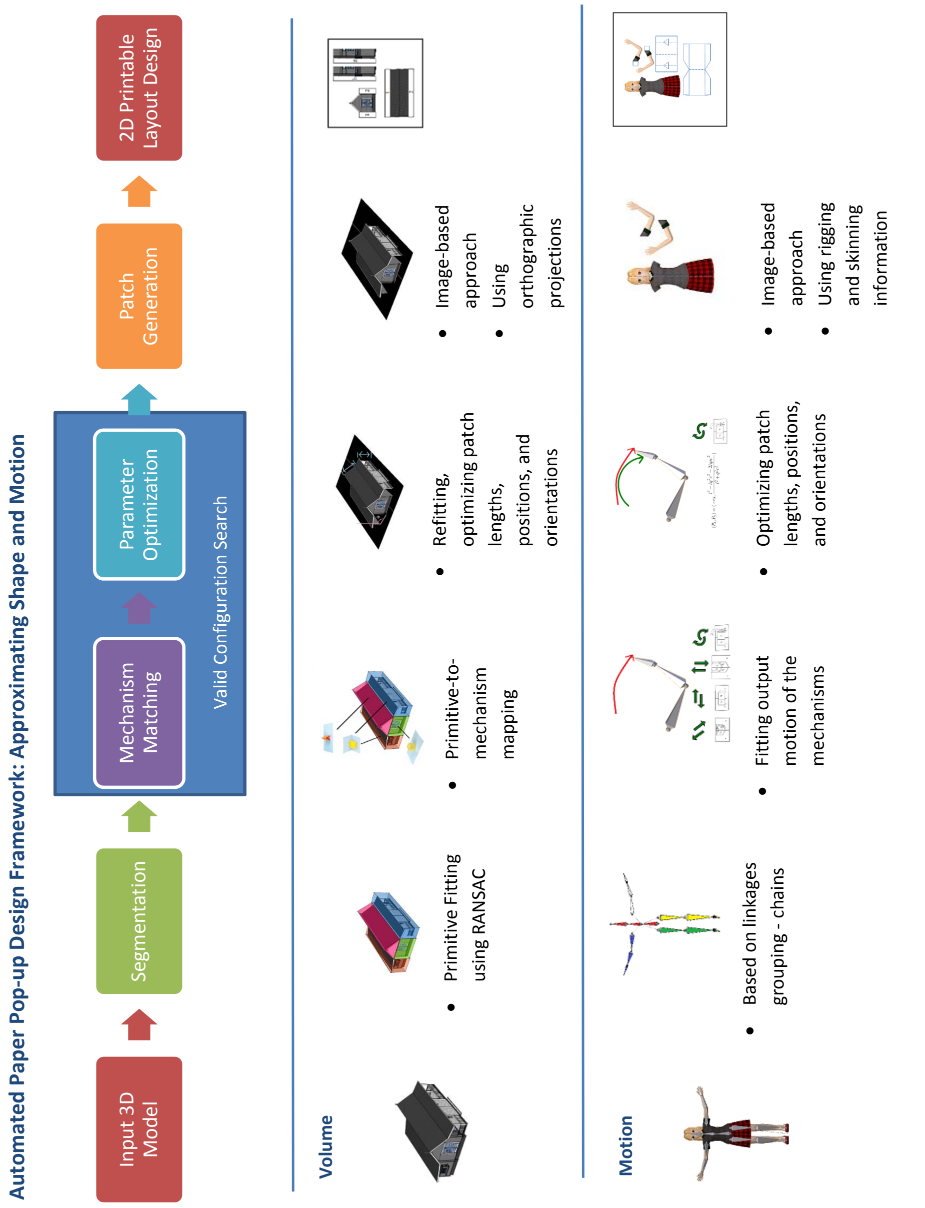

Automated Paper Pop-up Design Framework: Approximating Shape and Motion
Input 3D Model
Segmentation
Mechanism Matching
Parameter Optimization
Valid Configuration Search
Patch Generation
2D Printable Layout Design
Volume
Primitive Fitting using RANSAC
Primitive-to-mechanism mapping
Refitting, optimizing patch lengths, positions, and orientations
Image-based approach
Using orthographic projections
Motion
Based on linkages grouping - chains
Fitting output motion of the mechanisms
Optimizing patch lengths, positions, and orientations
Image-based approach
Using rigging and skinning information

Figure 9.1: Unified framework for computation paper pop-ups.

2D design layout. These specific implementations may be replaced with more complicated techniques in the future, such as better mesh abstraction, machine learning for motion matching, etc.

Recently, there has been much interest in the graphics community in the physical fabrication of 3D models. Our work could also be seen as such, without the need for specialized 3D printing equipment. Paper pop-ups also have the advantage of being economical and portable.

9.1 Contributions

One of the main contributions of this thesis is the novelty of the research problem itself. There are only a handful of automated methods and these are for very specific paper pop-up styles. This is also the first work to focus on generating animated paper pop-up designs.

Although, some of the individual steps (e.g. primitive fitting, simulated annealing) are well-known graphics or computer science techniques, but their combination and their use in similar context, with explicit consideration of pop-up validity, have not been done in previous work. We believe our present approach provides a novel and reasonable framework for combining multiple pop-up mechanisms and reproducing motion.

The specific contributions of this thesis are:

- A unified framework for automated pop-up design approximating shape and motion.
- A set of geometric constraints for individually valid mechanisms and the combinations thereof.
- An algorithm for automatic pop-up design generation using mesh abstraction by fitting primitives in order to incorporate multiple pop-up styles.

- A geometric study of the motion of pop-up mechanisms.
- A method for the automated design of animated paper pop-ups from the motion of articulated characters.
- A hybrid geometry- and image-based approach in generating the pop-up patches to preserve important texture and shape contours of the input model.

9.2 Future Work

1. **Interactivity**. Currently, the entire process is automated. For novice pop-up designers, this is favorable because it requires little skill and effort on their part. However, more expert users may prefer more control over the final design of the pop-up. As such, in the future we recommend incorporating the approach with a more interactive system. This will allow the user to alter some parts of the pop-ups according to their own specifications.
2. **Qualitative Measure**. A metric to measure the aesthetic quality of the pop-up can be very beneficial in creating more visually appealing paper pop-up designs. In our work, we have tried to minimize the volume, shape and motion difference of the input and the output pop-up. But actual pop-ups designed by artists are not necessary faithfully reproduction of a 3D object, but only its important and more salient features. Research in psychology and perception may also help in the automated design of pop-ups.
3. **More Complex Pop-up Mechanisms**. Our approach for approximating 3D shape has difficulty in approximating curved and rounded parts, since we only have rectangular and triangular prisms as primitives. Professional artists are able to approximate these shapes in their work using very

specialized rounded or curved mechanisms. These specific mechanisms however are not included because their formulations would require a different definition of stability, under the assumption of non-rigid and bendable paper.

For approximating motion, we only utilize basic pop-up mechanisms to generate 2D motions. Nonetheless, we believe that our framework is robust and flexible enough to incorporate more complex and 3D mechanisms in the future. We also do not consider other mechanical mechanisms, like strings and pull-out tabs, that may be interesting to consider in future work.

4. **Assembility**. There are some styles of paper pop-up designs that may encounter the problem of physically unrealizable designs, such as the lattice-style or sliceform pop-up. But for the general class of pop-ups, most of the concern is on the difficulty to construct pop-up books. It would be beneficial for book publishers to have a metric for measuring the difficulty to manufacture pop-up books. This metric could also be part of the error metric that we optimize along with the metric for shape and motion similarity.

5. **Physical Properties of Paper**. The presented geometric formulations here do not take into account the physical characteristics of paper. In actual pop-up design, the thickness, mass, strength and elasticity of paper are important considerations that paper engineers have to consider in their pop-up structures.

6. **Foldable Objects**. Similar to algorithms developed initially for origami that have eventually found applications in other fields. We hope that work in automated pop-up design could eventually be used for research pertaining to other foldable objects, such as foldable furniture. One example is the work of [LHAZ15]. Figure 9.2 shows an example of a seat made from a single sheet of plywood designed by Ufuk Keskin and Efecem Kutuk [Mic14], which is share similar properties with pop-up structures.

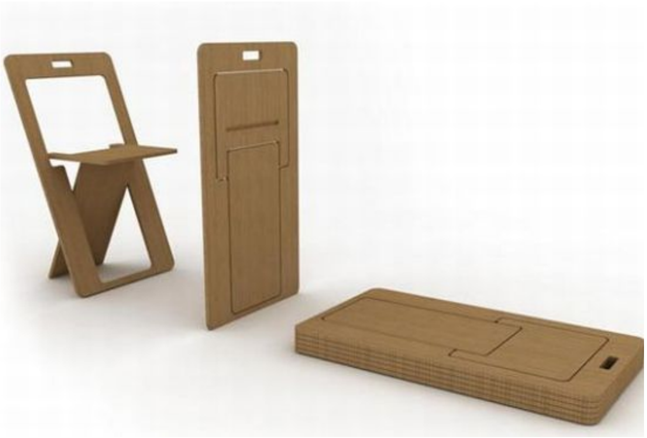


Figure 9.2: SheetSeat: a flat folding chair [Mic14].

Our preliminary work is just the first in this relatively new research area. As such, our main goal is to formally define the problem, provide a suitable framework and hopefully inspire other related work in papercraft research, as well as articulated structure design.

References

[ADD$^+$13] Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Anna Lubiw, André Schulz, Diane L. Souvaine, Giovanni Viglietta, and Andrew Winslow. Algorithms for designing pop-up cards. In *STACS*, volume 20 of *LIPIcs*, pages 269–280, 2013.

[APH$^+$03] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):828–837, July 2003.

[BH02] Sarah-Marie Belcastro and Thomas C. Hull. Modelling the folding of paper into three dimensions using affine transformations. *Linear Algebra and its Applications*, 348:273–282, 2002.

[Bir11] Duncan Birmingham. *Pop-Up Design and Paper Mechanics: How to Make Folding Paper Sculpture.* Guild of Master Craftsman, 2011.

[Can12] Canon. Paper craft - canon creative park website, 2012.

[Car08] David A. Carter. *Yellow Square: A Pop-Up Book for Children of All Ages.* Little Simon, 2008.

[CD99] David A. Carter and James Diaz. *Elements of Pop-Up.* Little Simon, 1999.

[Che05] Ron Chespak. Ron chespak paper sculpture. *The Computer, ChespakGroup*, 2005.

[CLM$^+$13] Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. Designing and fabricating mechanical automata from mocap sequences. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, 32(6):186:1–186:11, November 2013.

[COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 115–122, 1998.

[Cro11] Robert Crowther. *Amazing Pop-Up Monster Trucks.* Walker & Company, 2011.

[CS03] Lewis Carroll and Robert Sabuda. *Alice's Adventures in Wonderland: A Pop-up Adaptation.* Little Simon, 2003.

[CTN$^+$13] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. Computational design of mechanical characters. *ACM Transactions Graphics*, 32(4):83:1–83:12, July 2013.

[CZ06] Jyun-ming Chen and Yu-zhi Zhang. A computer-aided design system for origamic architecture. In *International Conference on Supercomputing*, 2006.

[DDSD03] Xavier Décoret, Frédo Durand, François X. Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, pages 689–696, 2003.

[Des] Lundstrom Design. Touch-3d [software].

[dF95] Luiz Henrique de Figueiredo. Adaptive sampling of parametric curves. *Graphics Gems V*, pages 173–178, 1995.

[DO07] Erik Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge Uni. Press, 2007.

[DSSD99] Xavier Décoret, Françis Sillion, Gernot Schaufler, and Julie Dorsey. Multi-layered impostors for accelerated rendering. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 1999*, 18(3):61–73, 1999.

[EPD09] Elmar Eisemann, Sylvain Paris, and Frédo Durand. A visibility algorithm for converting 3d meshes into editable 2d vector graphics. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009)*, SIGGRAPH '09, pages 83:1–83:8, 2009.

[GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, 1997.

[Gla02a] Andrew Glassner. Interactive pop-up card design, part 1. *IEEE Computer Graphics and Applications*, 22(1):79–86, 2002.

[Gla02b] Andrew Glassner. Interactive pop-up card design, part 2. *IEEE Computer Graphics and Applications*, 22(2):74–85, 2002.

[Goo13] Google. http://sketchup.google.com/3dwarehouse, 2013.

[GS09] Marivi Garrido and Ingrid Siliakus. *The Paper Architect: Fold-It-Yourself Buildings and Structures.* Potter Craft, 2009.

[HE06] S. L. Hendrix and M. A. Eisenberg. Computer-assisted pop-up design for children: computationally enriched paper engineering. *Adv. Technol. Learn.*, 3(2):119–127, 2006.

[HEH05] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)*, pages 577–584, 2005.

[HEH08] Derek Hoiem, Alexei A Efros, and Martial Hebert. Closing the loop in scene interpretation. In *IEEE CVPR*, 2008.

[Hen08] Susan Hendrix. *Popup Workshop: Computationally Enhanced Paper Engineering for Children.* PhD thesis, University of Colorado, 2008.

[Hin86] Mark Hiner. *Paper Engineering for Pop-Up Books and Cards.* Parkwest Pubns, 1986.

[Hin02] Mark Hiner. A short history of pop-ups - paper engineer. url:http://www.markhiner.co.uk/history-text.htm, 2002.

[HK90] Daniel P. Huttenlocher and Klara Kedem. Computing the minimum hausdorff distance for point sets under translation. In *Proceedings of the Sixth Annual Symposium on Computational Geometry*, SCG '90, pages 340–349, New York, NY, USA, 1990. ACM.

[HS09] Takuya Hara and Kokichi Sugihara. Computer-aided design of pop-up books with two- dimensional v-fold structures. In *Proc. 7th Japan Conf. on Computer Geometry and Graphs*, 2009.

[Hul06] Thomas Hull. *Project Origami: Activities for Exploring Mathematics.* A K Peters, Ltd, 2006.

[IEM$^+$11] Satoshi Iizuka, Yuki Endo, Jun Mitani, Yoshihiro Kanamori, and Yukio Fukui. An interactive design system for pop-up cards with a physical simulation. *Visual Computer*, 27(6):605–612, June 2011.

[Jac93] Paul Jackson. *The Pop-Up Book: Step-by-Step Instructions for Creating Over 100 Original Paper Projects.* Holt Paperbacks, 1993.

[Jac96] P. Jackson. *The Art and Craft of Paper Sculpture.* Apple Press, London., 1996.

[KGBS11] Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. Physics-inspired upsampling for cloth simulation in games.

ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011), 30(4):93:1–93:10, July 2011.

[Leo10] Su-Jun Leow. Automatic paper pop-up design. Master's thesis, National University of Singapore, 2010.

[LHAZ15] Honghua Li, Ruizhen Hu, Ibraheem Alhashim, and Hao Zhang. Foldabilizing furniture. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015)*, 34(4):90:1–90:12, July 2015.

[LJGH11] Xian-Ying Li, Tao Ju, Yan Gu, and Shi-Min Hu. A geometric study of v-style pop-ups: Theories and algorithms. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, 30(4):98:1–98:10, July 2011.

[LLLN$^+$14] Sang N. Le, Su-Jun Leow, Tuong-Vu Le-Nguyen, Conrado Jr. Ruiz, and Kok-Lim Low. Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics*, 20(2):276–288, February 2014.

[LNLRL13] Tuong-Vu Le-Nguyen, Kok-Lim Low, Conrado Jr. Ruiz, and Sang N. Le. Automatic paper sliceform design from 3d solid models. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1795–1807, November 2013.

[LSH$^+$10] Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju, and Shi-Min Hu. Popup: Automatic paper architectures from 3d models. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, 29(4):111:1–111:9, July 2010.

[LTS96] Y.T. Lee, S.B. Tor, and E.L. Soo. Mathematical modelling and simulation of pop-up books. *Computer & Graphics*, 20(1):21–31, 1996.

[LYLC14a] Chen Liu, Yong-Liang Yang, Ya-Hsuan Lee, and Hung-Kuo Chu. Image-based paper pop-up design. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 36:1–36:1, New York, NY, USA, 2014. ACM.

[LYLC14b] Chen Liu, Yong-Liang Yang, Ya-Hsuan Lee, and Hung-Kuo Chu. Image-based paper pop-up design. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 36:1–36:1, New York, NY, USA, 2014. ACM.

[LYMS07] Yan Li, Jinhui Yu, Kwan-liu Ma, and Jiaoying Shi. 3d paper-cut modeling and animation. *Computer Animation and Virtual Worlds*, 18(4-5):395–403, September 2007.

[Meg79] L. Meggendorfer. *International Circus*. The Metropolitan Museum of Art, 1979. Reproduction of the 1887 original.

[Mic14] C. Michaels. Sheetseat : The super flat folding chair, 2014.

[MS04a] Jun Mitani and Hiromasa Suzuki. Computer aided design for origamic architecture models with polygonal representation. In *Proceedings of the Computer Graphics International*, pages 93–99, 2004.

[MS04b] Jun Mitani and Hiromasa Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):259–263, 2004.

[MSM11] James McCrae, Karan Singh, and Niloy J. Mitra. Slices: A shape-proxy based on planar sections. *ACM Transaction Graphics (Proceedings of SIGGRAPH Asia 2011).*, 30(6):168:1–168:12, December 2011.

[MSU03] Jun Mitani, Hiromasa Suzuki, and Hiroshi Uno. Computer aided design for origamic architecture models with voxel data structure. *Transactions of Inf. Process. Soc. of Japan*, 44(5):1372–1379, 2003.

[oC15] "Library of Congress". World treasures: Beginnings, 2015.

[O'R11] Joseph O'Rourke. *How to Fold It: The Mathematics of Linkages, Origami & Polyhedra.* Cambridge Uni. Press, 2011.

[Rei05] Matthew Reinhart. *Cinderella: A Pop-Up Fairy Tale.* Little Simon, 2005.

[Rei07] Matthew Reinhart. *Star Wars: A Pop-Up Guide to the Galaxy.* Orchard Books, 2007.

[Rei13] Matthew Reinhart. *Transformers: The Ultimate Pop-Up Universe.* LB Kids, 2013.

[Rei14] Matthew Reinhart. *Game of Thrones: A Pop-Up Guide to Westeros.* Insight Editions, 2014.

[Rub13] E. Rubin. A timeline history of movable books, 2013.

[Sch97] J. Schreiber. *Bubenstreiche: Ein Verwandlungs-Bilderbuch [Schoolboy Pranks: A Transformation Picture Book].* Esslingen, 1997.

[SDB97] François Sillion, George Drettakis, and Benoit Bodelet. Efficient impostor manipulation for real-time visualization of urban scenery. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 1997)*, 16(3):C207–C218, 1997.

[SLCH11] Jau-Ling Shih, Chang-Hsing Lee, and Chuang Chao-Hung. A 3d model retrieval approach based on the combination of pca plane projections. *Journal of Information Technology and Applications*, 5(2), 2011.

[STL06] Idan Shatz, Ayellet Tal, and George Leifman. Paper craft models from meshes. *Visual Computer*, 22(9):825–834, September 2006.

[SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[Tac10] Tomohiro Tachi. Origamizing polyhedral surfaces. *IEEE Transactions Visualization and Computer Graphics*, 16(2):298–311, 2010.

[Tam07] TamaSoftware. Pepakura designer.
http://www.tamasoft.co.jp/pepakura-en/, 2007.

[TCG$^+$14] Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. Computational design of linkage-based characters. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)*, 33(4):64:1–64:9, July 2014.

[UT06] Ryuhei Uehara and Sachio Teramoto. The complexity of a pop-up book. In *18th Canadian Conference on Computer Geometry*, 2006.

[Wen10] Mak-Kin Weng. *Analysis, Modelling & Simulation of Pop-up Laminar Structures*. PhD thesis, Nanyang Technological University, Singapore, 2010.

[WL10] Jin Wei and Yu Lou. Feature preserving mesh simplification using feature sensitive metric. *Journal of Computer Science and Technology*, 25(3):595–605, 2010.

[XKM07] Jie Xu, Craig S. Kaplan, and Xiaofeng Mi. Computer-generated papercutting. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 343–350. IEEE Computer Society, 2007.

[YK12] Mehmet Ersin Yumer and Levent Burak Kara. Co-abstraction of shape collections. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)*, 31(6):166:1–166:11, November 2012.

[ZXS$^+$12] Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, and Baining Guo. Motion-guided mechanical toy modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)*, 31(6):127:1–127:10, November 2012.

[ZZ13] Kening Zhu and Shengdong Zhao. Autogami: A low-cost rapid prototyping toolkit for automated movable paper craft. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 661–670, New York, NY, USA, 2013. ACM.

Appendix A. Publications

Ruiz C., Le S.N., Low K.-L. "Generating Animated Paper Pop-ups from the Motion of Articulated Characters." Computer Graphics International 2015 (CGI'15), France (June 2015). *Visual Computer, Vol. 31, Issue 6-8.*

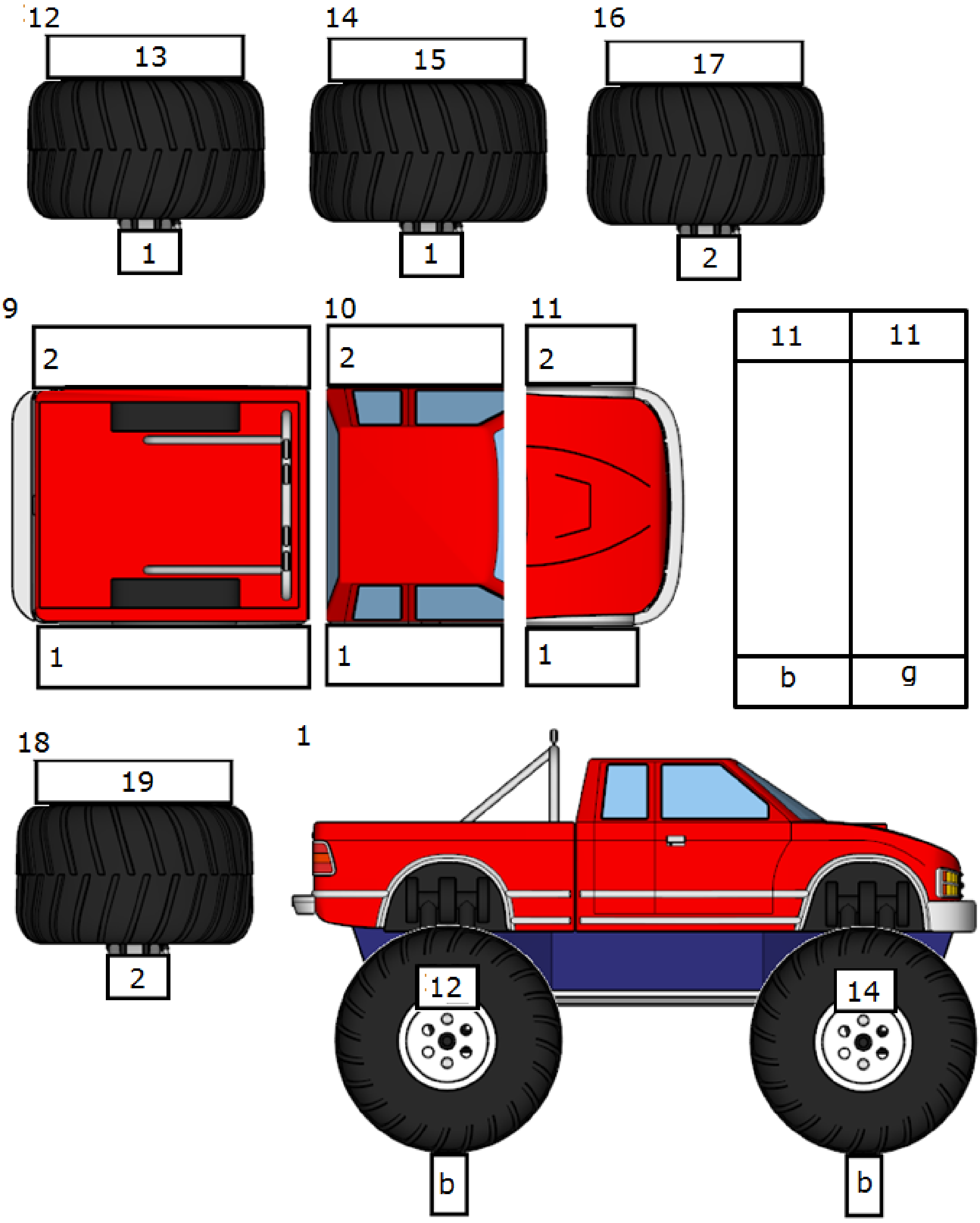
Ruiz C., Le S.N., Low K.-L. "Creating Animated Pop-up Books from the Motion of 3D Articulated Characters." ACM SIGGRAPH Asia 2014 (SA2014) - Poster, China (Dec 2014).

Ruiz C., Le S.N., Yu, J., Low K.-L. "Multi-style Paper Pop-up Designs from 3D Models." EUROGRAPHICS 2014, France (April 2014). *Computer Graphics Forum (CGF). Vol. 3, No. 2.*

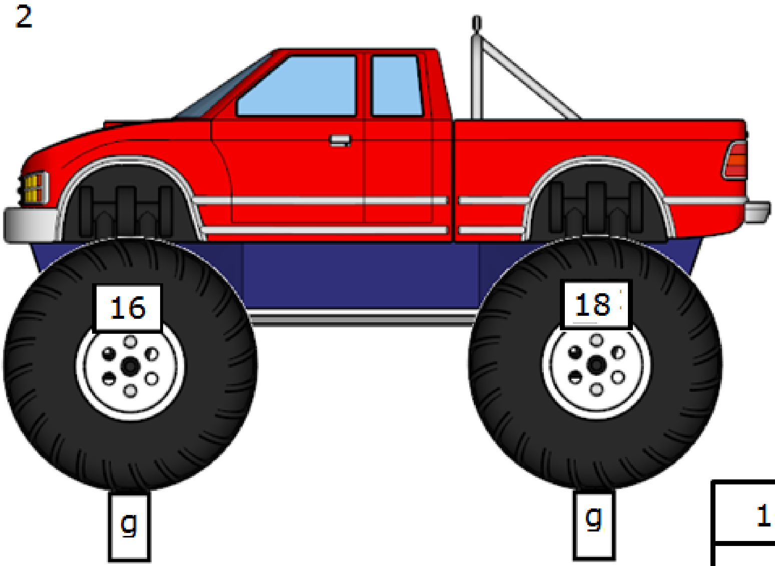
Ruiz C., Le S.N., Low K.-L. "Generating Multi-style Paper Pop-up Designs using 3D Primitive Fitting." ACM SIGGRAPH Asia 2013 (SA2013) - Technical Brief, Hongkong (Nov 2013).

Le S.N., Leow, S.-J., Le-Nguyen, T-V, Ruiz C., Low K.-L. "Surface- and Contour-Preserving Origamic Architecture Paper Pop-Ups." IEEE Transactions on Visualization and Computer Graphics (TVCG) 2013, Vol. 20, No. 2. *Also presented at PACIFIC GRAPHICS 2013.*

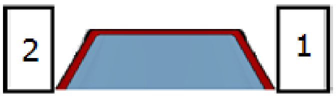
Le-Nguyen, T.-V., Low K.-L, Ruiz C., Le S.N. "Automatic Paper Sliceform Design from 3D Solid Models." IEEE Transactions on Visualization and Computer Graphics (TVCG) 2013, , Vol. 19, No. 4 *Also presented at ACM i3D 2014.*


12
13
1
14
15
1
16
17
2
9
2
1
10
2
1
11
2
1
11
11
b
g
18
19
2
1
12
14
b
b

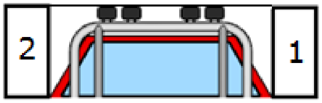
2



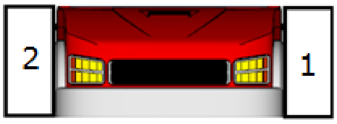
3



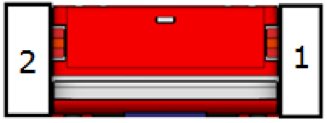
6



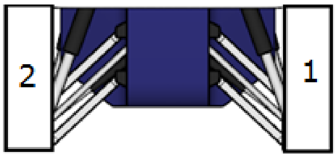
4



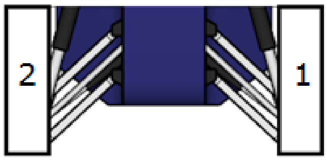
7

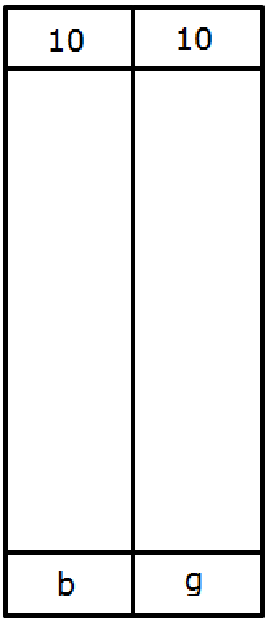


5

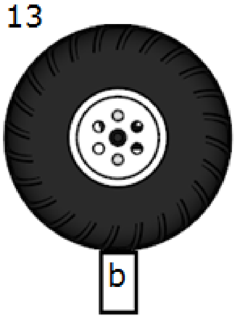


8

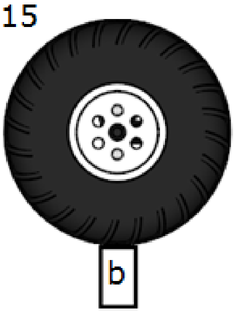




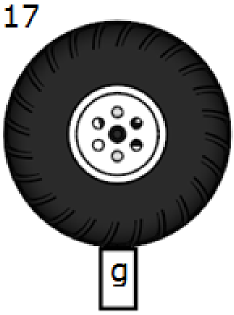
13



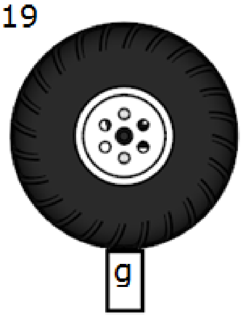
15



17



19



1

2

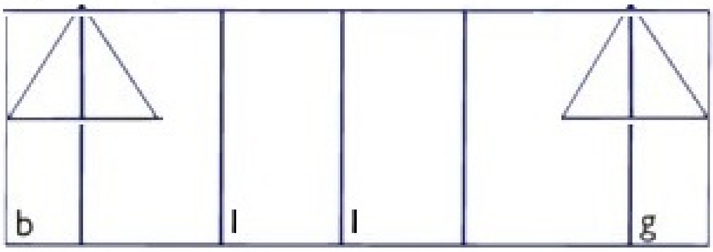
3

4

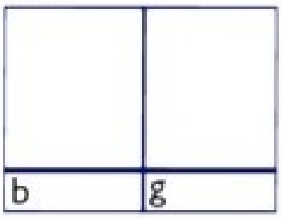
5

6

7



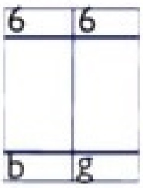
8



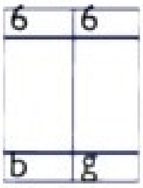
9

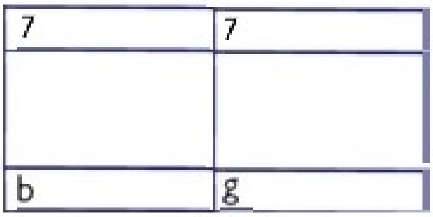
10

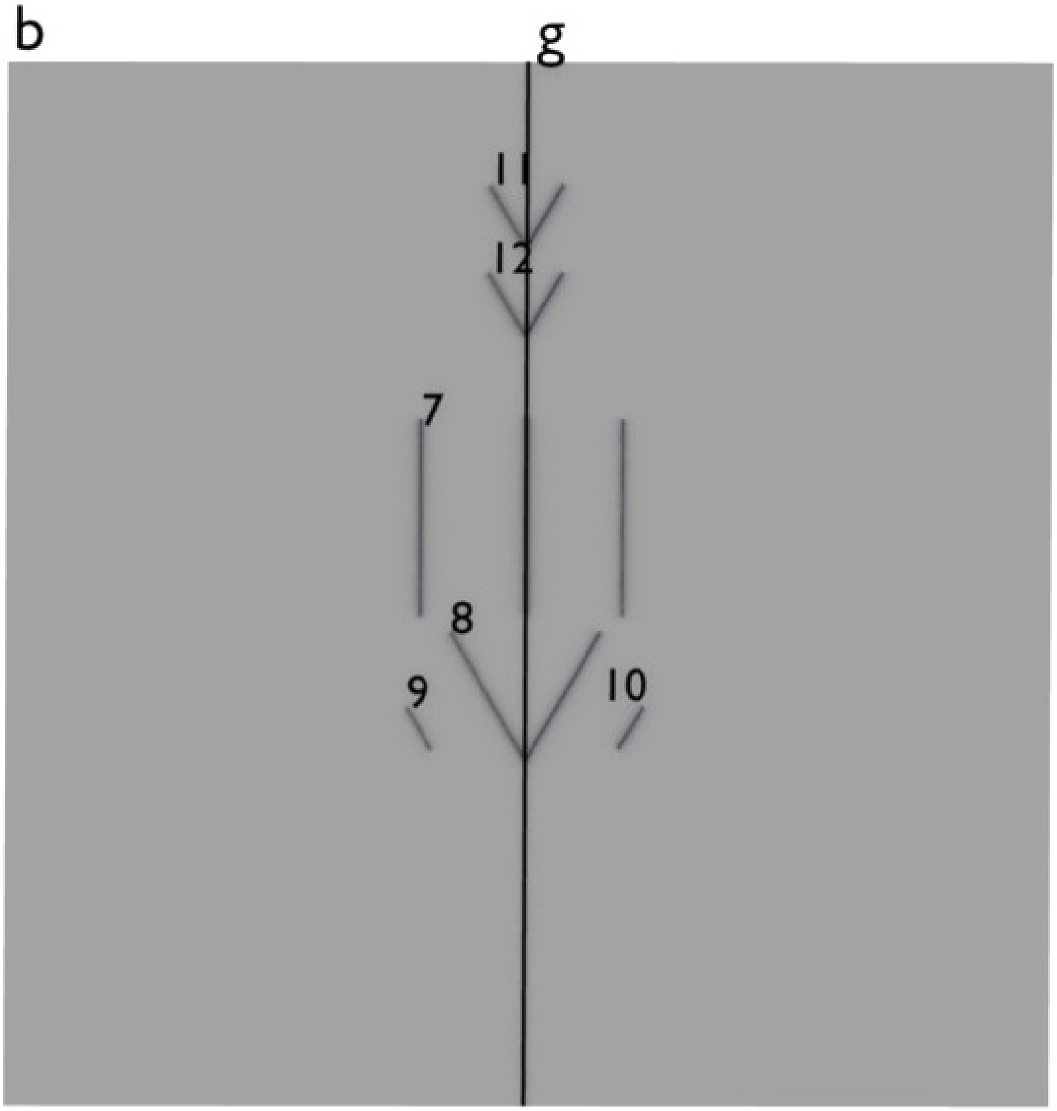
11



12






b
g
11
12
7
8
9
10

Appendix C. Resource Persons and Transcripts

1. Robert Sabuda

Author of *Alice's Adventures in Wonderland* [CS03]

2. Xian-Ying Li

Author of Popup: Automatic Paper Architectures from 3D Models[LSH$^+$10] and A Geometric Study of V-style Pop-ups: Theories and Algorithms[LJGH11]

Email Transcripts with Robert Sabuda

1. Can you briefly discuss how you currently design pop-ups? Do you start from 2D sketches? Do you make 3D mock-ups of the objects you plan to design? (No need to discuss the final physical construction of the book)

RS: I rarely make 2D sketches of my pop-up ideas because there is always the concern that what is created in 2D cannot be made a reality in 3D. So I do usually begin working in 3D as soon as I have a general idea of what I would like the pop-up to be and how the accompanying motions will be incorporated.

2. Do you consider the math or geometry of the paper pop-up during the design process? For example do you try to compute the angles and lengths to check for intersections and problems even before making physical models?

RS: I do not do any math or geometry at all when I'm in the beginning stages of designing in 3D. My years of experience as a paper engineer do allow me to make some cerebral calculations before I even commit to cutting cutting and folding the paper but this is still no guarantee that everything will work out!

2. Do you use any software to help you design the pop-ups? Do you use software to color or paint the patches?

RS: At this time I don't use any software to design the pop-ups but I do, on occasion, digitally color my 2D artwork that appears on the pop-ups.

3. Do you think that a paper engineer would find using computer-aided design tools helpful in designing pop-ups? What functionalities do you think would be helpful for designers such as yourself (e.g. automatic intersection/collision checking)?

RS: I'm afraid I'm probably too old at this stage in my career to really benefit from software programs like this, however I could see how they could be quite useful to younger generations of paper engineers. The reason I say this is because I've always noticed a certain reluctance (or even fear!) in beginning paper engineers regarding having there designs work EXACTLY the way they envision. This rarely happens (sometimes I'll make up to a dozen, physical 3D prototypes before getting one that really works the way I want it to) so I think digital tools might relieve some of that early anxiety.

4. Do you believe that a completely automated approach would be helpful for expert pop-up designers? (An automated approach in our case means that the designer looks for a 3D model from the internet, e.g. Google 3d warehouse or creates a 3d model himself and feeds it to the software, and the software automatically generates the 2D design layout for printing and cutting)

RS: That would be pretty amazing! The only challenge, as far as I can see (and I'm sure you've observed building actual, 3D paper pop-ups yourself), is that any real world, 3D paper pop-up model will always have size restrictions due to size of the paper pages it's attached to. A great deal of the work I do (and spend time on) is figuring out how to get very large pop-up structures to fit inside very small spaces. Currently that is all mental work and, as of yet, I've not seen any software that can do this.

Why do think that most paper engineers still design manually, what do you believe would make them consider using a software tool?

RS: See my comments above. The reader/consumer wants to be surprised at how large the pop-up is coming out of those little pages. It's magic between pages and the reader responds emotionally to that.

5. Do you believe that an automated approach would be helpful for novice designers or hobbyist?

RS: Absolutely, I think this would be a great tool to add to our tool box of paper, scissors and glue.

6. Do you think that a semi-automated approach is better? This means that after the initial design is created by the system the paper engineer can still go in and edit the designs. What aspects of the design process would you like to have control over?

RS: Good point. I think automation is good in the earlier stages of design but (as I'm sure you know), the experience of seeing a pop-up model unfold on a screen is COMPLETELY different from seeing the paper structure unfold in the real world. This is a VERY important point of emotional involvement by the viewer. The paper engineer would definitely want to be able to edit the real world pop-up structure as the design progresses.

7. Do you believe that our results are comparable to actual paper pop-up designs that paper engineers make?

RS: I think there are great similarities between the two processes. It can also help a beginning paper engineer to see, on a screen, how things can move before committing to the (often daunting) task of making it real world in paper.

Which models or parts of it would you have done differently? (Feel free to comment on the individual results, or if you don't have time then just the results in general)

RS: You must always remember that what paper engineers make is not a series of angles or movements but a real world experience. If we are designing a person or animal in some kind of action that action must be believable by the viewer. The human eye is VERY sensitive and will know instantly when something "is not quite right" in a design. Thinking about angles and movement is fine, but it is the final emotional response to what's been created (and being experienced) that counts. As long as this philosophy is always at the forefront of the work you're doing then success in your work will follow.