Virtual Origami: 
Investigation into Fold-Based 3D Modelling

Lewen Yu        Advised by Julie Dorsey and Min Kim

September 28, 2012

1 Motivation

Origami, or the art of paper-folding, is a popular pastime around the world that holds strong cultural significance in East Asian countries. The folder transforms a sheet of paper into a 3D model through a series of folds. The craft is relatively easy to pick up for beginners, but an expert folder can construct incredibly complex and intricate models.

We wish to explore a new geometric modelling paradigm inspired by this craft of paper-folding. In this paradigm, the modeller begins with a flat plane as the primitive, and constructs her model by applying a series of folding transformations. In contrast to some other 3D modelling paradigms, fold-based modeling1 is intuitive and has a flatter learning curve since its mechanics is grounded in a well-defined activity in the physical world. Folding in the virtual world also implies that modellers need not confine themselves to physical laws, thereby granting greater artistic control.

Though there have been significant work in the field of computer-assisted origami simulation, the recent explosion in the ubiquity of touch-based devices makes this a pertinent time to explore this modeling paradigm with touch-based gestural input.

1 “fold-based modelling” is a placeholder, which may change in future revisions of this proposal and the resulting paper.
2 Background

2.1 Survey of current modelling paradigms

Popular 3D modelling paradigms include polygonal modelling, curve modelling, constructive solid geometry, digital sculpting, and sketch-based modelling, among others.

Polygonal modelling is especially pervasive in the video games industry, where in some cases it is still necessary to minimize the number of polygons in the model while preserving information and aesthetics. In this workflow, artists interact directly with the vertices, edges and faces of the model through traditional 3D modelling software. Though this approach provides unparalleled artistic control, it can often be quite limiting and tedious. As a result, artists have specialized tools to alter and expedite their workflow in certain situations, such as the ability to cut, extrude, and stitch surfaces.

Curve-based modelling is often used in industrial design and engineering, where designers and engineers construct a solid model from a set of curves, which are usually modelled by Non-Uniform Rational B-Splines (NURBS). NURBS have the advantage of being invariant under scaling, rotation, translation and shearing [9], though they possess a steep learning curve. Curve-based modelling is particularly befitting for models with well-defined curvatures and arcs.

Another popular paradigm in CAD is constructive solid geometry (CSG), whereby solids are composed of various components via regularized set operations [10]. This is well-suited for models with repetitive components, such as teeth on a gear.

Recent development and commercialization of digital sculpting have led to strong adoption of the technique, particularly in the video games and 3D animation industry. Pioneered by Perry and Frisken as “digital clay” [8], this paradigm essentially has the artist carve away\textsuperscript{2} at a virtual bust to reveal the model underneath. Recent developments in Haptic technologies promise to augment digital sculpting with the usage of haptic pens or “chisels” [3].

Lastly, sketch-based modelling is a recent addition that emphasizes rapid construction and conceptualization of approximately 3D models by interpreting 2D brush strokes [12]. Recent improvements to sketch-based modelling use context as an integral yet unobtrusive component for 3D sketches [7].

\textsuperscript{2}Digital sculptors have the additional ability to smooth, mould, and extrude the bust at will.
2.2 Past work in origami simulation

Folding paper in the virtual world poses a number of unique challenges in user interaction. Even with elementary folds, intuitive mechanisms must be in place to specify the angle of the fold and the degree of the fold. An early origami simulator by Robert Lang implements a simple mouse-based graphical user interface (GUI) that allows users to carry out basic mountain and valley folds [4]. Around the same time, Miyazaki et al designed and implemented a similar system with the additional capability for nuanced folding techniques such as tucking and bending [5].

However, with more complicated techniques such as reverse, squash and sink folds, it is less apparent how a mouse-based GUI can maintain intuitivity. In fact, more traditional GUIs often introduce additional specialized tools, parameters, and modes to account for additional capabilities. In such cases, touch and gesture-based interfaces may potentially allow for more intuitive interactions. A recent origami simulator by Chang et al implements a touch-based interface with the capability for simple folds [1].

Aside from user interaction, virtual origami poses additional technical challenges such as self-intersection avoidance, physical simulation, and the need for a robust backend to quickly store and retrieve crease and fold history. The simulator by Miyazaki et al specifies a system where the resulting model is computed from a discrete steps of folds starting with a flat plane [6]. Tachi develops a more robust, physically-based origami simulator that enables more complex models that do not necessarily comprise of a series of discrete simple folds [11] (for example, the waterbomb—which often requires folders to inflate the model by blowing into it). Furuta et al introduces further improvements to existing rendering techniques [2].

3 Project Overview

Scope and non-goals

The goal of this project is to explore fold-based modelling techniques on a touch-based interface, so emphasis will be placed on developing strong user interaction mechanisms for manipulating models. Novel rendering, folding algorithms and simulation techniques are not the focus of this project, and neither is the construction of a faithful origami simulator. Though this modelling paradigm is strongly inspired by the craft, the final prototype will
likely deviate from traditional conventions of origami. A prototype will be developed on the iPad to gather feedback on the user interaction mechanisms we develop. Finally, a user study will be done at the end of the project to evaluate the overall effectiveness of this new modelling paradigm.

Implementation Notes

Platform: iPad
Languages: C++, ObjectiveC
Frameworks: OpenGL ES

The frontend and the wrapper for the backend of the iPad app will be written in Objective C. The backend shall be written in C++ for reasons of portability: should we decide to port the project to another platform (such as Android or Windows) to test its effectiveness, porting the core backend of the project will take relatively little effort.

Timeline

First Milestone: October 29
End of October break.

- Finish implementing rendering and simulation engine.
- Begin iterating on user interaction designs.
- Prepare to implement simple folding techniques.

Second Milestone: November 26
End of Thanksgiving break.

- Finish implementing designs for simple folds.
- Start implementing more complex folding techniques if time permits.
- Start writing report.
Final Deadline: December 12
End of reading period.

• Submit code and report.

4 Deliverables

Prototype and source code
The source code of the resulting prototype shall be made available to the Yale community on its CS490 web site and to the public on Github.

Documentation
Documentation for the code shall be written for potential follow-up work. This shall include a high-level architecture of the software, implementation details and decisions, and a list of known bugs and issues. This will be included as part of the report.

Report
I will write a report discussing the interim results of this prototype. In addition, the report shall cover the background, the documentation, and any deviations from the original proposal.

Paper
I will assist with the authoring of a paper based on the results of this project.

References


