

# Surface Manipulation Using a Paper Sculpture Metaphor

Glenn McCord  
Dept. of Computer Science  
University of Auckland  
Auckland, New Zealand

## ABSTRACT

The creation of 3D computer models is essential for many applications in science, engineering and arts and is frequently performed by untrained users. Typical interaction with a 3D modelling tool is in 2D, but an intuitive mapping between 2D input and 3D models is a non-trivial task and reflected in the difficulty novices have in using current 3D modelling software. Using metaphors of paper sculpture and pen sketching, our sketch based modelling tool simplifies this interaction mapping. More intuitive object manipulation means that an otherwise complex model can be rapidly created by an inexperienced, non-artistic user. To demonstrate this, we have chosen to model orchid flowers as they offer considerable challenges to the artist due to their complexity of shape and detail, especially the petal surfaces which vary a great deal in curvature.

## 1. INTRODUCTION

Traditional 3D modeling applications offer tools powerful enough to model a diverse range of creations but, unfortunately, many potential users can be overwhelmed by the enormous complexity associated with these applications' flexibility. One of the difficulties of these tools for novice users is that they are not based on any real world metaphor. Pencil and paper sketching, for example, is one of the most simplest yet effective ways to exercise artistry, yet few modelling tools support digital pens (styluses) to any significant degree. Other metaphors, such as paper sculpting, can provide an interaction that makes it easier for users to predict the results of an action. We are exploring a blend of paper sculpting and sketching (where sketched lines represent paper cutouts) as an aid to novice 3D modelling interaction.

The proposed interface combining sketching and paper sculpting has the goal of easing the transition from the initial conceptual design into the final 3D model. This proposal is supported by two observations: first, many users find it hard to create 3D shapes which correspond to multiple 2D views and second, they find it difficult to understand the

relationship between a surface's parameters (controls) and the resulting shape.

A 2D sketch is a quick, intuitive and easy way to represent 3D shapes. However, even if software was capable of comprehensively inferring a 3D model from a 2D sketch, some users don't have the artistic expertise to conceptualise and draw a 3D image. One advantage of a paper sculpting based metaphor is that interaction with paper is a natural process children perform from an early age and it facilitates the mental transition from a 2D to a 3D object since the paper used in this metaphor is a 2D object. A mixture of drawing and sculpting metaphors allows the user to intuitively interact with the model because they subconsciously predict the effect their actions will have on the model based on their real world experiences.

There is precedence for the use of 2D surfaces for modelling complex shapes. The primitives used in traditional modelling tools are composed of surfaces, metal workers construct using sheet metal and scaled architectural models are made with cardboard.

To realise the paper folding metaphor we have chosen to model orchid flowers as they offer great challenges to the artist due to their complexity of shape and detail, especially the petal surfaces which vary significantly in curvature (figure 1).

This report discusses two primary facets of our orchid modelling tool; how the paper sculpting metaphor can be used to create the surfaces that make up an orchid, and methods for repositioning these surfaces in order to edit the floral arrangement. Finally, we propose how the folding metaphors can be incorporated into other modelling domains.

## 2. PRELIMINARY INVESTIGATION

Paper is thin, which makes it an ideal metaphor for 3D modelling of objects that consist of thin plane-like objects such as flowers. Our approach uses metaphors of paper sculpting techniques whereby the user's sketch is a paper cutout that gets sculpted by folding, crimping and indenting it.

Paper is a widely used artistic medium, not just because of its prevalence but also because of its flexibility as a modelling medium. One of the most well known paper crafts is origami, but there is more to paper sculpting than just folding hard edges.



Figure 1: A screenshot of an orchid flower constructed with our modelling tool.

Paper can be cut, torn either with or against the grain, creased along either a straight or curved line, coiled/rolled, cut to form textured patterns by utilizing light sources, joined together using tabbing, layered in relief, crimped (forming curves by cutting the paper and then folding it in on itself), impressing the paper and by curling edges [10]. All of these can serve as metaphors for virtual modelling.

## 2.1 The Limitations of Sketching

Hand drawn sketches are a useful way of quickly conceptualising an object but using sketches as a means of generating 3D objects on a computer has several complications. Of most significance is that there is no depth information; a 2D sketch has infinitely many 3D interpretations but our real world experiences allow us to interpret the 3D information. However, there are many instances where our 2D to 3D perception breaks down, such as the artwork *Waterfall, 1961* by Maurits Cornelis Escher.

The software must either infer depth information by some of the visual clues and/or provide the user with a means of manipulating the depth information. Sketching is complex in that it is not just a simple silhouette; combinations of outlines represent object silhouettes, overlapping lines indicate 3D depth and shading depicts shadows and highlights from lighting on a 3D object.

Generally, sketching in software applications restrict the user to draw either like a child, with simple interpretations of objects, or like an architect by using structured lines.

## 2.2 Preliminary User Study

Before implementing a paper sculpture styled solution, we conducted an informal usability test to determine what type of sketching techniques could be used to assist in the modelling process. The participants comprised of twelve computer science and software engineering students and staff with varying drawing ability.

Each participant was given an A4 collage of thirteen photographs of different types of orchids at different orientations

Sketching Techniques Used (%)	
Sketched (multistroke) outlines	50%
Shading	42%
Surface contour lines (i.e. petal veins)	58%
Foreground object drawn first	50%
Mixed f/b-ground order but f-ground non-overlapping	42%
Mixed f/b-ground order	8%

Table 1: A summary of different sketching techniques used by participants in the preliminary user study.

(figure 2). The participants were then asked to draw an orchid, using the collage as a guide, keeping in mind the steps they took to draw the orchid.

From this study we wanted to determine what steps were taken to draw an orchid, such as the order the orchid components (petals, sepals, labellum, etc.) were drawn, and what drawing techniques were used, such as shading and multi-stroke silhouettes. Particular drawing strategies could then be investigated for inclusion into our sketch based orchid modelling interface.

Some of the observations are tabulated in table 1. Most participants drew object silhouettes with multiple pen/pencil strokes but less than half used shading. The most common strategy was to draw the foreground orchid parts first and background parts last. If non foreground parts were drawn first then they tended to not overlap the pen strokes that made up the orchid parts behind it. We suspect that users may draw differently on a digital medium so we propose that a similar sketching study be conducted with a digital medium using either a mouse or a digital pen/stylus.

The most important observation was that there was great variation in the way the participants drew the orchid flowers. We believe such variation in drawing styles make it very difficult to develop a ‘one size fits all’ sketch based orchid modelling interface that will be able to satisfy the desires of all users. The sketched orchids from this user study are shown in figure 3.

## 2.3 Paper Folding User Study

The diversity of drawing styles by the participants in the preliminary user study demonstrated the difficulty in creating a sketch based interface for the purpose of 3D modelling. To test how intuitive paper folding would be to assist a user in modelling in 3D, we conducted an additional experiment to prove a theory that geometric properties aide the user in predicting fold behaviour. As will be discussed further in section 5, we theorised that paper tends to fold about an axis between two concave areas in a surface silhouette.

Our small informal user study comprised of only six participants because it soon became clear what the benefits and challenges of paper folding were. Participants were given five different paper cutouts and asked to predict about which axis the paper would fold if picked up from a certain point. Each cutout was marked with three points from which the user would lift the paper (figure 4) and as the paper was lifted higher, the participants were asked to apply force in the direction indicated by the arrows. These arrows rep-



Figure 2: This collage of orchid photographs was given to the participants of the preliminary user test as a reference.





Figure 3: This collage of sketches from the preliminary user test shows how diverse the participant's drawing styles are.

Percentage of Correct Predictions (instances of)	
Type A	88% (7)
Type B	70% (5)
Type C	100% (3)

Table 2: A summary of results from the paper folding user study.

resented the direction the paper tended toward when lifted and was used to discourage participants from forcing the paper in alternative directions.

Each point can be categorised as one of three types depending on their position within the surface.

**Type A** Points that are in areas to the outside of the cutout that are isolated to their own foldable area as defined in section 5. This area must be smaller than the rest of the cutout.

**Type B** Points that are in areas that form a ‘C’ or ‘S’ like curve.

**Type C** Points that are in any other area not defined as either type A or B. This is generally in the middle of a cut out by being between a Type A or B point. Alternatively, it could be in an area that is large comparative to the rest of the cutout.

For each point lifted, the users’ prediction as to the axis about which the paper would fold was marked correct or incorrect accordingly. Table 2 summarises the results in relation to the type of points that were lifted. It was observed that folding the paper from type A points was much easier to predict than the other points, especially on surfaces with high-curvature.

The most common comment by the participants was “it depends on how you fold it.” This reflected the fact that folding the paper with a high arc placed the fold axis closer to the axis defined in section 5, whereas a tight/small arc placed the axis much closer to the lifting point.

Another observation was made in regards to the thickness and weight of paper. The theory set out in section 5 assumes an ideal were the paper is infinitely thin and of reasonable weight. However, thicker paper has a tendency to fold further away from the point of the paper being lifted so is sometimes in conflict with the proposed fold axes.

Paper weight also affects folding. When the centre of gravity of the area being folded is not adjacent to the proposed fold axis, the actual fold axis is twisted diagonally. This is a gravitational force versus tension force scenario.

This user study showed that it is possible to ‘over engineer’ the paper folding metaphor. The consequence though may be that users will not comprehend the physics behind it, thus making the interaction counter intuitive. The positive results for type A points supports the theory in section 5.

### 3. RELATED WORK

From our preliminary investigation we determined that our orchid modeller draws on work from four main areas of research: sketch based modelling, surface deformation, shape recognition and flower modelling.

#### 3.1 Sketch-based Modelling

Sketch-based tools have been explored for a number of 3D modelling domains such as transformation from sketch to structured CAD projects. SKETCH [21] was an early research project that turned a conceptualised sketch into a digitised 3D scene. It exploits the ease of design afforded by sketching and the ability to change viewing angles with the 3D digital medium. 3D primitives are constructed with basic pen strokes which are then extended to basic 3D objects. Complex objects are constructed using a combination of primitives.

Other research uses constraints offered by the perspective or orthographic viewing angles [11]. This allows users to actually sketch the model isometrically and have the system guess the hidden sides of the object by using symmetry. This means the object can be drawn more realistically instead of using interpreted gestures.

A common strategy for creating free form 3D objects from sketch input is to create simple objects and then either combine or deform them into other shapes. Igarashi’s Teddy application created a 3D object by inflating 2D sketches based on the width of the 2D object [6]. With a combination of cutting gestures and combining objects together, it is possible to create complex, inflated (blobby) shapes. Further research projects using similar inflation metaphors join objects smoothly together and afford shape alterations by re-sketching parts of the silhouette [12] or by inferring 3D geometry by interpreting overlapping sketch lines [13].

Other authors [16], [5] have shown that complex 3D objects can be edited using stylus strokes that retrace an object’s silhouette. The modification of a model’s silhouette subsequently rescales it so that it remaps itself to the new silhouette.

#### 3.2 Flower Modelling

The sketch-based modelling tools reviewed in the previous subsection are well suited for geometric objects with large planar areas such as plants and flowers. Leaf and petal surfaces have their own unique challenges such as their high detail, variation in curvature and overall layout.

One of the earliest plant modelling tools was an algorithmic approach by Prusinkiewicz and Lindenmayer whereby plant structures are constructed using rule based logic [19]. This abstract, bottom up approach is, however, non-intuitive and difficult to control without extensive experience, so alternative sketch-based modelling of plants has been explored [15], [1]. Constraints can help with automatically creating a 3D structure, such as the assumption that branches seek to be as far away from their neighbour branches as possible [18]. Ijiri et. al. have shown that an effective way of creating a realistic flower is to sketch and then edit each individual flower component (petals, flower head etc), and then combine them together to form the complete flower model [8].

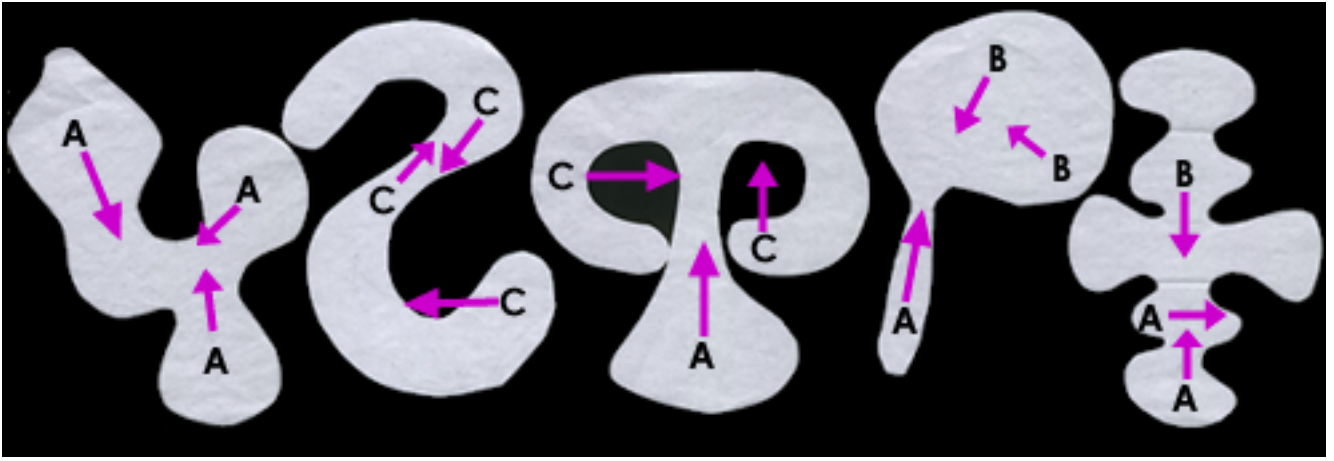


Figure 4: These five cutouts were used for the user study defined in section 2.3. The points that the user lifted the paper at are labelled by their type (A, B, C). The arrows represent the direction the user was to apply force once the paper was lifted at the designated points.

In order to make the flower modelling process more an artistic exercise, it has also been shown that a user can sketch a plant in its entirety, and then have each of its sketched components replaced with 3D equivalents [7].

Although petal like surfaces can be created from the Teddy ‘blobs’ by creating the blob and then cutting it like a potato chip, it deviates too far from what would be intuitive to a user. The flower modellers by Ijiri et. al. offer a much better alternative but their petals are restricted to a silhouette that doesn’t form large concavities; ideally a user can sketch a petal of arbitrary shape. Another limitation is that petal curvature can only be altered by a series of modifying strokes that displace the vertices. We believe that there are more intuitive ways of modifying the curvature of petal-like surfaces.

### 3.3 Surfaces and Shape Theory

The petal surfaces modelled in [8] and [7] are constructed with b-spline surfaces which, although very smooth, provide a representation that is mathematically complex and have a multitude of parameters. In contrast a Delaunay triangulated surface can be reshaped by any tool which modifies the point set representing it and the new triangulation is always unique not matter what order the point set is processed. Local details can be easily added by adding additional points and retriangulating it. In addition local details can be enforced by adding constraints. A defining characteristic of the Delaunay triangulation is that its triangles will have the highest possible aspect ratio [2].

Marr and Nishihara’s research into shape recognition [14] has proven useful for numerous domains such as 3D diagramming [9]. Of most relevance to our research is the observation of how people recognise shapes as a composition of smaller subparts. Conversely, a larger object can then be divided into subparts by identifying concave sections of a silhouette. The shape recognition strategies used in [14] are useful for modelling purposes because this is how users intuitively identifying relevant subsegments of 3D objects, including sub areas of individual surfaces.

## 4. PAPER SCULPTURE INTERACTION

Many of the paper sculpting techniques mentioned in section 2 can be applied to surface manipulation to facilitate predictable user interaction. The primary techniques are the ability to cut, curve and crease paper so we have explored how these metaphors can be used for manipulating surfaces on the computer.

Besides the inherent difficulties with managing 3D objects in a 2D space, there is also the problem of working with a single mouse cursor. We are essentially paper sculpting with one hand, we have to take a piece of paper, cut it to shape and then sculpt it by adjusting its curvature. By blending the sketching and paper sculpting metaphors together, simple interaction is achieved.

Here is an example of how the user could create a petal of an orchid. First the user draws the outline of the petal. The region enclosed by the sketch can be interpreted as a flat object cut out of a sheet of paper. Immediately the software generates and displays the cutout as a surface. The user selects part of the surface with the mouse cursor and drags/pulls at it. As a result, the selected subpart will fold about an axis formed depending on the geometry of the cutout (figure 5).

## 5. GEOMETRY

The user requires an intuitive way to fold their original shape. There are two generalized ways of achieving this:

1. The user defines a fold axis about which a sub part of the surface will fold about. This would be the equivalent of creasing paper and then folding about the crease.
2. The software infers fold axes about which areas of the surface can fold about.

Both of these ideas are perfectly valid for paper sculpting. We can achieve the first technique by creasing the paper, thus creating an artificial axis about which the two areas

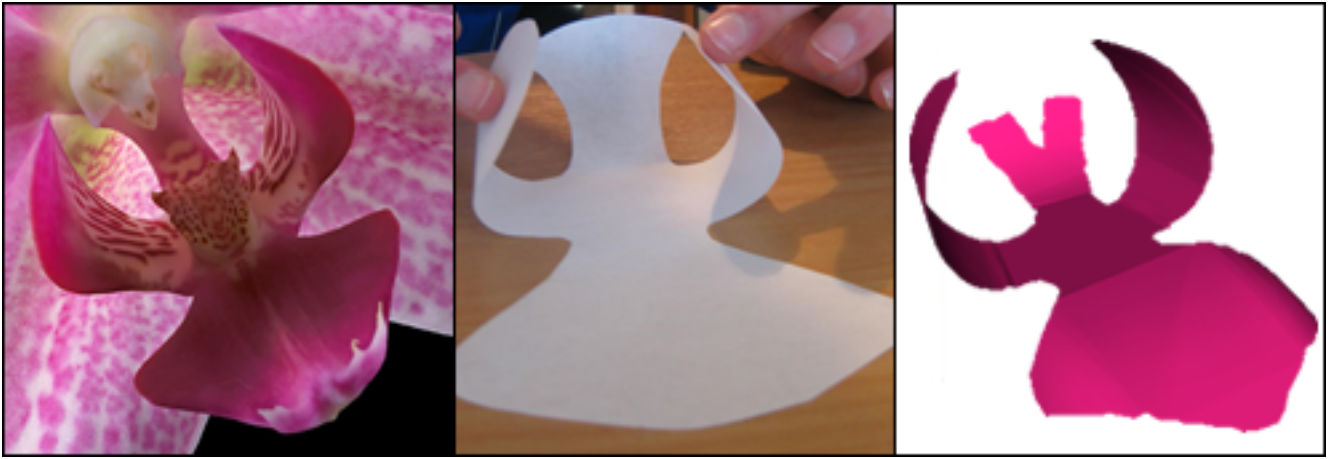


Figure 5: This selection of images shows how the curvature of a real orchid petal (left) can be represented with paper (centre) and with a digital model that uses paper folding properties.

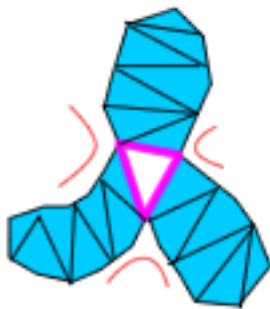


Figure 6: Lines between concavities in the surface are the fold axes of the surface.

can rotate about; creasing doesn't have to be restricted to a straight line either. The second technique takes advantage of a 2D geometric property that defines these foldable axes automatically. Halverston [4] determined that silhouettes are especially important in recognising objects, made evident when children draw the most salient silhouette of objects such as animals. By taking such a silhouette, Marr and Nishihara [14] noted that the concave sections of objects define the subparts of an object. Figure 7 (taken from [9]) demonstrates how the body parts of a donkey can be identified from the concavities in the silhouette.

This observation can be applied to identifying foldable areas of arbitrary surfaces. As can be seen in figure 6, the axes about which they fold is defined by a line that joins one concave curve to the other.

These geometric properties can be applied to paper sculpture. By picking up the paper from one of the subparts of the object, the subpart then naturally curves about the axis defined by the concave points of the silhouette that define that subpart (figure 5). The ability for the user to relate to this object subdivision and paper style folding is the basis for the interaction strategy used in our orchid modeller.



Figure 8: The triangulation strategy used by “Teddy” defines different triangles as Terminal (T), Sleeve (S) or Junction (J) depending on how many sides are shared with the silhouette [6].

## 6. IMPLEMENTATION

The implementation of our flower modeller comprises of three main parts. First, the user's sketch input (paper cutout) is turned into a surface. Then, this surface is analysed for axes about which certain areas of the surface would fold about, allowing the result of a user's interaction to be predictable. Finally, by providing the user with the ability to create multiple surfaces that can be folded and repositioned, orchid flowers can be modelled.

### 6.1 Delaunay Triangulation

We have used Delaunay triangulation to both create the surfaces and to identify foldable axes. Its usefulness is best demonstrated in Igarahi's ‘Teddy’ application, which is one of the most well known research projects for creating 3D models using sketch input [6]. The “Teddy” tool defines 3D shapes from 2D sketches by triangulating a closed sketched curve and computing a skeleton from it. The vertices of the sketched curve are then rotated around the skeleton, resulting in a 3D shape whose projection is the original sketch. During the process of constructing the skeleton the triangles that make up the triangulation are defined as either a terminal, sleeve or junction triangle where each triangle type has either one, two or no shared edges to the silhouette respectively (see figure 8).

The Delaunay triangulation is an efficient way of triangulation.

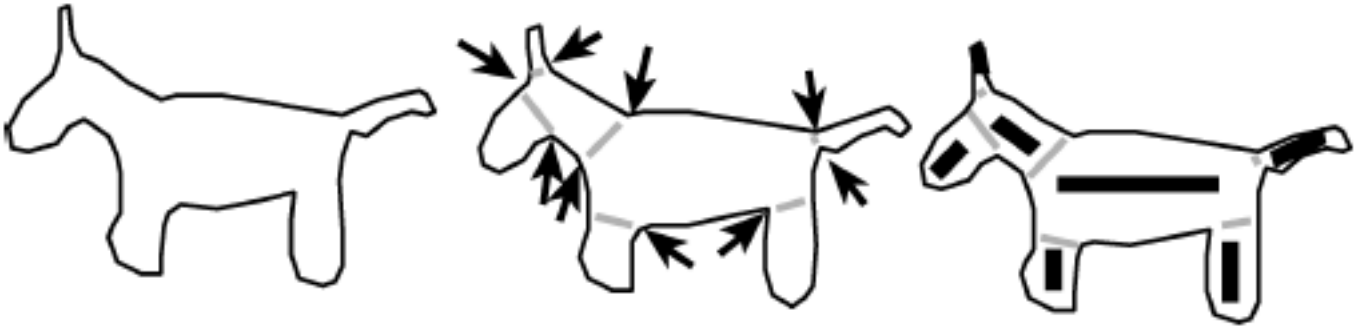


Figure 7: This image from [9] shows how Marr and Nishihara used concavity in a silhouette to isolate body parts of a donkey. We have adapted this technique to identify foldable areas of a surface.

lating a static set of vertices that ensures that the resulting triangles have the highest possible aspect ratio [2]. It is these properties that make the Delaunay triangulation an ideal method for calculating the primary fold axes of the surface because the fold axes are the same as the edges of some of the triangles. As discussed in section 5, these fold lines are the lines that join two concave points on a surface silhouette and are usually represented by edges from a junction triangle.

The Delaunay triangulation also allows the user to generate surfaces of arbitrary shape. By using NURBS or b-spline surfaces as in [8] and [7], the user is restricted to surfaces whose silhouette cannot have significant concave areas. Although b-spline surfaces afford very smooth curvature, we believe arbitrary surface shapes are more important in order to increase the versatility of the modeller.

## 6.2 Identifying Foldable Areas

Determining the foldable areas of a surface that the user can interact with first requires finding the fold axes. Each axis is then used to identify areas of the surface that correspond to them. However, this in turn requires the triangulated surface to be ‘pruned’ of triangles that would otherwise form foldable areas that would be too small to be folded. The left over triangles are then analysed for fold axes.

To achieve the pruning we use a simplified version of the ‘Teddy’ pruning method [6]. Whereas the Teddy application uses Delaunay triangulation for finding the skeleton of a surface, we have used it to discover fold axes. However, both require a means of ‘pruning’ the small, insignificant triangles that result from the triangulation. For the Teddy application this is useful for ensuring that there are no tiny bones in the skeleton, for surface folding it is important for ensuring that no insignificantly small foldable areas are identified.

Firstly all the triangles that make up the surface are classified as either terminal, sleeve or junction. Then, for each of the terminal triangles, we traverse through all the neighbouring triangles until a junction triangle is reached. Note that the nature of the triangulated sketch means that the traversal will never branch until a junction triangle is reached; on reaching the junction triangle, the traversal stops.

The shared edge between the junction triangle and the neighbouring traversed triangle (which could be sleeve or terminal) is a potential foldable axis. If the total area of the traversed triangles fits within a semi circle whose diameter is the potential foldable axis, then the area is too small and will not be deemed a foldable area; this is done by excluding these triangles from the fold axis identification step. An illustration of the pruning process is shown in figure 9.

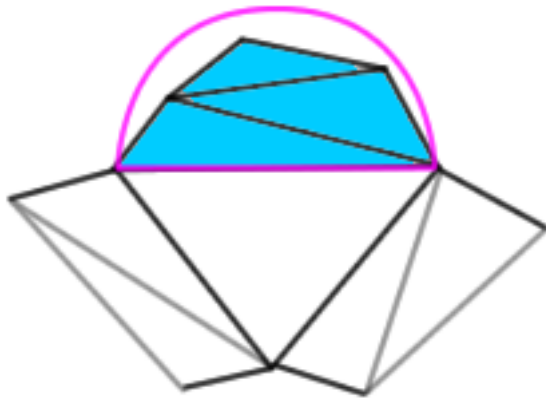
The remaining triangles are then utilised to find the fold axes by using a traversal process similar to figure 11. First the triangles have to be reclassified because the exclusion of the pruned triangles alter the silhouette. From a terminal triangle, we traverse through each neighbouring triangle, branching at junction triangles as required, until we reach another terminal triangle (end point). During this traversal, foldable axes are identified using the following criteria:

- The axis is a junction triangle edge that is between a junction triangle and a terminal triangle 11(b).
- The axis is the shortest non-silhouette triangle edge between two junction triangles 11(c)

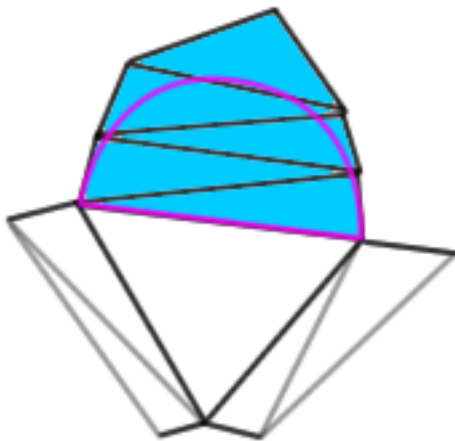
Using the original (non-pruned) set of triangles, each axis acts as a divider that separates the surface into two foldable areas. ‘Picking up’ one side of the surface will fold it about the other side along the respective axis. Determining what triangles are to what side of a fold axis is achieved by taking an end point of an axis and then following the silhouette around until the other side of the axis is met. The axis and silhouette segment form a perimeter that encircles a subset of triangles (figure 11(e)). This subset is one foldable area and the remaining triangles form the other. Note that a foldable area is a reference to a part of the surface and is not a surface in its own right.

It is common for a foldable area (‘parent’) to contain smaller foldable areas (‘children’). When the above strategy is applied to each axis, there are multiple overlapping foldable areas. However, if rendered according to size, smallest on top, when the user interacts with the surface, the smallest foldable area under their mouse cursor will be selected. It will be this area that will fold about its corresponding axis.





(a)



(b)

Figure 9: 9(a) shows a selection of triangles that will be pruned because they fall within the semi circle of the neighbouring junction triangle. 9(b) however, will become a foldable area.

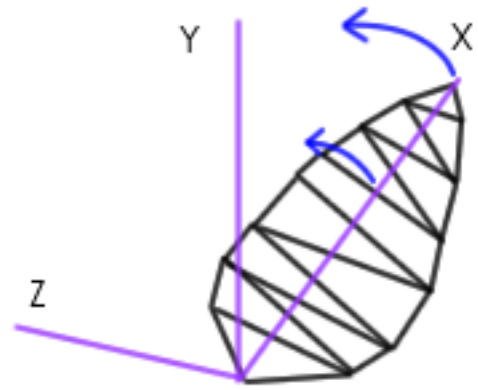


Figure 10: A surface with no calculated fold axes has a default fold axis about the Z axis pictured.

Unfortunately this is also a limitation in that if the user wishes to fold a parent foldable area then the user must select a point that is not within a child foldable area. This reduces the amount of surface area that the user can select.

It is possible for a surface to have no predetermined fold axes and therefore no foldable area. For this case we set the fold axis to be the line orthogonal to the vector that extends from the origin to the point on the surface that is furthest from the origin (figure 10). Attempting to fold the surface will therefore fold the entire surface about this axis.

### 6.3 Stroke Preprocessing

Before effective surface triangulation can take place, the user's stroke input must be processed by ensuring that input points are evenly spaced and relatively smooth. This can be achieved by line simplification and by controlling the input resolution.

One of the problems with our Delaunay algorithm is that three points in a line can sometimes result in an infinitely flat triangle that then leads to overlapping triangles. The risk of this occurring was reduced by simplifying the user's pen stroke points using a line simplification algorithm [17]. This greatly reduced the number of points along the same line, but was able to preserve the user's original stroke.

User input via pen strokes does not produce a perfectly smooth line and accepting too many points at close intervals results in a stroke comprising lots of small zigzags. This affects not only the smoothness of the silhouette but also the triangulation because it will create lots of small triangles toward the edge of the surface, adding to processing time. We controlled the resolution by making the inputted stroke points equidistant apart. It can be argued that the user may wish to create a highly zigzagged surface silhouette but because we are modelling orchid petals, which are generally smooth, we have made this assumption.

More complex shapes create complications and are outlined as follows:

- Sometimes the pruning algorithm eliminates all the junction triangles. However, there is usually at least one pre-pruning junction triangle that is larger and shaped closer to an equilateral. The shortest edge of this triangle is the significant fold axis. Note that size and closeness to being equilateral is a property of most of the junction triangles.
- If the width of a surface is consistent or has only one local maxima, then there will be no junction triangles or silhouette concavities and therefore won't be able to use the geometric properties discussed in section 5. In this case we use the method represented by figure 10.
- A surface in the shape of a letter 'S' or 'U' does have silhouette concavities and should be able to fold about certain points. However, its triangulation often doesn't result in junction triangles and can't be used by our algorithm to find appropriate fold axes. Because such shapes don't exist in orchid flowers, it has been ignored for now. Note that it was this shape that participants had the most difficulty with in the preliminary user study.

## 6.4 Folding

A foldable area can appear folded by rotating every vertex of the foldable area about the fold axis. The user can control this angle by clicking and dragging the foldable area with the mouse cursor.

We have currently implemented a simple folding mechanism whereby a mouse-drag to the left will fold the surface towards the user and a mouse-drag to the right will fold the surface away from the user. The angle the surface folds is related to the cursor displacement.

This angle represents the maximum angle the foldable area will fold about its axis because, in order to look like a paper fold and not a rigid cardboard fold, this angle can't be the same for all vertices. Instead, each vertex is rotated by an angle proportional to the distance it is from the fold axis (when of infinite length). For example, a vertex that is far away from the fold axis will rotate a greater angle than a vertex that is very close. The result is a fold that resembles a flexible piece of paper.

We have used a linear equation for determining the angle a vertex will fold about the axis. If  $d = \text{distance from axis}$ ,  $d_{max} = \text{max distance of any vertex from axis}$  and  $a_{max} = \text{max angle}$ , then:

$$\text{Angle} = d/d_{max} * a_{max} \quad (1)$$

This method does not preserve surface area and greater angles stretch the surface, however, it was rarely noticed in the user tests. Solutions are discussed in section 9.

As will be discussed in the user study in section 7, the left to right method of folding is non-intuitive and was intended to be a temporary implementation. Ideally, the direction the

mouse cursor is dragged should be orthogonal to the fold axis (when projected into screen coordinates) so that the user can fold the surface in the direction they wish it to go.

## 6.5 Folding Options

As observed from the user testing discussed in section 2.3, the nature of the fold can be influenced by the folder/user. If the fold arc is kept very low (figure 12(a)), the fold can be kept very flat. Alternatively a higher arc can produce a much rounder fold (figure 12(b)).

Equation 1 can be extended by using additional parameters to affect the height of the arc.

Additionally, the flexibility of the paper can be adjusted by adjusting the dependence the fold angle has on the distance to the fold axis. The more similar the angles are to each other, regardless of distance to the fold axis, the more the fold will appear like a rigid piece of cardboard.

The other extreme would be to apply a sagging effect due to gravity. However, this becomes counter intuitive as the user must comprehend the physics involved. These are all valid features that needn't be available by default.

As observed in the initial paper folding user study discussed in section 2.3, paper will continue to fold if forced beyond a certain point. This means that if a child foldable area is folded beyond a certain 'critical angle' when using a high folding arc, then the paper will start to fold further down the surface. However, we believe that this may be another example of over-engineering the folding interaction and the same effect could be achieved with a user defined fold axis.

## 6.6 Level of Detail

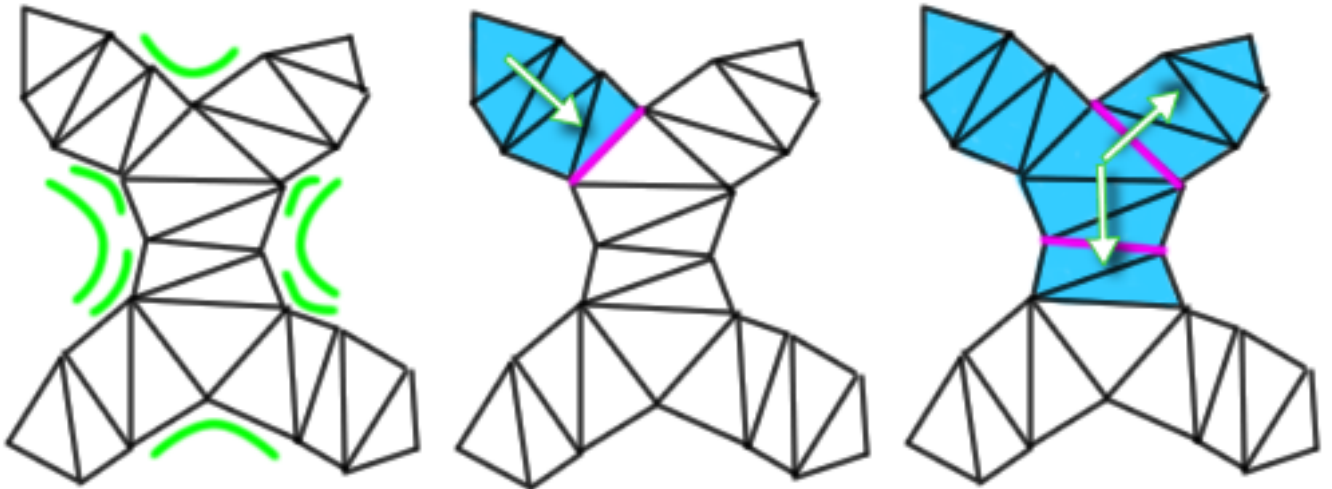
It is necessary to add interior vertices to a foldable region as this will not only ensure smoother folding but will also be useful for surface manipulation tools other than folding. Such tools are discussed in section 9.

Extra vertices are created while the surface is still 2D by laying out a vertex 'mesh' on top of the surface silhouette whereby each point in the mesh is equal distant apart. All the vertices that fall outside of the silhouette are excluded and the remaining vertices, including the vertices of the silhouette, are re-triangulated using the Delaunay triangulation.

## 7. DIGITAL FOLDING USER STUDY

To test the effectiveness of the folding component of our orchid modelling tool, we conducted a usability study. Participants were given five simple folding exercises to do and then given three questions in order to evaluate their experience with the tool.

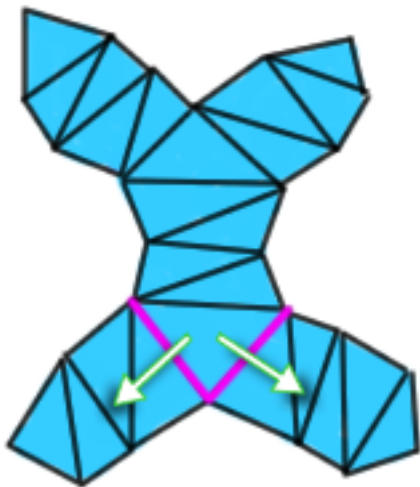
The first two exercises were training exercises to introduce the user to the application. For each one, the user was given a paper cutout of an arbitrary shape and asked to fold it. They were then shown the digital paper cutout equivalent and asked to interact with it by selecting areas with the mouse cursor and trying to fold it in much the same way as the paper version.



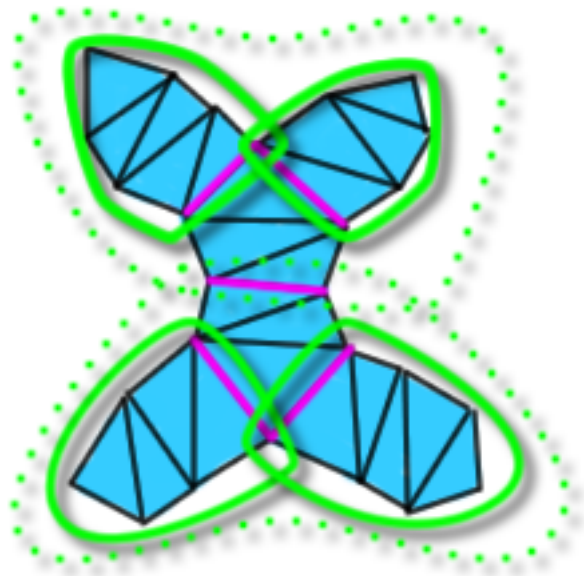
(a) The triangulated surface with the concavities indicated.

(b) Traversal starts at a terminal triangle. The first fold axis is found when a junction triangle is reached.

(c) The traversal branches. One fold axis will be the junction edge that is the same side as a terminal triangle. The other axis will be the shortest edge between the starting junction triangle and the next junction triangle.



(d) The traversal discovers two further fold axes; junction edges that are on the same side as a terminal triangle.



(e) The green outlines represent the overlapping foldable areas.

Figure 11: This series of images demonstrate the surface traversal used to determine fold axes and foldable areas.

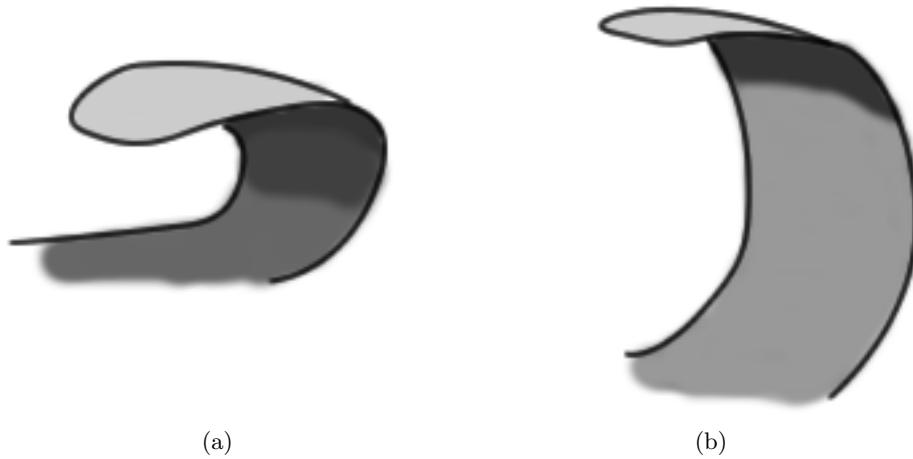


Figure 12: Paper folding can also take into account the folding arc. A low folding arc (12(a)) forces the actual fold axis further away.

For the final three exercises, the user was given an image of different orchid flowers and asked to model the labellum part of it. They were given a digital paper cutout version of the labellum as if it were spread out flat and then asked to fold it into shape using what they had learnt in the first two exercises.

On completion of the exercises, the participants were given three questions to answer. Answers were given on a scale of 1 to 10 where 10 corresponds to strongly agree:

- The folding tool was easy to use
- The folding tool allowed you to create satisfactory results
- The folding tool was fun to use

### 7.1 Results and Observations

Two factors were found to reduce the ease of use; the mouse-drag direction for folding, and incorrect shading on the surfaces.

At this stage we have implemented a folding technique whereby the user selects a foldable area and folds it by mouse-dragging the cursor to the left or right although subsequent updates will have the mouse-drag direction orthogonal to the fold axis. The user tests made it clear that this was an important improvement as every participant tried to mouse-drag in the orthogonal direction despite being informed of the controls.

Another source of confusion was the lack of realtime shading updates. As the surface is folded, the normals constantly change, but recalculating them at every frame is computationally expensive. The consequence of updating the shading on a mouse-release (after folding has finished), was a reduction in visual feedback of the surface position while being folded. Also, with the surfaces being only one colour

Level of Agreement (%)	
The folding tool was easy to use	70%
The folding tool facilitated satisfactory results	67%
The tool was fun to use	82%

Table 3: A summary of results from the usability study of our application.

(no textures) it was difficult to determine if the surface had been folded in front or behind.

There was another difficulty with the folding interaction in that the fold often went in the opposite direction than expected. A user would expect that if they were to select an area and then drag the cursor, that the fold will be in the direction of the cursor. But does the surface fold upwards in front of the surface or does it fold down behind the surface? For example, if the user was to select the surface in figure 13(b) with the cursor, one could reason that a mouse-drag in the left-down direction could fold the surface to form figure 13(b) or figure 13(c). Alternative folding strategies are discussed in section 7.2.

There was a desire for greater control of the surface folding when the surfaces did not fold where expected. Amongst the requests was the ability to define custom foldable axis (as if folding about a crease in the paper) and the ability to twist the paper or at least fold slightly off line to the fold axis.

Satisfaction with the results were related to the ease of use, especially in the desire to be able to fold the surfaces further. What was surprising though, was how much entertainment value their was with the tool. As can be seen by figure 14, some participants were keen to experiment with their own surfaces. The results from the user study is shown in table 3.



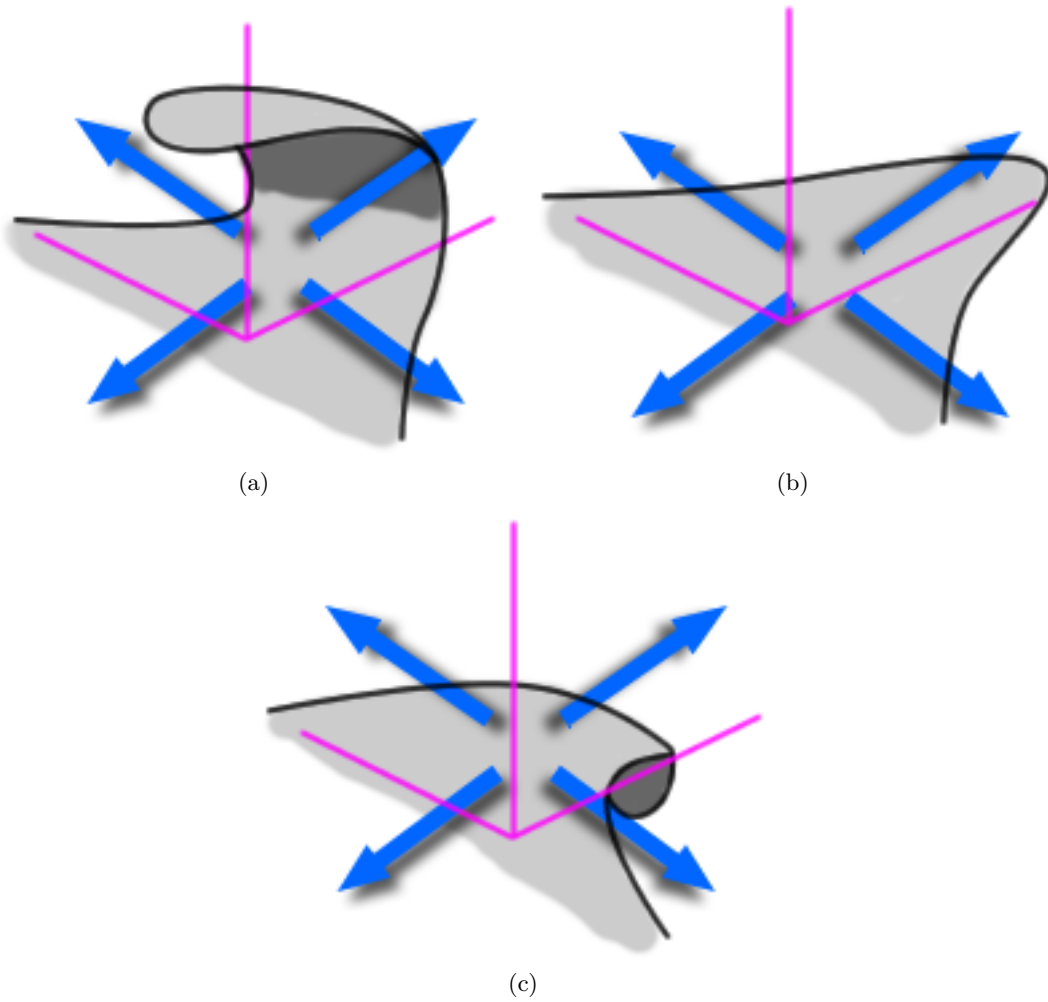


Figure 13: These three images demonstrate the challenges of folding a 3D object with a 2D interface. Folding each surface by mouse-dragging in a left-down direction for example, will intuitively result in opposite fold directions to each other.



Figure 14: One participant enjoyed using the tool enough to be compelled to model the Starship Enterprise. Unfortunately hard folding wasn't used.

## 7.2 Alternative Folding Strategies

In the user study discussed in section 7 it was observed that participants preferred to fold the surfaces by dragging the mouse cursor orthogonally to the fold axis. Unfortunately because we had only a limited folding implementation, the participants couldn't test how folding orthogonally could also have some issues associated with it.

If a surface can be folded a full  $180^\circ$  in both directions, there has to be a point where the fold ceases to be in the foreground and begins to move behind the surface into the background. One solution is to base the fold direction on both cursor direction and the orientation of the surface within the world. An interactive folding dilemma is demonstrated in figure 13. If the user wishes to fold the surface of figure 13(a) away from them, then a mouse-drag in the right-up diagonal direction would form figure 13(b). One might expect that a mouse-drag continuation in the right-up direction would continue to fold the surface; however, when looking at a right-up mouse-drag direction in figure 13(c), the user would probably expect the surface to fold in the opposite direction. This implies that there is a point at which the surface will stop folding if the cursor is only dragged in one direction. A possible implementation to address this could either require the user to reselect the surface and mouse-drag in the new direction, or require the user to change their cursor path in mid drag.

Our preliminary user study showed that a user induced effect that could be applied to paper folding, was controlling the arc height of the fold. A low fold arc keeps the the actual fold axis further away from the proposed fold axis and keeps the folded paper low, a high arc does the opposite.

One of the shortcomings of our folding implementation is that it doesn't capture fold arc information. A possible alternative though, would be to capture the angle of the fold by projecting the screen coordinates of the pre and post drag positions of the mouse cursor onto a plane that passes through the foldable area as in figure 15. By forming vectors between each point and the centre of the fold axis, the angle of the fold can be found. Capturing the start and end positions of the mouse-drag could infer information about the height of the folding arc as discussed in section 6.5.

## 8. ORCHID MODELLING INTERACTION

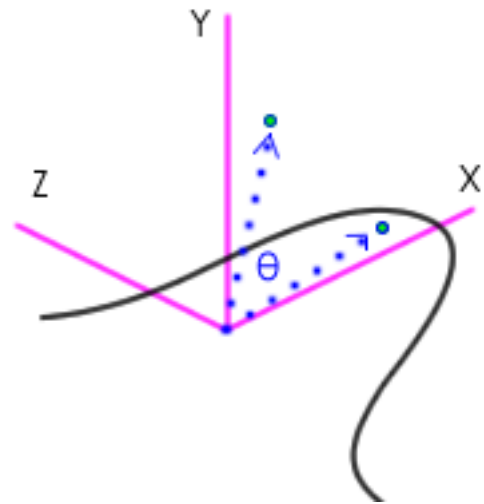


Figure 15: An alternative approach to determining the fold angle by projecting the mouse coordinates into an XY-plane.

Before discussing the modelling strategy, this section provides an introduction into the anatomy of an orchid. The structure is generally three outer sepals, three inner petals and a single large column [3].

**Sepals** The glorified remains of the flower bud. There is usually a dorsal (top) sepal and two lateral sepals

**Petals** Three petals of which two flank a large and flamboyant petal called the lip or labellum

**Column** Unlike other flowers, both the male and female reproductive organs (stamen and pistil) are combined into a single column (gynostemium)

**Operculum** The Column is located under the Operculum but it isn't visible on many species of orchid.

**Tepal** A name that refers to both petals and sepals.

Orchid flowers are bilaterally symmetrical (left and right half are symmetrical) and will bend so that the labellum is pointing downwards (except for those rare species where the labellum points straight up). A typical orchid flower can be seen in figure 16 although one of the common variations has the lateral sepals fused together.

At this stage, our application only has a means for creating petals, sepals and specific types of labellum; the bilateral symmetry must be enforced by the user. A library of flower components that can be added to the model and a means for auto-arranging tepal spacing as in [7] and [8] respectively, would be ideal.

### 8.1 Floral Structure

One of the problems with the flower modeling technique in [8] is that by drawing each individual flower component individually, the user loses an appreciation of how all the components relate to each other. For example, if the user was

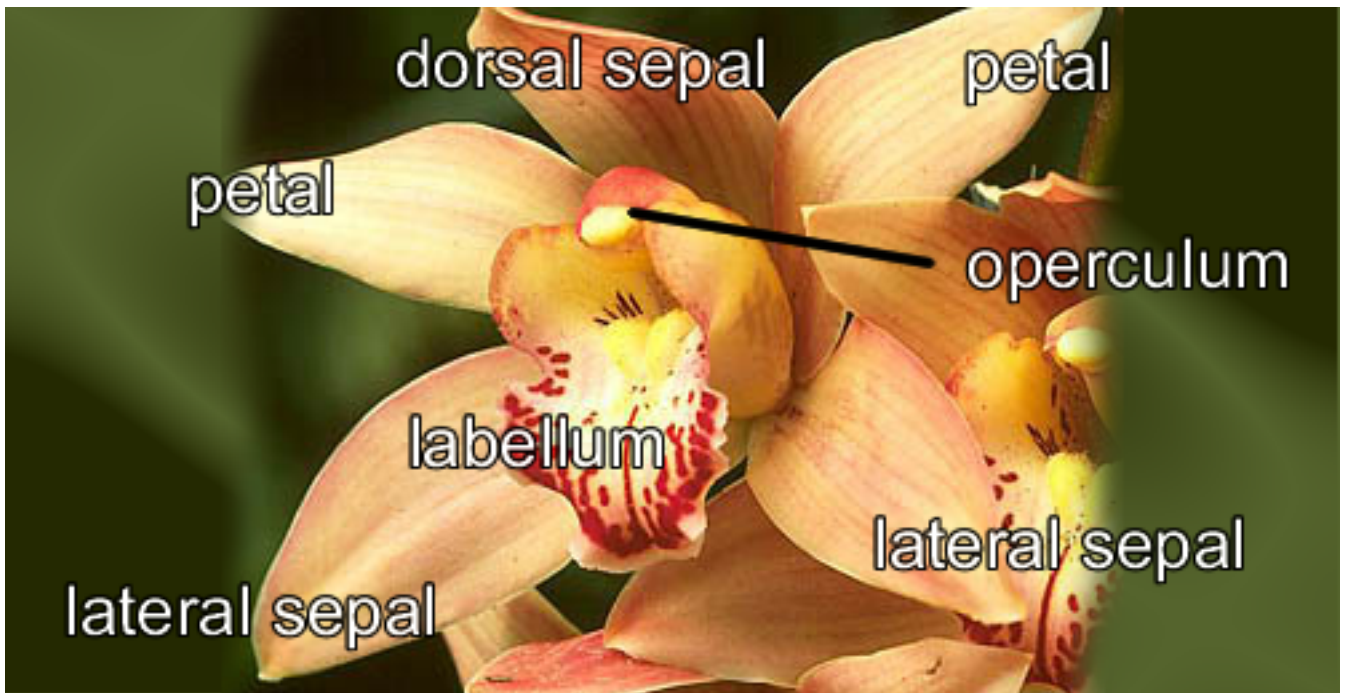


Figure 16: This photograph specifies the prominent parts of an orchid flower

to create a petal and a sepal in isolation to each other, the user would lose an appreciation of scale and orientation; how much larger or smaller the petal is from the sepal and how well it relates to the rest of the flower. Ideally the user can draw the flower in its entirety as if drawing a flower on a piece of paper.

[7] assists by allowing the user to draw plants and their flowers all in one complete sketch. The user then replaces each sketched 2D plant component with a 3D equivalent. The benefit of this approach is that the user can ensure that the 3D model preserves the scale and orientation of the 2D sketch that it is replacing.

The floral arrangement (of petals and sepals etc.) can be modified in [8] using a 2D representation and in [7] by directly interacting with the 3D objects.

Our approach is to generate the 3D model of the flower components as soon as the initial sketch is drawn. These 3D components can then be directly interacted with by moving them about the floral structure or by curving and folding them.

We exploit the flower modelling domain by assuming that all petals and sepals are anchored to the flower head. Because the user's stroke is unlikely to anchor to the flower, the sketch and subsequent 3D model can be 'snapped' to the centre. This domain knowledge is leveraged when editing the floral arrangement because flower components will always have an end anchored to the flower head. This assists in altering the position of the petals, sepals and labellum because they can be rotated about this anchor point.

## 8.2 Onion Skinning

As observed in user tests discussed in section 2.3, participants tend to sketch in layers by drawing foreground objects first and background objects last.

We propose a modelling method whereby the user draws each of the orchid components (petals, sepals, labellum) as a separate layer. When drawing components of a particular layer, the other layers become transparent, bringing focus to the current layer. This way the user can easily distinguish between the different components types but still appreciate how the active components relate to the rest of the orchid flower (figure 17).

This could be beneficial for editing features such as automatic spacing of components within a layer, or modifying strokes that affect all the components of a particular layer. All editing effects would be in context with the rest of the flower.

## 8.3 Tepal Repositioning

The repositioning of orchid tepals after they have been drawn can be achieved by the user by selecting and dragging the target tepal with the mouse cursor. We have implemented a virtual trackball approach [20] for the rotation, allowing tepals to be rotated to any angle about the flower head (origin). We suspect though, that this flexibility is difficult to control because it depends on an object's orientation; so rather than being able to rotate it on six axes (anchored at the origin) we suggest it may be better to exclude the ability to 'twist' it. Note that this isn't a corkscrew effect but a width-wise rotation. A four button control, either via keyboard or the GUI, could be a suitable alternative to the semi-unintuitive mouse controlled virtual trackball. How-

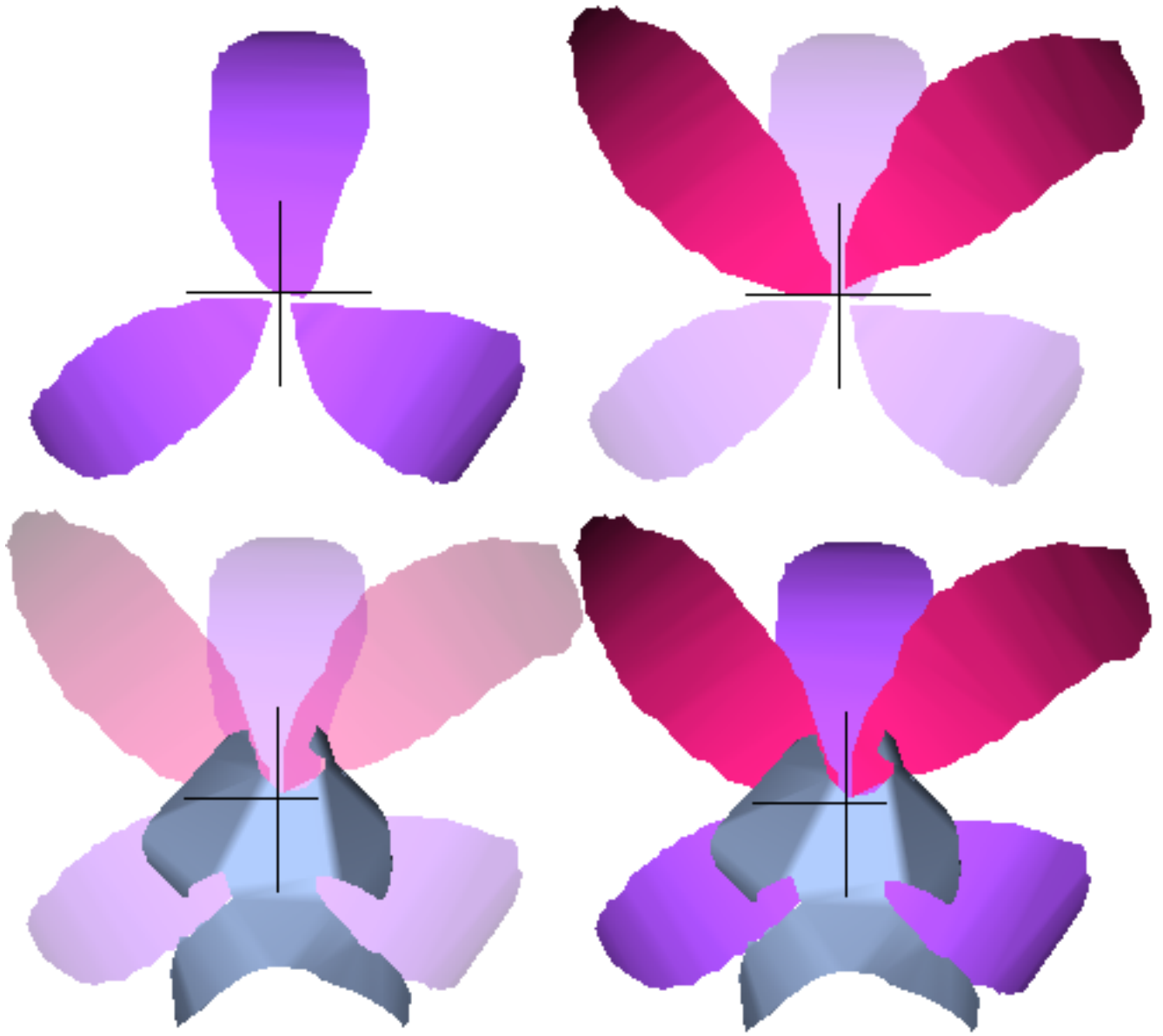


Figure 17: This series of images show how the onion skinning approach can be used to model an orchid. It separates each layer, making other layers transparent in order to bring focus to the current layer.



ever, this should remain as a ‘power feature’ for experienced users.

The virtual trackball utilises the domain specific assumption that the petals are attached to the flower head and thus rotate about it. Also required is the ability to ‘snap’ the user’s tepals to the flower head in the event of them not being able to draw it exactly in the correct spot. This requires a simple translation of the tepal by a vector formed between the origin and the closest point on the tepal.

## 9. DISCUSSION

By adopting shape recognition theory and a paper sculpting metaphor, we believe we can facilitate the task of creating and manipulating the flat surfaces required by 3D flower models. However, there are still some points that should be addressed with regards to expanding the functionality and highlighting some of the issues with the interaction strategies. Once resolved, there is potential for paper sculpting techniques to be useful in other modelling applications.

Our usability studies suggested that a paper folding interface offered an approach that was not just easy to use but enjoyable. Currently our tool facilitates the ability to ‘cutout’ a surface, fold it about automatically generated axes and rearrange to orchid floral structure using domain specific knowledge.

There are a number of extensions that can be implemented to enhance the application. The most important one, as identified in the user study of section 7, is a means for the user to define their own fold axis. This would be the equivalent of creasing paper and then folding it about the crease, even if that crease is curved. However, there is a question of whether this extra level of control would marginalise the existing automatically generated foldable areas. We suspect not as there is often a need to separate simple controls from ‘power features.’ What would be more valuable in this regard, is a refinement in the way the fold axes are determined.

Section 3 mentions research that incorporates many different curving strategies. By increasing the level of detail as discussed in section 6.6, curving strategies that involve displacing the surface vertices can be used. Localised and global curvature modifications to a surface are used to good effect in [8] and [7] where inputted pen strokes depict curvature changes. For subtle curvature that cannot be achieved by vertex displacement, bump maps would provide an effective solution.

Additional curvature modifications have a flow on effect that affects the ability to fold the surface further. For example, take a piece of A4 paper and curve it in the length-wise direction like a sine wave oscillation. It would now be incredibly difficult to fold the paper along an axis that extends length-wise. However, in many instances, it is impossible to flatten out orchid petals, especially the labellum, without splitting it. Real orchid petals can form a curvature that is very difficult to sculpt with paper unless one was to ‘collapse’ the paper by crumpling it or by using multiple tiny zigzag-like folds. Modelling such orchid surfaces would therefore break the paper sculpting metaphor because simple paper folds maintain surface area. This demonstrates the need to

be able to ‘do the impossible’ and allow for non paper-like folding so that the surface can take a shape that reflects that of many orchid labellums. With our current implementation the fold axis is a straight line, so the possibility of a curved axis that follows the curvature of the surface should be investigated.

Such observations suggest it is not necessary to be too concerned with surface area preservation. However, too greater warping does detract from the intuitiveness of the interface. Our current rotation implementation stretches the foldable area and becomes quite noticeable for angles beyond 90°. Divergence free surface deformation and incompressible materials need to be further researched.

The paper folding metaphor may be useful for domains beyond orchid modelling. However, because such models are likely to be more complex it would be unreasonable to expect the user to first conceptualise the model as a flattened out piece of paper. Instead many simpler surfaces could be ‘glued’ together like in the Teddy application. Another technique may be to keep retracing the silhouette like in [16] and [5] to make the surfaces progressively larger.

## 10. CONCLUSION

We have shown that paper sculpting is a promising metaphor to assist in the creation and manipulation of complex surfaces such as those used in flower modelling. By utilizing geometric properties and the sculpting qualities of paper, it is possible to make a more intuitive 2D to 3D mapping. Traditional 3D modelling tools are complex so such techniques are important to assist both novice and expert users to rapidly create complex models.

We have implemented the cutting and folding metaphors but there are a multitude of other paper sculpting techniques that can enrich the user interaction possibilities. As suggested by the user tests, the ability to define fold axes is the next most important addition.

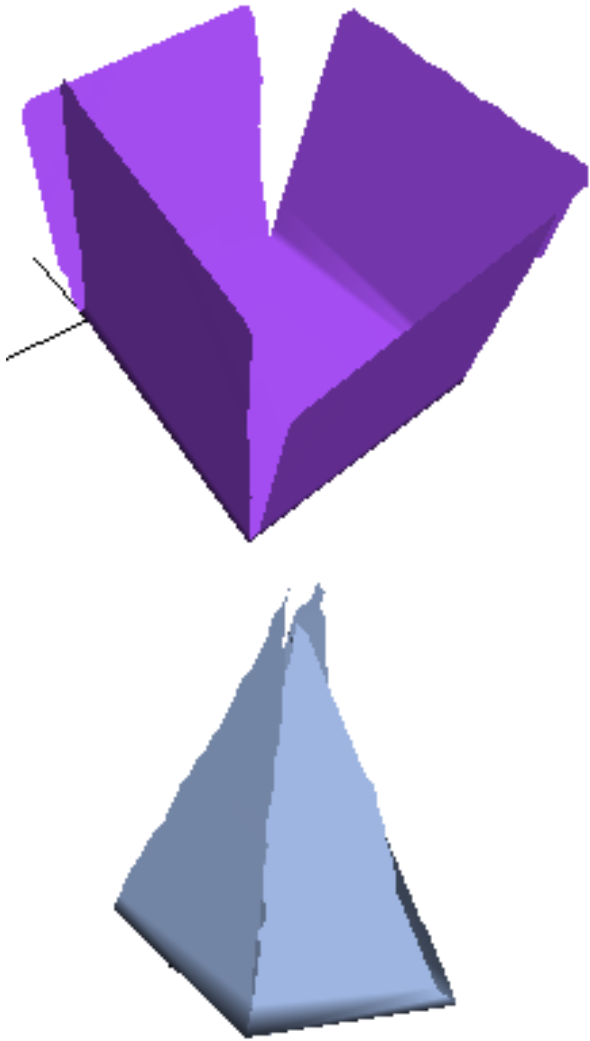


Figure 18: Cardboard like folds can be used to construct objects such as boxes and pyramids



Figure 19: Two orchid flowers that were modelled using our tool.

## 11. REFERENCES

- [1] Fabricio Anastacio, Mario Costa Sousa, Faramarz Samavati, and Joaquim A. Jorge. Modeling plant structures using concept sketches. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 105–113, New York, NY, USA, 2006. ACM Press.
- [2] Paul Bourke. Triangulate. Retrieved 2007 from <http://local.wasp.uwa.edu.au/~pbourke/papers/triangulate/index.html>.
- [3] Linda Fortner. Anatomy of an orchid flower. Retrieved 2007 from [http://www.orchidlady.com/pages/encyclopedia/orchid\\\_flower\\\_anatomy.html](http://www.orchidlady.com/pages/encyclopedia/orchid\_flower\_anatomy.html).
- [4] J. Halverston. The first pictures, perceptual foundations of paleolithic art. *Perception*, 21:389–404, 1992.
- [5] Jing Hua and Hong Qin. Free-form deformations via sketching and manipulating scalar fields. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 328–333, New York, NY, USA, 2003. ACM Press.
- [6] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [7] T. Ijiri, S. Owada, and T. Igarashi. Seamless integration of initial sketching and subsequent detail editing in flower modeling. *Computer Graphics Forum*, 25(3):617–624, 2005.
- [8] Takashi Ijiri, Shigeru Owada, Makoto Okabe, and Takeo Igarashi. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. *ACM Trans. Graph.*, 24(3):720–726, 2005.
- [9] Pourang Irani and Colin Ware. Diagramming information structures using 3d perceptual primitives. *ACM Trans. Comput.-Hum. Interact.*, 10(1):1–19, 2003.
- [10] Paul Jackson. *The Art and Craft of Paper Sculpture*. Apple Press, 1996. Quarto Publishing Plc.
- [11] Levent Burak Kara, Chris M. D'Eramo, and Kenji Shimada. Pen-based styling design of 3d geometry using concept sketches and template models. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 149–160, New York, NY, USA, 2006. ACM Press.
- [12] O. Karpenko, J. Hughes, and R. Raskar. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum*, 21(3):585–594, 2002.
- [13] Olga A. Karpenko and John F. Hughes. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Transactions on Graphics*, 25(3):589–598, July 2006.
- [14] D. Marr and H. K. Nishihara. Representation and Recognition of the Spatial Organization of Three-Dimensional Images. *Proceedings of the Royal Society of London, Series B*, 200:269–294, 1978.
- [15] Lars Mundermann, Peter MacMurchy, Juraj Pivovarov, and Przemyslaw Prusinkiewicz. Modeling lobed leaves. *Computer Graphics International*, pages 60–65, 2003.
- [16] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1142–1147, New York, NY, USA, 2005. ACM Press.
- [17] Department of Geomatics: The Univeristy of Melbourne. Line simplification algorithms. Retrieved 2007 from <http://www.sli.unimelb.edu.au/gisweb/LGmodule/LGSimplification.htm>.
- [18] Makoto Okabe, Shigeru Owada, and Takeo Igarashi. Interactive design of botanical trees using freehand sketches and example-based editing. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 18, New York, NY, USA, 2006. ACM Press.
- [19] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990.
- [20] Burkhard Wuensche. Department of computer science. Retrieved 2007 from <http://www.cs.auckland.ac.nz/compsci372s1c/burkhardLectures/>.
- [21] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 9, New York, NY, USA, 2006. ACM Press.