Animated Surface Pasting

Clara Tsang and Stephen Mann
Computing Science Department
University of Waterloo
200 University Ave W.
Waterloo, Ontario
Canada N2L 3G1

e-mail: clftsang@cgl.uwaterloo.ca, smann@cgl.uwaterloo.ca

Abstract

In computer animation, the animator spends a lot of time modeling. Surface pasting is a composition method in which a feature surface is attached to a base surface to provide details on the underlying surface. With surface pasting, the user can quickly and easily model faces and other objects with small details. Thus, surface pasting may be a useful animation technique because it reduces the modeling time. However, it is unclear whether pasted surfaces will have problems such as distortion in animation. In this paper, we generated a key-frame animated short to see how pasted surfaces behave in animation. In general, while pasted surfaces behaved as desired in animation, there are some situations where they behave poorly. Most of these situations can either be avoided or they can be fixed with simple intervention by the animator.

Keywords: Surface Modeling, Key-frame Animation

Introduction

Surface pasting is a new method for modeling. With this technique, a user can model, for example, a puppet by selecting features from a database and pasting them on the base model without having to model all the details of each feature. Previous work has shown surface pasting to be a useful modeling technique. In the animation process, an animator spends a large amount of time modeling. Surface pasting can reduce modeling time because the animator can access a database of features, and thus does not need to remodel all the detailed features.

Side Effects Software Inc.[1] has recently incorporated surface pasting technique into *Houdini*, but the method is new enough that Houdini users have had little time to use and test the technique, and it is unclear how pasted surfaces will behave at the inbetween frames in a key-frame animation system. Potentially, the pasted surfaces may distort or separate from the base surface, or they may not resize proportionally to the size changes of the base surface. The goal of this paper is to investigate the behavior of pasted surfaces and features at inbetween frames, and determine if surface pasting is a useful technique in animation. To test how pasted surfaces behave, we implemented a simple key-frame animation system to animate a puppet modeled with surface pasting.

[1]Houdini is trademark of Side Effects Software Inc.

Tensor Product Surfaces

Surface pasting is based on tensor product B-spline surfaces. A tensor product surface is determined by a rectangular mesh of three-dimensional control points, where a tensor product B-spline surface $S(u, v)$ can be expressed as a double summation:

$$S(u, v) = \sum_{i=0} \sum_{j=0} P_{i,j}\, B_i^m(u) B_j^n(v)\,.$$

Here, the $P_{i,j}$ are the control points and $B_i^m(u)$ and $B_j^n(v)$ are univariate B-splines basis functions. The shape of the tensor product B-spline surface is determined by control points $P_{i,j}$, and one method of modeling with tensor product surfaces is to interactively manipulate these control points (see [3] for further discussion of tensor product B-splines).

Surface Pasting

Surface pasting is a process in which a feature surface is attached on a base surface as a detail. The base surface can be a single surface or a composite (pasted) surface. After pasting, the original underlying surface is not changed, while the feature surface retains the character of its unpasted shape and reflects the topography of the underlying surface as well. The surface pasting method was developed by Barghiel [1, 2] from a technique proposed by Forsey and Bartels [4]. This section first presents the diffuse coordinate system and Greville Displacement B-Splines of Barghiel, and outlines the procedures of surface pasting.

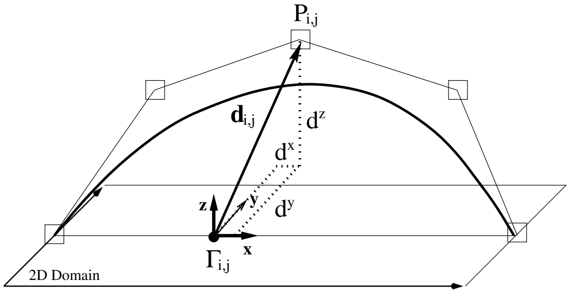


Figure 1: Greville Displacement B-Spline

In a diffuse coordinate system there is a diffuse coordinate space (*DCS*) and each control vertex has its own associated local coordinate frame $\mathcal{F}_{i,j} \in DCS$. Each control vertex can be written as an origin $O_{i,j}$ plus an offset vector $\boldsymbol{d}_{i,j}$ in the diffuse coordinate space:

$$
\begin{aligned}
S(u,v) &= \sum_{i=0}\sum_{j=0} P_{i,j}\, B_i^m(u) B_j^n(v) \\
&= \sum_{i=0}\sum_{j=0} (O_{i,j} + \boldsymbol{d}_{i,j}) B_i^m(u) B_j^n(v)
\end{aligned}
$$

For each control vertex $P_{i,j}$ of a B-spline surface, there is an associated domain point, called a *Greville point*, where the surface point at this domain point is maximally influence by $P_{i.j}$ [5]. The Greville point $\gamma_{i,j}$ associated with control vertex $P_{i,j}$ is defined as a pair of *Greville abscissae*, one for each parametric direction:

$$\gamma_{i,j} = (\gamma_i, \gamma_j)\,,$$

where the Greville abscissae γ_i and γ_j are the average of d (surface degree) knots in parametric directions i and j, respectively.

The two-dimensional Greville point (and the entire feature domain) can be embedded into a three-dimensional space by adding an additional coordinate (e.g., 0) to each point. Then the three-dimensional Greville point is

$$\Gamma_{i,j} = (\gamma_{i,j}, 0) = (\gamma_i, \gamma_j, 0)$$

The three-dimensional Greville point $\Gamma_{i,j}$ is chosen to be the origin $O_{i,j}$ of the local frame for $P_{i,j}$. Then the displacement vector $\boldsymbol{d}_{i,j}$ will be a vector from the origin $\Gamma_{i,j}$ to the control vector $P_{i,j}$:

$$\boldsymbol{d}_{i,j} = P_{i,j} - \Gamma_{i,j}\,,$$

which is called the *Greville displacement*. Thus, the diffuse representation of a B-spline point is

$$S(u,v) = \sum_{i=0}\sum_{j=0} (\Gamma_{i,j} + \boldsymbol{d}_{i,j}) B_i^m(u) B_j^n(v)$$

as illustrated in Figure 1.

The basis $(\boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{z})$ for the local frame is constructed with $\boldsymbol{x}$ and $\boldsymbol{y}$ as the two parametric directions of the embedded domain, and $\boldsymbol{z} = (0,0,1)$. The Greville displacement $\boldsymbol{d}_{i,j}$ can be expressed as a linear combination of the coordinate vectors of the local frame:

$$\boldsymbol{d}_{i,j} = d_{i,j}^x\, \boldsymbol{x} + d_{i,j}^y\, \boldsymbol{y} + d_{i,j}^z\, \boldsymbol{z}$$

where $d_{i,j}^x$, $d_{i,j}^y$ and $d_{i,j}^z$ are the x, y and z components of the Greville displacement vector, respectively.

Figure 2 shows the process of surface pasting. There are at least two surfaces involved in the pasting procedure: a *base surface* S_B on which to paste, defined over the base domain, and a *feature surface* S_F that pastes onto S_B, defined over the feature domain. Surface pasting is an operation that manipulates the location of the feature's control vertices, provided the displacement vectors remain unchanged.

The process of pasting a feature surface onto a base surface starts with mapping the feature domain into the base domain by a transformation T. This transformation T is used to determine the location and the size of the feature surface relative to the base surface.

After mapping the feature domain into the base domain, the origin $\Gamma_{i,j}$ of the feature local frame becomes $T(\gamma_i, \gamma_j) = (\gamma_i', \gamma_j')$. The base surface point $S_B(\gamma_i', \gamma_j')$, which is the image of the mapped $\Gamma_{i,j}$, is used as a new origin for positioning the feature displacement vector $\boldsymbol{d}_{i,j}$ during pasting. The basis $(\boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{z})$ maps to $(\boldsymbol{x'},\, \boldsymbol{y'},\, \boldsymbol{z'})$ where $\boldsymbol{x'}$ and $\boldsymbol{y'}$ match the embedded feature domain's parametric directions, and $\boldsymbol{z'}$ remains perpendicular to the embedded feature domain.

Before the feature displacement vector $\boldsymbol{d}_{i,j}$ is applied onto the mapped origin $S_B(\gamma_i', \gamma_j')$, a local base coordinate frame $\boldsymbol{\mathcal{F}_B}$ at $S_B(\gamma_i', \gamma_j')$, where $\boldsymbol{\mathcal{F}_B} = \{S_B(\gamma_i', \gamma_j'), \boldsymbol{v}(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})\}$ has to be constructed. To reflect the curvature of the base surface at the mapped origin $S_B(\gamma_i', \gamma_j')$, the components $\boldsymbol{i}$ and $\boldsymbol{j}$ of the vector $\boldsymbol{v}$ are calculated as the partial derivatives of the base surface S_B at $S_B(\gamma_i', \gamma_j')$ in the $\boldsymbol{x'}$ and $\boldsymbol{y'}$ directions ($\frac{\partial S_B}{\partial x'}$ and $\frac{\partial S_B}{\partial y'}$), respectively, and $\boldsymbol{k}$ is the cross product of $\boldsymbol{i}$ and $\boldsymbol{j}$.

Finally, the pasted control vertex $P_{i,j}'$ results from embedding the displacement vector $\boldsymbol{d}_{i,j}$ of the feature into the local base coordinate frame $\boldsymbol{\mathcal{F}_B}$. The pasted control vertex $P_{i,j}'$ is calculated by a point-vector addition performed in the given frame context:

$$
\begin{aligned}
P_{i.j}' &= S_B(\gamma_i', \gamma_j') + \boldsymbol{d}_{i,j} \\
&= S_B(\gamma_i', \gamma_j') + d_{i,j}^x\, \boldsymbol{i} + d_{i,j}^y\, \boldsymbol{j} + d_{i,j}^z\, \boldsymbol{k}
\end{aligned}
$$

Note that the feature surface and the base surface do not meet with C^0 continuity. However, they do meet with

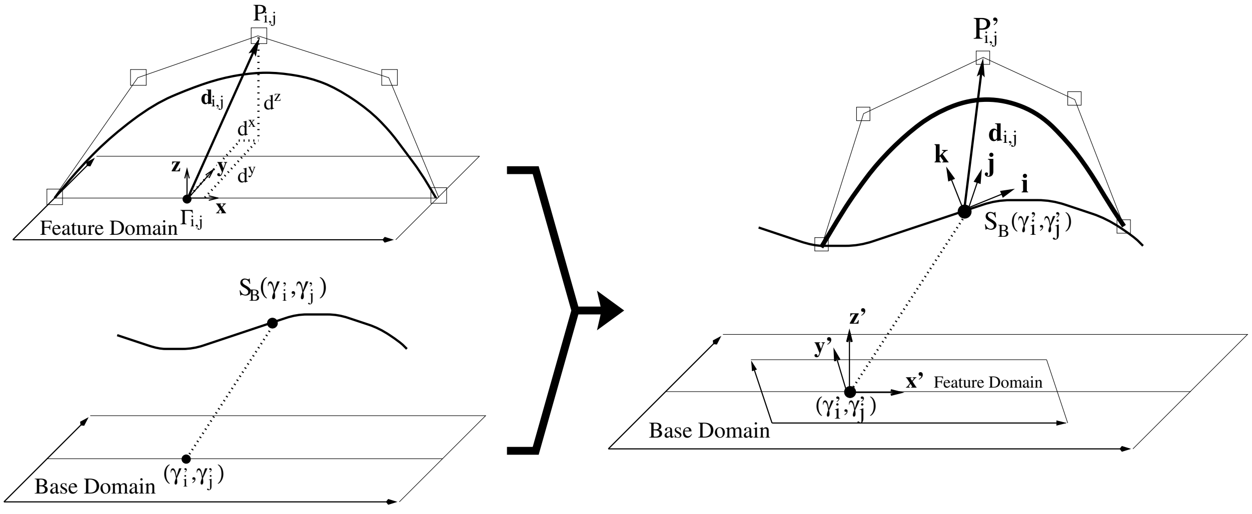


Figure 2: Surface Pasting Procedure

ε-C^0 continuity, and the feature can be placed upon the base so that ε (the maximum distance between the feature boundary and the base surface trim curve) can be as small as desired. The feature can also be pasted onto the base so as to minimize the discrepancy with the normal field of the base surface (along the trim curve). Thus, while the two surfaces fail to meet with mathematical C^0 continuity, in practice they appear to meet with G^1 continuity.

Our goal in this work was to test pasted surfaces in a key-frame animation system, to gather empirical evidence on how pasted surfaces behave at inbetween frames, and to decide what (if any) changes should be made to improve the behavior of pasted surfaces at inbetween frames. The remainder of this paper describes our pasted model and animation, and analyzes how these pasted surfaces behaved in the animation.

Pasted Details

For our model, we built a simple puppet with arms, hands, and facial features. The eyes, ears, nose and lips are pasted on the skin of the head. The ears, nose and lips are each a single surface while each eye is formed of three surfaces using hierarchical pasting. Figure 3(a) shows the three layers of the eye without pasting and Figure 3(b) shows the result of the eye after pasting. The lips are formed by one surface. There are five lip models, which we used to animate the lips for talking. To show the inner part of the mouth, part of the head surface (under the lips) has been trimmed.

On each hand, the fingers are modeled as two separate surfaces, one for the thumb and the other for the rest of the fingers. The fingers can perform a grasping motion by moving the domain of the finger surface. Note that this is a poor way to animate such a motion. Normally, you would model the fingers with a skeleton, and key-frame their joint angles. We modeled this motion by sliding the domain only because we wanted to test moving features in an animation system.

Description of Animation

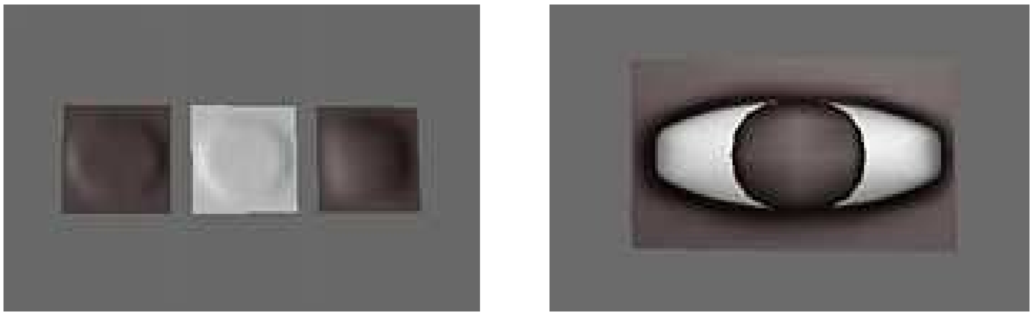
The idea of this animation is based on The Scream by Edvard Munch. The animation starts with a man standing on a bridge. The man seems to be looking for something. Suddenly, there is a bright. The man does not know what to do. He just holds his head with his hands. Alas, his head gets deformed because he squeezes too hard and his head does not return to its normal shape after he removes his hands. So, he grabs his head with his hands and stretches it until he restores its original shape.

For this animation, we built forty-two key-frames and generated one hundred and ninety-one inbetween frames. The total run-time of the animation is approximately thirteen seconds. Figure 4 shows the scream frame of this animation. In the following sections, the discussion is based on inbetween frames extracted from the animation.

How Pasted Surfaces Behave

The main task of this research is to test how pasted surfaces behave in animation. Thus, we have to look at the facial features, such as eyes, nose, ears and lips, and fingers to see how they behave when the head is deformed, and when the fingers move on the hands.

For the most part, the facial features behave properly when the head is deforming. Figure 6(a) shows the facial features in the original size and the head without any deformation. When the head is squished, the eyes, ears,



(a) Without Pasting (b) After Pasting

Figure 3: Eye Model

nose and lips narrow correctly as shown in Figure 6(b). Next, when the head is stretched sideways, the facial features are elongated in width and are shortened in height appropriately as shown in Figure 6(c). Finally, the heights of the facial features and the head are lengthened at the same time (see Figure 6(d)).

Although the facial features behave reasonably on head deformation, some problems did arise. Figure 5(a) shows the puppet with the normal head and wide opened mouth. However, there is a problem after the head of the puppet has been stretched sideways. Figure 5(b) shows that the lip sticks into the nose. This occurred because the height of the head has been reduced and every pasted surface has been compressed along the y-axis to such an extent that the lips touches the nose. Thus, an animator of pastes surfaces would have to watch for such problems when producing the animation, and possibly modify inbetween frames to avoid penetration of surfaces.

We also noticed a problem with the eyes. Normally, the pupil of the eye should be inside the white of the eye as shown in Figure 7(a) and Figure 7(c). However, in Figure 7(b) and Figure 7(d), the pupil of the eye is bigger than the white of the eye. This artifact occurred because the shape of the three eye surfaces are bumps. The displacement vectors of the second boundary layers are pointing outside. When the head is stretched sideways, the height of the eye surface is shortened and the displacement vectors point "outside" more. Since the three eye surfaces are similar in shape and pasted hierarchically, the effect of the displacement vector pointing outside is accumulated. Thus, the last pasted surface, the pupil, lays outside the white of the eye. Figure 7(e) illustrates this problem. On the top figure, it shows the white of the eye and the displacement vectors of the pupil on the unstretched head (which clearly project onto the white of the eye) while the bottom figure shows that the displacement vectors of the pupil point outside the white of the stretched eye.

A third problem we noticed was a distortion of the fingers during the grasping motion. The grasping motion is implemented by sliding the fingers on the hand. Unfortunately, the fingers enlarge when the fingers are slid to one side of the hand as shown in Figure 8(b) (Figure 8(a) shows the initial pose of the hand and the fingers).

The fingers get fatter because of the shape of the base surface – the hand. In the first version of our animation, we kept the width of the feature domain relative to the base domain constant. However, the hand has a non-uniform parameterization (i.e., equally spaced points in the domain mapped to non-equally spaced points in the range). Thus, as we slide the finger domain across the hand domain, the fingers will span different areas on the hand, the boundaries of the fingers get further apart, and the fingers get thicker.

Since the fingers get bigger because the width of the base surface covered by the feature surface is lengthened, we fixed this problem by shortening the width of the fingers domain. After experimenting, we found a constant factor that can be combined with the sliding offset to form a compression factor. This compression factor is used to rescale the finger domain when the fingers are sliding, with the result that the pasted fingers maintain nearly constant width while sliding. This method worked in our tests (see Figure 8(c)). However, in a real system, the software would need to be extended to maintain automatically the actual width of the fingers no matter where the fingers are on the hand.

Conclusions

In our animation, we found that in general the facial pasted features behave like we think and expect, and that pasted surfaces behave properly under deformation of a base surface. As the base stretches and deforms, the features stretch and deform in a similar fashion without separating from the base. However, there were some problems with the facial features. First, some of the features that did not touch on the undeformed head would interpenetrate on the deformed head. This is a problem with many animation techniques, and it highlights the importance of allowing the animator access to the inbetween frames to modify them as needed.

Second, although simple features behaved well under distortion, the one hierarchical feature (the eye) did not behave as well. As the eye narrowed, the pupil bulged beyond the white of the eye. This bulging was a result of a change in the directions of the normals to the sides of the eye that caused the feature to distort in an unwanted manner. This suggests that care should be taken when designing features, especially hierarchical features, to avoid pasting over regions of high curvature. Alternatively, we could use a different pasting process for the interior control points to minimize distortion.

The second test we conducted for animated features was to slide the fingers over the hand to make a grasping motion. Initially, we performed this motion by sliding the feature (finger) domain within the base (hand) domain, where we kept the relative sizes of the two domains constant. With this motion, the fingers became greatly distorted, increasing in width by more than a factor of two. An analysis of the base surface (the hand) revealed that this distortion problem was a result of the base surface having a non-uniform parameterization.

To fix this problem with the fingers, we rescaled the size of the finger domain within the hand domain, and were able to keep the fingers at nearly a constant thickness. One more general approach would be to allow the user to key-frame the domain width. However, it might also be possible to have the system automatically adjust the relative domain sizes in an attempt to keep the feature width constant. Note that a similar problem occurs when modeling with pasted surfaces. However, in a modeler, the user can easily adjust the features manually. In a key-frame system, it is more important to try to maintain constant (or key-framed) size at inbetween frames. As a side note, these size problems are alleviated if the base surface has a reasonable parameterization, so base surfaces should be designed that way when possible.

It is difficult to compare our work directly to other animated modeling techniques, but we expect it to compare well, as surface pasting should reduce modeling time and leave the animator free to concentrate on the animation motion. However, this can only be determined once surface pasting is fully integrated into an animation package and tested by production animators.

This animation is available on the Web at `http://www.cgl.uwaterloo.ca/Projects/Splines/pasting/` animate98/index.html. A full description of the animation, skinning, and key-framing software, as well as a further analysis of the pasted animation can be found in Tsang's masters thesis [6], which is also available on the web at `ftp://cs-archive.uwaterloo.ca/cs-archive/CS-98-19/`

Acknowledgments

Many thanks to Mark Riddell who drew the "Scream-like" background, helped sequence the animation and made the video titles, and to Patrick Gilhuly who recorded the video.

This research was funded by the Natural Sciences and Engineering Research Council of Canada.

*

References

[1] C. Barghiel. Feature oriented composition of b-spline surfaces. Master's thesis, University of Waterloo, Computer Science Department, Waterloo, Ontario, March 1994. (Available as Computer Science Department Technical Report CS-94-13).

[2] C. Barghiel, R. Bartels, and D. Forsey. Pasting spline surfaces. In L. Schumaker, M.Daehlen, and T.Lyche, editors, *Mathematical Methods for Curves and Surfaces*, pages 31–40. Vanderbilt University Press, 1995.

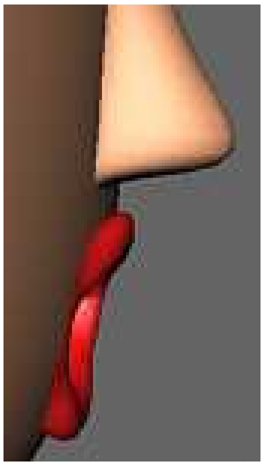
[3] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc., 1987.

[4] R. Bartels and D. Forsey. Spline overlay surfaces. Technical Report CS-92-08, University of Waterloo, Computer Science Department, Waterloo, Ontario, 1991.

[5] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Inc., 3rd edition, 1994.

[6] C. Tsang. Animated surface pasting. Master's thesis, University of Waterloo, Computer Science Department, Waterloo, Ontario, July 1998. (Available as Computer Science Department Technical Report CS-98-19).

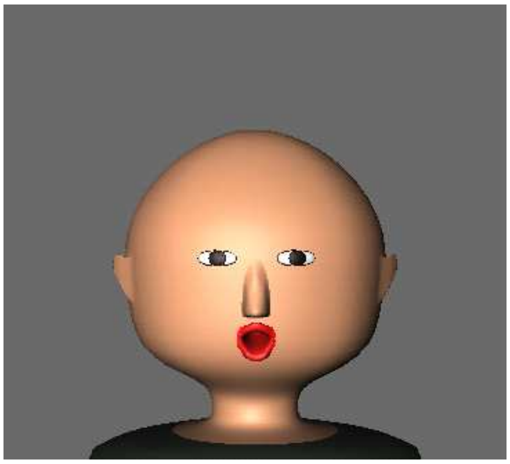
Figure 4: The Scream Frame



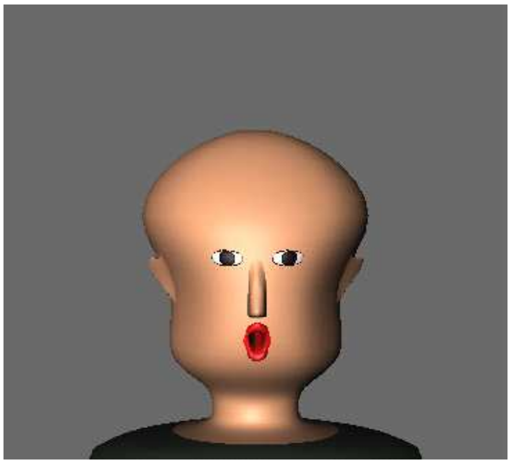
(a) Normal Head

(b) Stretching Head

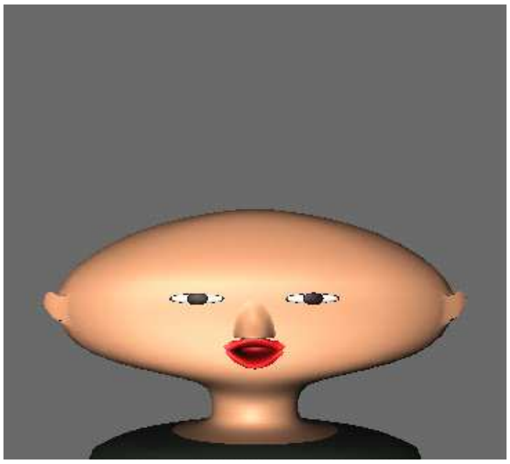
Figure 5: Modeling Problem



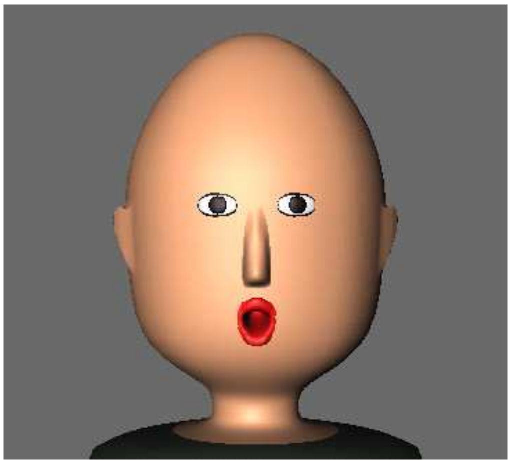
(a) Initial Pose



(b) Squishing Pose

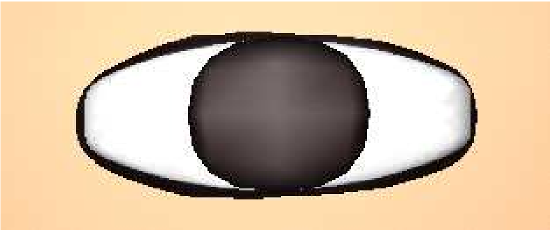


(c) Stretching Aside Pose

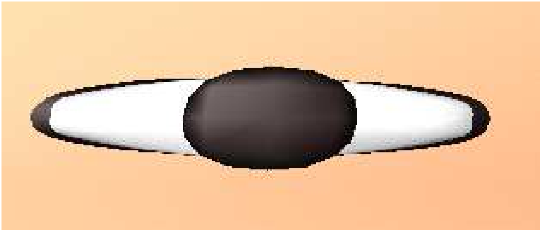


(d) Stretching Up Pose

Figure 6: Different Poses of Head and Facial Features



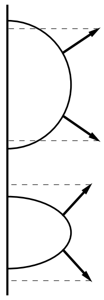
(a) Normal Head Front View



(b) Stretching Head Front View

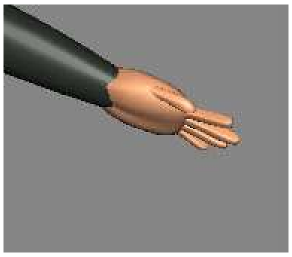
(c) Normal Head Side View

(d) Stretching Head Side View

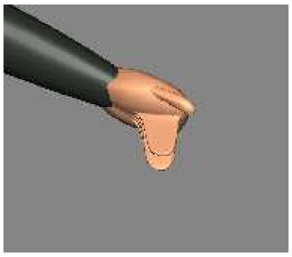


(e) Problem Illustration

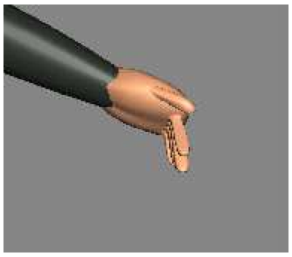
Figure 7: Bigger Pupil of Eye Problem



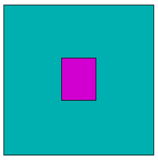
(a) Initial Pose



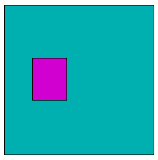
(b) Grasping Pose With Problem



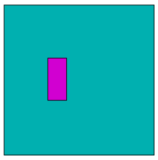
(c) Grasping Pose With Correction



(d) Initial Pose Domain



(e) Grasping Pose Domain With Problem



(f) Grasping Pose Domain With Correction

Figure 8: Fingers Distortion Problem