Making Extreme Caricatures with a New Interactive 2D Deformation Technique with Simplicial Complexes

Ergun Akleman*, James Palmer, Ryan Logan
Visualization Program
Department of Architecture
Texas A&M University

Abstract

In this paper, we present a new interactive 2D deformation technique to make caricatures with extreme exaggerations. In our technique, we use simplices as deformation primitives. User defined source and target shapes of each simplex primitive define local coordinates. These local coordinates determine a unique translation vector for any given point in 2D space. We also provide a set of blending functions to effectively interpolate these translation vectors from each simplex primitive to compute a combined translation. These combined translation vector are used to transform each vertex of a texture mapped grid. Deformation of the grid provide interactive deformation of the texture that is mapped to the grid.

Keywords: Applications in Art, Image Synthesis, Warping, Deformations, Caricature.

1 Introduction

The human mind has an amazing ability to distinguish between hundreds of thousands of faces it encounters throughout a lifetime of interaction. We can subconsciously detect the slightest variation in proportion, color or even movement of facial features. The success of caricature as an art comes from this ability of the human mind. We use the small differences in every face to distinguish one face from another. Often these differences are not obvious. Calder et al. showed that it is possible to differentiate between subjects by exaggerating a face's relative difference in proportion from a so-called average face [15]. According to Mauro et al. this effect is due to the fact that faces are encoded based on their deviation from a norm [17].

*Address: 216 Langford Center, Visualization Laboratory, Texas A&M University, College Station, Texas 77843-3137. emails: ergun@viz. tamu. edu, james@viz. tamu. edu, stuck@viz. tamu. edu. Supported in part by the Texas A&M, Interdisciplinary Research Initiatives Program.

The success of caricaturists come from their ability to conciously identify any deviation from a normal face. The most important part of caricaturing a given face is the identification of the facial features which are different than most people. Akleman showed that when using a 2D deformation tool it is easy to identify the features that are different from a normal face [1]. The identification procedure with a 2D deformation tool consists of three stages: (1) Start with an extremely simple template, (2) Exaggerate only one feature at a time, (3) If exaggeration does not create a likeness, go to the opposite direction. The exaggerations that lead to a likeness represent the deviations from a normal face. Akleman called this process discovery of proportions.

Once proportions are discovered, caricature can be made by simply exaggerating the features that are different than normal. In 2D, caricatures can be obtained with a 2D deformation tool by deforming photographs.

Figure 1: Extreme exaggerations can work if and only if the deviations from normal face are correctly identified.

The caricaturists' goal is to make extreme exaggerations, since such exaggerations are the ones that create dynamic and interesting caricatures. It is important to note that extreme exaggerations can create a likeness if and only if deviations from the normal face are *correctly* identified. Figure 1 illustrates the importance of correctly identifying deviations. The top-right image in Figure 1 is a real picture of President Ronald Reagan. The top-left image was created by slightly exaggerating the features in wrong direction. It is clear from the image that it is almost impossible to recognize the President from this image. On the other hand, the extreme exaggeration shown in the bottom-right image still resembles him since exaggerations are in the correct direction.

As we have discussed above, it is important to correctly identify the features that distinguish one face from another. The identification process depends on the intuitiveness and speed of deformation tools. There are various interactive deformation tools, but they are not intuitive for dicovering proportions. Beier and Neely's Feature-Based Image Metamorphosis (FBIM) technique [3] is the most intuitive 2D deformation method we have found. However, it uses inverse mapping and does not support realtime deformations.

In this paper we develop a new technique to provide both intuitive and interactive deformations. Our new technique is motivated by Crespin's Implicit Free-Form Deformations (IFFD) [7] and it is a generalization of FBIM technique [3]. In addition to FBIM's line primitives, we also use points and triangles as deformation primitives. Our deformation technique can scale easily to n dimensions since it is based on simplicial complexes [9]. A major advantage of using triangle primitive in addition to line primitive is that triangles can effectively express a shear from the source image to the target image. This power is essential to discover the correct proportions and to create extremely exaggerated caricatures.

We have implemented the system for 2D deformations by using C++, Fltk and OpenGL. In our system, user defined source and target shapes of each simplex define local coordinates. These local coordinates determine a unique translation vector for any given point in 2D space. We also provide a set of blending functions to effectively interpolate these translations. The interpolated translation vectors are used to transform each vertex of a texture mapped grid. Instead of using FBIM's inverse mapping technique, we use forward mapping functions to manipulate the geometry of grid. The deformation of the grid also deforms the texture that is mapped to the grid. This approach takes the advantage of 3D acceleration hardware that supports texture mapping.

2 Previous Research

Various fields such as cognitive science, computer graphics, computer vision and psychology have recently started to investigate faces and caricatures [1, 21, 13, 20, 22, 23, 17, 24, 25, 27, 26, 28].

Recently it has been determined that faces are coded with respect to a general model. The way we interpret a face, depends on the way we, as individuals, categorize facial features [19]. Studies have been done between faces of the same ethnicity, across ethnic boundaries and even across separate species [16]. In order for a face to stand out, it must compete with other faces grouped within its category. Facial expressions have also been shown to follow a similar pattern [15], and the expressions can be enhanced for recognition very effectively with the use of caricature [12]. The individuality of a face helps us to store and retrieve information related to its owner or associated subject matter [14]. We can increase this connection of visual stimulus and resulting association by caricaturizing a face. Caricature is a process in which a face's distinguishable features are amplified and average features are minimized [13]. This coding of facial features seeks to bring out the underlying character of a face and therefore make it more unique. This process also seems to support long-term recognition and retrieval of a face. Caricatures have even been considered for forensic purposes in finding suspects [18].

The process of automating caricature has already been researched and implemented with both line drawings and later with continuous-tone representations [13]. Both attempts were extremely successful at capturing the seemingly subjective analysis methods involved in caricature, however, both attempts had major drawbacks in the initial step of identifying specific features of the face. In both cases, crucial points on a face were identified and then compared to a face representing the mean in order to determine the amount of exaggeration. These mean faces vary and are typically divided based on racial boundaries due to generalized differences in proportion between race. Both of these methods require large databases of random faces with hundreds of features identified in everyone. By limiting the number of features accounted for, the process could be significantly streamlined. Instead of treating hundreds of individual features as separate entities, features could be divided into major regions accounting for mouth, eyes, nose etc. By limiting variables, great efficiency can be achieved and the process becomes much less tedious.

Akleman suggested an art oriented approach to create caricatures. He developed a procedure to discover the proportions of a given face. He showed the effectiveness of this procedure by creating various caricatures of people including some presidents of United States of America and some Computer Graphics researchers [1]. Akleman's deformation based technique is based on deformations. His technique requires intuitive and interactive deformation tool for effective and correct discovery of proportions.

The first deformation operations in computer graphics were introduced by Alan Barr in 1984 [2]. In 1986, Sederberg and Parry introduced "Free-Form Deformations" [10]. The Free-Form Deformation technique has been a popular technique that has been extended and adapted in a number of ways. The basic deformation process includes: initially positioning the deformation object, computing the local coordinates of the effected object (freezing), using the tool to deform the object, and finally computing the new coordinates of the effected object based on the local coordinates of the deformation object (melting). The last two steps are repeated interactively until the user is satisfied. The deformation is based on a lattice structure composed of tricubic Bezier patches [11].

Over the years a number of variations on this work have been introduced. These include Extended Free-Form Deformation (EFFD) [6], axial deformations [8], Scodef [4], Implicit Free-Form Deformations [7], etc. In 1999, Crespin introduced Implicit Free-Form Deformations (IFFD). IFFD provides a frame work in which most of the Free-Form Deformation techniques can be used but is based on deformation primitives defined by a local tool and a blending function [7]. Within the IFFD model, a set of n deformation primitives act on a model. Each deformation primitive is characterized by a local coordinate space, an invertible mapping function, an invertible transformation function and a potential function. These deformation primitives are then applied to geometry using a process similar to FFD: positioning the tool, freezing, deformation, melting, and combining [7].

Crespin's work in turn, inspired some of our initial investigations with using implicit functions and simplicial complexes to deform geometry. We also observed a number of similarities between our technique and the feature-based image metamorphosis technique introduced in 1992 by Beier and Neely [3].

Feature-Based Image Metamorphosis (FBIM) utilizes a series of intuitive lines to effectively warp or deform pixel location based on a technique called "field morphing" [3]. Although the end transformation is in many respects different from a deformation, the underlying transformation algorithm shares many similarities to deformation algorithms. This technique has been used as a visual effect in a number of motion pictures as well as the Michael Jackson video "Black and White". The feature-based approach uses a distance and blending function to deform pixel location and could be considered basically implicit in nature.

An important difference between our technique and FBIM technique is that deformation objects are based on simplices. A simplex is a set of d+1 points whose convex hull has dimension d. The points of the simplex may exist in a space whose dimension is larger than d. In 2D, simplexes includes points, lines and triangles. A simplicial complex is composed of a number of simplices. The intrinsic dimension of the complex is the dimension of each simplex in the complex. The embedded dimension of the simplicial complex is the dimension of the space of the points in the simplicial complex.

Simplicial complexes represent a straightforward and well defined data structure that allow one to take advantage of linear programming methods for the solution of geometric problems, boundary evaluation, affine transformations, subdivision, and constructive solid geometry operators. Simplicial complexes also provide a very simple and general method for expressing geometry in n-dimensional space. [9]

3 Methodology

Our approach is based on functions that are constructed by the operations that are used in implicit surface construction. The overall deformation is described by a set of simplex pairs (point, line and triangle pairs). Each simplex pair consists of one source and one destination simplex. For each source and destination simplex a local coordinate is computed to describe the translation described by change of the position and shape of the simplex. In addition, for each simplex a weight function that is described by the distance to the simplex is given. Then, the translations are combined by using these weight function. The following subsection describes how local coordinates are computed for a given simplex.

3.1 Translation Described by a Simplex Pair

As we have discussed above, in order to describe a deformation, the user defines a set of simplex pairs. Each simplex pair consists of two simplices: the source simplex and the destination simplex. Each one of these pairs uniquely describe a translation vector for any given point in the source image. In this section, we provide the equations to compute the translation vector for a simplex pair.

The translation given by each type of simplex pair must be computed differently. The simplest of these simplices are point pairs. Let p_0 be the source point and p_1 be the destination point. Let p_{source} be a point in source image. Translation vector T is independent of p_{source} and it is given as

$$T = p_1 - p_0.$$

Let $p_{destination}$ be the position of the point p_{source} in destination image. The new position of any point in the destination image will simply be

$$p_{destination} = p_{source} + T = p_{source} + p_1 - p_0.$$

Note that a point simplex does not provide scaling and rotation.

Beier and Neely have already provided the equation for the translation described by line pairs [3]. Lines can describe scaling and rotation up to 360 degree. However, they cannot support shear transformation.

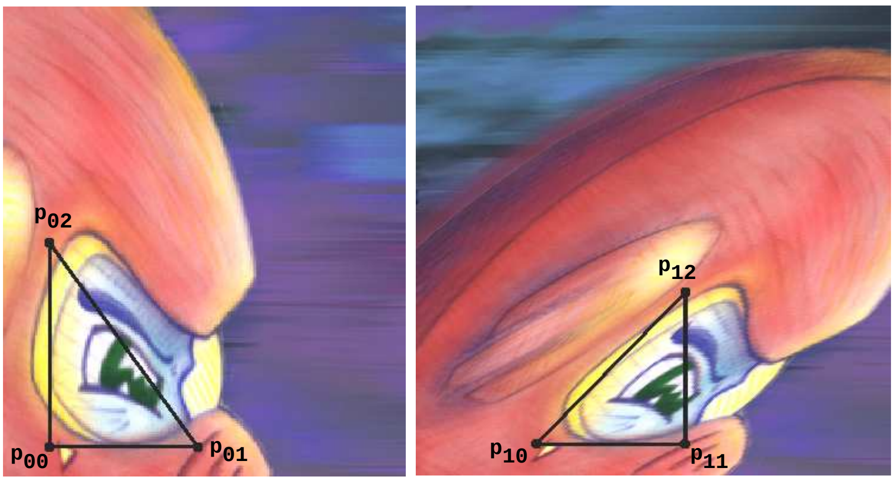

Figure 2: An example of shear transformation obtained by using triangle pairs.

Triangles provides shear transformation, scaling and rotation up to 360 degree. An example of shear transformation that is obtained by a triangle pair is shown in Figure 2. Let $p_{0,0}$, $p_{0,1}$ and $p_{0,2}$ denote three vertices of source triangle and $p_{1,0}$, $p_{1,1}$ and $p_{1,2}$ denote three vertices of destination triangle. In order to compute the translation, we first find the local coordinate described by the source triangle. When p_{source} is a point in the source image, the transformed local coordinates (x, y) of this point are computed by the following equations:

$$x = \frac{(p_{source} - p_{0,0}) \bullet v_2}{(p_{0,1} - p_{0,0}) \bullet v_2}$$

$$y = \frac{(p_{source} - p_{0,0}) \bullet v_1}{(p_{0,2} - p_{0,0}) \bullet v_1}$$

where

$$v_1 = (p_{0,1} - p_{0,0}) \times (p_{0,2} - p_{0,0}) \times (p_{0,1} - p_{0,0})$$

$$v_2 = (p_{0,2} - p_{0,0}) \times (p_{0,1} - p_{0,0}) \times (p_{0,2} - p_{0,0})$$

and $\bullet$ is scalar multiplication (dot product) and $\times$ is vectoral multiplication (cross product). Based on these local coordinates the new position of p_{source} at the destination image, is given by the following equation:

$$p_{destination} = p_{1,0} + x(p_{1,1} - p_{1,0}) + y(p_{1,2} - p_{1,0})$$

The translation vector T in this case is computed as a difference between source and destination:

$$T = p_{destination} - p_{source} = p_{1,0} + x(p_{1,1} - p_{1,0}) + y(p_{1,2} - p_{1,0}) - p_{source}$$

3.2 Combining a Set of Simplex Pairs

If there exists more than one simplex pair, the problem is to appropriately combine the translations decribed by each pair. In order to compute the combined translation, we simply calculate a weighted average of translations given by each simplex pair. Let T_i denote the translation vector function described by simplex pair i where $i = 0, 1, \ldots, n$. Then the combined translation vector function $T_{combined}$ is

$$T_{combined} = \frac{\sum_{i=0}^{n} w_i T_i}{\sum_{i=0}^{n} w_i}$$

where w_i is a positive valued function that is computed by using the distance from the source simplex of simplex pair i.

Let d_i be the distance from the source simplex of simplex pair i. Any monotone decreasing always positive function of d_i can be used as w_i. Examples of such functions are

$$w_i(d_i) = exp(-d_i/\mu_i)$$

$$w_i(d_i) = (1 - d_i/\mu_i)^2(1 + 2d_i/\mu_i) + a_i(d_i/\mu_i)^2(3 - 2d_i/\mu_i)$$

where μ_i is any positive real number and $1 > a_i > 0$. Larger μ_i values gives smoother warping effects. Beier and Neely uses the following function for w_i:

$$w_i(d_i) = \left(\frac{l_i}{a_i + d_i}\right)^{b_i}$$

where l_i, p_i, a_i and b_i are real constants. Beier and Neely suggest that the values of b_i in the range of $[0.5, 2]$ are the most useful values. The value of a_i must always be positive. Smaller values provide more precise control, larger values yield smoother warping effect. The scalar l_i can be used to give different importance to each simplex pair. In our system, we give all of the weights given above as options.

Distance function d_i will be computed differently for each type of simplex. For point simplex, d_i is the euclidean distance to the source point p_0. For the line simplex, d_i is a distance to the source line that is given by its two endpoints $p_{0,0}$ and $p_{0,1}$. This distance is computed as

$$d_i(p_{source}) = \begin{cases} |p_{source} - p_{0,0}| & \text{if } (p_{0,0} - p_{0,1}) \bullet (p_{source} - p_{0,0}) \geq 0, \\ |p_{source} - p_{0,1}| & \text{if } (p_{0,1} - p_{0,0}) \bullet (p_{source} - p_{0,1}) \geq 0, \\ z & \text{otherwise,} \end{cases}$$

where

$$z = |p_{source} - p_{0,0} - \frac{(p_{source} - p_{0,0}) \bullet (p_{0,1} - p_{0,0})}{|p_{0,1} - p_{0,0}|}|.$$

Figure 3 shows how to compute the distance to a triangle.

If we use only line primitives, FBIM's weight function and inverse maps instead of forward ones, our deformation technique will be exactly same as the FBIM technique. In other words, our technique can be considered as a generalization of the FBIM technique. Our generalization comes from (1) using triangle and point primitives in addition to line primitives, (2) using new weight functions and (3) using forward maps instead of inverse ones. By using triangles in addition to lines, we provide users with shear transformations in addition to translation, scaling and rotation. Because of the shear transformation, our technique is more intituitive for making extreme caricatures. And by using forward maps, we are able to develop an interactive system by taking advantage of texture mapping hardware.

4 Implementation and Results

The implementation of the technique is done using C++, Fltk and OpenGL. OpenGL takes advantage of 3D acceleration hardware that may be present. This allows us to use forward mapping functions to interactively manipulate texture mapped grids. Figure 4 shows an example of extreme caricatures created by using our system. It is interesting to compare this caricature with the caricatures in Figure 5. Akleman created these caricatures in Figure 5 with FBIM by deforming the same photograph of President Clinton [1]. It is clear that the new extreme caricature of President Bill Clinton is much more dynamic and 3D looking than his caricatures that are shown in Figure 5.

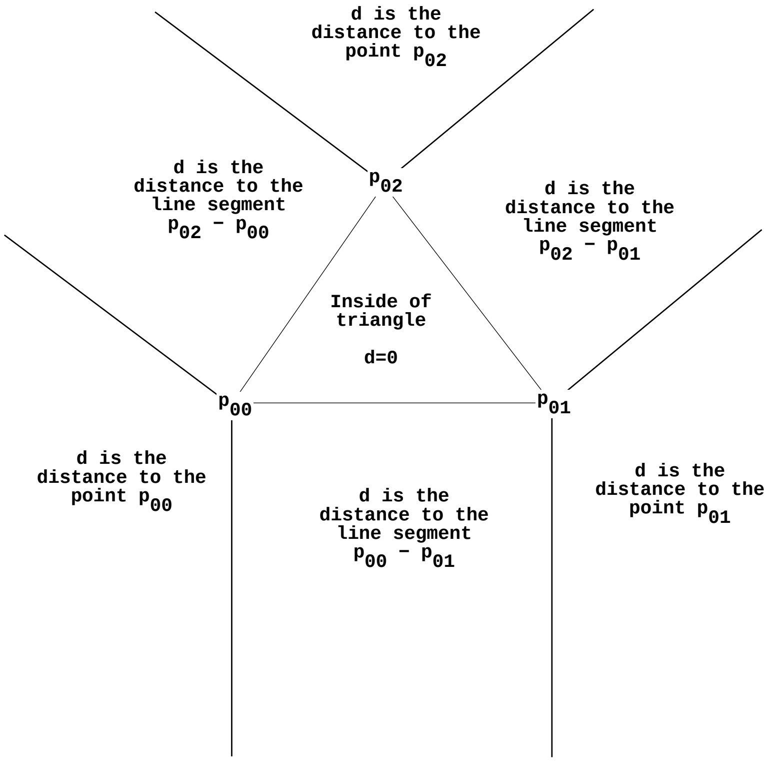

Figure 3: Computation of the distance to a triangle.

5 Future Work

One limitation of the local coordinate based approaches such as this technique or FBIM is that the coordinate transformation can not express a rotation of more than 360 degrees. If we define a coordinate mapping transformation in terms of scale rotation and translation we can preserve and express additional information about the rotation. While we lose some properties, we are able to express rotations of greater than 360 degrees.

By using C++'s object oriented inheritance we can easily define new deformers that can be plugged into our application and will blend seemlessly with existing deformers. We intend to extend our current work to higher dimensions and then we intend to define distance functions for more complex simplicial complexes. Simplicial complexes and operations on simplicial complexes are well defined in higher dimensions. By building our algorithms' framework on these structures our algorithm should easily extend from 2D to 3D and potentially to even higher dimensions.

6 Conclusions

We developed a new interactive deformation technique. This technique improve existing techniques and provide a powerful framework for future work. The simplicial complex based deformation algorithms that we are working with should work well across n dimensions and have a plethora of applications. Applications for this work include 2D morphing and warping, as well as 3D warping, modeling, and animation.

References

[1] E. Akleman. Making Caricatures with Morphing. *Visual Proceedings of ACM SIGGRAPH'97*, 134, Aug. 1997.

[2] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics SIGGRAPH '84 Proceedings*, 18(3):21-30, 1984.

[3] Beier and Neely. Feature-Based Image Metamorphosis. *Computer Graphics*, 26(2):35-42, 1992.

[4] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics*, 13(2):137-155, 1994.

[5] E. Brisson. Representing geometric structures in d dimensions: topology and order *ACM Symposium on Computational Geometry*,218-227, 1989.

[6] S. Coquillart. Extended Free-Form Deformation : A Sculpturing Tool for 3D Geometric Modeling. *Computer Graphics SIGGRAPH '90 Proceedings*, 24(4):187-196, 1990.

[7] B. Crespin. Implicit Free-Form Deformations. *Implicit Surfaces*, 1999.

[8] F. Lazarus, S. Coquillart, and P. Jancene. Axial deformations: an intuitive deformation technique. *Computer Aided Design*, 26(8):607-613, 1994.

[9] A. Paoluzzi, F. Bernardini, C. Cattani and V. Ferrucci. Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics*,12(1):56-102, 1993.

[10] T. W. Sederberg, S. R. Parry. Free-form Deformations of solid geometric models. *Computer Graphics SIGGRAPH '86 Proceedings*, 20(4):151-160, 1986.

[11] Wyvill, B. and Wyvill, G. Field Functions for Implicit Surfaces. *Visual Computer*, 5:75-82, 1989.

[12] Benson, P. , R. Campbell, T. Harris, M. Frank and M. Tovee. "Enhancing Images of Facial Expressions" *Perception and Psychophysics* Feb. 1999: 259-274.

[13] Brennan, Susan E. "Caricature generator: the Dynamic Exaggeration of Faces by Computer. " *Leonardo* 18. 3 170-178.

[14] Calder, A. , A. Young, P. Benson and D. Perett. "Self Priming From Distinctive and Caricatured Faces" *British Journal of Psychology* Feb. 1996: 141-162.

[15] Calder, A. , A. Young, D. Rowland and D. Perrett. "Computer-Enhanced Emotion in Facial Expressions" *Proceedings of the Royal Society of London Series B-Biological Sciences* Jun. 22 1997: 919-925.

[16] Campbell, R. , O. Pascalis, M. Coleman, S. Wallace and P. Benson. "Are Faces of Different Species Perceived Categorically by Human Observers?" *Proceedings of The Royal Society of London Series B-Biological Sciences* Oct. 22 1997: 1429-1434.

[17] Mauro, R. and M. Kubovy. "Caricature and Face Recognition" *Memory and Cognition* Jul. 1992: 433-440.

[18] Rhodes, G. , G. Byatt, T. Tremewan and A. Kennedy. "Facial Distinctiveness and The Power of Caricatures" *Perception* (1997): 207-223.

[19] Valentine, T. and V. Bruce. "The Effects of Distinctiveness in Recognizing and Classifying Faces" *Perception* (1986): 525-535.

[20] Burton, A. , V. Bruce and P. Hancock. "From Pixels to People: A Model of Familiar Face Recognition" *Cognitive Science* Jan. 1999: 1-31.

[21] Byatt, G. and G. Rhodes. "Recognition of Own-race and Other-race Caricatures: Implications for models of Face Recognition" *Vision Research* Aug. 1998: 2455-2468.

[22] Cox, D. "Caricature, Readymades and Metamorphosis, Visual Mathematics in the Context of Art" Leonardo 25. 3 295-301.

[23] Lewis, M. and R. Johnston. "Understanding Caricatures of Faces" *Quarterly Journal of Experimental Psychology Section A-Human Experimental Psychology* May 1998: 321-346.

[24] Moskovitch, M. , G. Winocur and M. Behrmann. "What is Special about Face Recognition?" *Journal of Cognitive Neuroscience* Sep. 1997: 555-604.

[25] Otoole, A. , T. Vernon, H. Volz and E. Salter. "Three-dimensional Caricatures of Human Heads: Distinctiveness and the Perception of Facial Age" *Perception* (1997) 719-732.

[26] Rhodes, G. , S. Carey, G. Byatt and F. Proffitt. "Coding Spatial Variations in Faces and Simple Shapes: A Test of Two Models" *Vision Research* Aug. 1998: 2307-2321.

[27] Rhodes, G. , and T. Tremewan. "Averageness, Exaggeration, and Facial Attractiveness" *Psychological Science* Mar. 1996: 105-110.

[28] Valentin, D. , H. Abdi, A. Otoole and G. Cottrell. "Connectionist Models of Face Processing - *A Survey" Pattern Recognition* Sep. 1994: 1209-1230.

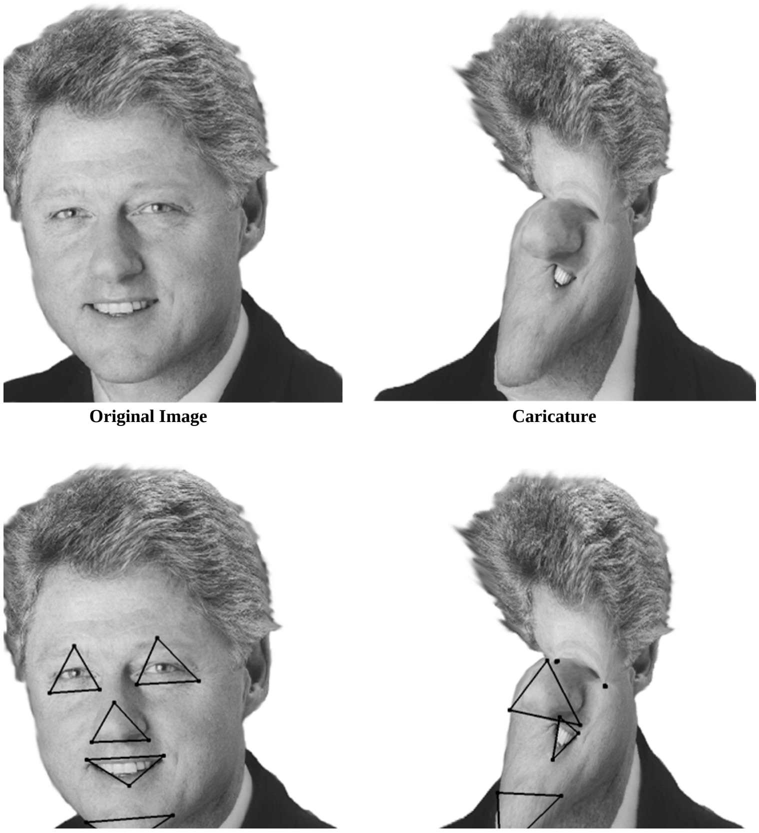

Figure 4: Caricature of President Bill Clinton with our interactive technique.

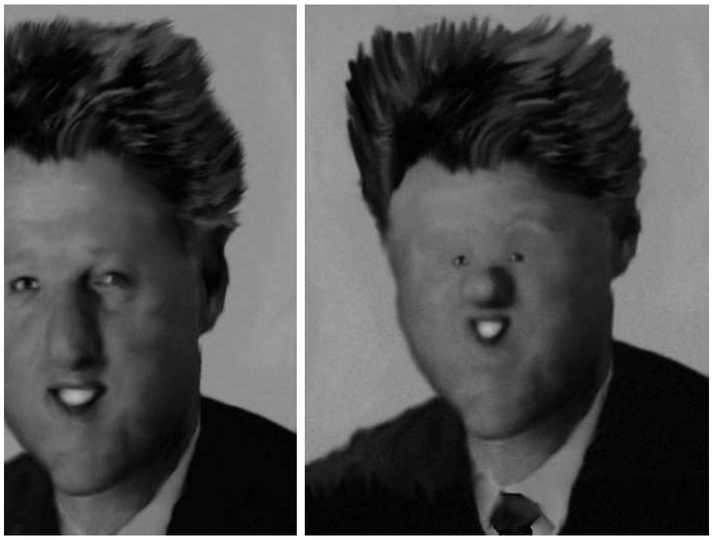

Figure 5: Earlier caricatures of President Bill Clinton by using FBIM. These images are enhanced by using photoshop.