An Implementation of Feature-Based Image Metamorphosis

Term Project

Submitted in partial fulfillment
of the requirements of
CS F401

By

ISHAN SHARMA
ID No. 2016B2A70773P

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI,
PILANI CAMPUS

March - April 2021

Original Paper

Link Reference: Princeton CS Repository Link / ACM Digital Library
Project Area: (773 % 14) + 1 = 4. Graphics Media

Computer Graphics, 26, 2, July 1992

Feature-Based Image Metamorphosis

Thaddeus Beier

Silicon Graphics Computer Systems
2011 Shoreline Blvd, Mountain View CA 94043

Shawn Neely

Pacific Data Images
1111 Karlstad Drive, Sunnyvale CA 94089

1 Abstract

A new technique is presented for the metamorphosis of one digital image into another. The approach gives the animator high-level control of the visual effect by providing natural feature-based specification and interaction. When used effectively, this technique can give the illusion that the photographed or computer generated subjects are transforming in a fluid, surrealistic, and often dramatic way. Comparisons with existing methods are drawn, and the advantages and disadvantages of each are examined. The new method is then extended to accommodate keyframed transformations between image sequences for motion image work. Several examples are illustrated with resulting images.

Keywords: Computer Animation, Interpolation, Image Processing, Shape Transformation.

2 Introduction

2.1 Conventional Metamorphosis Techniques

Metamorphosis between two or more images over time is a useful visual technique, often used for educational or entertainment purposes. Traditional filmmaking techniques for this effect include clever cuts (such as a character exhibiting changes while running through a forest and passing behind several trees) and optical cross-dissolve, in which one image is faded out while another is simultaneously faded in (with makeup change, appliances, or object substitution). Several classic horror films illustrate the process: who could forget the hair-raising transformation of the Wolfman, or the dramatic metamorphosis from Dr. Jekyll to Mr. Hyde? This paper presents a contemporary solution to the visual transformation problem.

Taking the cutting approach to the limit gives us the technique of stop-motion animation, in which the subject is progressively transformed and photographed one frame at a time. This process can give the powerful illusion of continuous metamorphosis, but it requires much skill and is very tedious work. Moreover, stop-motion usually suffers from the problem of visual strobing by not providing the motion blur normally associated with moving film subjects. A motion-controlled variant called go-motion (in which the frame-by-frame subjects are photographed while moving) can provide the proper motion blur to create a more natural effect, but the complexity of the models, motion hardware, and required skills becomes even greater.

2.2 3D Computer Graphics Techniques

We can use technology in other ways to help build a metamorphosis tool. For example, we can use computer graphics to model and render images which transform over time.

One approach involves the representation of a pair of three-dimensional objects as a collection of polygons. The vertices of the first object are then displaced over time to coincide in position with corresponding vertices of the second object, with color and other attributes similarly interpolated. The chief problem with this technique is the difficulty in establishing a desirable vertex correspondence; this often imposes inconvenient constraints on the geometric representation of the objects, such as requiring the same number of polygons in each model. Even if these conditions are met, problems still arise when the topologies of the two objects differ (such as when one object has a hole through it), or when the features must move in a complex way (such as sliding along the object surface from back to front). This direct point-interpolation technique can be effective, however, for transformations in which the data correspondence and interpolation paths are simple. For example, the technique was successfully used for the interpolation of a regular grid of 3D scanned data in "Star Trek IV: The Voyage Home" [13]. Methods for automatically generating corresponding vertices or polygons for interpolation have been developed. [5][6]

Other computer graphics techniques which can be used for object metamorphosis include solid deformations [1] [12] and particle systems [10]. In each case the 3D model of the first object is transformed to have the shape and surface properties of the second model, and the resulting animation is rendered and recorded.

2.3 2D Computer Graphics Techniques

While three-dimensional object metamorphosis is a natural solution when both objects are easily modeled for the computer, often the complexity of the subjects makes this approach impractical. For example, many applications of the effect require transformations between complex objects such as animals. In this case it is often easier to manipulate scanned photographs of the scene using two-dimensional image processing techniques than to attempt to model and render the details of the animal's appearance for the computer.

The simplest method for changing one digital image into another is simply to cross-dissolve between them. The color of each pixel is

Implementation Details

The main objective of this paper is to metamorph Image 1 into Image 2 via seamless transition.
Metamorphosis doesn't only mean to cross dissolve one image into another but also to retain the features in the image at a particular location without displacing it too much.

Conventional morphing techniques involve changing / interpolation of pixels but this paper entails on the *computer graphics aspect* of determining the *interpolation of frames using line segments* that emphasis the features in the images, hence the same concept can be extended to 3D using 3D line segments to interpolate voxels.

This paper introduces a new technique called **Field Morphing.** This morphing is influenced by the surrounding two-dimensional control primitives.

Distortion of a single image is achieved using inverse mapping in which every pixel in the destination image gets set to some appropriate source pixel.

How to determine the source pixel?

Line segments are drawn on two images to highlight the feature correspondence between two images. **[Ref Figure 4]**

The paper introduces the **Transformation With Multiple Pair Of Lines**

```
For each pixel X in the destination
    DSUM = (0,0)
    weightsum = 0
    For each line Pi Qi
        calculate u,v based on Pi Qi
        calculate X'i based on u,v and Pi'Qi'
        calculate displacement Di = Xi' - Xi for this line
        dist = shortest distance from X to Pi Qi
        weight = (length^p / (a + dist))^b
        DSUM += Di * weight
        weightsum += weight
    X' = X + DSUM / weightsum
    destinationImage(X) = sourceImage(X')
```

Figure 3. [Taken from paper] Algorithm using multiple line pairs and distance error and weighted sum.

Using the above algorithm we find the source pixel from the destination pixel.

Approaching The Solution

First I would need to design a GUI that displays the two images that we want to metamorph.
This GUI would need you to draw the corresponding line segments to highlight features.

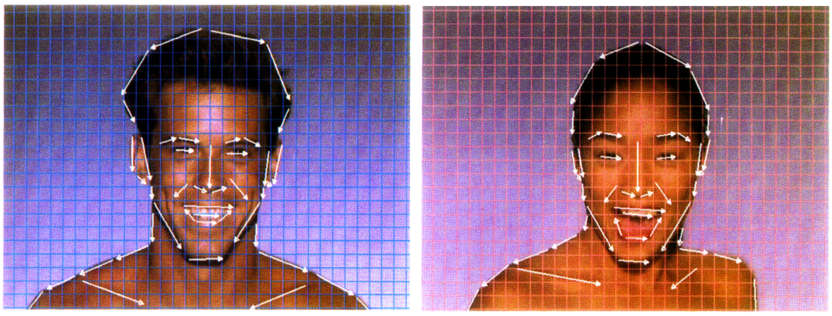

Figure 4. [Taken from paper] Shows the line segment feature highlighting correspondence to the images to be metamorphed.

These line segments would be stored in a separate text file.

Assume we are having two images **Image1** and **Image2**. And the resultant image at any given **time = t** is **ResultImage**.

The image metamorphosis algorithm could be normalised to a scale of time between time = 0 to 1.

At time = 0, **the ResultImage is exactly Image1.**

At time = t, the ResultImage consists of distorted Image1 translucent and distorted Image2 translucent.

At time = 1, **the ResultImage is exactly Image2.**

As time progresses we need to metamorph this Image1 into Image2.

This process according to the algorithm described by Beier-Neely consists of three steps -

1. Distort Image1.
2. Distort Image2.
3. Blend Distorted Image1 and Distorted Image2.

Two functions need to be designed for the above steps - DistortImage and BlendImage.

DistortImage Function: This function takes in the following parameters:

1. Image
2. TimeStamp (Value between 0 to 1)
3. Three Parameters [a, b, p] specified in the algorithm for weight calculation.
4. Line Segments in Image1 and Image2.

This function distorts an image according to the algorithm described in the paper using linear interpolations of the line segments.

BlendImage Function: This function takes in the following parameters:

1. Image1
2. Image2
3. TimeStamp (Value between 0 to 1)

This function linearly blends (cross fades) the two images based on the timestamp.

The resultant pixel (x,y) at time = t would be the weighted average (parameterized with t (time)) of the two image pixels at the location (x,y).

The main driver program shall take the two images Image1 and Image2 and progressively generate the interpolated frames of two images morphed together at different time intervals (time-step of 0.25 units).

All of the resultant images generated can then be combined and converted into a GIF hence achieving the metamorphosis task.

References

[1] Neely S & Beier T, *“Feature-based image metamorphosis”.* SIGGRAPH '92, July 1992, Pages 35-42

[2] https://web.cs.wpi.edu/~matt/courses/cs563/talks/morph.html

[3] https://en.wikipedia.org/wiki/Morphing