Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Implementation of Image Morphing Through Mesh Morphing Algorithm

Urvashi Bhushan*
M.Tech ISM
DIT, Dehradun
India

Dr. G. P. Saroha
Prof&Head IT Dept.
DIT,Dehradun
India

Disha Tiwari
M.Tech ISM
DIT,Dehradun
India

***Abstract*-The term "morphing" is used to describe the combination of generalized image warping with a cross-dissolve between image elements. The term is derived from "image metamorphosis". Morphing is an image processing technique typically used as an animation tool for the metamorphosis from one image to another. The idea is to specify a warp that distorts the first image into the second. Its inverse will distort the second image into the first. As the metamorphosis proceeds, the first image is gradually distorted and is faded out, while the second image starts out totally distorted toward the first and is faded in. Thus, the early images in the sequence are much like the first source image. The middle image of the sequence is the average of the first source image distorted halfway toward the second one and the second source image distorted halfway back toward the first one. The last images in the sequence are similar to the second source image. The middle image is key if it looks good then probably the entire animated sequence will look good. For morph between faces, the middle image often looks strikingly life-like, like a real person, but clearly it is neither the person in the first nor second source images.The morph process consists of warping two images so that they have the same "shape", and then cross dissolving the resulting images. Cross-dissolving is the major problem as how to warp an image. In this paper the image morphing is implemented using mesh warping algorithm and the software is implemented in java using netbeans and images are used from the internet and from digital camera. The program will output N images that perform the morphing. These images can then be edited together into a short animated sequence.**

***Keywords*— Morphing, Distorted, Faded, Cross-Dissolve, Warping.**

I. INTRODUCTION

Image morphing is referred as the animated transformation of one digital image to the other. It is a powerful tool and has widespread use for achieving special visual effect in the entertainment industry. Morphing is achieved by coupling image warping with color interpolation. As the morphing proceeds, the first image (source) is gradually distorted and is faded out, while the second image (target) starts out and is faded in. Thus, the early images in the sequence are much like the first image. The middle image of the sequence is the average of the first image distorted halfway towards the second one and the second image distorted halfway back towards the first one. The last images in the sequence are similar to the second one. Then, the whole process consists of warping two images so that they have the same "shape" and then cross dissolving the resulting images. Before the development of morphing, image transitions were generally achieved through the use of cross-dissolves, e.g., linear interpolation to fade from one image to another. The result is poor because without real warping, the double-exposure effect will be apparent in misaligned regions. Warping must be used in order to achieve a fluid transformation and since the cross-dissolving is very simple, warping becomes the major problem of morphing techniques.

II. PROBLEM DEFINITION

In this paper we are developing an Image Morphing software application in Java for morphing two images.Morphing is the process by which one picture smoothly transmutes into another. The intermediate images that bridge the transition are calculated from the source and destination images using a mathematical formula. The software would be implemented using the Mesh Morphing algorithm.

III. Mesh Warping

The mesh-warping algorithm relates features with non-uniform mesh in the source and destination images, i.e., the images are broken up into small regions that are mapped onto each other for the morph.

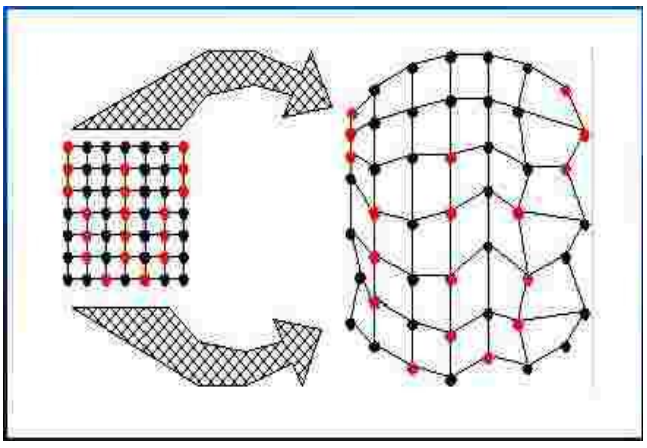


Figure 1 Mesh Warping

We refer to the source and the target images as IS and IT, respectively. The source image is associated with mesh MS. It specifies the coordinates of control points, or landmarks. A second mesh MT specifies their corresponding positions in the target image.

Meshes MS and MT are respectively shown overlaid on IS and IT in the upper left and lower right images of the figure. Notice that

landmarks such as the eyes, nose, and lips lie below the corresponding grid lines in both meshes. Together, MS and MT are used to define the spatial transformation that maps all points in IS onto IT. The meshes are constrained to be topologically equivalent, i.e., no folding or discontinuities are permitted. Furthermore, for simplicity, the meshes are constrained to have frozen borders.

All intermediate frames in the morph sequence are the product of a four-step process:
For each frame f do
Linearly interpolate mesh M, between MS and MT
Warp IS to I1, using meshes MS and M
Warp IT to I2, using meshes MT and M
Linearly interpolate image If, between I1 and I2
End

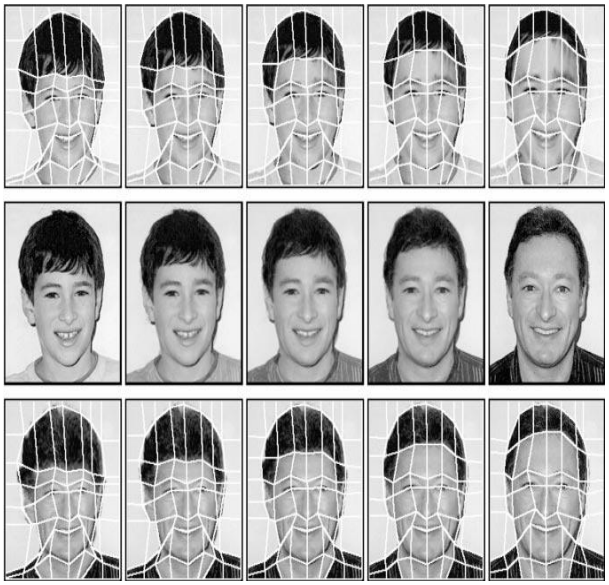


Figure1.1 Mesh Warping

Figure 1.1 depicts the process of mesh warping. In the top row of the figure, mesh MS is shown deforming to mesh MT, producing an intermediate mesh M for each frame f. These meshes are
Used to warp IS into increasingly deformed images, thereby deforming IS from its original state to those defined by the intermediate meshes. The identical process is shown in reverse order in The bottom row of the figure, where IT is shown deforming from its original state. The purpose of this procedure is to maintain the alignment of landmarks between IS and IT as they both deform to some intermediate state, producing the pairs of I1 and I2 images shown in the top and bottom rows, respectively. Only after this alignment is maintained does a cross-dissolve between successive pairs of I1 and I2 become meaningful, as shown in the morph sequence in the middle row. This process demonstrates that morphing is simply a cross-dissolve applied to warped imagery.

IV .Transformation between one pair of lines

A pair of lines (one defined relative to the source image, the other defined relative to the destination image) defines a mapping from one image to the other.

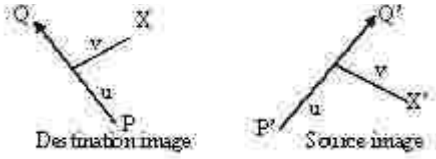


Figure1.3

Where u is the position along the line, and v is the distance from the line.

$$u = \frac{(X-P)\bullet(Q-P)}{\|Q-P\|^2}$$

$$v = \frac{(X-P)\bullet Perpendicular(Q-P)}{\|Q-P\|}$$

$$x' = P' + u\bullet(Q'-P') + \frac{v\bullet Perpendicular(Q'-P')}{\|Q'-P'\|}$$

The algorithm transforms each pixel coordinate by a rotation, translation, and/or a scale, thereby transforming the whole image.
Transformation between multiple pairs of lines.Normally there are many features in images where transformation between multiple pairs of lines are applied. It specifies more complex transformations. A weighting of the coordinate transformations for each line is performed. The weight is determined by the distance from X to the line.

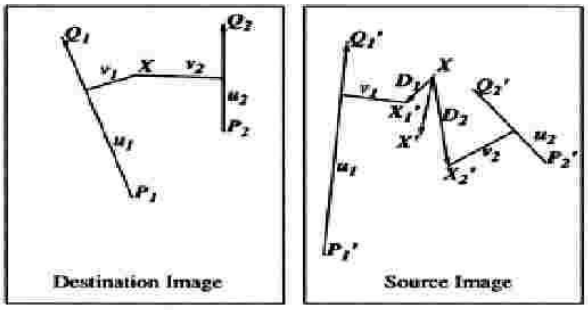


Figure 1.4

$$weight = \left(\frac{length^p}{(a+dist)}\right)^b$$

Where length is the length of a line, dist is the distance from the pixel to the line, and a, b, and p are constants that can be used to change the relative effect of the lines
The multiple line algorithm is as follows:
For each pixel X in the destination
DSUM=0
weightsum = 0
For each line piqi
calculate u, v based on piqi
calculate X'i based on u, v and p'iq'i
calculate displacement Di=x'i-Xi for this line
dist= shortest distance from X to piqi
weight = (lengthp / (a + dist))b
DSUM += Di * weight
weightsum += weight
X' = X + DSUM / weightsum
destinationimage(X) = sourceimage(X')

V. Implementation and Results

The Mesh Morphing algorithm has been implemented as our paper. We have divided our project into following steps explained below.

Step 1- Loading Of Two Images

1. **getImage(),**
2. **createImage(),**
3. **PixelGrabber class including functions getPixel() and grabPixel(),**
4. **drawImage().**

Step 2- Getting Corresponding Points

ControlPoint c=new ControlPoint(X1,Y1,X2,Y2)

Step 3- Creating The Mesh
Triangles are formed by the **morph()** function as a intermediate step. The Triangle is formed by the instance of **Triangle** class which takes a parameterized constructor as the values of the instances of **ControlPoint** class. The particular triangles are then split into 22 **SimpleTriangle** i.e the Traingle which has the linear base(two of the coordinate of the three point Triangle have the same 'y' coordinate).

Step 4- Getting The Number Of Frames
This step will be implemented by clicking the particular Menu Item. More the number of frames, clearer will be the sequence of the morphed images when we see the video(as the sequences of the morphed image).The number of frames will be entered by the user with the help of a dialog box opened asking for the same.

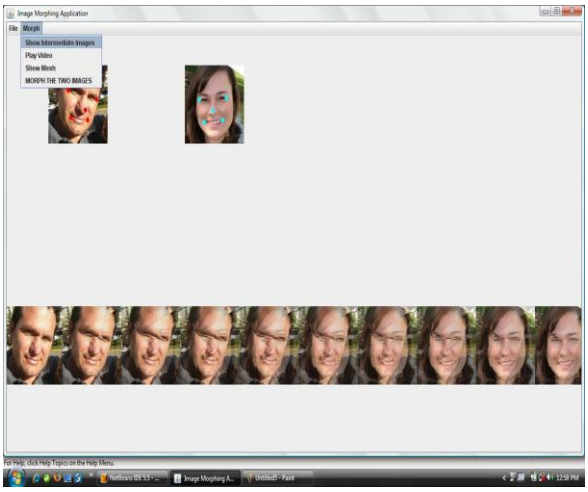
Step 5- Intermediate Points
Intermediate points are the points which are formed by mapping the corresponding points of Imagesrc and Imagedest. They are generated by the following formula:

```
ratio = (double)f / (double)(framecnt-1)
x0 = (int)Math.round((double)cp[0].x1 + ((double)(cp[0].x2 – cp[0].x1) * ratio))
y0 = (int)Math.round((double)cp[0].y1 + ((double)(cp[0].y2 - cp[0].y1) * ratio))
 t = new Triangle(x0, y0, x1, y1, x2,y2);
```

Step 6- Generation Of The Morphed Images
This function is implemented by the **morphtriangle1()** and **morphtriangle2()** function, which basically interpolate the pixel values at both the images and assign the value to the intermediate image.

Step 7- Displaying The Morphed Image This images generated in the previous step would be displayed in the sequence with the **drawImage()** function

VI. RESULT



VII. CONCLUSION

Our paper has successfully implemented Image Morphing using Mesh Warping algorithm combined with the cross dissolving technique. The algorithms used were fast and intuitive, which efficiently computed the mapping of each pixel from the source image to the destination image.
The mesh was formed of triangles obtained from the manually specified control points. It was noticed that more the number of frames better were the morphed results.

REFERENCES:

[1]. Herbert Schildt 2002 The Complete Reference, 5th Edition
[2]. George Wolberg, Image morphing: a survey, Department of Computer Science, City College of New York, New York, USA
[3]. Stephen Karungaru, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, Image Morphing and Warping: Application to Speech Simulation Using a Single Image, Department of Information Science and Intelligent Systems, University of Tokushima, Japan.
[4]. wikipedia.org/wiki/Morphing

[5.] Daniel J. Palma, IECEP-UE Introduction to Image Morphing

[6]. pixel manipulation - Java - Forums at ProgrammersHeaven.com.html
[7]. Internet.com.Processing-Image-Pixels-using-Java-Getting-Started.htm

