CS 631 Assignment #1

Due Date : **Feb 22th 1999**

Introduction

In this assignment, you will implement a morphing technique that was used in creating the face morphing sequences in Michael Jackson's Black or White music video. The technique is first described in a paper titled *Feature-Based Image Metamorphosis* by Beier & Neely. You will read the paper and implement a program that produces image morphing sequences based on methods mentioned in the paper.

The paper can be found on-line at http://www.hammerhead.com/thad/morph.html.

Feature-Based Image Metamorphosis in a Nutshell

Morphing involves two main operations : warping and cross-dissolving. To morph from image I1 to I2, one would try to warp I1 to the shape of I2 and I2 to the shape of I1, and perform cross-dissolve between the sequence of intermediate warping images. When the dissolved images are put together and played back quickly, they will create the illusion of I1 gradually changing to I2.

The essence of this technique lies in how the warping is done, and the paper calls its approach *field morphing*. Field morphing involves using pairs of corresponding line segments to determine the answer to the question : "Which pixel coordinate in the source image do we sample for each pixel in the destination image?". How this question is answered is well detailed in section 3 of the paper. You should read and understand this section before you start to code your assignment.

Section 3.4 describes how morphing between two images can be done. It tells you how you can interpolate the line segments for intermediate frames, but it does not say how the cross dissolving should be done. However, it is not hard to come up with something sensible. One thing that you can do is to determine the color based on the frame number of the output image. For eg., if there are 10 frames total, and you're doing output frame 3, the pixels will be 3/10th of frame 3 in warped I1 and 7/10th of frame 7 in warped I2.

Implementation Details

Your program again will have a simple command line interface that looks like the following :

```
> morph <start-image> <end-image> <line-seq> <out file name> <nii>
```

where <start-image> is the image that will morph into <end-image>, <line-seq> is the name of an ASCII text file will contain pairs of corresponding line segments defined as pairs of coordinate points, and <out file name> is the base name for the output frames.
Your program will output the number of intermediate frames as specified by <nii> and their frame numbers will be appended to the base <out file name>. For eg., if my <nii> is 10, then 10 intermediate frames will be produced, and let’s say <out file name> is “face”, then my output frames will be called face1.pgm, face2.pgm, face3.pgm… face10.pgm. Note that the start and end images do not count as intermediate images.
Both the start and end image will be in the binary PGM format and they will have similar dimensions. You don’t have to worry about ASCII PGM images.

The <line-seq> file should look like the following :

<number of corresponding line pairs>
$x_{L1,src,A}$ $y_{L1,src,A}$ $x_{L1,src,B}$ $y_{L1,src,B}$ $x_{L1,dest,A}$ $y_{L1,dest,A}$ $x_{L1,dest,B}$ $y_{L1,dest,B}$
$x_{L2,src,A}$ $y_{L2,src,A}$ $x_{L2,src,B}$ $y_{L2,src,B}$ $x_{L2,dest,A}$ $y_{L2,dest,A}$ $x_{L2,dest,B}$ $y_{L2,dest,B}$
………………
………………

Each row will represent a pair of corresponding line segments. For the above file, ($x_{L1,src,A}$ $y_{L1,src,A}$ $x_{L1,src,B}$ $y_{L1,src,B}$) will represent a line segment AB in the source image and it will correspond to a line segment AB in the destination image represented by the coordinates ($x_{L1,dest,A}$ $y_{L1,dest,A}$ $x_{L1,dest,B}$ $y_{L1,dest,B}$).

For example, the following is a line segment text file with 3 line segments :

3
10 10 60 60 18 20 75 74
55 20 120 140 60 32 130 145
23 40 23 100 44 40 44 100

For this file, the line (10,10,60,60) on the start image will correspond to line (18,20,75,74) on the end image.

Instructions:

- This assignment must be done in groups of 2 students.

- You should use C or C++ and please make sure that your code compiles on the Windows platform. You are free to use whatever libraries that are available to manipulate the images (short of actually doing the assignment), however these libraries must be statically linked into your executable.

- We will provide you with a set of source and destination images on the web. The minimum requirement is for your morphing implementation to work on these images. However, you are free to choose up to 2 additional set of source/destination images to submit your coolest morphs.

- You are required to send "hw1.zip" file as an attachment containing the following:
  1. Source code
  2. Readme.txt containing instructions on how to compile and run your program. Please include information on *parameters* such as the line segment file.
  3. Makefile (that can be run using nmake)
  4. Extra source/destination images that you used. Please do not send the images that are already provided to you by us instead, you only need to submit the line segment file for these images.
  5. Line segment file(s): one per morph.

- How to submit?
  - Send *hw1.zip* as an attachment to warkhedi@cs.cornell.edu
  - The subject header of your e-mail must contain the string "CS631: HW1" (ignoring the whitespaces & case).
  - The assignment should be submitted on or before 11:59pm Feb 22th 1999.

- Grading
  - This assignment will be graded on a scale of 10.
  - The grade will primarily be based on correctness of your morphing implementation.
  - For each extra morph (i.e. different src/dest images), you can earn 1 extra credit point. If your morph is truly "dazzling" [to us] you will be awarded 2pts. Remember you can only submit up to 2 extra morphs. Excess will be ignored.

- Enjoy!