Introduction

The goal of this project is to provide a user interface and backend driver that supports the specification and rendering of transitions between two images (the start image and the end image). These transitions will be rendered as a sequence of images that can be wrapped as a video (AVI, for example) and used in standard video editing software.

Each student is expected to develop a program in Java that provides the following functions:

- **Pixel Processing**: including YIQ adjustment, median filtering, low-pass filtering, and edge detection. These operations can be applied to either the start image or the end image as a pre-processing step.
- **Image Pan**: An image pan is a rectangular, axis-aligned region that “moves” from a starting location in the start image to an ending location in the end image. The two images are blended linearly during the transition.
- **Piecewise Image Warp**: In this case, you only need to work on a single image. A piecewise image warp, defined by a triangulated mesh, specifies an initial position and an ending position. The initial position is always on a rectilinear grid. The resolution of the grid can be hard-coded, but must be greater than 5 by 5. The user can interactively change the position of the grid points. Gradually warp the image according to the grid points in the triangle mesh.

*Details about these functions are discussed in class. If you need further classification, contact the instructor during his office hour.*

Extra Credit

Two additional functions (also discussed in class) can be implemented for extra credit:

- **Speed control**: For both image pan and image warp, the speed of transition is controlled by a Bezier curve.
- **Advanced Image Warp**: This allows the initial grid position to be changed as well, in addition the start and the end image can be different.

Final Demonstration

You should schedule a 10 minute slot with the instructor to demonstrate the functions of your program. The date for the final presentation is April 26, 2006. This date is firm. Sign-up sheet will be available in class.

What to Turn In

You must turn in three things on or before April 30th, 2006:

- Complete java source code;
- External documentation that includes
  - A simple user manual;
  - Software architecture: class hierarchies and their implementation, lessons learned etc.
  - Roles each group member had in designing and implementing the project.
• An example set of transitions created with your tool, which can be the same as the ones demonstrated in class. The example transitions should be an AVI file. MS Moviemaker (or any other similar tool) can be used to convert images to a video file.

**Grading Guideline**

Total: 100 + 30 (extra credit) points

• 15pts: documentation (HTML page and final video clips)
• 20pts: image filters (there are four of them, 5pts each)
• 30pts: image pan
• 35pts: mage warp
• 15pts: speed control
• 15pts: advance image warp

**Hints:**

- A piece of sample code to load, modify, and write an image is available on the course web page.
- A JAVA class to compute the homography (3x3 perspective warp matrix) will be also available on the course web page by April 9, 2006.
- For the advanced image warp part, one needs to determine which pixel is in which quad patch. One simple solution is to draw each quad with different colors on the image. This color-coded image can serve as a look-up table to find out which perspective transform to apply.