CS6640 — Project 3
Assigned October 06, 2009
Due October 27 (Midnight)

A PDF version of this assignment is here.

Goals

The purpose of this assignment is to learn about image transformations (warping) and image mosaicing. Please read following instructions carefully.

1 Assignment

You are to build a function in matlab executed as follows:
`mosaic(parameter-file-name)`
where `parameter-file-name` is of type string, and it specifies a file of parameters that will be read. An example of the particular type of files your program should read is given here. The parameter file specifies image files and control points. **Please do not hard code your examples in a Matlab or C++ file — you should read specific mosaicing examples from a parameter file.** It also specifies the name of an output file, which will contain a mosaic image that includes all of the input images represented in the domain of the target image, as specified in the parameter file. Your program should properly read a file like this and properly use the parameter. Your code can read more parameters, but it cannot require them. The parameter files should allow for C++ comments, so that you can keep track of your correspondences and experiments. In Matlab, you can read easily text from a file ignoring C++ comments by setting flag (search the help). If you are using Vispack/C++, there is a set of code in place to help you deal with reading parameters and setting up transformations, you can find that you need here, and you can start by looking at the README in that directory.

2 Grading and Report

To get a good grade, your report should show that you have experimented with the mosaicing algorithm, analyzed its behavior, etc. Your report should explicitly answer all indicated with underlines. You should also build your algorithm to handle some tricky details that are not specified.

2.1 Experiments

- Test and experiment with your application on the example given in mosaic example.
- Try to build mosaics with your own data sets, use both planar and panoramic data sets.
- Experiment with different numbers of control points etc., and report your results, with pictures.

2.2 Questions

- How many control points does it take to get a "good" transformation between images?
- How does the algorithm behave at the theoretical minimum of the number of control points?
- From your experiments how does the accuracy of the control points affect the results?

2.3 Details

To the extent that your algorithm handles these details in a good way, it will improve your grade.

Contrast: A good algorithm should automatically adjust for major intensity differences.

Feathering: A good algorithm should alleviate sharp transitions between images.

Image size: A good algorithm should should produce an output image that is a good size and shape to display all of the images.

Cascading: A typical set of images and control points might not include connections directly back to the target output image. Hopefully you can always get from any given image to the target by a series of projective transformations. This is a graph traversal problem (not a hard one). A good algorithm would automatically find a transformation back to the target (if one exists).

Mosaic Example: An example input data set is given in mosaic example.

3 Hints

Inverse Transforms: Remember that the projective transformation need for image mosaicing is described by a 3x3 matrix A. The inverse of this matrix describes the inverse of the associated projective transformation.

Cascading: The net result of cascading two consecutive projective transformations is described by multiplying their associated matrices (in homogeneous coordinates), i.e. AB.

Sizing Outputs: In order to figure out the size and extent needed for the output, you can look at where the corners of an image map. The image maps to a quadrilateral defined by these four points.