CPSC 4040/6040
Computer Graphics Images

Assigned: Nov. 12, 2015
Due: 11:59pm, Tues., Nov. 24, 2015

Programming Assignment 7 - Removing Artifacts in Warps *(Grading: 0–10 points)*

Project Description

This week you will test out different strategies for artifact removal in the presence of nonlinear warps. This will give you experience with the tools you need to clean up your warped images. You are to use one or more of the techniques covered in class to overcome the aliasing and/or the reconstruction artifacts that are introduced by the warp.

Basic Requirements

You can start this assignment by modifying your pa06 code. You will implement a new key command:

`p` *px py* perspective warp

That allows you to build a full perspective matrix. In doing so, you will have to adjust your pipeline to support producing w components which are not equal to 1, and successfully normalizing the x and y components. The remainder of your pipeline should stay identical.

Using this new warping matrix, you can now produce nonlinear warps. This means that you can create both magnification and minification artifacts. Learning how to remove those artifacts is the purpose of this project.

After displaying the warped image, your task is to write code that repairs magnification artifacts using a scheme of your choice. This repaired imaged should be displayed (and toggled to compare with the unrepaired image) by pressing the `r` key. Upon pressing this key, you should rewarp the image using a repair strategy of your choice. Fixing magnficiation artifacts requires better reconstruction schemes, and you are encouraged to implement a scheme of your choice (e.g. bilinear or bicubic interpolation). Any of the ideas discussed in class, as well as ideas of your own, are allowed.

For this assignment I would like you to perform a very specific test case. In particular, you must repair the artifacts on the included input image, `construction.tif`, given a perspective warp where $px = 0.002$ and $py = -0.001$. This warp should both magnify and minify the image, repairing the warp should improve the quality of the image in the magnified region.

Be sure to include the output, repaired image in your submission, and describe what scheme you used to repair it.

Code Documentation Please see Programming Assignment 01 for the expected requirements. Your code will be evaluated based on code structure, commenting, and the inclusion of a README and build script.

Advanced Extension (worth 2 points)

Specifically, you must implement a fix for the minification/antialiasing problem using supersampling. You are welcome to use whatever strategy you choose for adapting the sample, whether it is procedural, based on the evaluating the gradients, or using jittering. Your code MUST compute multiple samples per pixel, where appropriate, and it should lead to removing the aliasing artifacts in the minified region.

When pressing the `r` you should apply both supersampling and whatever other strategy you employed for the basic requirements. This is straightforward – for each sample in your supersample, you will draw it from the reconstructed sample.

In addition to testing supersampling with the perspective warp from the basic requirements, you should also test your repair procedure using the twil warp from pa06, if you implemented it.

Submission

(Please read all of these instructions carefully.)

Please write the program in C or C++, and I recommend using OpenGL and GLUT graphics routines for the display. Use OIIO for image input and output. Please take extra care to make sure that your homework compiles on the School of Computing's Ubuntu Linux cluster—testing on your own home machine, even if it runs Ubuntu, may not be sufficient. Remember:

> both a working build script and README must be provided

Consequently, to receive *any credit whatsoever* this code must compile on the cluster, even if it does not accomplish all of the tasks of the assignment. The grader will give a zero to any code that does not compile.

Submit using the handin procedure outline at `https://handin.cs.clemson.edu/`. You are welcome to use the commandline interface, but the web interface is sufficient. The assignment number is `pa07`.

Finally, since we are using multiple files, please only submit a single file which has aggregated everything. This file should be named `[username]_pa07.tgz` where `[username]` is your Clemson id (please remove the brackets []). For example, mine would be `levinej_pa07.tgz`. Please make sure your build script is at the top level directory.

Rubric for Grading

4040 students will be graded for the following requirements:

I. +3 Reading, Writing, and Display

Code correctly reads the image, displays the warped image, and writes out the repaired warped image if a second filename is specified.

II. +2 Implementing the perspective warp

Code correctly inputs a perspective matrix and applies the warp appropriately. This includes allocating the correct image size and handling perspective warps in combination with other affine warps. Code also correctly scales by the w component when projecting back to the $w = 1$ plane.

III. +3 Repaired Warp

Code responds to the `r` or `R` key and warps the image while repairing magnification artifacts in the warp. Code correctly implements the chosen strategy for repair. Subjective quality of the chosen repair strategy, based on a submission of a "repaired" version of the input image.

IV. +2 Clarity

Does the code have good structure? Is the code commented and is a README provided? Is a working build script provided?

6040 students will receive 8 points for completing the above requirements, scaling the above by 80%. To achieve 10 points they must also complete:

I. Advanced Extension

Code correctly implements some form of super sampling to repair aliasing, which is run with the other repairing strategy used by the basic requirements. when the `r` or `R` key is pressed. Subjective quality of the repaired image using this technique.

Going above and beyond these requirements may result in extra credit at the discretion of the instructor and grader. Please note any extra features you implement in the README.