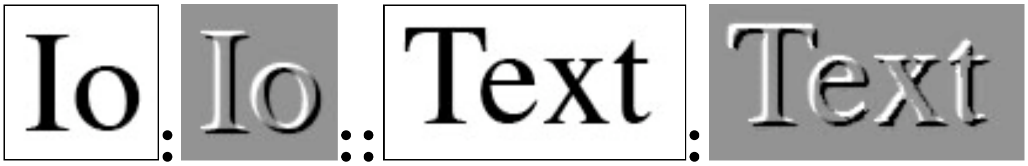
CS371 Spring 2007
Prof. Morgan McGuire

Assigned: Fri. 3/16/07
Due: Wed. 4/4/07 8:00 pm

Project 0x05: NPR



For this project you will experiment with the Image Analogies and Paint by Numbers algorithms and use them to simulate varying artistic styles. The programming for this assignment is intended to be minimal. You may work with one other person on this assignment.

The primary goal of this assignment is to build your presentation skills. In computer graphics a technically correct solution is insufficient. You must also be able to present your solution through carefully chosen applications and data sets so as to visually impress the viewer and effectively demonstrate the algorithm.

You will be graded on the quality of your written submission **document** and of your ability to present a small piece of your code at a brief one-on-one **code review** during the Thursday lab period. You will not know which part of your code you will be presenting ahead of time—I will randomly select one method and ask you to explain and defend the implementation. Your partner will not be available during the code review and you are expected to be prepared to present any piece of code, regardless of who wrote it.

NPR Specification

A. Implement a simple paint-by-numbers system in the style of Haeberli 1990 with the following features:

1. Accept an input image as a filename on the command line.

2. Allow the user to draw strokes on a blank canvas using the mouse.
    a. A single stroke has a single color based on the average color of all pixels from the input image under the stroke.
    b. The stroke has thickness given by a variable that is increased by pressing the '+' (i.e, '=') key and decreased by pressing the `-' key.
3. Save the current canvas to a new file on disk when the 'S' key is pressed.
4. Do not innovate in your code beyond these instructions—spend additional time improving your results and not the algorithm.

B. Experiment with the Image Analogies algorithm, an implementation of which is provided for you in the starter code. Demonstrate results comparable to those from the paper in a variety of styles and applications. You are not required to modify the provided code.

Implementation Tips

In the onUserInput method, use the UserInput mouse methods to find the mouse location. When the mouse button (which is treated as a key by G3D) is pressed, record the position. Then when the mouse is released, draw the line between them.

You can either use Bresenham's algorithm (see RTR) or simply linearly interpolate between the pixel endpoints to draw the line. To make a line thick you need to draw parallel lines that are separated by about one pixel perpendicular to the line direction. The perpendicular to a 2D line in direction *d* is (-*d*.y, *d*.x).

When saving, use Image3::save instead of RenderDevice::screenshot.

The image analogies algorithm is quadratic in the size of the A and A' images. If you use small images you will get results much faster.

Submitting Your Solution

1. Put your name, e-mail address, and the name of the file in a doxygen comment at the top of **each file**.

2. Create a "doc-files" directory that contains a **readme.html** file with your name, e-mail address, partner's name (if you worked in a pair) and anything you'd like to point out to me when I'm reviewing your program. This is where you'll present your results.

3. Delete all generated files using "icompile --clean". Do not hand in a build this week.

4. Change to the **parent** directory of your project and run the FreeBSD command:

```
submit371 npr
```