

Mobile Image Analysis

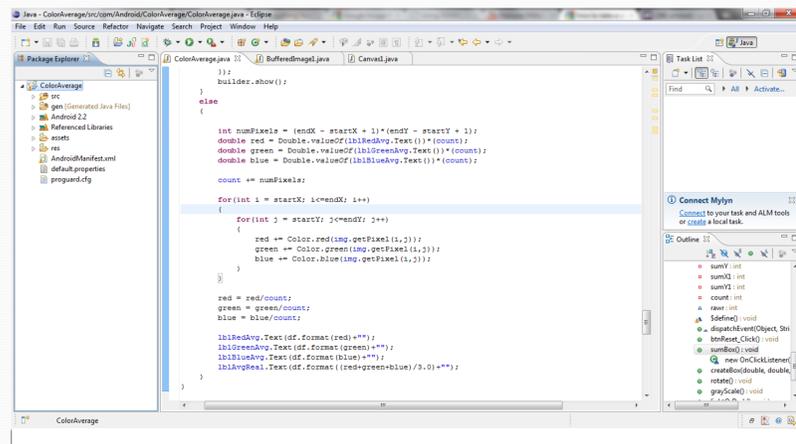
Abstract

Image analysis is a useful tool in many fields and sciences. It can be used for a variety of things such as detecting movement, counting objects in an image, and returning color information about an image. Smartphones (Android, iPhone, and Blackberrys) provide a highly mobile and convenient medium for an image analysis tool. This research project describes the need and development of an Android application that provides the user with a suite of image analysis tools.

The application's original purpose was to return the red, green, and blue values of the pixels in an image. This information is returned by processing the pixels in an image taken with the phone's camera (or loaded from the phone's SD card). This has many applications, since many experiments/tests require the comparison of an image's red, green, and blue values to another image's values (usually a standard, or control image). For example, a test measuring ground level ozone levels uses special paper strips that become darker depending on how much ozone is in the air. These differences in color can be quantified and compared to a standard scale using the average red, green, and blue values. Other applications include measuring albedo (the relative reflectivity of light from a surface) and water quality. Some other avenues of investigation for the application include useful image manipulations, object counting using edge detection, and sectioning images by color. Using smartphone applications to analyze images allows people to quickly and conveniently collect image data from tests or experiments.

App Inventor Java Bridge

The App Inventor is a browser-based Android development environment that utilizes a simple drag and drop user interface. This simple interface allows many students (or others) to create applications without having much programming experience. The Java Bridge translates the App Inventor components into actual Java code (the Eclipse IDE was used for Java programming). The App Inventor hides many of the complexities and inner workings of the Android framework (which can be good and bad); the Java Bridge provides an intermediary between the App Inventor environment and regular Android SDK development. The Java Bridge gives the programmer more flexibility and power than the App Inventor, while not being as complex as regular Android development.



Purpose

The original purpose of this project was to develop an Android application to conduct basic image analysis involving the pixels of an image. The idea came as a request during an Android programming camp. An image is composed of many pixels, and each pixel (usually) has a red, green, and blue component that determines its color. It's often useful to analyze these color component values. Scientists at the University of Massachusetts were interested in an application that could return the average red, green, and blue values of a colored test strip from an experiment; these values would then be compared to standard values. This was the main idea behind the project.

Development

The necessary programs and kits were installed on a laptop. Android applications are written in Java, so prior knowledge of Java syntax from taking an AP Computer Science course was useful. In order to use the Java Bridge, knowledge of App Inventor components and their uses was necessary. Experience with using the Java Bridge to make applications was also needed; although Android applications are written in Java, there are still many unfamiliar classes (even when using the simplified Java Bridge method). An Android Programming Camp attended during the summer greatly helped with these things and marked the formation of the idea and start of the project.

All of the coding for this project followed the same basic pattern: code, test, fix, and repeat. A basic version of the application was completed at the camp as a final project. In order to return information about the pixels, a special BufferedImage class was used, since no App Inventor components support this function. This version only had the basic idea implemented, but it was a good exercise and established the scope of the project.



First Version:

- User only able to select a single pixel at a time.
- Pre-selected Image
- Kept running average of RGB values.

Android Programming Tools

- HP Laptop – used for coding and development.
- HTC Hero Android Smartphone – typical Android smartphone with camera and touch-screen capabilities; used for development and testing.
- Eclipse IDE with Android ADT Plugin – provided efficient coding environment.
- Android SDK – software development kit necessary for developing the application.
- Java SE Development Kit 6 and JRE 6
- libSimpleRuntimeWithBufferedImage JAR File (Java Bridge) – contained numerous classes used in developing the application including the BufferedImage class from App Inventor

Work after the summer camp focused on adding more functionality to the application. The ability to take a picture or load a picture from the phone and analyze that image was added, as well as the ability to calculate the average RGB values of a region selected using the smartphone's touch-screen.

After the application's original purpose was completed and had basic functionality, other ideas were explored. The original BufferedImage class and Canvas class (the Canvas class is part of the Java Bridge JAR file) were modified to support basic image manipulations, such as lightening and rotating an image. A rudimentary edge detection manipulation was added to the application.



Current Version:

- User able to select a rectangular region of pixels.
- User able to analyze a taken photo or load a picture from phone.
- Basic image manipulations such as lightening and rotating added.
- Basic edge detection manipulation added.

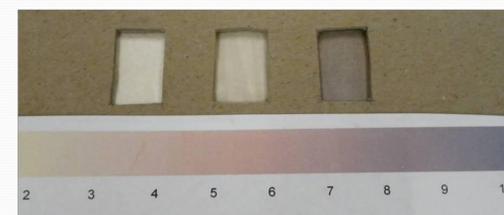
Applications

The application's RGB value function can be used in many different experiments and tests involving the comparison of color shades. When comparing two shades of color, the difference can be quantified using the differences in their RGB values. These differences can be converted to a percentage and then some concrete information regarding the experiment or test. A few possible applications are:

- Measuring Ozone Levels
- Measuring Albedo (relative reflectivity of light)
- Measuring Diffusion Rates
- Assessing Water Quality

Future Work

Future work includes, but is not limited to: looking into how the basic RGB value function can be enhanced or added upon, other uses (besides the ones already noted) of the application, image manipulations that might be useful for scientists, and basic edge detection and its applications (such as counting objects in an image). The user interface could also be improved.



An example of Schoenbein papers, which are used to measure ozone levels.