Samantha DiMaio

Professor Rivas

Semester Project Final Writeup

10 May 2017

Abstract:

The goal of this project is to develop an image processing Java code that will apply blur, sharpening, and edge detection filters to an image. The code will use a variety of Java classes and methods from the java.awt and java.io packages. Image Expression allows the user to create his or her own unique filtered image with a variety of personalized features. Some of the features include loading the image of their choice, adding a text of the user's name that will appear on the image, and choosing what color background the user would like. These options and features will be applied through the user's input in the terminal once the program runs. It will then be displayed in a graphical window which appears as the user adds input into the terminal.

Introduction:

Filtering is a technique used to modify or enhance an image. Someone can use a filter to highlight or remove certain aspects of image. Image Expression incorporates a variety of filters which include edge detection, sharpening, and blurring. By definition, blurring means to make hazy or indistinct in appearance. Blurring an image is a distortion of the pixels in the image. From a mathematical standpoint, it can be seen as an array kernel of numbers is being shifted

across the image and each is multiplied by each of the pixels of the image. Then each multiplication result is added together to create the new pixel value. This is known as a mathematical process called convolution. Thus, depending on what these original kernel of numbers are will determine what filter will be used on the image. Various standard kernels exist for each specific filter application. The values of the matrix, the width of the kernel, and the height of the kernel is then used in the Kernel() and ConvolveOp() methods to apply the data. Below are examples of kernels to be used to access the different effects:

Blur

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Edge Detection

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

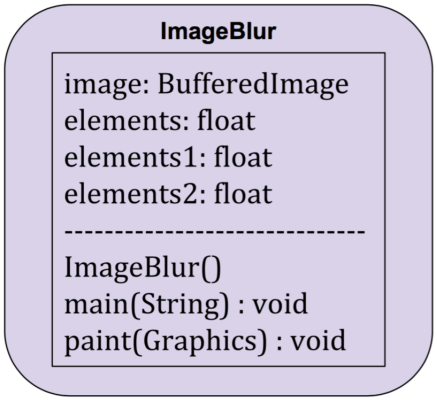
Sharpening

	-2	-1	0	
	-1	1	1	
	0	1	2	

Java also includes many packages with classes and different methods to assist in producing code that will blur an image, draw a frame, and personalized attributes such as text and background color. The Frame class, the Graphics class, and the Color class are all examples of classes that are implemented into the source code and used through different methods. The code will also ask for input to design an image that appeals to the user.

Detailed System Description:

UML
Class Diagram



Code Requirements:

The java source code will have one constructor and two methods. In the main method, a Frame class object will call different methods to set the frame of the image, such as the frame's dimensions, the window's background color, and the frame's visibility status. It will also ask the user to input the dimensions of their frame window. The constructor ImageBlur() will be used to load the original image onto the program. It will ask the user to input the file name of their image, then it will call methods to obtain the image dimensions and set that image as a new BufferedImage.

The paint(Graphics create) method will be used to load the image, and ask the user what filter they would want to apply on their image. It will also ask for user input on the name they would like to place onto the image. It will consist of a variety of methods from the Graphics class which is part of the java.awt package. These methods include ConvovleOp(), which takes in a kernel, and Kernel(), which takes in the matrix height, the matrix width, and the data float

values from the matrix. Then an instance of the Graphics class called create is used to call different methods of the Graphics class such as drawString() and drawImage(). The drawString() method is used to draw the text and the position of the text onto the graphical window.

User Manual:

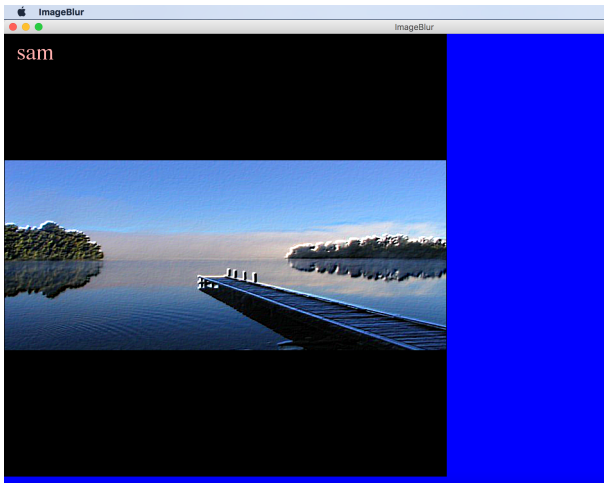
This section contains a list of guidelines to help the user run Image Expression efficiently

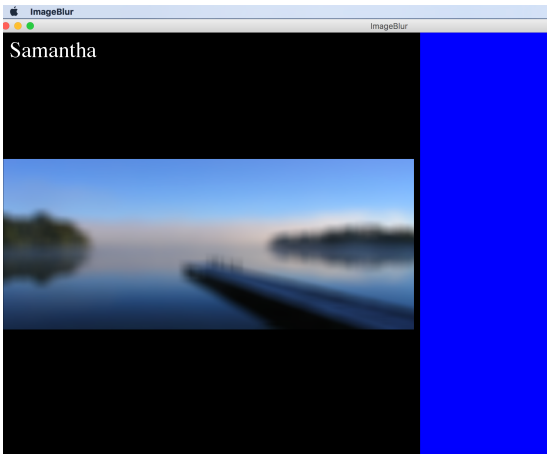
- When setting the dimensions for the frame size, they must be the exact or larger dimensions of the image itself. It is recommended to keep the frame window at these dimensions for the laptop full screen view- Height: 1600 & Width: 1600.
- The user must enter the filename of the image they would like to use including the .jpg Ex: "ocean.jpg".
- The image file **must** be in the same folder as the source code.
- The user must follow instructions on what number they would like to enter to choose the desired filter

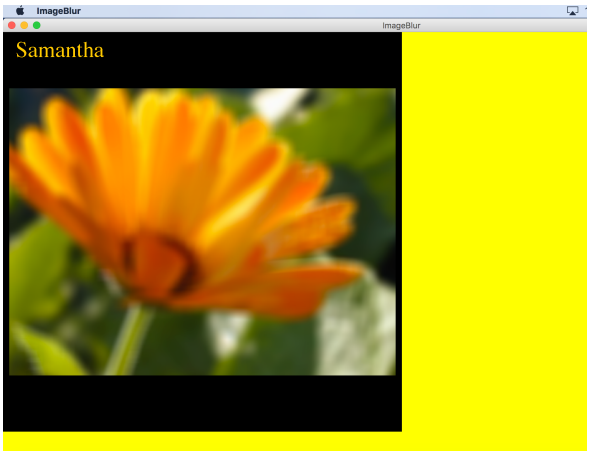
Conclusion:

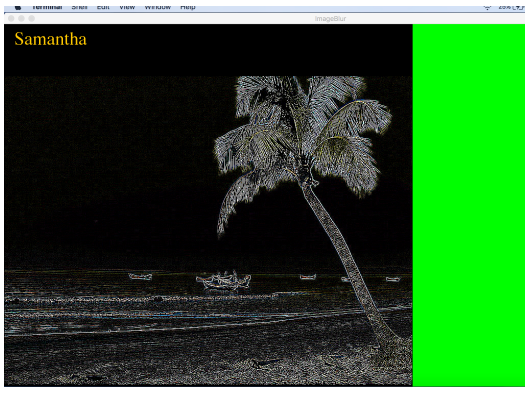
Blurring, edge detection, and sharpening are only a few of the many image processing filters that can be developed with java code. Convolution is the mathematical process behind the art of image processing. Java packages assist in providing the methods used to apply these filters onto the image. Image Expression allows the user to enter his or her own input to create their desired image by using methods that invoke convolution. In using the code, the user has the freedom to design a unique image that can be different every time they run the program.

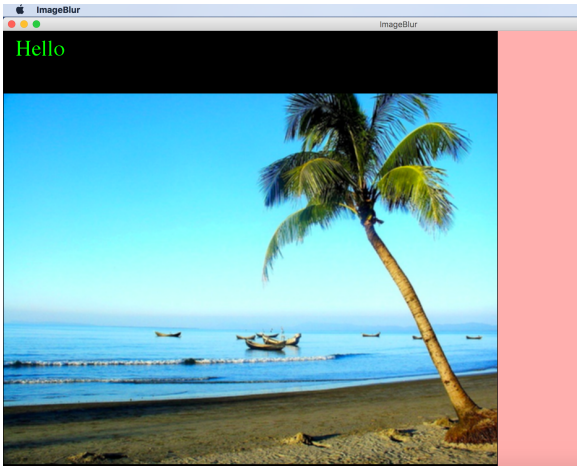
Image Expression Examples:











Resources

Huxtable, Jerry. "Blurring For Beginners." *JH Labs*. N.p., n.d. Web. 2017. <http://www.jhlabs.com/ip/blurring.html>.

Java Oracle. "Package java.io." *Java Platform SE 7*. N.p., 11 Jan. 2016. Web. May 2017. <https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html>.

"Package java.applet." *Java.applet (Java Platform SE 7)*. N.p., 11 Jan. 2016. Web. May 2017. <https://docs.oracle.com/javase/7/docs/api/java/applet/package-summary.html>.

"Package java.awt." *Java.awt (Java Platform SE 7)*. N.p., 11 Jan. 2016. Web. May 2017. <https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html>.

"Package java.awt.image." *Java.awt.image (Java Platform SE 7)*. N.p., 11 Jan. 2016. Web. May 2017. <https://docs.oracle.com/javase/7/docs/api/java/awt/image/package-summary.html>