Computational Photography Project Non Photo-realistic Rendering

MEMBERS: Akshay Joshi
Jacob Steele
Ritambhara Singh
Samyukta Jadhwani.

ORIGINAL PROPOSAL:

For the computational photography final project, we are proposing to create an Nvidia tablet application to take an input image and apply a artistic style to this input to produce a non-photorealistic output image.

These artistic styles will be predefined for the user, but will be highly extensible. We each expect to implement different artistic styles and combine them for the completed project. We expect to be able to work on any form of input image and a wide range of size.

To achieve these non-photorealistic rendering we are going to compile several papers on the subject and implement their strategies. The true difficulty in this project is the creation of several different styles as options for our users. Currently, we have several ideas such as modeling the gradient, luminance, and edge styles of the artist from several images, but this is the area which we have least understanding in terms of what is necessary.
It is expected that each member will contribute at least one stylistic model along with contributing to the base tablet app in some way.

The presentation for original, mid-phase and final are included as Project Proposal, Midphase Prezi and Final Presentation respectively. the prezi can also be viewed HERE.

COMPLETED TASKS WITH MEMBER CONTRIBUTION:

Target Tasks	Status	Member(s)
Effects		
Cartoonize	Completed	Samyukta
Painterly	Completed	Akshay
Pencil Sketch	Completed	Jacob
Pointillism	Completed	Ritambhara
Extra effects		
Comic, Glass, Minima Filtering, Oil Paint, Vampirize, Rastafarian, Warhol	Completed	Samyukta
Tablet Implementation		
App Development	Completed	Ritambhara
Pen and Ink	Completed	Jacob
Painterly	Completed	Akshay
Website	Completed	Akshay, Samyukta

RUNNING THE CODE:

Painterly:

Call jar file and pass it an image path as input. It will save the output as painterly_source image name in the same folder as the source image.

Other filters:

All other style names describe the effect they apply, and are matlab files and can be run as follows:

Using Matlab as FILENAME('IMAGE.EXTENSION')

OR

The executables can be run as **FILENAME IMAGE** where the files are FILENAME.EXE and IMAGE.EXTENSION . The executables are in a folder called Executables.

TABLET:

myTablet.zip contains the application code for out project's tablet application.

In order to run the app, you should have Eclipse and required Android SDK installed on your computer. After building the correct libraries, run the Project "MyTablet", while connected to the android device with a USB cable (with Developer Settings ON). This ports the application onto the device and after running it on the device once, you can disconnect the cable, and use "MyTablet" as a stand-alone application.

You can upload an image either from the existing image library or by clicking a photograph and apply one of the two available filters, ie. painterly or pen and ink.

(We have left the MyTablet app on the NVIDIA tablet for you to use).

WEBSITE:

The website is currently hosted on one of our computers but due to firewall issues you may or may not be able to access it. The link is : http://128.143.63.69/filter.html

Website.zip contains all the files you need to run the website. Install and run WAMP Server and extract the contents of Website.zip into the wamp/www directory. Add the wamp directory to the matlab path. check if localhost is working and then call your_ip/Filter.html OR localhost/Filter.html

TAKE-AWAYS:

- Code Runtime is bad
- Matlab Doesn't port well to Android SDK
- Java is best for programming for portability
- Android SDK
  - Doesn't fully support java.awt/java.swing
  - Doesn't work with Matlab-converted to C/C++

RESULTS:

Feel free to use the results as desired. They are in the folder called Results.