Image Processing for Boost GIL

Gaurav Hoskote
Pune Institute of Computer Technology.
Bachelor of Engineering (BE).
Electronics and Telecommunication.
gaurav.hoskote@gmail.com
Homepage: https://github.com/gauravhoskote/codes

Availability:
1)The project will take 3 months to get over.
2) I will be busy preparing for my Insem and Endsem exams which will be held in March and May respectively. Insems start from 11the March and should end within a week. Endsem exam timetable is not out yet. I will keep the mentors informed about my availability, the duration of the exam period should not be more than 2 weeks.

About Me:

I am currently in my final year of Engineering (Electronics and Telecommunication). I have been passionate about coding throughout my entire course of engineering. I got recruited by HSBC for the role of Software Development Engineer (SDE) few months ago through on-campus placements. I chose Image processing as my elective in last semester and have a firm grip on all concepts and algorithms as I have implemented them before.

I have always been comfortable coding in C++ and also used the same language to crack my interview. I noticed that people who knew Java had an upper hand because of their in-built libraries. Boost is trying to solve this problem by making available many interesting libraries which makes me want to contribute to it.

Also I have worked on a project which required knowledge of IOT and Image processing. The project used Haar Cascades Algorithm to recognize the faces in a room and switch the lights ON and OFF accordingly.

Even after GSOC I intend to contribute to Boost [GIL] library and optimize it as much as possible.

Knowledge Rating: (in range 0 to 5)

C++ 98/03 (traditional C++): 4

C++ 11/14 (Modern C++): 4

C++ Standard Library: 4.5

Boost C++ Libraries: 3

Git: 2

I will be using Xcode (Mac OS) for my projects.

Overview

The Boost GIL library consists of many mainstream algorithms, but lacks image processing algorithms like thresholding, morphology etc. In my project I intend to add a few header files to the GIL library. These Header files can be placed inside a folder called 'imageprocessing' which can be then placed in the GIL library. This folder will contain header files like 'threshold.hpp', 'morphology.hpp' etc. If someone wishes to perform thresholding on images, he/she will have to import the header using : #include "boost/gil/imageprocessing/threshold.h"

Goals

1. To implement the headers "threshold.h", "morphology.h", "normalize.h", "filter.h", and "fft.h" which can be placed in a folder.
2. To optimize the algorithms and make them generic.

Milestones

I. Week 1:

Discuss with the mentors the specifications of the threshold.h header. Also find out efficient ways to implement simple and adaptive thresholding.

II. Week 2:

Implementing the thresholding algorithms and placing them in the header file "threshold.h".

III. Week 3:

Discuss with the mentors and implement dilation and Erosion for morphology.h header file.

IV. Week 4:

Discuss with the mentors and implement smoothening and sharpening algorithms.

V. Week 5:

Discuss various efficient ways to implement normalization with mentors and also fix bugs if any in the code developed till date.

VI. Week 6:

Implement the normalize.h header file.

VII. Week 7:

Discuss with mentors the algorithms to be implemented and implement Wiener filter.

VIII. Week 8:

Implement the Average and Median filters.

IX. Week 9:

Discuss the issues with the current header kernel.h and make necessary changes to it.

X. Week 10:

Fix issues in the convolve.h header and discuss with mentors.

XI. Week 11:

Discuss the ways fft.h can be implemented. Also fix any existing issues.

XII. Week 12:

Implementing the fft.h header file and optimizing it.

Competency Test:

I have implemented a sample header file “threshold.h”. (Path: “codes/gsoc2019/boost/gil/imageprocessing/”) It contains a simple function which performs thresholding operation on an 8 bit grayscale image. The arguments to this function are image source and destination views as well as the threshold value. This is just a sample code and the actual header file will consist of generic versions of it.

This sample header file will also be found in the attachments of the mail.