DSP Project Abstracts
CE 101
February 2009

Project Code:

CE101-feb2009-01

Title:

Generalized Masks with Matrix Convolution

Proponents:

Jeuz Ser Aragon
Omar Jerico Filoteo
Daniel Jeffrey Lagazo

Abstract:

Convolution is the single most important technique in Digital Signal Processing. In essence, it is a way of combining two signals to form a third signal. It is indispensable precisely because it relates the input signal and an impulse response to generate an output signal. For audio signals, you can for example compute a filter's response to an entering signal by convolving it to the filter. Likewise audio signal, we can also compute image convolutions, and this is what matrix convolution is all about. If you want to know the outcome of a filter to an image, you will simply have to convolve the entering image with a n x n mask representing the impulse response of the image filter. If we know a system's impulse response, then we can calculate what the output will be for any possible input signal. This project presents the impulse response of an image filter or simply mask.

Notes:

Project Code:

CE101-feb2009-02

Title:

Image Stitching via Difference

Proponents:

Kevin Armstrong Cua
Datu Em Ibn Saud C. Sinsuat

Abstract:

This image stitching application serves the purpose of matching two images with the intent of stitching these two to have a single image.

Ease of use is a prime endeavour of the proponents; therefore, the application was done using wxDevC++, which enabled a Graphical User Interface (GUI). With this, selection and customizing are done with just clicking and dragging. Moreover, users are able to view selected images to be stitched, as well as the stitched images.

Additionally, this application supports the following image formats: bmp, gif, jpeg, and png. Matching can be done against two different image formats for a larger scope of use.

Notes:

Project Code:
CE101-feb2009-03
Title:
Color Counting
Proponents:
Konde Luciano Domingo
Nathaniel John Ramos
Abstract:
Color Counting is a C++ program developed using wxDev and the wxWidgets library (specifically wxImage and wxString), which aims to count the number of colors present in a given image. This program is an extension of the previous project Image Viewer by authors Paolo Olayres and Grace Benedicto. The difference is an added "Perform Color Counting" option to the Edit menu, which is the project itself. The program counts the number of colors it detects in the loaded image based on the user's radius selection.

The program first asks for a radius to determine its sensitivity to colors being encountered. The program then creates a histogram of colors of the chosen image. The program looks next for the most common color and then proceeds to clear the area around it according to the user's chosen radius. After this, it then looks for the next occurring color until none are left. The program keeps track of how many colors it has counted as well as each color's occurrence while doing this.

The program creates a text file of the summary of the colors, their respective occurrences in the image and the number of pixels that were counted in the process. It also creates a BMP image of the colors it encountered, ranked from most common to rarest. It also presents the color's thickness according to that color's ratio in the output image.

Notes:

Project Code:

CE101-feb2009-04

Title:

Fast Fourier Transform Using Cooley-Tukey Algorithm And Bluestein's Algorithm

Proponents:

David Joseph N. Tan
Raymund Gerard B. Velasco
Melvin V. Victoria

Abstract:

This project aims to construct the FFT algorithms, specifically, Cooley-Tukey Algorithm and Bluestein's Algorithm. Generally, Cooley-Tukey Algorithm is used if a given divisor is in powers of 2 or is specified by the user; otherwise, the Bluestein's Algorithm is used. In addition, these algorithms are implemented in the DFT class which can be reused for future applications. Moreover, this project also includes an interface to the DFT class with the use of the Sound Data class. The Sound Data class gets its data from a file or a vector and calculates the DFT as specified by the programmer or the user. This class also includes the option to divide the data gathered in chunks and to use the Cooley-Tukey Algorithm with powers of 2 or a specific divisor. Lastly, the main function is used as a sample program to illustrate how to use the Sound Data class.

Notes: