DSP Project Abstracts
ECCE Department
Ateneo de Manila University

DSP Project Abstracts
ELC 152
October 2006

Project Code:
ELC152-oct2006-01
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-02
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-03
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-04
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-05
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-06
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-07
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-08
Title:

Proponents:

Abstract:

Notes:

Project Code:
ELC152-oct2006-09
Title:

Proponents:

Abstract:

Notes:

DSP Project Abstracts
CE 101
February 2007

Project Code:

CE101-feb2007-01

Title:

WavPro 1.0

Proponents:

Maris Alano
Vernice Casareno
Rachelle de la Rama

Abstract:

The WavPro 1.0 Program is a project for the CE101 Class entitled Digital Signal Processing under the Electronics, Computer, and Communications Engineering Department of the Ateneo de Manila University. This program is written in Sun Microsystem's Java Programming Language and is developed in Netbeans IDE.

This program can process the audio signals produced by Sony MZ-NH 700 Recorders. These 16-bit audio signals are *.wav files with sampling Frequency of 44.1 kHz and follow the canonical wav format.

From these wav files, graphs are presented in a Graphical User Interface. These graphs, which are shown on different tabs are representative of the Amplitudes, Power and Histogram of the said wav file. The user can adjust the range of the graphs, as well as zoom it, according to her/his preferences.

These graphs, as well as text areas indicating the wav file details can be saved as image and text files, respectively.

This project has its significance in rain monitoring that use acoustic recordings.

Notes:

Project Code:

CE101-feb2007-02

Title:

XMCS Player / Recorder

Proponents:

Anna Veronica C. Baterina

Jeric V. Macalintal

Abstract:

A console-based XMCS Player / Recorder is implemented using C/C++. Along with this is the design and implementation of an XMCS class that provides recording and playback functionalities, through the Windows Waveform-audio API. Two other classes, the Utility class and the UserInterface class, are also created to provide the other functionalities needed to complete this project. The player is capable of playing 8-bit and 16-bit single- and dual-channel audio, with sampling rates of up to 44.1 kHz. The recorder is capable or recording 8-bit and 16-bit single-channel audio with sampling rates of up to 44.1 kHz.

Notes:

Project Code:

CE101-feb2007-03

Title:

Data Acquisiton via Parallel Port

Proponents:

Ma. Fatima Johara de Lara

Ma. Theresa Joy Rocamora

Abstract:

This project aims to utilize the alternative uses of latex, specifically prophylactics (PRO), as a dielectric for Electrostrictive Actuator Polymers, EAPs, or artificial muscles, and eventually develop them for Biomedical Applications.

Information is sent to and from the computer using the parallel port. The first part of the system involves a Digital to Analog Converter (DAC) which is hooked onto a High Voltage Amplifier (HVA) which in turn feels the voltage into the muscle system. The second part feeds back the information into the computer via Analog to Digital Converter (ADC) which is fed into a Random Access Memory (RAM) chip, that is, in turn, connected back to the parallel port.

The active bandage system contracts and expands based on voltage command. An additional feature is the capability of the system to get the feedback voltage from the artificial muscles. This system can be used in a therapeutic exercise system in conjunction with ECG monitoring of the patient or in therapies like ECCP in reducing coronary heart diseases [1].

Notes:

Project Code:

CE101-feb2007-04

Title:

Multiple Message Steganography System

Proponents:

Carlos Ezequiel
Rogelio Naval Jr.
Jerwin Louise Uy

Abstract:

The study involves the design and testing of a digital image steganography system for secure data transmission over a network. The system has applications in securely transmitting sensitive data such as medical records, personal accounts and secret codes. Digital image steganography is the technique of hiding a message inside a digital image. Multiple messages would be hidden inside a Portable Network Graphics (PNG) image using a PC encoder program written in Java. A Java-based PC decoder program would then extract one of the messages using a password. Different passwords would extract different messages from the image. Further improvements include adding a public key encryption layer and using JPEG image-based steganography.

Notes:

Project Code:

CE101-feb2007-05

Title:

Digital Stethoscope and ECG Waveform Generator and Analyzer

Proponents:

Jovilyn Therese B. Fajardo

Ma. Rowena Grace Q. Predas

Abstract:

Heart diseases are prevalent nowadays. Cardiac centers are few and not easily accessible, especially to those living in the provinces and remote areas. In this respect, this project aims to create an application that allows digitized data from a digital stethoscope and an electrocardiogram device to be graphed as a waveform in a personal computer. Analog heart signals from a patient are acquired using the digital stethoscope and electrocardiogram hardware devices. These signals are then converted to digital form using an analog-to-digital (ADC) circuit. Data acquisition is also at regular intervals, which is accomplished by using a random access memory (RAM) chip. Moreover, an analysis of the acquired digitized signals is done to detect heart abnormalities if present and this is done by comparing the signal values with the values acquired from the available medical databases.

Notes:

Project Code:

CE101-feb2007-06

Title:

RGBText Image Composer

Proponents:

Feliz Grace D. Flores
Mary Ann N. Pelecio

Abstract:

This project is a DEV-C++ console-based implementation primarily intended for an image composer in RGBText format. This will include the basics in composing an image files such as making a new image, resizing the image, setting/changing the color background, setting/changing the pen color, plotting a point, drawing a line, and saving the file in RGBText Format.

The RGBText format consists of a three two-dimensional arrays which corresponds to the r, g, and b respectively. Each array will contain a value from 0 to FF depending on the user's input. These values denote the color value for each allocation.

Error checking is also implemented in the program as to assure the correctness of the user's input and to prevent malfunction in the program.

The user could view the finished product through the RGBText Viewer, a project done by Dalino, Escalante, Favorito, Onglao, and Tiamzon.

Notes:

Project Code:

CE101-feb2007-07

Title:

XMCS Real-Time Recorder and Fast Fourier Transform Analyzer

Proponents:

Edgardo P. Guevarra, Jr.

Fatima Aiza F. Medina

Abstract:

This project aims to integrate two past DSP projects last 2nd semester 2006 namely as XMCS Recorder for Windows and Implementation of a Chromatic Tuner using Fast Fourier Transform. This is in line with the group's thesis, Characterization of an Acoustic Laser (SASER).

From previous projects, XMCS file type has proved to be a very convenient file format in terms of the valuable data that can be extracted from it. Its capability to show the sampling rate, bits per sample, radix, and the recorded data written in binary or hexadecimal form enables it to efficiently play audio waveforms and signals. Because of the great features of an XMCS file, this project adopts a DSP group's project last semester in creating XMCS real-time recorder to capture the audio waveform from the acoustic laser (SASER) as an XMCS file format. The data from the recorded XMCS file will be then used in preparation for the computation of the fundamental frequency using a Fast Fourier Transform algorithm, which is also adopted from a second past DSP project that uses FFT to implement a chromatic tuner. The changes in the computed fundamental frequencies of the waveforms can be graphed to see the beam profile of the SASER, which is essential in the characterization of the acoustic laser.

Notes:

Project Code:

CE101-feb2007-08

Title:

Histogram Display for the RGBText Editor

Proponents:

Beverly Danielle T. Marquez

Abstract:

A histogram is “a graphical display of tabulated frequencies”. Each point of the data spread is separated into a particular category or “bin.” The data is then displayed as a graph. In photography, a histogram is taken of the image's shadows, midtones and highlights. This is an important qualitative measure of the image's exposure level. A good exposure would represent something similar to a Bell curve with a large amount of midtones and an equal amount of shadows and highlights.

A previous project, the RGBText Editor, submitted by Mssrs and Mdmes Jose Dalino, Arlan-Z Escalante, Cathy Favorito, Luigi Onglao, and Claire Tiamzon, was a graphical histogram and channel editor for RGBText files. The program was written in C++ using Bloodshed's Dev-C++ Compiler with WX plug-ins.

The goal of this project is to incorporate a histogram display to the user interface. The histogram of the original picture is installed on the side of the screen. When a change is made to the picture, data is refreshed, but instead of refreshing the original histogram, a second histogram is shown where the changes have been made. In other words, the user may view the image's histogram as he edits the picture. The project is merely a concatenation to the project of Dalino et al., but will improve the usability of the software vastly in favor of the user.

As the function would be merely an add-on to existing programs, emphasis would first be placed on the existing code and on how it works. Other tasks would focus on the algorithms required to calculate and display the histogram.

Notes:

Project Code:
CE101-feb2007-09
Title:

Proponents:

Abstract:

Notes:

DSP Project Abstracts
ELC 152
October 2007

Project Code:

ELC152-oct2007-01

Title:

Musical Note Identifier

Proponents:

Mark David G. Abat
Faye Janelle A. Aninipot
Mary-Clare C. Dy

Abstract:

The musical notes are identified by the project from a recorded tone file loaded in the XMCS Recorder and Player. Using a Fast Fourier Transform Algorithm, the frequencies and magnitudes will be acquired from the samples. Having this information, a table for the different frequencies of musical notes will be compared with the samples to acquire the fundamental frequencies of the inputs. This project is able to identify one musical note from a tone file.

The recorded tone files utilized in this program is based from another project by the class: Digital Piano. The said project is able to acquire harmonic samples in hexadecimal format, and the task of the Musical Note Identifier is to convert the data into decimal format so that the values can now be compatible with the FFT algorithm with fixed point classes. From the FFT algorithm, the magnitude and frequency are identified; for the magnitude, the two highest amplitude will be taken into account, and within that range of frequencies, the fundamental frequency can be identified using the compare algorithm

The idea of the Musical Note Identifier is very similar to the previous project, XMCS Real-Time Recorder and Fast Fourier Transform Analyzer, since the previous project is also interested with the different frequencies of the musical notes, the group adopted its algorithm for the comparison of frequencies and just made some adjustments like including the octave in the output and removing unnecessary parameters. The XMCS Player and Recorder was also used in the project.

The main component of the project is the improvement of the Fast Fourier Transform algorithm. What made this project unique is that a fixed-point class was incorporated in the FFT algorithm to make the calculation of the frequencies and amplitudes of the sampled notes more precise. Also, this project is able to output satisfactory results as opposed to the previous project.

Notes:

Project Code:

ELC152-oct2007-02

Title:

Cross-Correlation of Acoustic Files in XMCS Format

Proponents:

Gemalyn D. Abrajano
Dani Joy J. Famarin
Jane Louie F. Zamora

Abstract:

The Cross Correlation of Acoustic Data in XMCS Format processes two XMCS files for cross correlation. The XMCS files are obtained using the XMCS Player and Recorder taken from a previous project. The file is in 16 bit binary format at 44.1 kHz sampling rate, the standard format of the acoustic WAV file data from which the converted XMCS file is taken from. The acoustic data were produced from recording rain sound using a Hi-MD player and was retrieved as WAV. For our analysis in this project we converted the WAV file to XMCS by internally recording the sound while the original file is being played in VLC Media Player, a recent discovery we made.

Cross correlation formula enables one to compare two or more data samples to see whether there exists a high or low similarity between samples. If the correlation is high then there is a high comparison between the samples processed and if not then the correlation is low. In the case of this program, it is limited to comparing only two acoustic data. Another limitation of the program is that it cannot handle a very large file with a lot of samples because processing takes a long time.

The program will be outputting a summary of the peak and the minimum correlation of both samples. The user is free to choose whether he or she would want to view the results in console or in text form. The results of this program may be used in the analysis of the rain data in the rain thesis later on.

Notes:

Project Code:

ELC152-oct2007-03

Title:

Image Manipulation Using Edge Detection

Proponents:

Raiza Pamela Ante
Christine Boongaling
Rommel Regonas

Abstract:

This project aims create a working program in C++ that performs edge detection upon any RGB text file the user inputs. Before performing the edge detection, the program will ask the user to set the outline color for the detected edges and the preferred threshold value. The program outputs three separate RGB text files—(1) the original image in greyscale, (2) the original image with its edges marked with the color specified by the user and (3) a black outline of all the edges amidst a white background. The thickness/thinness of the outlined edges can be controlled through the given threshold value—for instance, setting a higher threshold indicates a more discriminating detection of edges, thereby producing an output image with less marked edges.

Among the various techniques for edge detection, this project makes use of the difference operator, one of the simplest and most straightforward methods, which basically performs a series of pixel subtractions (on the converted greyscale data) to obtain the maximum difference of neighboring pixel pairs per window. These absolute value of the maximum differences are the potential edges of the image. Screening out the real edges from the weak/fake ones involve image thresholding wherein the magnitude of the maximum difference is compared to the set threshold value. If greater than or equal to the threshold value, the center pixel will be decided upon as an edge. Consequently, the RGB pixel value (of the original image and the blank image) will be consequently changed in accordance with the user input. The newly changed images, along with the greyscale representation of the image will then be written to a separate RGB text file.

Notes:

Project Code:

ELC152-oct2007-04

Title:

XMCS Steganography

Proponents:

Jose Raphael C. Arenas
Adrian Joseph C. Mozo
Christian John D. Rigor

Abstract:

The XMCS Steganography Project is a console-based software that enables the user to embed a message from a text file first converted to its binary representation into an XMCS file and extract that same message from the steganized XMCS file. There is no significant difference in the sound between the original XMCS file and the steganized XMCS file. In addition, this program is implemented using the C++ Programming Language. The program is capable of embedding/extracting messages from 16-bit single-channel XMCS files. Particularly, the program embeds the text message using 1, 2, 4 or 8 least significant bit substitutions in each XMCS file voice sample.

Notes:

Project Code:

ELC152-oct2007-05

Title:

Image Pattern Recognition Using Template Matching

Proponents:

Roy Khristopher Bayot

Jan Lester Gerard Lofranco

Abstract:

A program has been developed to perform image pattern recognition using template matching. This program accepts two images that uses the rgbtext file format. One of the files is presumed to be the larger image while the smaller file is the template image. The template is the image being searched for in the larger image. The program developed uses the convolution algorithm wherein pixels are matched with another. The result of the convolution is stored in an output text file. Locations of the match are also stored in another output text file.

Notes:

Project Code:

ELC152-oct2007-06

Title:

Digital Image Filtering Using Matrix Convolution

Proponents:

James Nicole L. Brojan
Lionel Oliver R. Juinio
Joan Frances N. Peralta

Abstract:

This paper discusses the theory, methodology and final output of a signal processing project utilizing RGBText files. The program is designed to manipulate images using convolution in the RGB plane. Image manipulation involves adjustment of brightness, edge detection, blur, and other similar filters. Image convolution is done on fixed point data values. The original floating point values are converted to fixed point before convolution. After convolution, data are converted back to floating point. The program was constructed using Dev C++ and the graphic user interface from a previous project entitled “Image Editor using RGBText” is used to view the RGBText Files being used.

Notes:

Project Code:

ELC152-oct2007-07

Title:

RGBText Stitching

Proponents:

Emmanuel Alpheaus M. Ching
Gerald Earvin M. Valentino

Abstract:

This project is a C++ console-based implementation primarily intended for image stitching in RGBText format. This will include the basics in stitching image files and saving stitched files in RGBText or text format.

The RGBText format consists of a three two-dimensional arrays which corresponds to the r, g, and b respectively. Each array will contain a value from 0 to FF depending on the user's file input. These values denote the color value for each allocation.

Error checking is also implemented in the program as to assure the correctness of the user's input and to prevent malfunction in the program.

The user could view the finished product through the RGBText Viewer, a project done by Dalino, Escalante, Favorito, Onglao, and Tiamzon.

Notes:

Project Code:

ELC152-oct2007-08

Title:

Voice-to-Text Converter

Proponents:

Retcher Hans N. Ching
Zent Casimir L. Lim
Leandro Alejandro S. Soriano

Abstract:

Two console-based voice-to-text converter is implemented using C/C++. The first console is the design and implementation of an XMCS that gives the output of one's voice in a decimal text file. The second console uses the output decimal text file as the input and would compare the peaks of the graph in order to output the respective letter coefficient. This mainly uses the Fast Fourier Transform in order to be able to compare the voice patterns. The converter is capable of playing16-bit single- and dual-channel audio, with sampling rates from 22k up to 44.1 kHz.

Notes:

Project Code:

ELC152-oct2007-09

Title:

RGBText Video CODEC

Proponents:

Ted Angelo T. Chua
Emarc Perri D. Magtanong
Juan Carlos Y. Yu

Abstract:

This paper discusses the details regarding a digital image processing that uses the format of RGBText files. The program is designed to allow the user to compress a series of RGBText file images. The program also provides the user the ability to decompress the compressed output file. The efficiency of compression varies depending on the images being used. The program was created using Dev C++.

Notes:

Project Code:

ELC152-oct2007-10

Title:

Tone Generator

Proponents:

Kevin Cristobal
Federic Karl-magne Pulma
Richard Tiu

Abstract:

This project is a tone generator implemented using C++. The generator would use the XMCS format as its output, while the input would be a text file. The program would generate unique tones for the letters, numbers, and some punctuation marks included in the input text file. The input text file should contain at most 50 characters; any other excess would be disregarded. The program would have a given table of values for each character. The values would then be inputted to an oscillator and then outputted to a file. The output file can be sent to another computer or played to relay the message.

Notes:

Project Code:

ELC152-oct2007-11

Title:

RGBText Compression Using Haar Wavelets

Proponents:

Adriell Matthew Julius A. Dagasuan

Abstract:

This project involves the compression and decompression of RGBtext image files. The compression and decompression algorithms are implemented using the combination of Haar Wavelet Transforms and Run Length Encoding. The pixel values from the image files are separated into R, G, and B values, then each of the values are run through the transform operation and also in the RLE encoder. Similarly, the decoder takes in the RLE data from the compressed file and then lets the values run through the inverse transform operation. Image compression ratios vary with the data of the RGBtext image file. Some image files are fairly successful in being compressed while others are not.

Notes:

Project Code:

ELC152-oct2007-12

Title:

Night-Time Traffic Density Image Analyzer

Proponents:

Jimson G. Ngeo
Dave Francis B. Rayel
Jan Michael G. de Villeres

Abstract:

This paper discusses the theory, methodology and final output of the signal processing project involving night-time traffic density analysis of images. The program is designed to process night-time traffic images and display the number of cars based on the detection of the number of possible car head or tail lights in an image. As part of the whole process of the traffic density analyzer, the program can also produce the gray-scale equivalent of an image and can also produce single colored blobs from an image while filtering out the rest of the colors. The program was constructed using Dev C++.

Notes:

Project Code:

ELC152-oct2007-13

Title:

LSB Substitution Steganography on RGBText

Proponents:

Adrin V. Del Rosario

Archie Q. Dolit

Abstract:

This project is a DEV-C++ console-based implementation of LSB Substitution Steganography. The algorithm is able to hide text into a cover image represented in RGBText format. The text to be hidden can be typed from the screen or can come from a specified text file. The text which is transformed into an array of bit pairs is embedded in the last two bits of the cover image. The output of the encoding operation would be a stega image. The cover image and the stega image should not have visual differences. The algorithm is also able to decode the hidden text in a stega image. The text can then displayed in the screen or be saved in a text file.

Error checking is also implemented in the program to assure that user inputs are of the correct format. This prevents usual program crashes.

Notes:

Project Code:

ELC152-oct2007-14

Title:

Digital Piano

Proponents:

Jennifer Mae S. Fabros

Gilbert Z. Peren

Abstract:

This project is a console-based implementation of a piano, limited to a single octave where the "middle C" can be found. The program used to write the code for this application is Dev-C++. This project can be divided into five main parts (as far as the note playing concerned): 1) fetching the correct note frequency given that the correct key is pressed, 2) generating the samples that represent the different notes with the use of an IIR filter oscillator, 3) translating the samples into a format that can be read by the sound card, 4) shaping the samples so that it could imitate the sound of a piano using the ADSR envelope, and 5) playing of the samples using the XMCS player.

Notes:

Project Code:

ELC152-oct2007-15

Title:

XMCS Tab Generator

Proponents:

Chrisandro Favila
Erwin Ray Hapal
John Lester Lim

Abstract:

Guitar tablature, or guitar tab for short, is a music sheet that show which notes are being played on the guitar. Guitar tab is useful for learning how to play a variety of songs without even having formal guitar training.

With this project, any music recorded with a sampling rate of 4096Hz, and in 16-bit hexadecimal XMCS file format, can be translated into a guitar tab in plain text format. The XMCS data sequence is fed into a Fast Fourier Transform algorithm to allow spectral analysis of the audio signal recorded. Peak detection and filtering methods are used to determine which frequency component will be translated into the guitar tab file. The program is then tested on simulated waveforms and actual audio recordings. Results are interpreted by analyzing the discrepancies between the tab generated and the actual note played or the expected output.

Notes:

Project Code:

ELC152-oct2007-16

Title:

Bitmap Image to Editable Text Format

Proponents:

Daryl Aaron C. Gaerlan
John Michael C. Rono
Karen Angela R. Tiambeng

Abstract:

The project is a Dev C++ console-based implementation which aims to convert a simple bitmap file containing text into an editable text file format. The program will analyze the characters seen in the bitmap file and compare this to the project's character database. The output will be the determined text placed in a text file document.

The program consists of several parts: (1) conversion of the bitmap file into Portable Pixel Map (ppma) format; (2) conversion of the ppma file fomat into an RGBText file; and (3) streaming the bit values from the RGBText file and comparing it from an RGBText file containing all the sample characters. This sample file will serve as the database upon which the original scanned bitmap file will be based.

Notes:

Project Code:

ELC152-oct2007-17

Title:

Steganography on RGBText Files

Proponents:

Michael U. Siapno
Ronell B. Sicat
Kelvin Chryzth P. Velasquez

Abstract:

This project revolves in the art and science of steganography. The Greek origin of this word is "covered" and "hidden writing". For digital signal processing, it is basically embedding data on another data without significantly changing the latter. In this project, a text file can be hidden on an image file, specifically an RGBText file, without changing the image's appearance.

Since pixels in the RGBText file are represented by the combination of different magnitudes of red, green and blue, the colors of the pixel can be changed to some extent in order to hide the message in the image. The change in color is so slight that the naked eye will not notice the changes of color in the image. The group selected the least significant bits of the RGBText file (LSBs) as the pad where the message will be hidden. So the message to be hidden is converted to its binary equivalent and then written on the red, greed and blue values of each pixel by changing the last bit of each value and for each pixel. And for the extraction of the hidden message from an RGBText file, the LSBs of each color value in each pixel is taken and stored. The collected LSBs are then converted to the character equivalents, which are then written into a new text file specified by the user.

In addition to these processes, the project utilizes the fixed point method of representing values. This is of utmost importance so that all projects in Signal Processing have a universal way of representing values such as pixels of an image or samples of a sound file. This will enable projects to utilize other projects' fixed point manipulation functions.

The steganography program was compiled and tested afterwards by hiding a sample text file in an RGBText file. The new image was viewed and compared with the original image. No significant differences were seen. Then, the message was extracted from the written RGBText file into a new text file. The text file was opened and contained the original message written into the original RGBText. Aside from these tests, the group tested the capabilities of the program to detect input errors and verified the program's efficiency.

Notes:

DSP Project Abstracts
CE 101
February 2008

Project Code:

CE101-feb2008-01

Title:

Audio Equalizer

Proponents:

Jonathan Ryan C. Ang
Jose Lorenzo L. Losantas
Moses Brahms Y. Tiu

Abstract:

Implementation of an audio equalizer using C++ was done in this project. The design and implementation of an equalizer class was done to carry out the program's equalization functionalities. Other classes (file class, parser class, interpolator class, windower class, fourier class, combinator class, and scaler class) were also created to provide the other functionalities needed to complete this project. The program is capable of implementing a 10-channel audio equalizer for XMCS files that are 8-bit or 16-bit single-channel audio, with sampling rates ranging from 8 kHz to 44.1 kHz. A graphical interface is supported by the program. Graphical display of the input and output files' frequency spectra are also supported.

Notes:

Project Code:

CE101-feb2008-02

Title:

Pitch Shifting

Proponents:

Federico M. de la Cruz
Lea Cristina D. Macaraig
Oliver L. Rivera

Abstract:

A pitch-shifting program is implemented through the C++ language. The application accepts data in XMCS format and converts the samples into their fixed point equivalents. Short Time Fourier Transform is used to obtain the frequency domain representation of the input signals. The algorithm for the actual pitch shifting process follows the phase vocoder technique, which performs sample rate conversion while retaining the duration of the signal. The user is allowed to manipulate the extent of the pitch shifting through the pitch shift factor. The program outputs data using the same format as the input.

Notes:

Project Code:

CE101-feb2008-03

Title:

Edge Detection For Color Images

Proponents:

Daphne Dale Domingo
Paula Nessa Godoy
Galilee Semblante

Abstract:

A program that performs edge detection on RGB text file formats was constructed using C++. This edge detection program asks for user input to specify the degree of thresholding desired, thus controlling the level of edge detection performed. For example, specifying a higher threshold will result to thinner output edges since the program will employ a more discriminating detection of edges.

The use of the difference operator is employed in the program. The difference operator is basically a series of pixel subtractions of neighboring pixels per window. Through this method, the maximum differences are gathered and thus the possible edges are detected. If the maximum difference is greater than or equal to the threshold value, it is considered an edge. Thresholding filters out the real edges from the non-edges. All the calculations of the difference operator will be performed in the RGB color scheme.

Notes:

Project Code:

CE101-feb2008-04

Title:

DSP Audio Processor

Proponents:

Dennis Ted P. Duran
Ma. Angeline U. Sansolis
Jacqueline Patricia Michelle L. Velasco

Abstract:

An audio processing and simulation project is proposed. An audio file is the program's expected input. The program will process the input file as stated by the user by implementing an FIR filter via a graphical user interface (GUI). This interface contains graphs of input and output signals and the specified filter's frequency response. After processing, the expected output is the audio file either played through a speaker or saved to an output format, and corresponding filter coefficients as exported to a text file.

Notes:

Project Code:

CE101-feb2008-05

Title:

Image Resizer

Proponents:

Joshoua C. Esmenda
Mark Andrew S. Mateo
Jerome Vergil S. Paez

Abstract:

An image resizer is implemented in C++ employing the fixed-point format. Images handled are in the RGBtext file format, and are resized using bilinear interpolation. The project also includes a graphical user interface, where the input image and output image (after resizing) can be viewed. The program also supports opening and resizing of BMP, JPG, PNG, and GIF files. These files can be saved in RGBtext, BMP, JPG, and PNG formats.

Notes:

Project Code:

CE101-feb2008-06

Title:

Audio Combination

Proponents:

Rafael R. Moreno
Camille Anne S. Paras
Jesus Ramon A. Ronquillo

Abstract:

A program that combines two XMCS audio files is created using the C++ programming language. As an added feature, a sampling rate converter is implemented.

The implementation is done through creation of a fixed point class that converts the values in the XMCS files into fixed point. Also, an interpolate function is created for converting the sampling rates of the files into the desired sampling rate. Lastly, a combine function is also created for the combination proper of the audio files.

The program can process XMCS files with sampling rates of 8000 – 44100 Hz, and having 1 – 16 bits per sample.

Notes:

Project Code:

CE101-feb2008-07

Title:

Manipulation of Color Resolution

Proponents:

Giselle Mae M. Pacot
Karina P. Palileo
Clarisse Eileen V. Sabulao

Abstract:

This project is a wxDEV C++ implementation of a program that manipulates the color resolution of an image that is in RGBText file format. The function is integrated into the Image Editor program originally developed by Dalino, Escalante, Favorito, Onglao, and Tiamzon in October 2006 and further developed by Marquez in March 2007. Specifically, the function developed for this project prompts the user to enter the number of bits assigned for a particular color component (red, green, and blue), allowing the user to increase or decrease the color resolution based on the image on the canvas. The histogram and the file and image manipulation functions that were available in the previous project are still accessible in this new development. Equivalent BinText files of the old and new images are also generated to compare the actual number of bits assigned for each color component.

Notes:

DSP Project Abstracts
ELC 152
October 2008

Project Code:

ELC152-oct2008-01

Title:

XMCS Recorder and Player Modification

Proponents:

Katrina Eunice B. Aganon
Melissa Frances L. Balmes
Chariz Sandra H. Ramos

Abstract:

The XMCS Player / Recorder is a console-based project implemented using Dev C++ that is capable of audio recording and playback. It includes three classes: XMCS class, UserInterface class and Utility class that perform the complete functionalities of the project. It can record an 8-bit and 16-bit single-channel audio with a sampling frequency limited to values: 8000 Hz, 11025 Hz, 22050 Hz or 44100 Hz. The player can playback 8-bit and 16-bit single and dual-channel with a sampling frequency within the range of 8 kHz and 44.1 kHz.

Derived from the XMCS Player / Recorder is the XMCS Recorder and Player Modification, which is also implemented using Dev C++. The main feature of the project is a binary output file in contrast to the the text file of the previous project . The new Recorder and Player provides an option for saving a recorded audio with a different resolution. Lastly, the available values for the sampling frequency is increased but still within the range of 8000 Hz to 44100 Hz. The project is implemented with a 16-bit single channel recording and playback.

Notes:

Project Code:

ELC152-oct2008-02

Title:

Image Matching via Edge Detection

Proponents:

Luis Antonio A. Aguja
Francis Paolo B. Dy
Stephanie R. Tan

Abstract:

This image processing application serves the purpose of matching two images, via edge-detection algorithms, with the intent of determining the degree of confidence that one image is similar or identical to the other image.

Users are given the privilege of choosing a desired Edge-Detection Algorithm needed for matching. The following are the three algorithms of choice made available to users: Sobel, Laplacian, and Kirsch Edge Detection. Additionally, the threshold of perceived edges can be customized by the user, as well as the level of confidence of matched images.

Ease of use is a prime endeavor of the proponents, therefore, the application was done using wxDevC++, which easily 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 matched, as well as the edge-filtered 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:

ELC152-oct2008-03

Title:

Digital Audio Reverberation Using the Schroeder Reverberator

Proponents:

Kalil Christian B. Almonte
Dick Sanny P. Español
Lester U. Vinzons

Abstract:

A software implementation of digital audio reverberation in XMCS format is developed via simulation of the Schroeder reverberator in Dev-C/C++. Source codes and classes of a previous project entitled, "XMCS Player/Recorder," are utilized and modified accordingly to provide support functionalities, such as audio recording and playback. Three new classes are created for the primary functions of the project: two subclasses for the comb and all-pass filter implementation and one primary class for the main Schroeder reverberator, which includes functions to modify, store, and retrieve filter parameters. The result is a console-based application which accepts an XMCS file, runs it through the simulated Schroeder reverberator configurable by the user, and produces another XMCS file with the desired reverberation effects.

Notes:

Project Code:

ELC152-oct2008-04

Title:

Complex Fast Fourier Transform Analyzer for XMCS Recorder

Proponents:

John Neslie Ang

Jose Rafael Dimaano

Abstract:

The program takes the XMCS output file and produces a text file that would give the amplitude of the frequencies.

The program is created using the C++ programming language, with the usage of the Dev C++ Compiler specifically. The process is implemented by the use of the Complex Class available in the library of the compiler.

The program is capable of analyzing with different and specified number of chunk division, N, in the Time Domain, and later on to choose the number of samples per chunk, M, in the Frequency Domain. The program utilizes the general form of Fourier Transform and utilizes techniques in decreasing the number of computations to make the computations fast.

Notes:

Project Code:

ELC152-oct2008-05

Title:

Philippine Banknote Recognition

Proponents:

Angelo Carlo E. Arceta
Barbra Francine V. Te
Michelle Christine L. Yap

Abstract:

Philippine banknotes are commonly recognized in automated vending machines using their magnetic components. Another possible way to detect the identity of a banknote involves image processing. With a sample image of a Philippine banknote, a specific pattern matching algorithm is implemented in C++ to determine its identity. Such technique deals with similarity measures between the sample image and template images stored in a database. In the sum of the square differences method, low similarity measure values indicate a possible match. As for a modified version of a normalized cross-correlation, similarity measures approaching the value of three suggest potential match. The project includes a graphical user interface (GUI) which has the image brightness and similarity measure functions. The program supports JPG images. Generated information from the pattern matching involving the sample image can be saved in text file.

Notes:

Project Code:

ELC152-oct2008-06

Title:

Image Viewer

Proponents:

Grace Ann B. Benedicto

Paulo J. Olayres

Abstract:

The Image Viewer project is a continuation of the previous Image Resizer project. Its aim is to improve the quality of the resizing algorithm as well as to improve the image viewing pleasure of the user. Using C++ and the wxWidgets library, the Graphical User Interface (GUI) was restructured to remove space-consuming window elements such as the toolbar, sidebar, and status bar that were previously found in the Image Resizer. The structure of the source code was also modified so that it uses wxImage, a native image container, which allowed the image to be rendered faster. The software also contains utility functions such as sequential viewing of pictures, image dragging, rotating, copying to clipboard, and saving to different popular file formats.

As for the resizing algorithms, three options were provided: bilinear interpolation, preblurring method, and Lanczos' resampling algorithm. The bilinear interpolation was derived from the previous group but was modified to use the wxImage class. This algorithm proved useful when enlarging an image but was insufficient when shrinking it. Because of this, the preblurring method was introduced. This works by applying a blurring filter on an image before resizing it using bilinear interpolation. Lastly, the Lanczos' resampling algorithm was implemented and proved superior to both bilinear interpolation and preblurring method when shrinking an image. However, when the image is enlarged, bilinear interpolation is still better because of its faster execution time.

Notes:

Project Code:

ELC152-oct2008-07

Title:

An Algorithm for Sound Brightness Determination
via XMCS Player and Recorder

Proponents:

Lovelyn Faith L. Cabiles
Christel Anne R. Fitero
Lord Alec M. Pasion

Abstract:

A software that can calculate for the spectral centroid of music files is implemented using C++ language. The process includes an algorithm for the Discrete Fourier Transform of time-domain audio samples. The music files that can be processed by the software are limited to samples recorded from the XMCS Player and Recorder which is in .txt file format, with data listed in decimal form. Both the number of samples that are analyzed per chunk of the file and the sampling frequency are user-defined and can be varied.

Applying the program to different music samples, it is observed that the spectral centroid of pure voice is generally lower than the spectral centroids of musical instruments such as guitar, violin, and piano. However, when samples with combined voice and instruments are compared with instruments alone, the spectral centroids of the combined vocals and instruments tend to have higher centroid values. These results confirm that the algorithm constructed is able to discriminate voiced music samples from instrumental ones and can be applied in masking techniques usually incorporated in vocals remover applications.

Notes:

Project Code:

ELC152-oct2008-08

Title:

Eye Recognition, Tracking and Blink Detection for Car Safety Applications

Proponents:

Mitchell Angelo J. Cabrera
Manuel Carlo C. Cristobal
Wyndale P. Wong

Abstract:

This project uses eye recognition for car safety applications by creating a program that can recognize if a person has fallen asleep behind the wheel by successfully detecting if a person's eyes are closed. For this purpose, the program is equipped with eye-searching, eye-tracking, and blink-detection capabilities. This is implemented using the Dev C++ programming language, utilizing the necessary predefined commands and functions from the Open Source Computer Vision Library (Open CV), to make image and video processing possible. The program is able to perform a search for eye presence, to spot the location of the eyes, to record its color gradient, to track the moving eyes and to identify if the eyes are closed or opened again. The source code is also tailored to allow the program to accept live video input from a web camera, to establish a graphic user interface, and to send a visible output signal on the computer screen with each successful blink-detection.

Notes:

Project Code:

ELC152-oct2008-09

Title:

Image Cross-Fader

Proponents:

Ama Cara M. Cagampan

Shella Joyce P. Villanueva

Abstract:

The Image Cross-Fader is implemented using C++ programming language employing a fixed-point format. Images handled are converted in RGBtext file format. There are two algorithms used in performing the cross-fader technique: cross-fading by pixels and cross-fading by color through the R, G, and B values of the images. The project includes reading from two images and producing a series of frames cross-fading from the first input image to the second input image. The program is able to read RGBtext, JPEG, BMP, and PNG file formats. The output images can be saved individually in RGBtext, JPEG, BMP, and PNG file formats.

Notes:

Project Code:

ELC152-oct2008-10

Title:

Change Original Color

Proponents:

Jromed C. Cheng
Daphne C. Landoy
Lorlynn A. Mateo

Abstract:

A selective color changer, Change Original Color, is implemented in C++ with the use of the fixed-point format and algorithms namely, color masking and color matrix transformation.

By using color matrix transformation, the program, aside from employing selective color change, is also capable of carrying out global color changes. Hence, two options are provided to handle these features: the first option for color masking and the second option for color matrix transformation.

The images handled by the program are in the RGBtext file format, but can also support BMP, JPG, PNG, and GIF files. Processed images can be saved in RGBtext, BMP, JPG, and PNG formats.

Finally, the project can be accessed using a graphic user interface, where the original picture and the processed pictures can be viewed altogether.

Notes:

Project Code:

ELC152-oct2008-11

Title:

FIR Filter Designer

Proponents:

Kinsley C. Chua
Josef Carlo A. Pusta
Edgar Marko S. Trono

Abstract:

A console-based Finite Impulse Response (FIR) filter designer is implemented using the C++ programming language. The program asks the user for the filter parameters, namely, the type of filter, the sampling frequency, the desired cutoff frequency/frequencies and the width of the transition band(s). Using the Windowed-Sinc algorithm, the program then calculates the filter kernel. A text file containing the values of the samples of the designed filter's kernel is then provided by the program.

Notes:

Project Code:

ELC152-oct2008-12

Title:

XMCS Audio File Filtering Using LTI Systems

Proponents:

Jason Paul M. Cruz
Hansel C. Dy
Zyra S. Landrito

Abstract:

A program that filters an XMCS audio file is created using the C++ programming language, with the help of the Dev-C++ software.

The implementation is done through the use of fixed point classes from the Audio Combinator Project by Moreno, Paras, and Ronquillo (February 2008) that converts the values in the XMCS files into their respective fixed point representation (FPR). For filtering, the coefficients for the LTI specfiles may be either arbitrarily defined and/or obtained from the current project entitled "FIR-D", by Chua, Pusta, and Trono, and/or from IIR Digital Filter Design Applet. The input XMCS files are filtered by integrating the FPR of these hexadecimal values with the coefficients from LTI spec files using the difference equation. The modified FPR is then written on an output XMCS file defined by the user.

The XMCS files mentioned, both input and output, may be played using the XMCS Player and Recorder.

Like other XMCS programs, it can process 8 kHz to 44.1 kHz sampling rate, and at 8 or 16 bits per sample.

Notes:

Project Code:

ELC152-oct2008-13

Title:

Variable Image Resolution

Proponents:

Jaime Paulo S. Dumaliang
Monika G. Pagkanlungan
Mark Noel C. Runas

Abstract:

This project is a C-based program, created through Dev-C++ version 4.9.9.2, which converts JPEG, BMP, and RGBText formats into a binary file format. This was specifically designed to filter the Red, Green and Blue color dimensions/components (RGB values) of all pixels of the input file format. Based on the number of bits that the user indicates, the program converts every four filtered bits into its hexadecimal equivalent. The new hexadecimal characters are saved in the binary file.

The program also creates the equivalent RGBText version of the binary file. Both the RGBText file and the binary file are created simultaneously. The RGBText Viewer created by a previous project can be used to open the new RGBText file and view the new image produced. According to the number of bits that the user indicates for the RGB components, the resulting image will differ in color from that of the original picture. The nearer the RGB user inputs are to 8, the more similar it is to the original image. The nearer the RGB user inputs to 1, the resulting RGBText when viewed will have a very different and blurred resolution. If the RGB user inputs are all equal to 8, the resulting RGBText file must contain the original RGB values of the input image.

The project also consists of another console, a binary file checker which checks if the resulting binary file is indeed correct. This checker analyzes the binary file and produces another RGBText file, which must be the same as the first RGBText file produced. Both RGBText files must have the same size, and same contents, and same image when viewed using the RGBText Viewer.

Notes:

Project Code:

ELC152-oct2008-14

Title:

XMCS Cropper and Stitcher

Proponents:

Dale Derrick Y. Dy
Maria Margarita Therese M. Pineda
Angeli C. Silang

Abstract:

A console-based XMCS Cropper and Stitcher, which was originally adapted from the XMCS Player / Recorder developed the previous year, is implemented using C/C++. It continues to use the originally designed and implemented XMCS class that provides playback functionalities, with now the added features of audio cropping, stitching, slow motion playback, and a display of the time elapsed for both slow-motion and normal playback. The UserInterface class was modified accordingly to continue to provide both the new and the original functionalities needed to complete this project, whereas the Utility class was not modified anymore. The Player is capable of playing 8-bit and 16-bit single- and dual-channel audio, with sampling rates of 8, 11.025, 22.05, and 44.1 kHz, showing the time that has elapsed as it plays back. The Cropper is capable of cropping any loaded XMCS file given a starting and end point and the Stitcher is capable of combining up to five XMCS files. The Slow Motion Player is capable of playing back the audio at half the frequency of the original while its time display still follows that of the original.

Notes:

Project Code:

ELC152-oct2008-15

Title:

Improvement on the Audio Equalizer

Proponents:

Roma Lynne G. Espiritu
John Vianney Z. Isiderio
Ray Edwin T. Ocfemia

Abstract:

In this project, an improvement on the design and structure of the graphic audio equalizer software is implemented using C++. The creation of a windows-based Graphical User Interface, or GUI, through the utilization of the wxWidgets toolkit is done as part of the modifications of the project. In addition, the GUI is designed to support the integration of the audio equalizer and the XMCS player/recorder into a single program. The purpose of combining the two projects is to provide flexibility and ease of use by being able to access both in the same graphical user interface. Another major development is the replacement of the direct computation for the Discrete Fourier Transform to the utilization of the Fast Fourier Transform algorithm to enable a quicker and more efficient data processing. The program is able to provide a 10-band graphic audio equalizer for 8-bit or 16-bit single-channel audio XMCS files having specified sampling rates from 8 kHz to 44.1 kHz. Furthermore, the amplitude interval for each of the frequency band is increased to more levels. Moreover, an XMCS player/recorder that is capable of playing 8-bit and16-bit single- and dual-channel audio, as well as recording 8-bit and 16-bit single-channel audio with specified sampling rates of up to 44.1 kHz is included. Lastly, predefined equalization settings and an optional display of the XMCS data in text format are made available as part of the features of the program.

Notes:

Project Code:

ELC152-oct2008-16

Title:

Optical Character Recognition of Inclined Text

Proponents:

Andy S. Gimpaya Jr.
Thom Arjay C. Jaucian
Jan Reinhard M. San Luis

Abstract:

The project is a Dev C++ console-based implementation which uses the Rotational Matrix equation in rotating inclined scanned text. The program works for RGBtext files, therefore several file conversion must first be accomplished before the image can be rotated. Included in this project are programs that allows conversion from BMP file format to PPMA and from PPMA to RGBtext. These programs can first be used before rotating the image.

The group used the rotation matrix equation given by

$$x' = x \cdot \cos ø + y \cdot \sin ø$$
$$y' = y \cdot \cos ø + x \cdot \sin ø$$

in rotating text images that were scanned inclined. After rotating the image the group uses Optical Character Recognition through pixel by pixel comparison to get the Character equivalents of the text. This identified characters are then written in a TXT file.

Notes:

Project Code:

ELC152-oct2008-17

Title:

FFT Graph Imager

Proponents:

Aldrin F. Khan
Mark Baldwin B. Martinez
Tal Miguel A. Navarro

Abstract:

This project aims to create images of graphs of Fast Fourier Transforms of sound files recorded and processed with the previous DSP project, XMCS Real-Time Recorder and Fast Fourier Transform Analyzer.

The XMCS audio file format is very useful owing to the many utilities it provides us in analyzing the audio. From an XMCS file the XMCS player can extract the the sampling rate, bits per sample, radix, and the recorded data written in binary or hexadecimal form. This makes it very convenient for audio file analysis.

One integral part of the XMCS player's signal analysis repertoire is its ability to find the Fast Fourier Transform of an audio file. In Digital Signal Processing, the frequency spectrum of a signal refers to the way energy in the signal is distributed over its various frequency components.

The FFT Analyzer as an improvement to the XMCS player was implemented in 2007. It breaks down an audio file into numerous samples, then takes the FFT of each sample. The output of the FFT analyzer is a series of numbers which represent the magnitudes of the frequency spectrum of the sample.

This project is a further improvement on the XMCS Player – FFT Analyzer. The group takes the output of the FFT analyzer and graphs it using the image-manipulating library EasyBMP. This gives the user a clearer picture of the frequency spectrum of a sound file. Also, the graphs are labeled appropriately so that the values are clearly displayed.

Notes:

Project Code:

ELC152-oct2008-18

Title:

Transparency of Two Bitmap Images Using EasyBMP

Proponents:

Carlos Miguel M. Lacson
Marc Ericson C. Santos
Marty Peterson L. Tan

Abstract:

This project aims to perform a transparency effect on two bitmap images at a time. The program is designed to accept user inputs of image filenames, pixel coordinates, transparency value and output filename.

This project is an application of the EasyBMP Library. The EasyBMP is an open source software that contains libraries to perform manipulation of bitmap files. Through the use of the EasyBMP library, the program is able to load images and extract their RGB values. These RGB values are used in computing the new RGB values of the transparent image. Using the EasyBMP library, the transparency is placed on top of the background and the output image is created in to a separate file.

Notes:

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:

Project Code:

Title:

Proponents:

Abstract:

Notes:

Project Code:

Title:

Proponents:

Abstract:

Notes: