DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

The 1996 Mississippi State University Conference on

Digital Signal Processing

What: EE 4773/6773 Project Presentations
Where: Simrall Auditorium, Mississippi State University
When: December 2, 1996 — 1:00 to 4:00 PM

SUMMARY

The Department of Electrical and Computer Engineering invites you to attend a mini-conference on Digital Signal Processing, being given by students in EE 6773 — Introduction to Digital Signal Processing. Papers will be presented on:

- parallel implementations of fast Fourier transforms;
- real-time audible frequency detection and classification;
- analysis of forestry images for scenic content.

Students will present their semester-long projects at this conference. Each group will give a 12 minute presentation, followed by 18 minutes of discussion. After the talks, each group will be available for a live-input real-time demonstration of their project. These projects account for 50% of their course grade, so critical evaluations of the projects are welcome.

Session Overview

1:00 PM — 1:10 PM: J. Picone, Introduction

1:15 PM — 1:45 PM: Michael Balducci, Ajitha Choudary, and **Jon Hamaker**, “Comparative Analysis of FFT Algorithms In Sequential and Parallel Form”

1:45 PM — 2:15 PM: **David Gray**, Craig McKnight, and Stephen Wood, “Audible Frequency Detection and Classification”

2:15 PM — 2:45 PM: Yaquin Hong, **Nirmala Kalidindi**, and Liang Zheng, “An Algorithm To Determine The Scenic Quality Of Images“

3:00 PM — 4:00 PM: Demonstrations in 434 Simrall

AUTHOR INDEX

Volume 2

Digital Signal Processing

Table of Contents

Comparative Analysis of FFT Algorithms in Sequential and Parallel Form

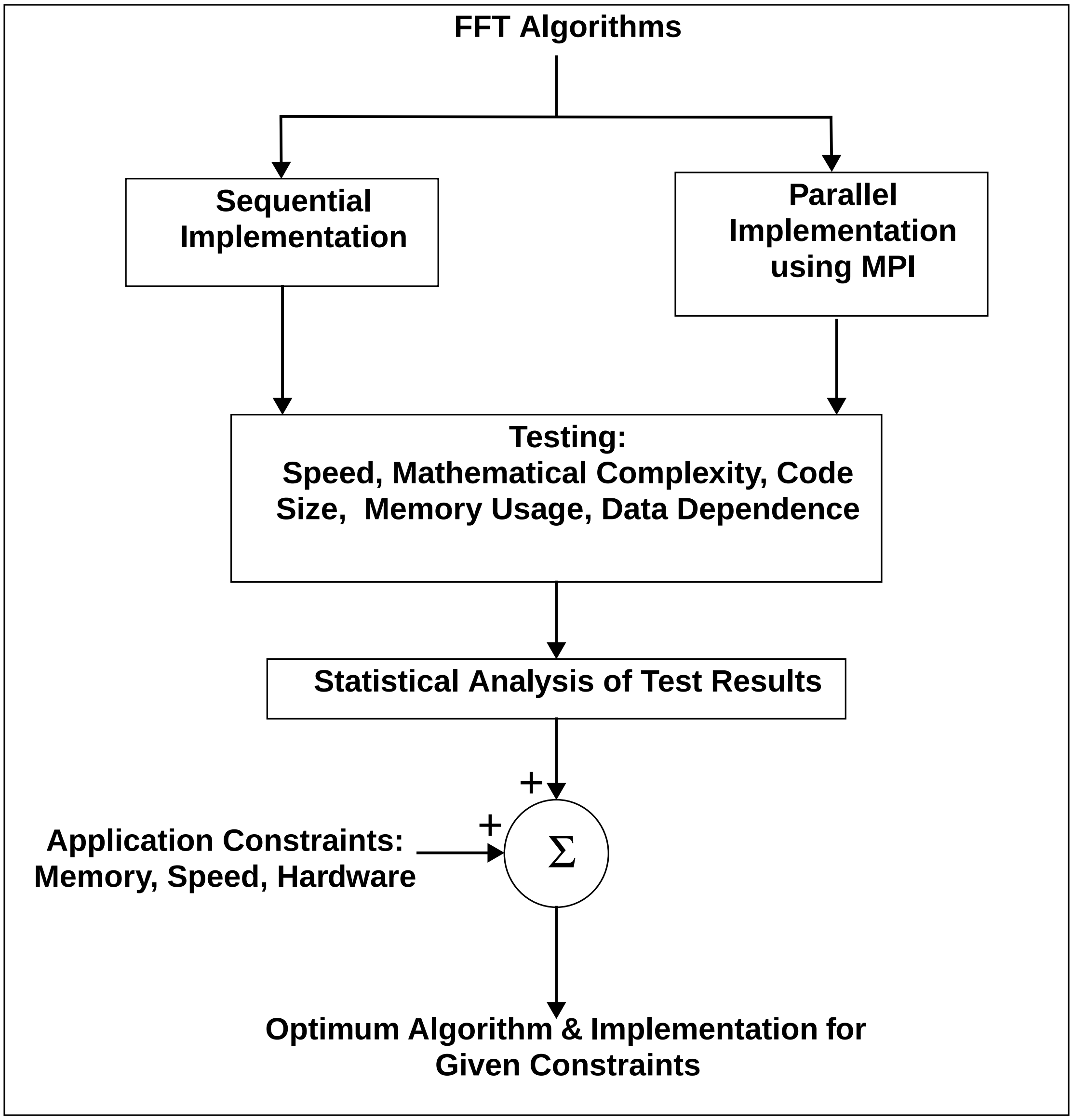
by

Michael Balducci balducci@erc.msstate.edu
Ajitha Choudary ajitha@erc.msstate.edu
Jonathan Hamaker hamaker@isip.msstate.edu

ABSTRACT

- Motivation: Need for efficient FFT implementations which fit the specific constraints of an application.
- We merge a large number of FFT algorithms into a common, object-oriented framework.
- We have produced a public-domain collection of sequential and parallel implementations of a variety of FFT algorithms
- Performed statistical analysis of each algorithm based on speed, mathematical complexity, and memory usage.
- Future Goals: Develop routines which use our statistical results to pick the FFT algorithm implementation which best matches a user's application and hardware constraints.

Strategy



FFT Algorithms

Sequential Implementation

Parallel Implementation using MPI

Testing: Speed, Mathematical Complexity, Code Size, Memory Usage, Data Dependence

Statistical Analysis of Test Results

+

+

Σ

Application Constraints: Memory, Speed, Hardware

Optimum Algorithm & Implementation for Given Constraints

Theory

Discrete Fourier Transform (DFT)

- Yields frequency spectrum, given N time samples of a signal

$$X(k) = \sum_{n=0}^{N-1} x(n)\ W_N^{kn} \qquad k = 0, 1, ..., N-1$$

$$W_N = e^{-\frac{j2\pi}{N}}$$

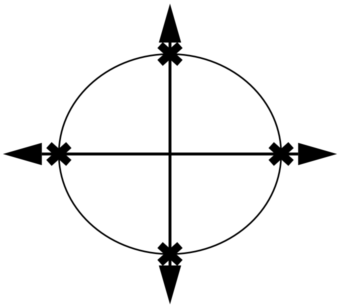
Q: We have the DFT...why do we need the FFT.
A: Complexity reduction...we reduce the number of operations from $O(N^2)$ to $O(N * \log_2 N)$.

For a 1024 point transform we reduce our operations from ~ 10^6 to ~ 10^3 - a 98% reduction!!

The Fast Fourier Transforms (FFTs)

- FFTs use symmetry and periodicity of W_N to eliminate redundancy

$N = 4$
$k = 0, 1, 2, 3$
$n = 0, 1, 2, 3$



Only four unique values!

- Values of W_N can be pre-calculated to attain higher speeds.

- What’s the catch?...We **trade speed for memory (money)**

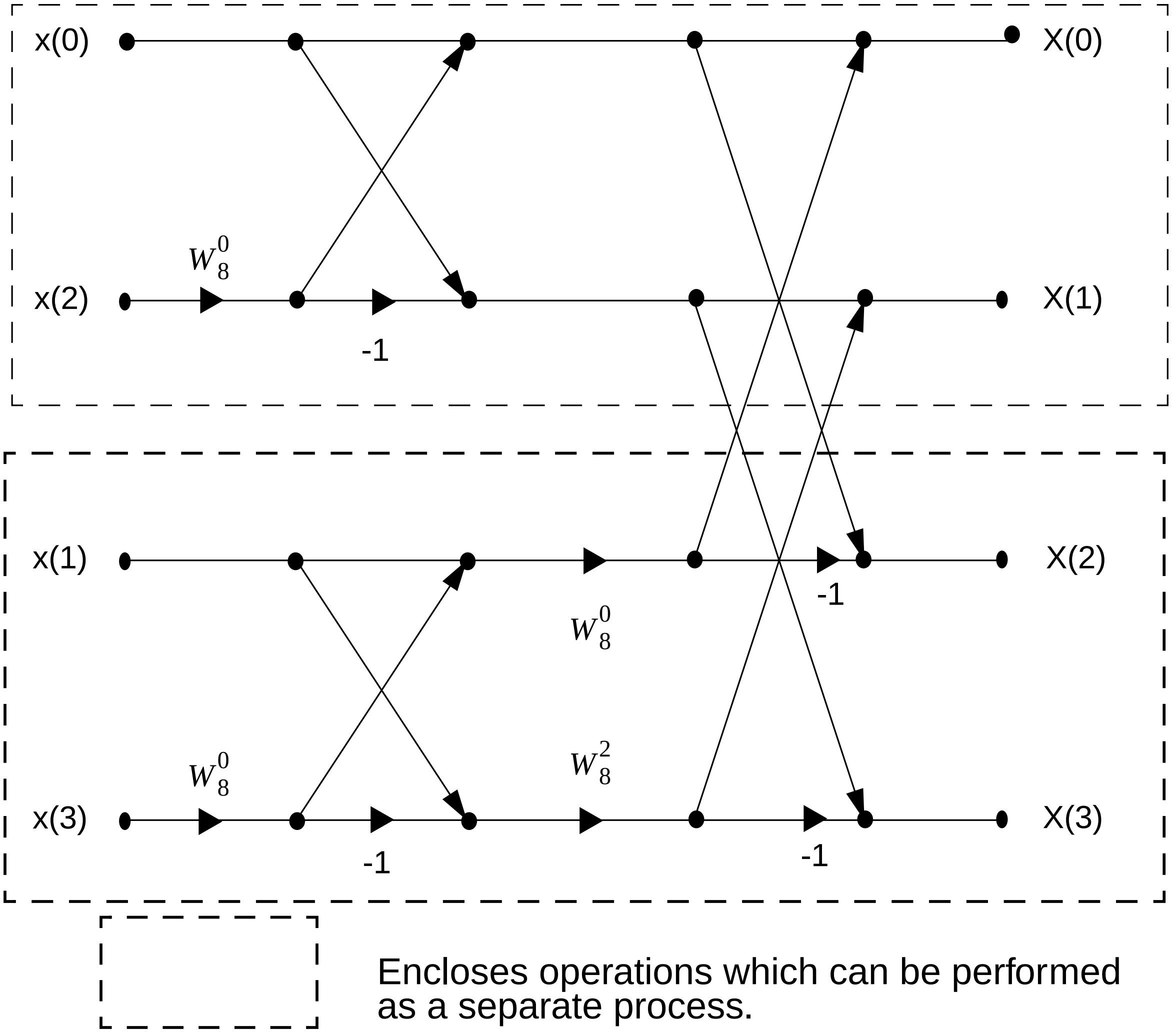
Radix Algorithms

- ☞ If we limit our data size to be of the form R^V we can gain efficiency
- ☞ Most popular form of Radix algorithms are Radix-2 and Radix-4

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n\ even} x(n) W_N^{kn} + \sum_{n\ odd} x(n) W_N^{kn}$$

- ☞ We break the DFT into a sum of smaller DFTs to reduce complexity.

Parallel Strategy for Radix Algorithms



Encloses operations which can be performed as a separate process.

Split-Radix Algorithm

- DFT can be broken into a single N/2-point and two N/4-point DFTs.
- Using the Radix-4 for the N/4 portions reduces the number of calculations - increasing efficiency.
- Split-Radix uses both Radix-2 and Radix-4

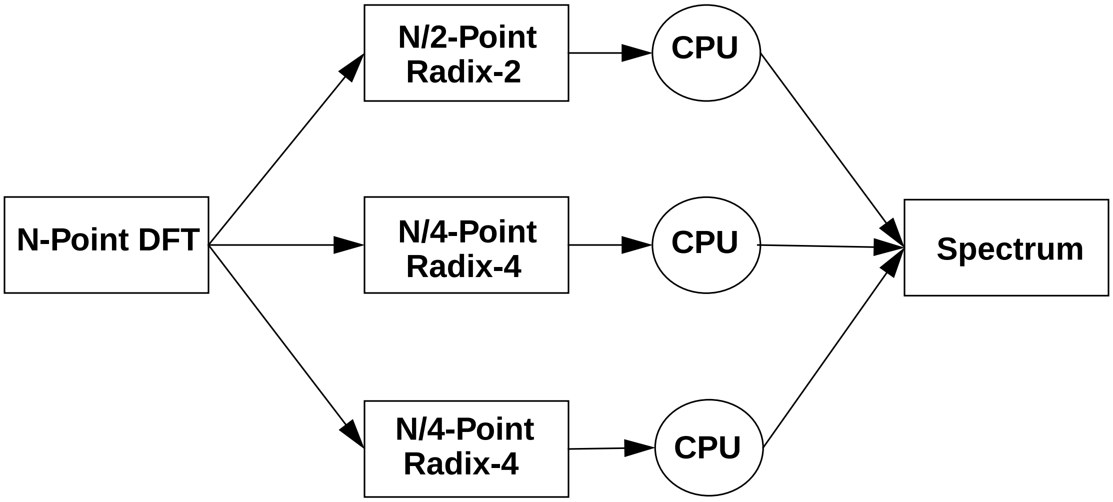
Hartley Transform (FHT)

$$X(k) = \sum_{n=0}^{N-1} x(n)\left(\cos\frac{2\pi}{N}nk + \sin\frac{2\pi}{N}kn\right)$$

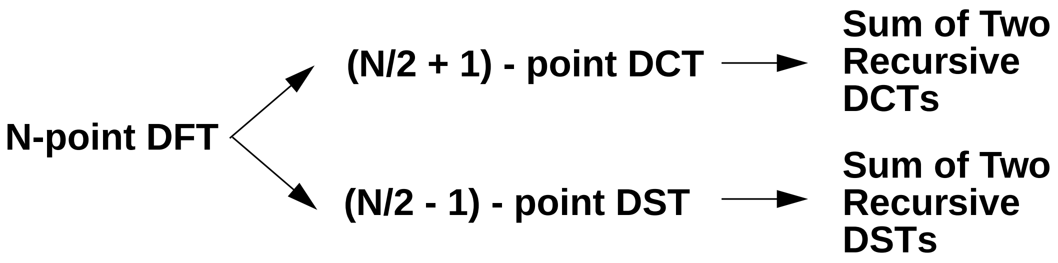
- **No complex arithmetic!**
- Can still use Radix-2, Radix-4, Split-Radix, etc.
- Conversion factors necessary to produce Fourier coefficients.

Parallel Strategy for Split-Radix Algorithm and Split-Radix FHT

- Similar to Radix...use separate processor for independent butterflies



Quick Fourier Transform (QFT)

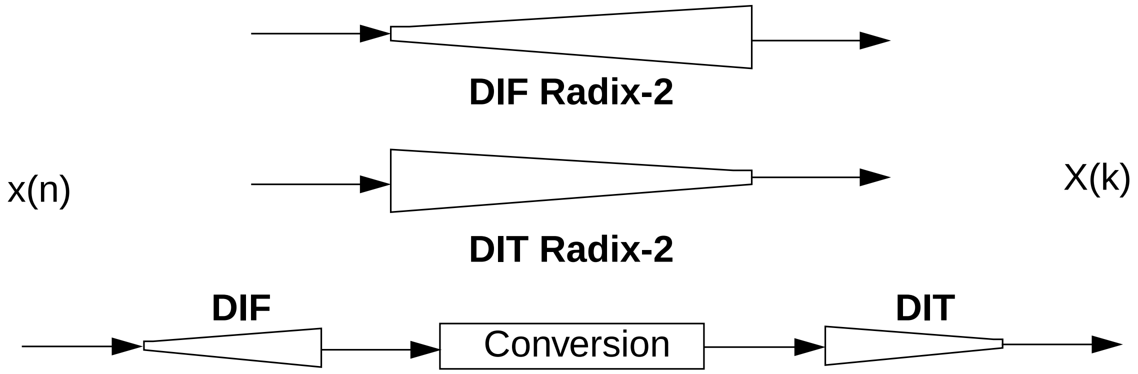


N-point DFT

(N/2 + 1) - point DCT → **Sum of Two Recursive DCTs**

(N/2 - 1) - point DST → **Sum of Two Recursive DSTs**

Decimation-in-Time-Frequency Algorithm (DITF)

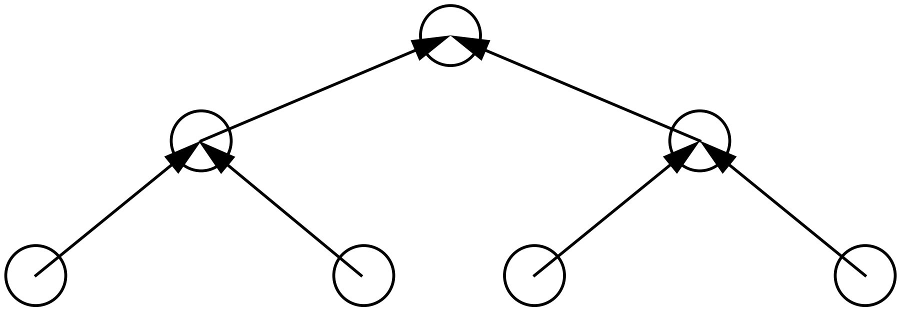


DIF Radix-2

DIT Radix-2

DITF

Parallel Strategy for Recursive QFT and DITF Algorithms



○ **Indicates calculations which can be performed by one processor.**

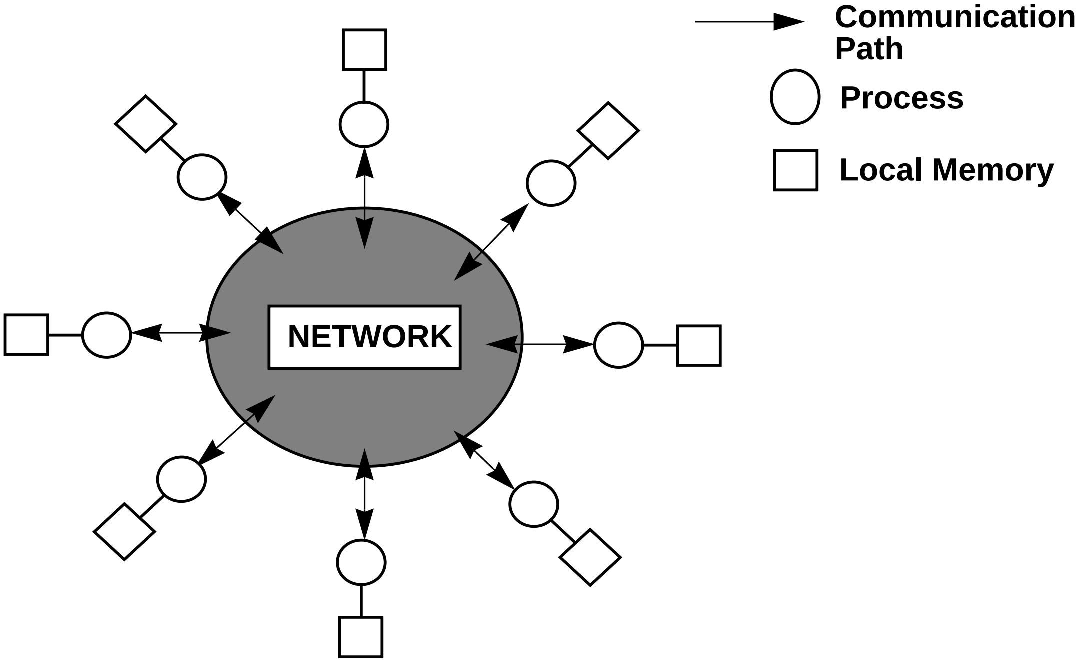
→ **Indicates direction of communication between processors**

Parallel Communications

Q: How do we communicate between processors?
A: Message Passing Interface (MPI)

What is MPI?

- ☞ Portable set of Library functions
- ☞ Largely hardware independent
- ☞ Provides distributed computing
- ☞ Local memory and a shared communication network

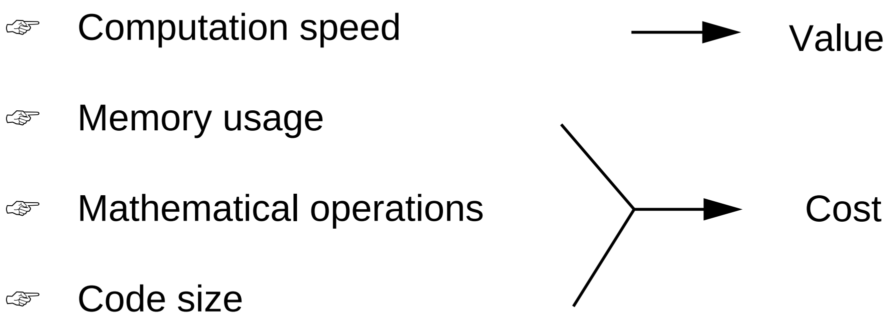


Why use MPI?

- ☞ Portability
- ☞ Allows collective communications
- ☞ Ease of use - do not need special knowledge of hardware

Evaluation Methodology

Evaluation Criteria



- ☞ Computation speed → Value
- ☞ Memory usage
- ☞ Mathematical operations → Cost
- ☞ Code size

A typical application will need the algorithm with the **fastest possible speed**, but which has the **lowest hardware requirements**.

We desire to determine the trade-off between speed and hardware for each algorithm.

Iterative Approach

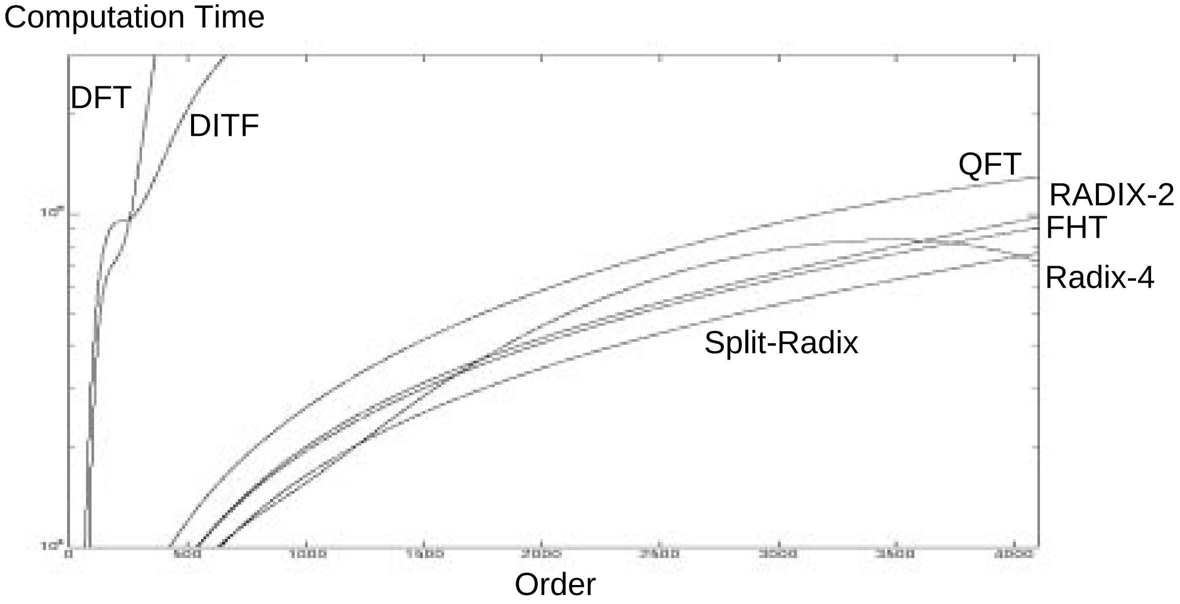
Statistics are taken as averages over a large number of iterations.

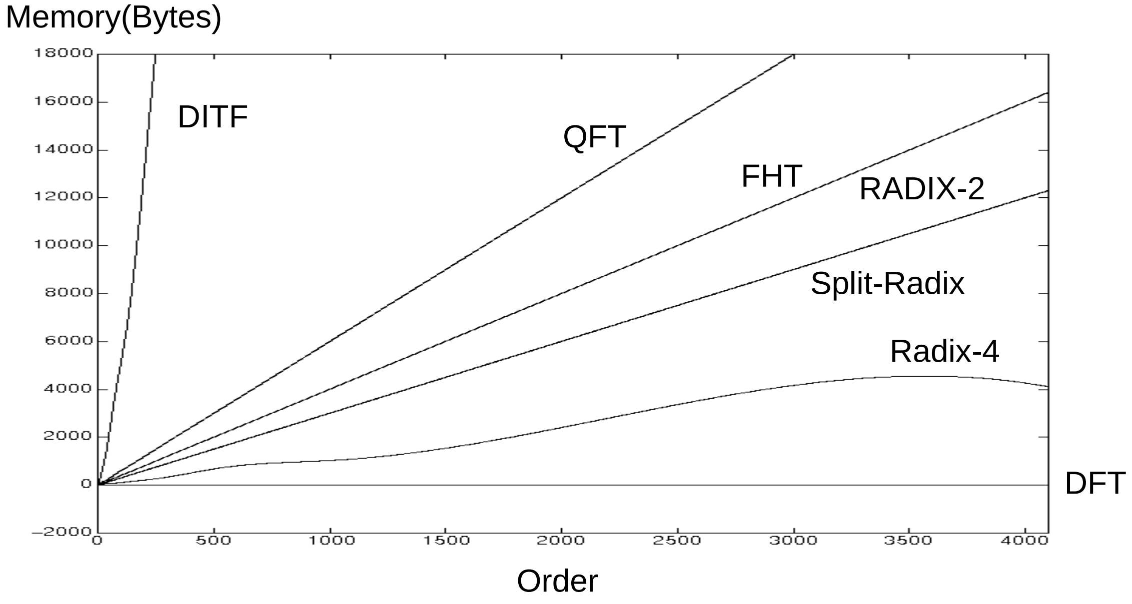
Why use an iterative method?

Eliminate transients caused by processor loading

Convergence of time statistics

Results -> Sequential



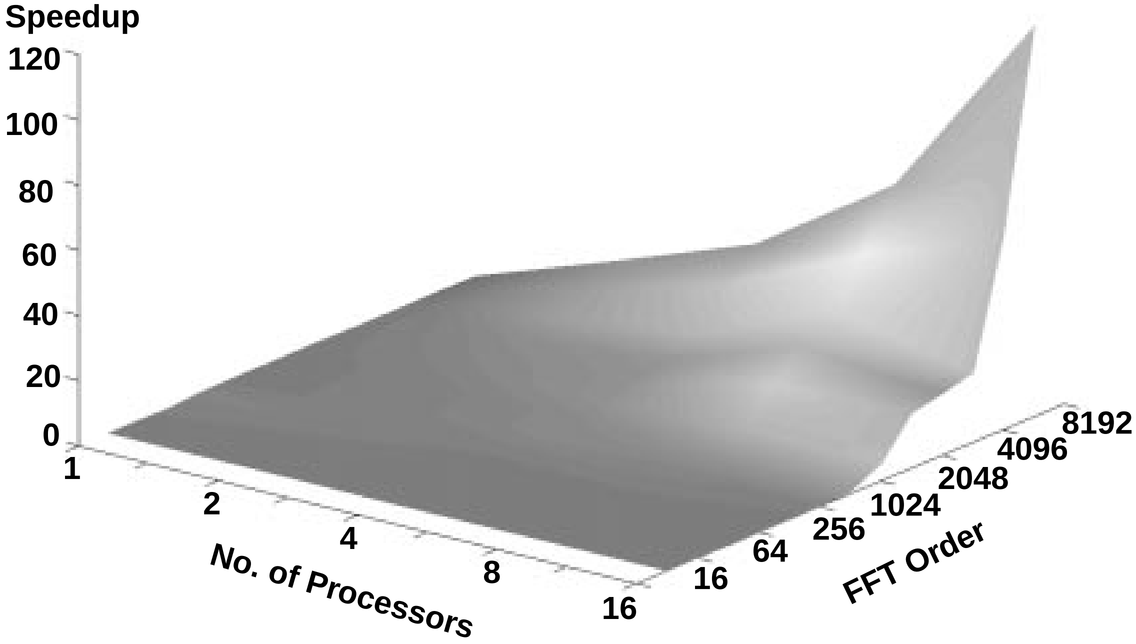


Mathematical Complexity (1024-points)

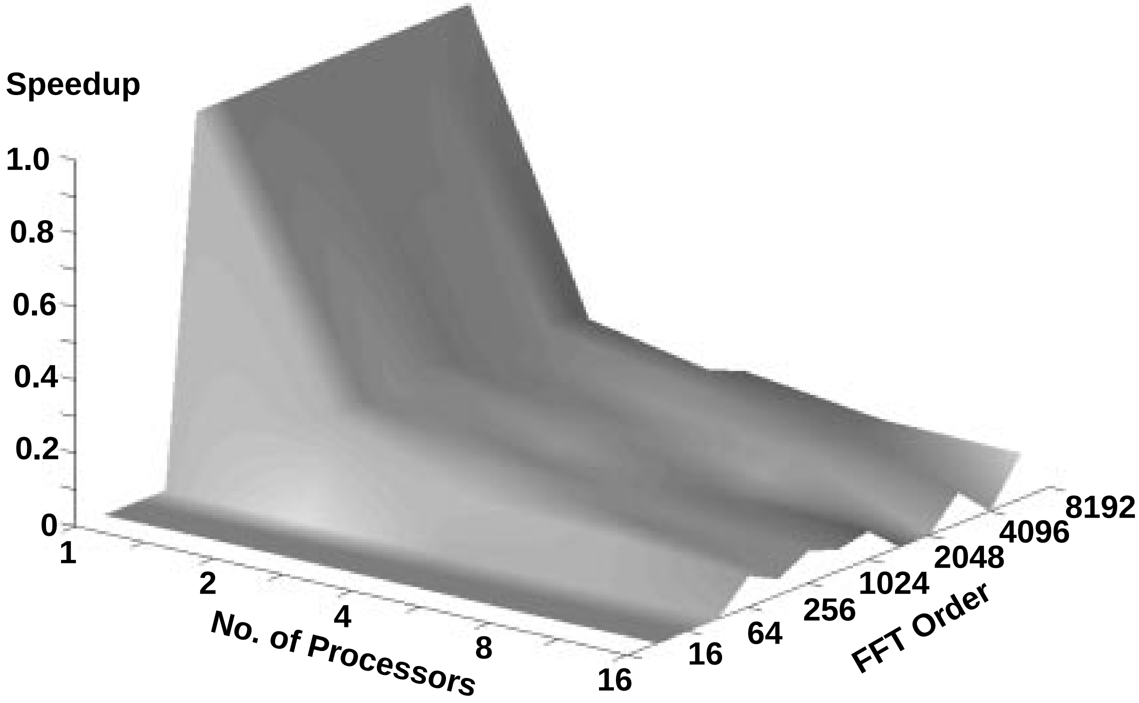
Rank	Algorithm	Total Operations
1	srfft	30764
2	fht	31176
3	qft	53628
4	rad4	56494
5	rad2	67581
6	ditf	112985
7	dft	5243904

Results -> Parallel

Best Case Results - Data Splitting

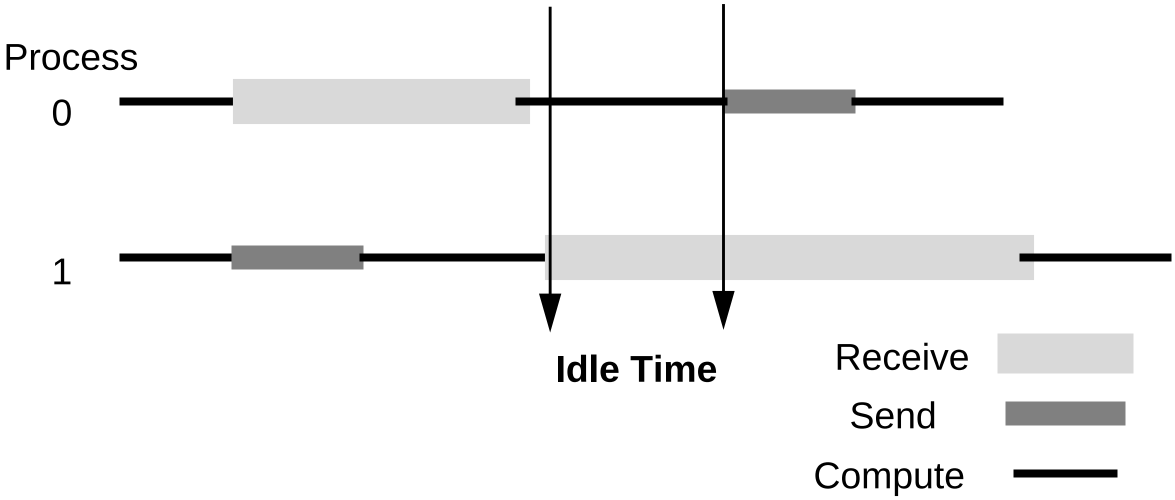


Typical Results



The Bottom Line

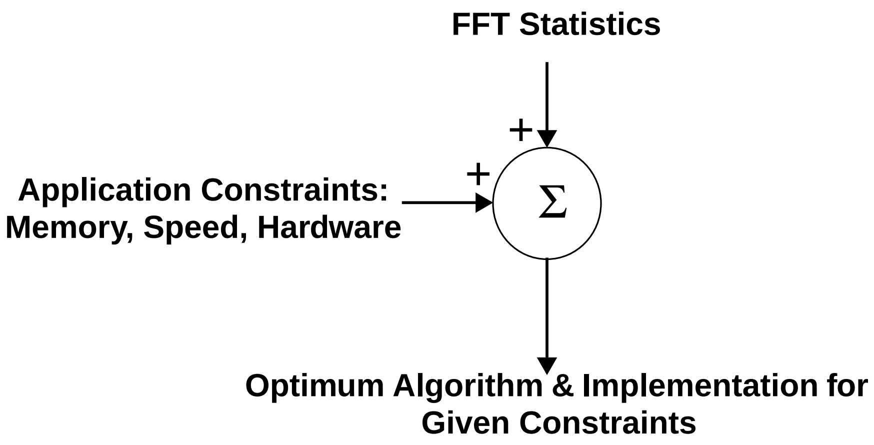
- **Increasing the number of processors did not always increase the speed of the algorithms.**
- **Sequential code usually gave the best case results.**
- **WHY?**
  - Not enough overlap in communication and computation



  - Processors were idle while waiting for communication.
  - Formed parallel code from tight sequential code.
- **How do we overcome these problems?**
  - Redistribute the loading
  - Rewrite routines to take better advantage of the symmetric algorithms

Future Implications...

- **What have we accomplished?**
  - First unified collection of parallel FFTs in the public domain.
  - Compiled complexity figures which include the overhead involved in coding. (loop counters, intermediate operation, temporary variables, etc.)
  - Have framework for application and constraint driven software.
- **The Next Steps...**
  - Rewrite code with a parallel structure as opposed to converting sequential code
  - Explore other parallel processing techniques (shared memory, threads, etc.)



FFT Statistics

Application Constraints: Memory, Speed, Hardware

Optimum Algorithm & Implementation for Given Constraints

References

1. Oppenheim, Alan V., Ronald W. Shafer. *Discrete-Time Signal Processing.* Prentice Hall. Englewood Cliffs, New Jersey 1989. pp 587-610.

2. P. Duhamel and H.Hollomann, “Split radix FFT algorithm,” *Electron. Lett.*, vol. 20, pp. 14-16, Jan. 1984.

3. R. Bracewell. *The Hartley Transform* Oxford, England: Oxford Press, 1985, chapter 4.

4. Patrick Bridges, Nathan Doss, William Gropp, Edward Karrels, Ewing Lusk, and Anthony Skjellum. “Users guide to MPICH, a Portable Implementation of MPI”, 1994. *MSU Argonne Joint Documentation.*

5. H. Guo, G.A. Sitton, C.S. Burrus. “The Quick Discrete Fourier Transform.” *ICASSP94 Digital Signal Processing.* vol III. Institute for Electrical and Electronics Engineers. pp. 445-447, 1994.

6. Saidi, Ali. “Decimation-In-Time-Frequency FFT Algorithm.” *ICASSP94* *Digital Signal Processing.* vol III. Institute for Electrical and Electronics Engineers.

7. Martin Vettereli and Pierre Duhamel, “Split-Radix Algorithms for Length-p^m DFT’s,” *IEEE Trans. on ASSP*, vol. 37, no. 1, January 1989. pp 57-64.

8. Brigham, E. Oran. *The Fast Fourier Transform and Its Applications.* Prentice Hall. Englewood Cliffs, New Jersey 1988. pp 131-156.

9. Peled, Abraham, Bede Liu. *Digital Signal Processing, Theory, Design, and Implementation.* John Wiley & Sons. New York, 1976. pp 142-171.

Audible Frequency Detection and Classification

by

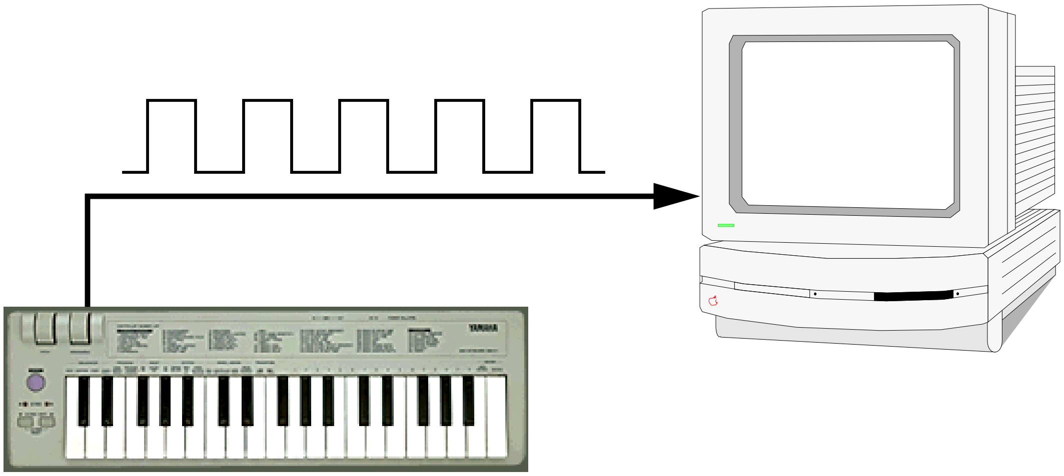
David E. Gray gray@ece.msstate.edu
W. Craig McKnight wcm1@ece.msstate.edu
Stehpen R. Wood srw1@ece.msstate.edu

Audible Frequency Detection and Classification Group
Department of Electrical and Computer Engineering
Mississippi State University
216 Simrall, Hardy Rd.
Mississippi State, Mississippi 39762

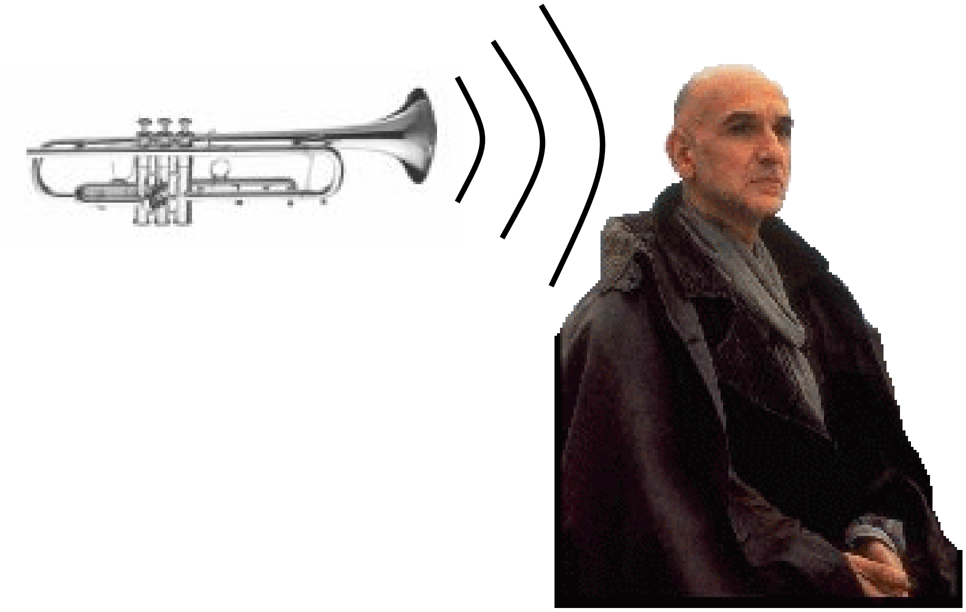
ABSTRACT

Current music software relies on external input from MIDI capable devices. Because traditional musical instruments are inherently analog, the interaction of musicians and computers is rare. The purpose of this project is to develop a software package for music education utilizing an acoustical instrument interface so that players of all instruments can begin to utilize the computing power of today's world. Musicians who play tones into a microphone will see those tones analyzed in the areas of relative and absolute pitch.

A Computer's Musical Experience: MIDI

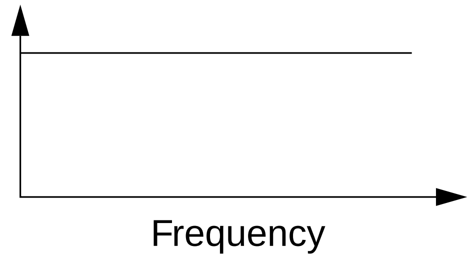


A Human's Musical Experience: Analog

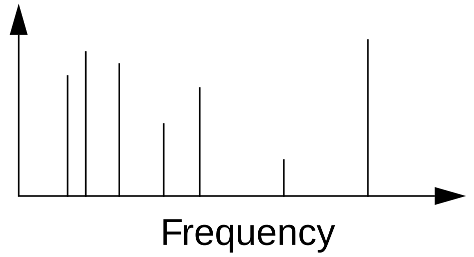


Characteristics of a Musical Signal

Not Music

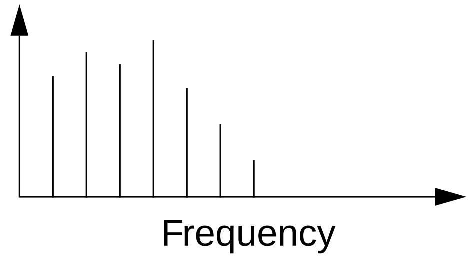


- Signal Energy is Evenly Distributed Across All Ranges of the Frequency Spectrum.



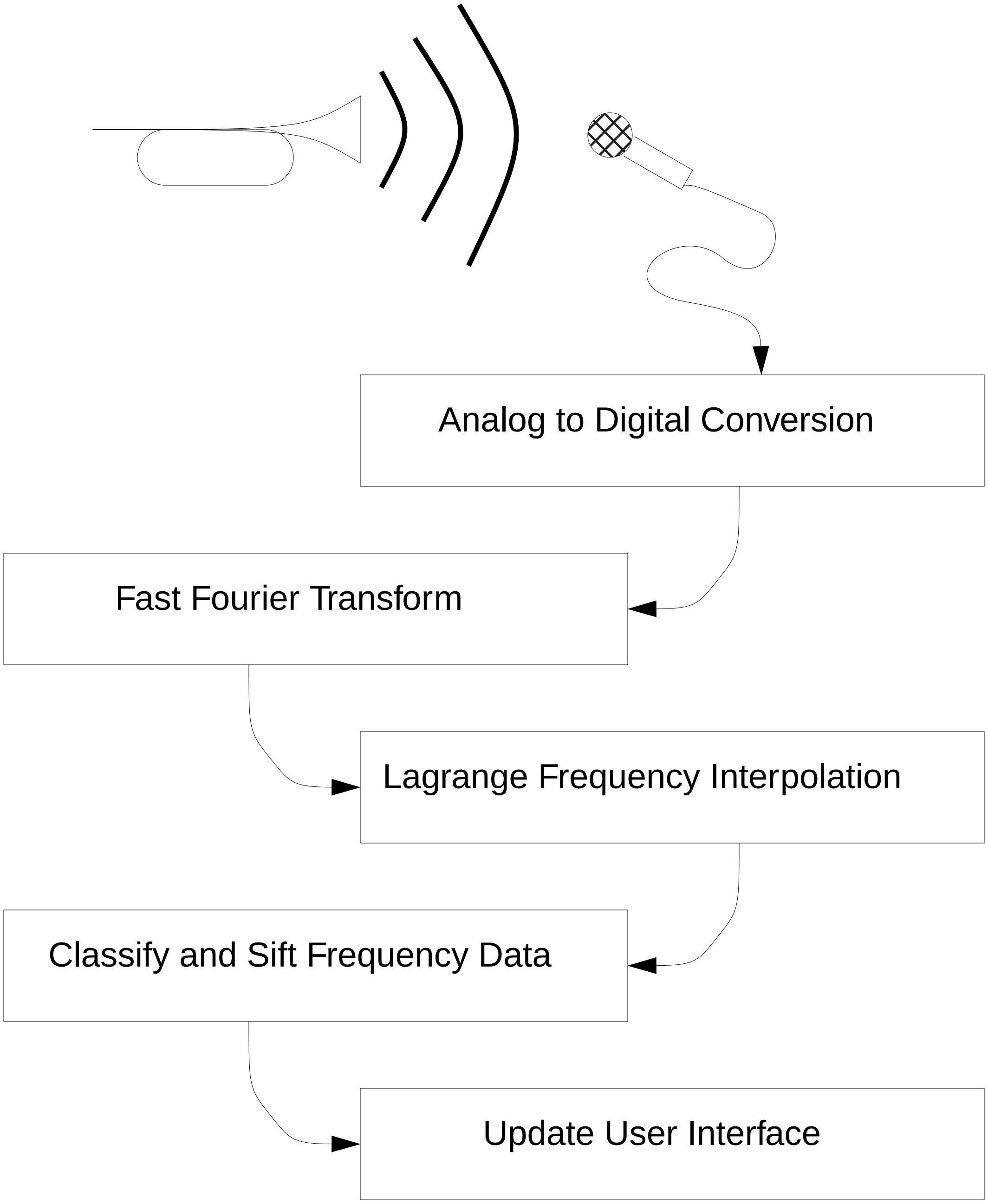
- Signal Energy Is Concentrated at Random Frequencies of the Spectrum.

Music



- Signal Energy Is Concentrated in Regular Discreet Portions of the Frequency Spectrum.
- Pattern of Frequency Spectrum Varies Among Different Instruments, But Signal Energy Is Always Concentrated at Integer Multiples of the Lowest (or Fundamental) Frequency, f_0, of a Note.

Digital Analysis of Musical Sounds



Analog to Digital Conversion

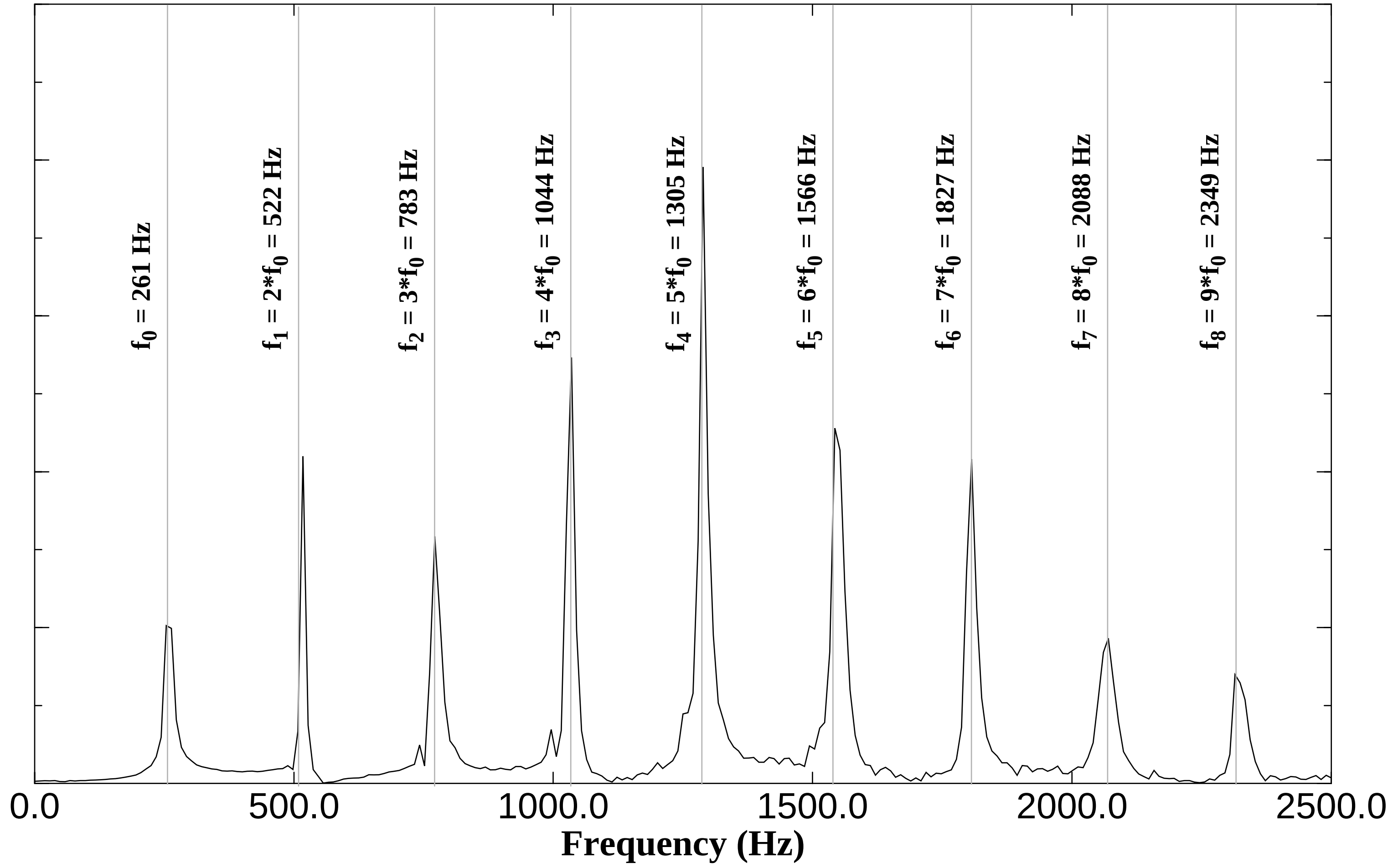
Fast Fourier Transform

Lagrange Frequency Interpolation

Classify and Sift Frequency Data

Update User Interface

C4 Played on Trumpet



The Even-Tempered Scale and Its Derivation

The frequency, f_{octave}, of a note, N_{octave}, played one octave above a note, $N_{fundamental}$, is twice as high as the frequency of the fundamental, $f_{fundamental}$. That is,

$$f_{octave} = 2 \bullet f_{fundamental} \tag{1}$$

Overtones occur at frequencies that are integer multiples of a note's fundamental frequency. Because of this, the same note name does not represent a unique frequency for all keys.

Even-Tempered Tuning is a system where each half-step can be approximated equally well in any scale without retuning the instrument. It is the standard system for tuning musical instruments in the Western world.

The Even-Tempered Scale is designed so that the frequency of each semitone is a factor K larger than the previous semitone. That is,

$$f_1 = K \bullet f_0 \tag{2}$$

There are 12 semi-tones in one octave. According to equation (1), the thirteenth note, the octave, must be twice the frequency of the first note. Dividing the octave into equal semi-tones, it can be shown that

$$K = \sqrt[12]{2} \approx 1.05946 \tag{3}$$

Restrictions of FFT: Frequency Resolution

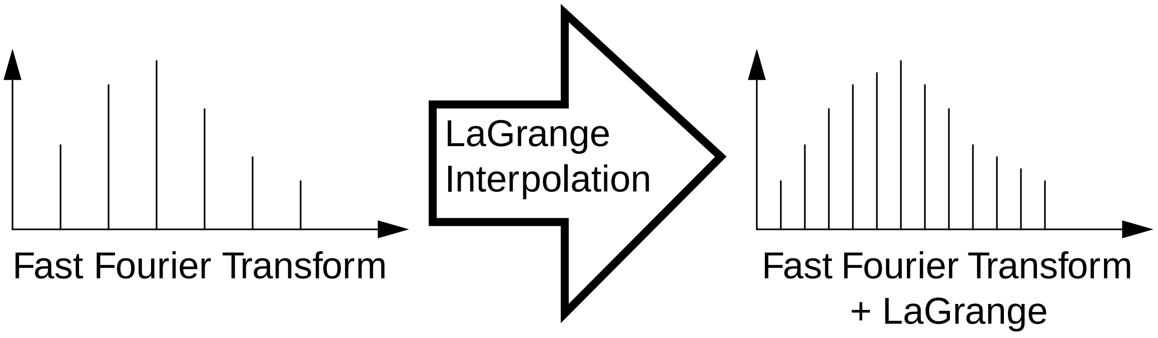
The frequency resolution, Δf, of an N point FFT with sample frequency, f_s, is defined as

$$\Delta f = \frac{f_s}{N} = \frac{10000}{1024} = 9.765625 Hz \tag{1}$$

Recalling the distribution of notes using Even-Tempered Tuning, we realize that for low scale tones, several semi-tones can occur in a 10 Hz window. This is an unacceptable resolution for low frequency pitches. We can improve the resolution of the FFT by interpolating between points.

Given m data points, the LaGrange Interpolation Technique calculates a unique polynomial of order (m-1) that passes through all m points.

Using a 5-point LaGrange Interpolation, the resolution of the data in the frequency domain can be roughly doubled.



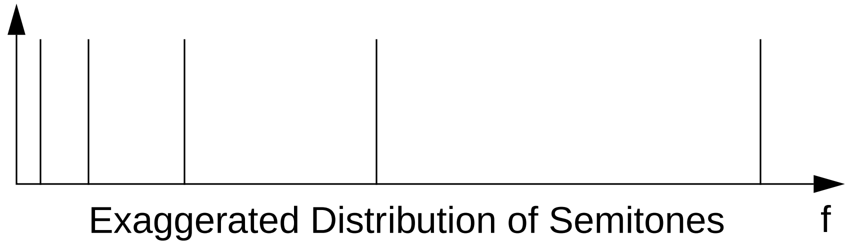
LaGrange Interpolation

Fast Fourier Transform

Fast Fourier Transform + LaGrange

Tuning a Note's Overtones to Increase Accuracy

Notes in Even-Tempered Tuning are not evenly spaced throughout the frequency domain. Each semitone is K times larger than the previous one. Therefore the number of Hertz between semitones increases as the pitch of the note increases.



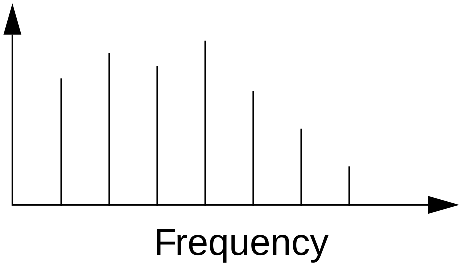
Exaggerated Distribution of Semitones

So the discrepancy between the actual frequency of a peak and the interpolated frequency of that peak becomes less significant as the note's frequency increases.

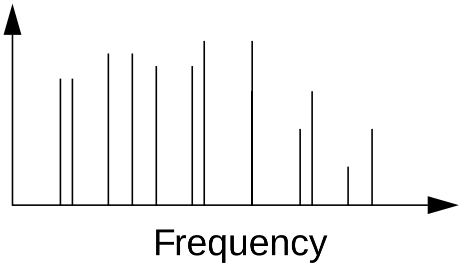
Because a note produces overtones at integer multiples of the fundamental frequency, a direct relationship exists between the frequency of the overtone and the frequency of the fundamental.

Interpolating a note's overtones using LaGrange, we can be within a constant error that becomes less significant for a higher overtone. We can also correct for the intrinsic error introduced by Even-Tempered Tuning.

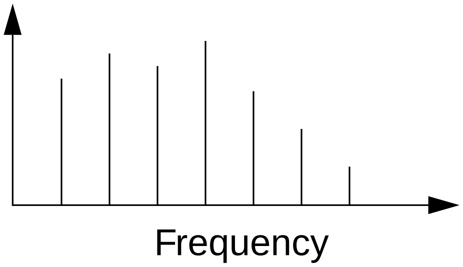
Project Database



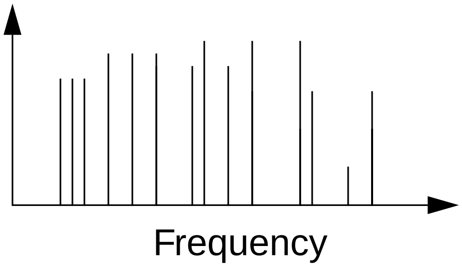
- **Phase I:** Single note Played for entire file. Each note played for three cases: flat, sharp, and in tune. Built for Tuning.



- **Phase II:** Two notes played together. Each interval within an octave recorded. Used for Interval Detection.



- **Phase III:** Three notes played sequentially in time. Used for Tuning and detecting note changes.



- **Phase IV:** Three notes played at once in a triad. Three different qualities of triads for each instrument.

Results

- Running our software, musicians can play tones into a microphone and see those tones analyzed in the areas of relative and absolute pitch.

- **Using our software to evaluate the Project Database:**

  - The software correctly identifies the fundamental in all files of Phase I.

  - The software tunes the Phase I Database to a great deal of accuracy compared to a commercial tuner.

  - The interaction of multiple notes makes classification of fundamentals more difficult than in Phase I. However, the software classifies the fundamentals properly in most cases.

  - The fundamental was correctly identified in most of Phase III.

- **Using our software to evaluate real-time data:**

  - The software performs in real time using Network Audio as an Analog to Digital Converter.

  - The software performs in real time as well as it does on the Project Database except for data that is very loud or that contains many notes.

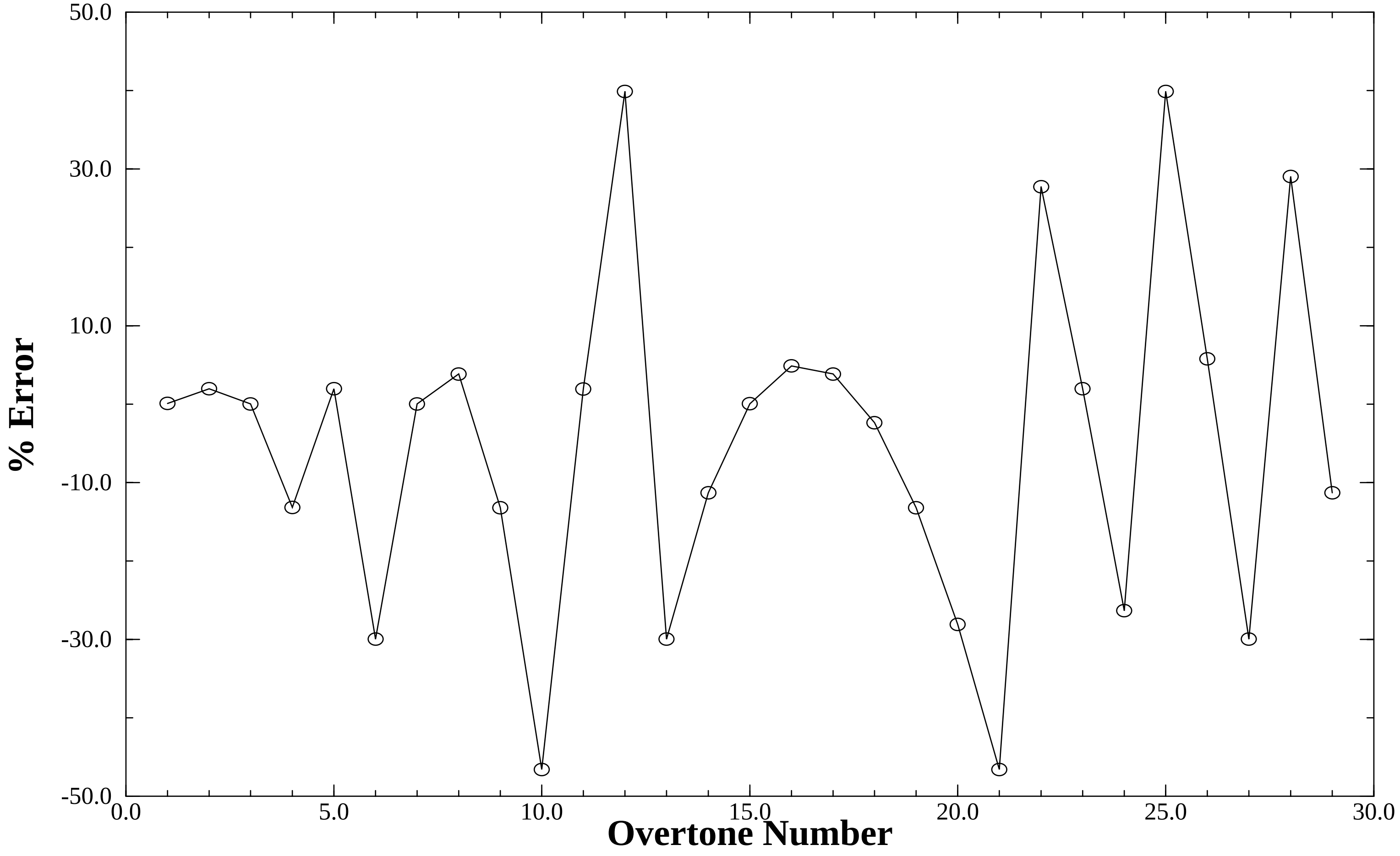
Summary and Areas For Future Research

- We successfully built a tone analyzer that works in real time with Network Audio as an Analog to Digital Converter.
- This research could be used as a basis for music notation or music education software designed to be used with instruments that are not MIDI capable.
- The software analyzes the musical data in the areas of relative and absolute pitch.
- Adjustment of the noise floor filter could improve the performance of the software.
- Adjustment of the “tolerance” for classifying overtones could improve the performance of the software. This could eliminate some unwanted high-frequency spikes by classifying them as overtones instead of as notes.

REFERENCES

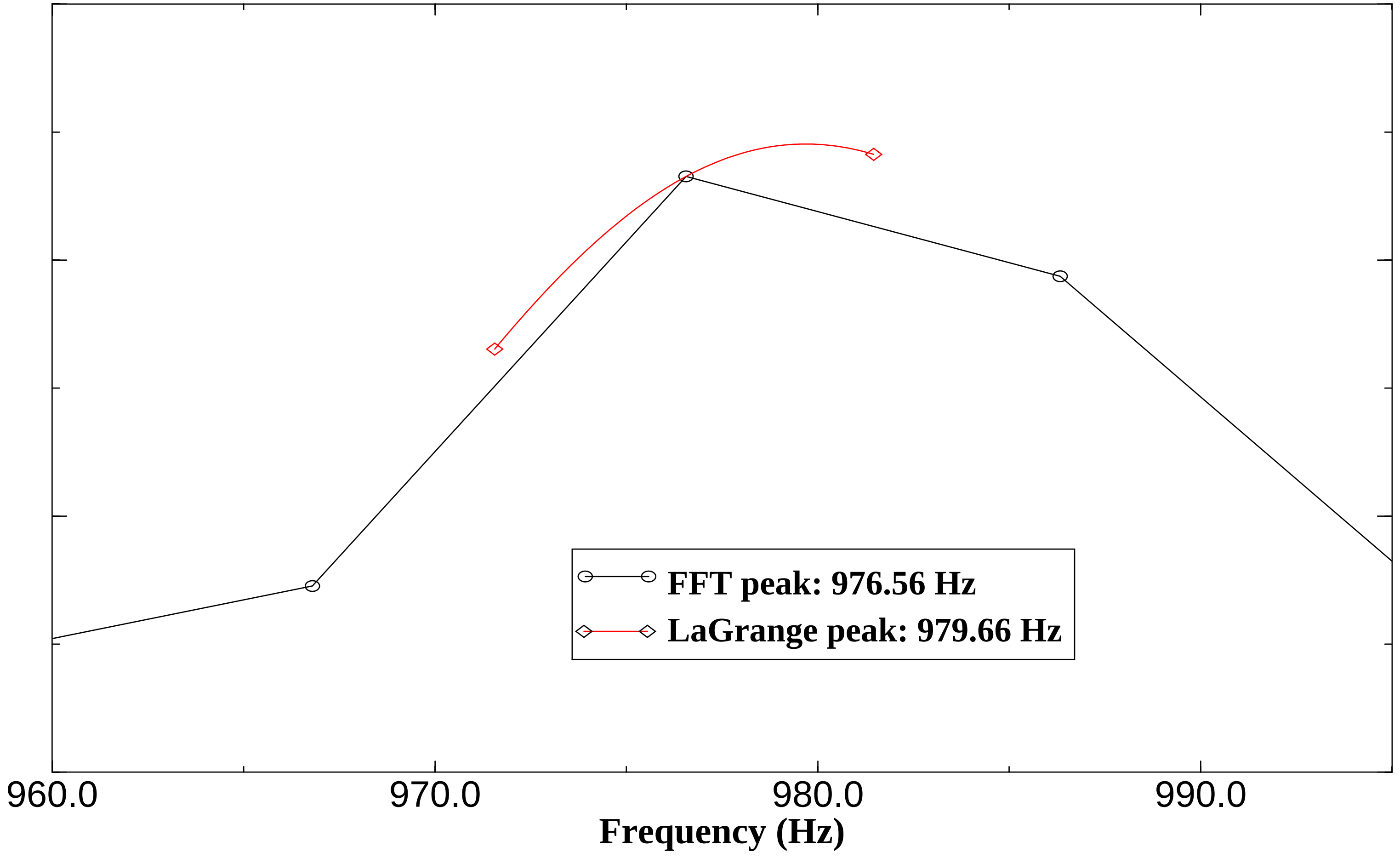
1. B. Vercoe, *MIT Media Lab Csound Manual*. http://www.leeds.ac.uk/music/Man/Csound/pref.
2. *Listen's Sound and MIDI Features*. http://www.imaja.com/Listen.html.
3. H. Chamberlin, *Musical Applications of Microprocessors*. Hayden Books., Hasbrouck Heights, NJ, 1985.
4. W. W. Smith and J. M. Smith, *Handbook of Real Time Fast Fourier Transforms*. Institute of Electrical and Electronic Engineers, Inc., New York, NY, 1995.
5. R. E. Ziemer, W. H. Tranter, and D. R. Fannin, *Signals and Systems: Continuous and Discrete, 3rd Edition*. MacMillan, New York, NY, 1993.
6. M. B. Priestley, *Spectral Analysis and Time Series, Vol. 1*. Academic Press, London, 1981.
7. S. L. Marple, Jr., *Digital Spectral Analysis With Applications*, Prentice Hall, Englewood Cliffs, NJ, 1987.
8. A. H. Benade, *Horns, Strings, and Harmony*. Doubleday, Garden City, NY, 1960.

Overtone Error Using Even-Tempered Scale



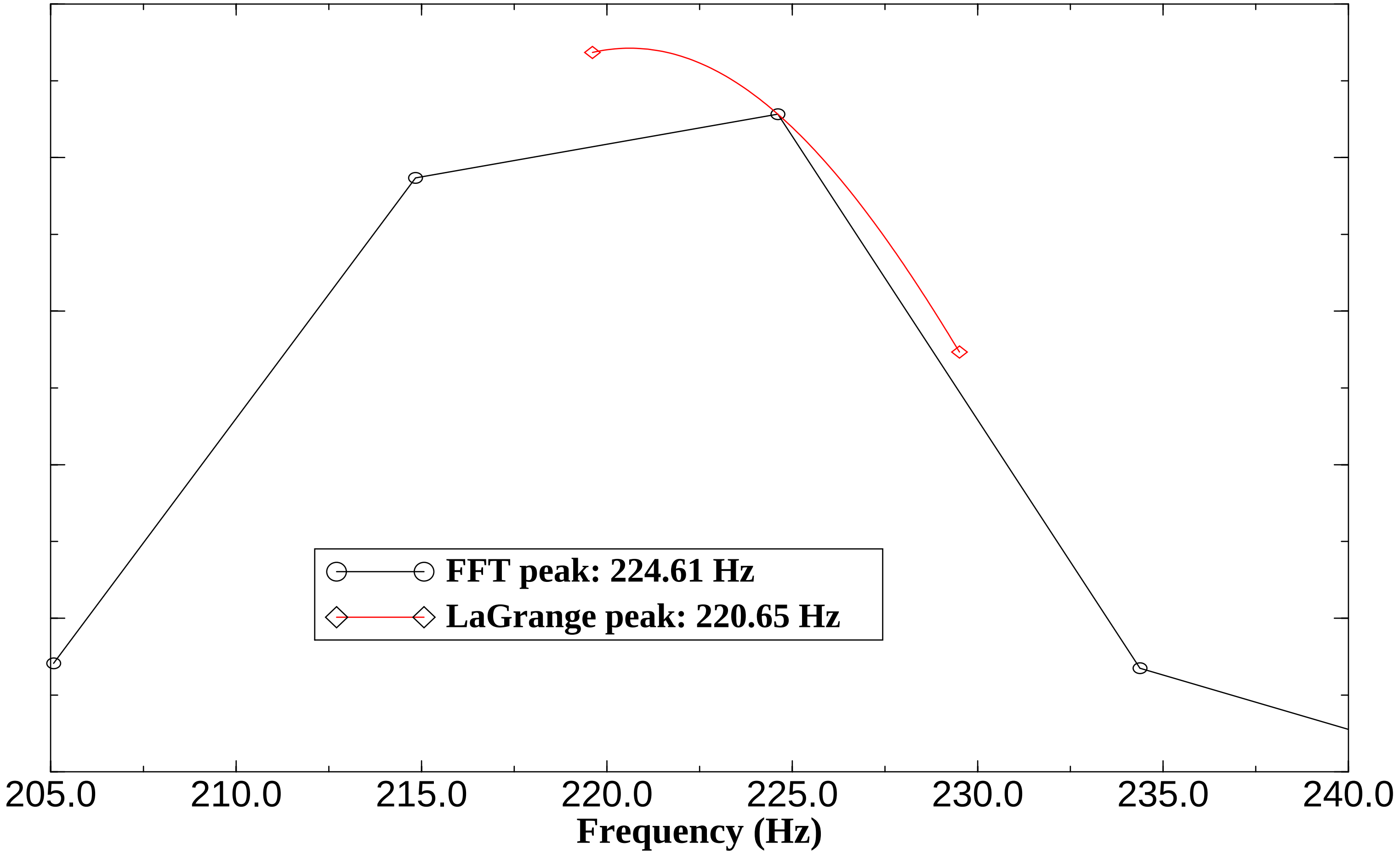
981 Hz Peak

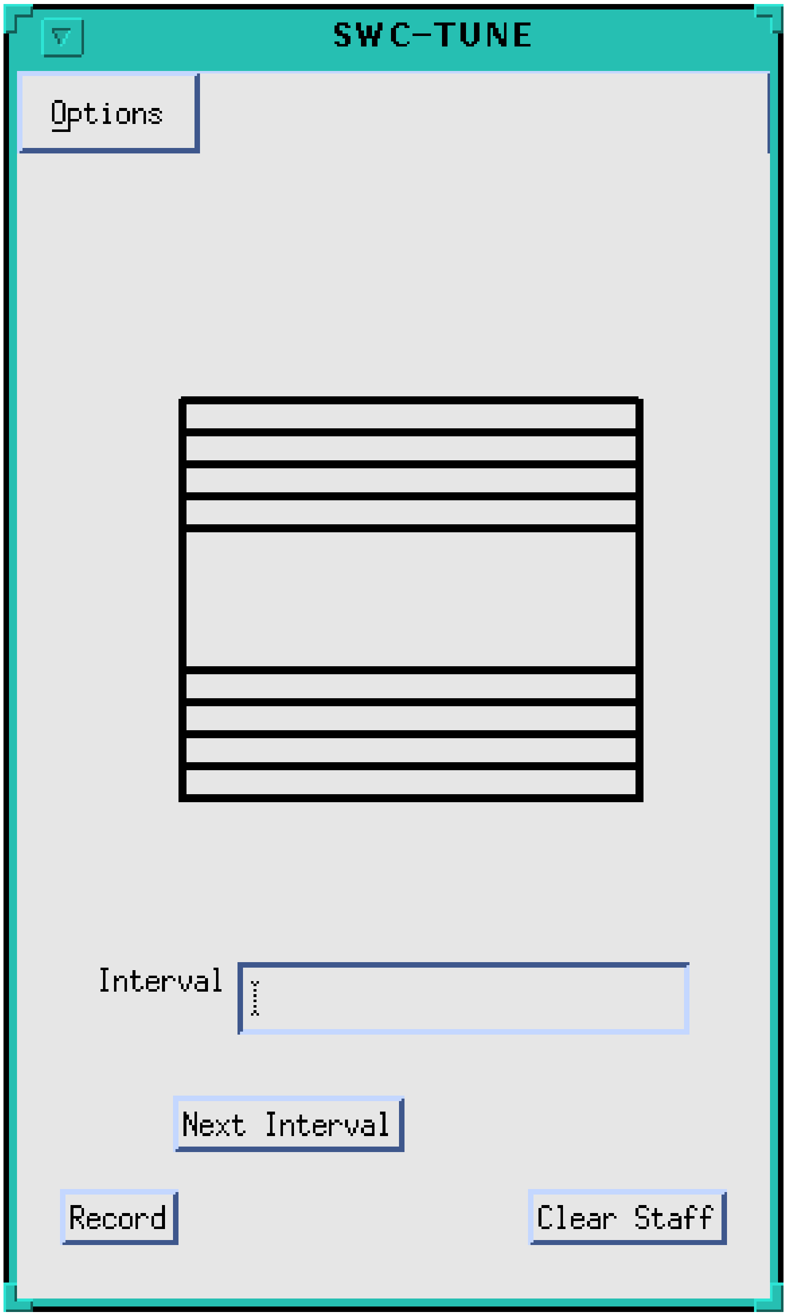
Increased Accuracy Using LaGrange Interpolation

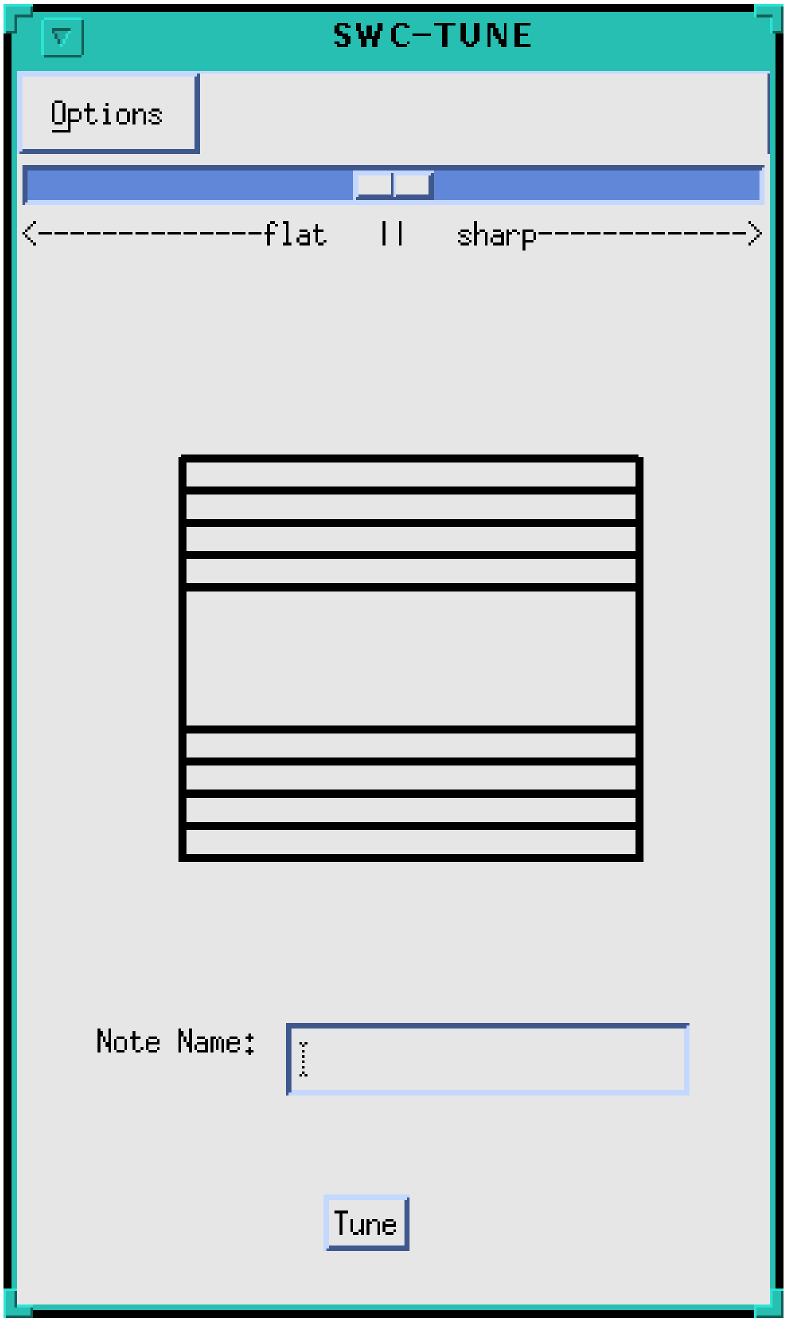


A3 Played on Bass Guitar

Increased Accuracy Using LaGrange Interpolation




SWC-TUNE
Options
Interval
Next Interval
Record
Clear Staff


SWC-TUNE
Options
<-------------flat || sharp------------>
Note Name:
Tune

An Algorithm to Determine the Scenic Quality of Images

by

Nirmala Kalidindi kaldindi@isip.msstate.edu
Yaqin Hong yh4@ra.msstate.edu
Zheng Liang zl3@ra.msstate.edu

Image Processing Group
Mississippi State University

ABSTRACT

The United States Forest Service wishes to determine the scenic quality of images to maintain the beauty of the forest inspite of cutting and also for recreation purpose. This project determines the scenic quality of image on a scale from “0” to “1”. We have the database consisting of 680 unique images given by the forestry department. The database has four pictures of the same image taken during all the seasons of the year. This will also help to study the effect of seasons on the scenic quality of the image.

There are subjective scenic beauty ratings available for each of the images in the database. Some of the parameters which effect the scenic quality of the image are the color, number of vertical lines, texture of the image and entropy. For this project we are dealing with only color and number of vertical lines. Histogram is developed to compute the mean of each of the color in the image and edge detection is done to calculate the number of vertical lines.

Overview

❒ Scenic Beauty Estimation

❒ Decision making in forest planning

☞ To preserve recreation

☞ To preserve aesthetic resources

What to implement

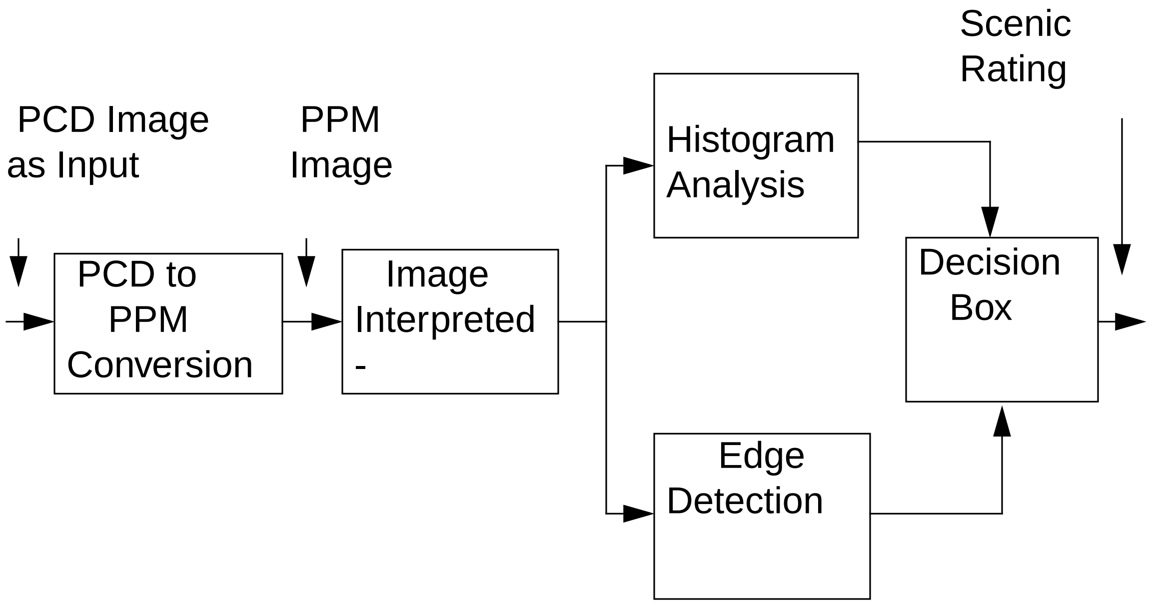
❐ Parameters to be determined in scenic beauty determination

☞ Color: Histogram

☞ Vertical Lines:Edge Detection

☞ To compare the derived scenic rating with the actual value

Block Diagram of the System



Scenic Rating

PCD Image as Input

PPM Image

PCD to PPM Conversion

Image Interpreted -

Histogram Analysis

Edge Detection

Decision Box

Explanation of the Algorithm

- Calculating the mean of each of the color
- Converting the color image to gray-scale image
- Performing the edge detection
- Calculating the number of vertical lines
- Evaluating the scenic beauty

Mathematical Equations

☞ Color to Gray Conversion

Y = 0.299 R + 0.587 G + 0.114 B

☞ Mask used in Edge Detection

$$G_x = (I_7 + 2I_8 + I_9) - (I_1 + 2I_2 + I_3)$$

$$G_y = (I_3 + 2I_6 + I_9) - (I_1 + 2I_4 + I_7)$$

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

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

❐ Evaluation

☞ Computing the scenic beauty based on mean and vertical lines

☞ Evaluating the program on the existing database

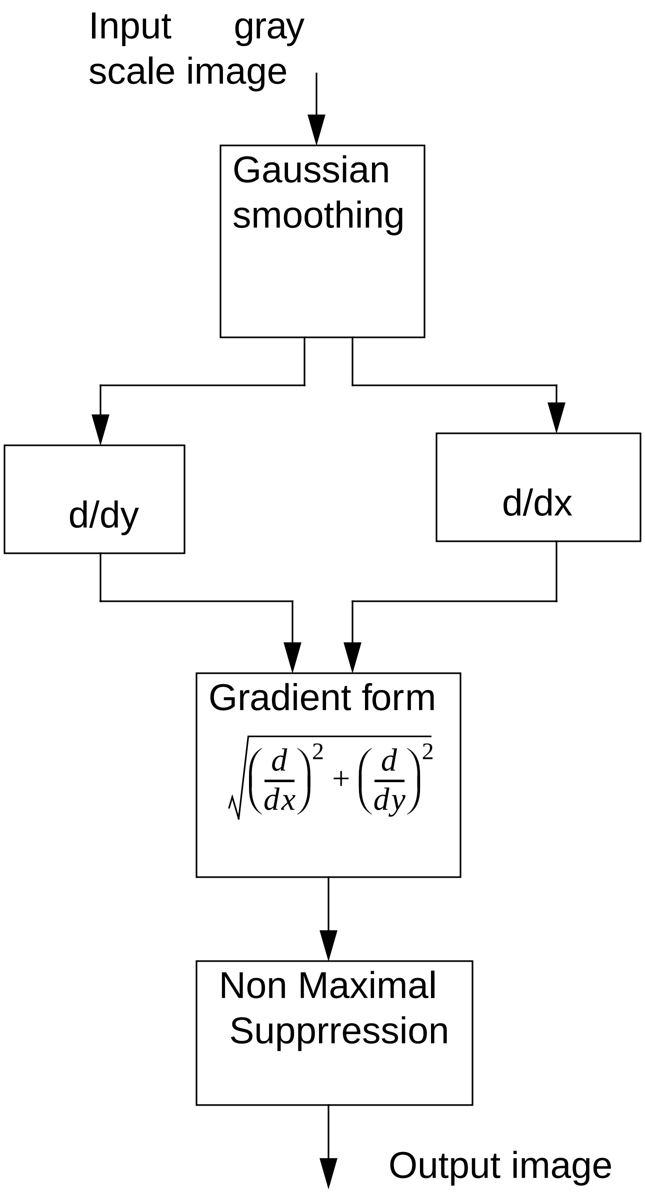
☞ Comparing the derived scenic beauty and the actual beauty

❐ Database

☞679 Unique Images with subjective ratings

☞ Standardized ratings available for each image

Edge detection Algorithm



Input gray scale image

Gaussian smoothing

d/dy

d/dx

Gradient form

$$\sqrt{\left(\frac{d}{dx}\right)^2 + \left(\frac{d}{dy}\right)^2}$$

Non Maximal Supprression

Output image

Results

Original Image

Gray Image

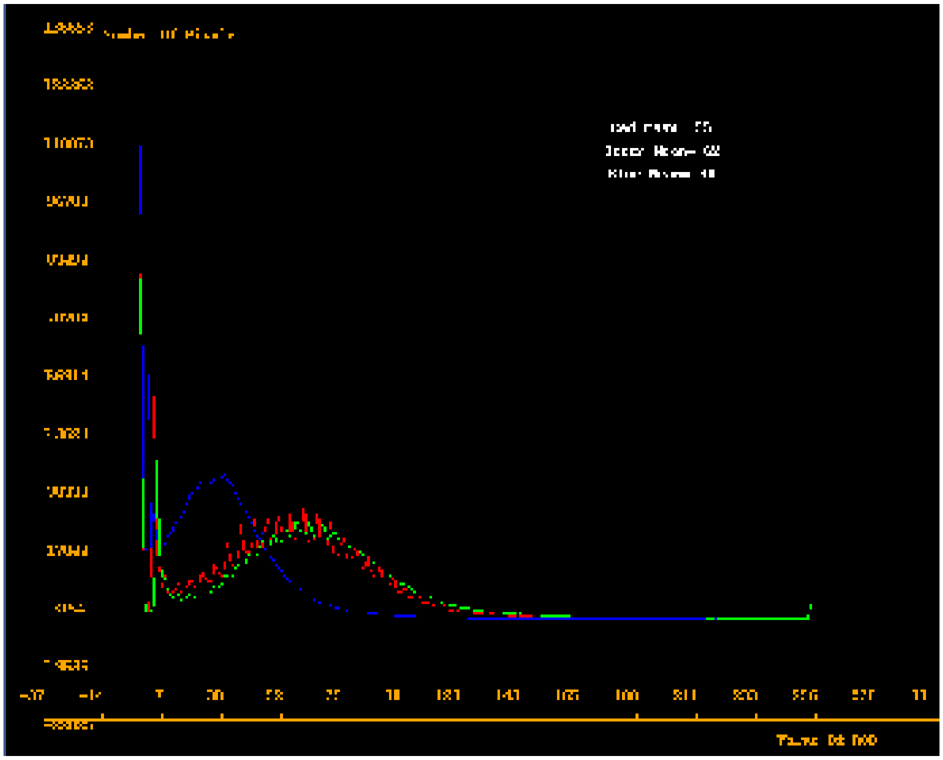
Results

- SBE depends on the mean of each of the color, number of vertical lines
- The length of the lines is a direct indication of the long trees and bushes

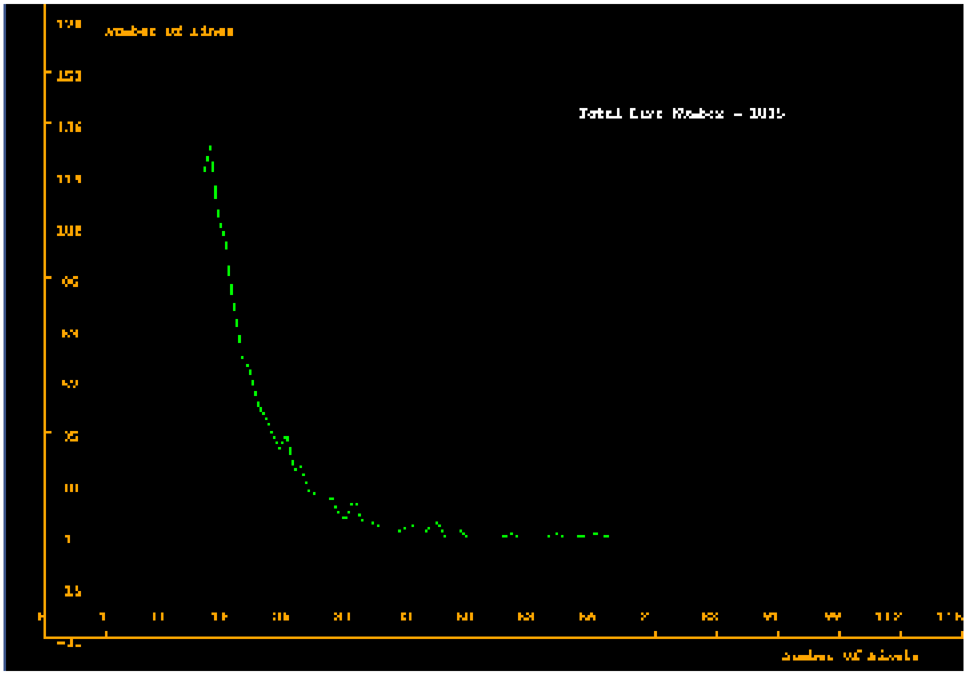
Table 1: table showing the derived means and the SBE rating

Redmean	Greenmean	Bluemean	%longlin	%shortli	der_sbe	act_sbe
66	72	36	1.97	78.00	0.40	4.37
46	62	36	2.10	76.10	0.52	4.80
64	78	45	2.20	73.60	0.50	5.16
55	62	30	2.02	77.10	0.42	4.68
46	62	36	2.27	75.60	0.47	4.80

Graph showing the number and value of each of the colors



Graph showing the number and size of the vertical lines



Summary

- Dependency of color on the scenic beauty
- Dependency of vertical lines on the scenic beauty
- Well organized database

Future Research

- Using neural network to evaluate the scenic beauty rating
- Develop algorithm to check the effect of texture, landform position on the scenic beauty
- To study the future growth of the trees with the present data

1. REFERENCES

[1] T. A. Herrick, V. A. Rudis, “Visitor Preference for Forest Scenery in the Ouachita National Forest”, *Proceedings of the symposium on Ecosystem Management Research in the Ouachita Mountains: Pretreatment Conditions and Preliminary Findings*: 1993 October 26-27, pp. 212-222

[2] J. H. Gramann, V. A. Rudis, “Effects of Hardwood Retention, Season of the Year, and Landform on the perceived Scenic Beauty of Forest Plots in the Ouachita Mountains”, *Proceedings of the Symposium on Ecosystem Management Research in the Ouachita Mountains: Pretreatment Conditions and Preliminary Findings*: 1993 October 26-27; pp. 223-228

[3] V. A. Rudis, J. H. Gramann, and T. A. Herrick, “Esthetics Evaluation”, *Proceedings of the Symposium on Ecosystem Management Research in the Ouachita Mountains: Pretreatment Conditions and Preliminary Findings*; 1993 October 26-27; pp 202-211

[4] J. H. Gramann, W. Yhang, “The Effect of Forest Color on the Perceived Scenic Beauty of Recently Treated Pine-Oak Plots on the Ouachita National Forest, Arkansas”, Final Report Prepared for Forest Inventory and Analysis unit, USDA Forest service.

[5] V. A. Rudis, “Sampling and Modeling Visual Component Dynamics of forested Areas”, State of the Art Methodology of Forest Inventory; A Symposium Proceedings; 1989 July 30 - August 5; pp. 84-85

[6] T. C. Brown, T. C. Daniel, “Scaling of Ratings; Concepts and Methods”, Research Paper RM-293, pp. 6-18.

[7] W.K.Pratt, *Digital Image Processing*, Wiley interscience publication,Newyork,1978

[8] K. R. Castleman, *Digital Image Processing*, Prentice Hall, Inc. Englewood Cliff, NJ, USA, 1979

[9] Rafael C. Gonzalez, Paul Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, California, USA, 1987