SIPL SIGNAL & IMAGE PROCESSING LAB

Guitar Tuner for Symbian OS

Students:

Reut Ochayon
Gilad Cohen

Supervisors:

Yevgeni Litvin
Yoav Shechtman

01.07.2009

Spring semester 2008

Outline

- Motivation
- Project goal
- Requirements
- General architecture
- HPS Pitch detection algorithm
- Live demonstration

Motivation

- Why tune a guitar?
- What existed before?

- Why do we need something better?

Motivation

- Modern cellphones are strong enough to run a tuner program

- This implementation has major advantages over the rest
  - No need to carry a special device
  - Easily updated

Project goal

- To create a “Guitar Tuner” application for symbian OS
  - Sybmian is an OS designed for mobile devices, it is very commonly used today

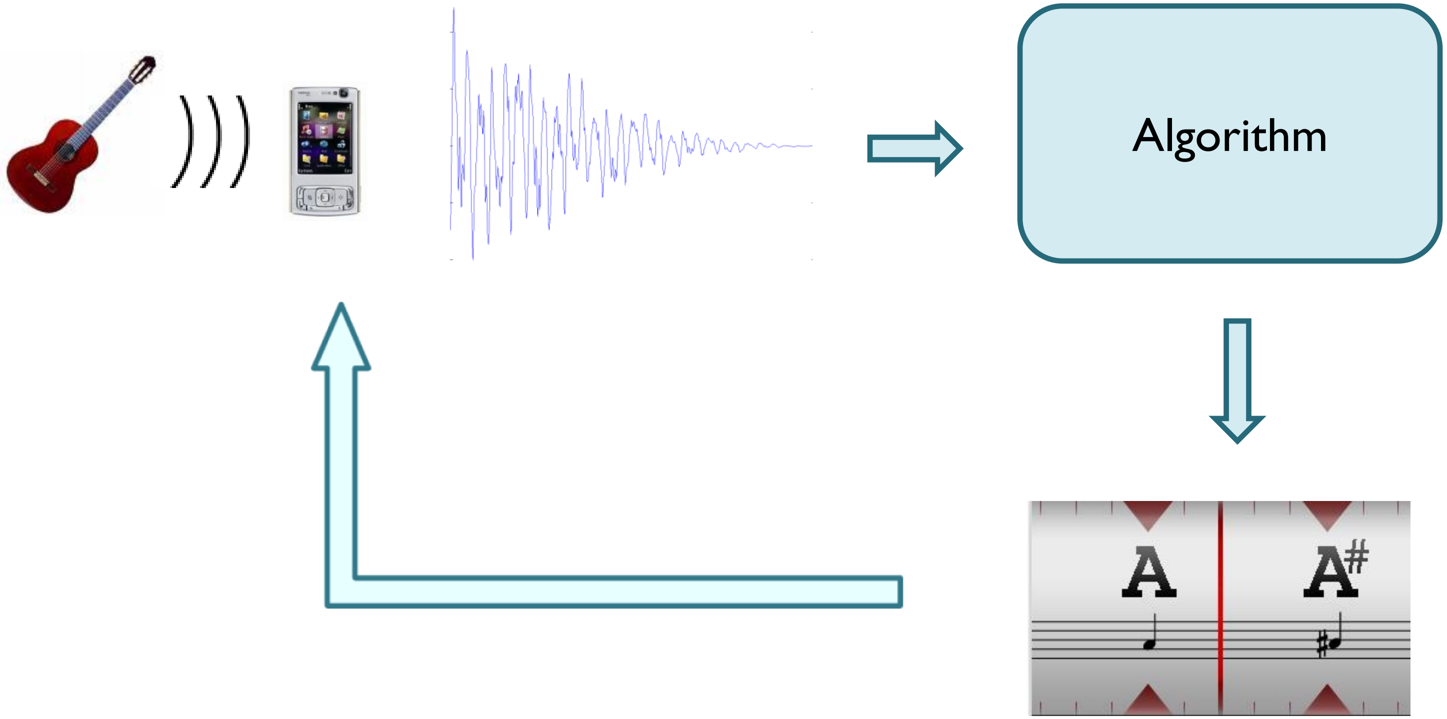
Requirements

- 8-48KHz sampling rates, we will work with 8KHz
- The desired pitch range is [C2,~E5] – frequency range is [65,700] Hz

Requirements

- Minimal frequency resolution : 25 cents =~ 1Hz around the low frequencies
- RealTime processing
- Fixed point arithmetic

General architecture

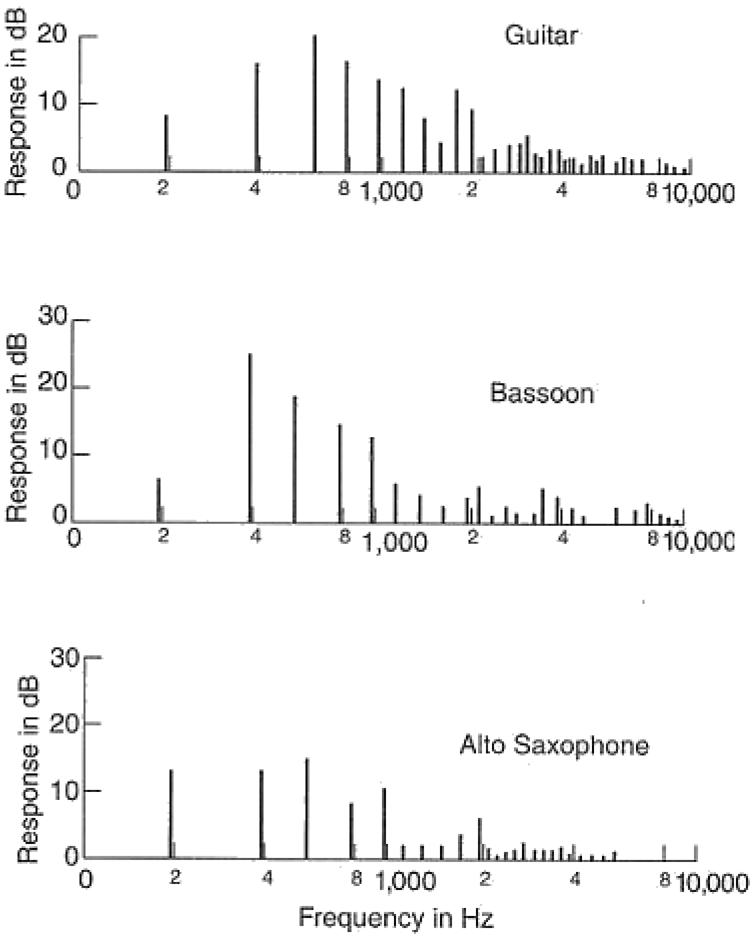


Algorithm

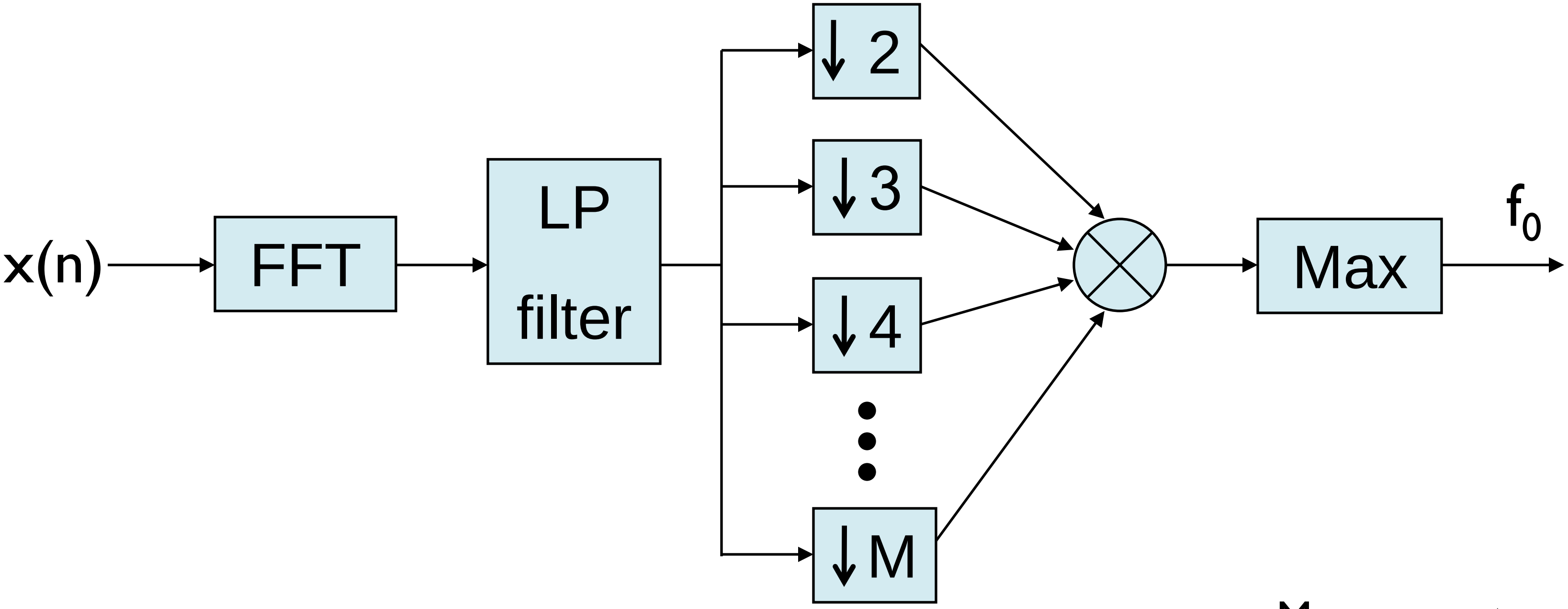
Harmonic Signal Structure

- Fundamental frequency (f_0) and its products

- Different energies



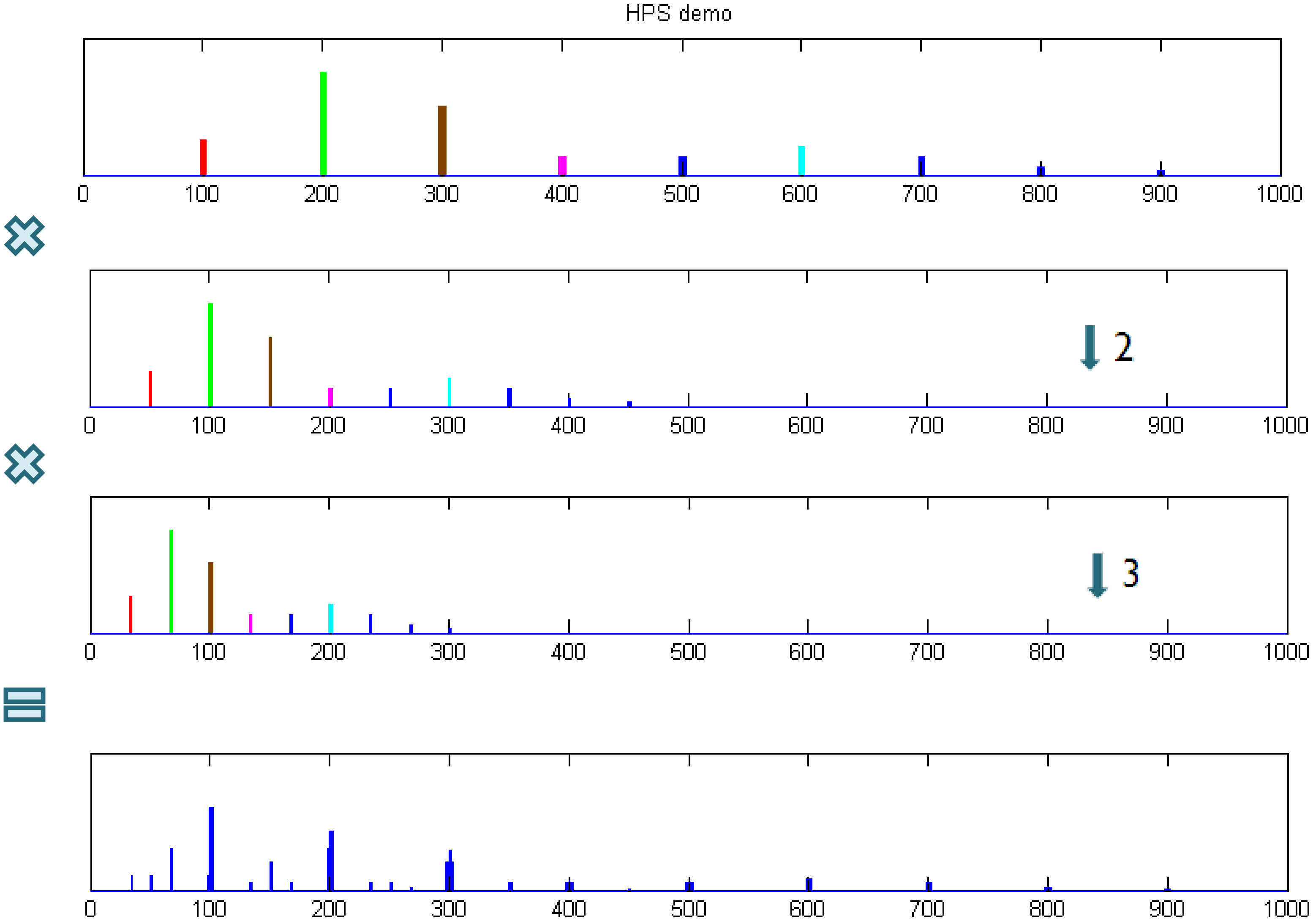
Harmonic Product Spectrum



M - parameter

* Patricio de la Cuadra, Aaron Master, Craig Sapp. Efficient Pitch Detection Techniques for Interactive Music. [Center for Computer Research in Music and Acoustics, Stanford University]

Harmonic Product Spectrum



Implementation Problems & Solutions

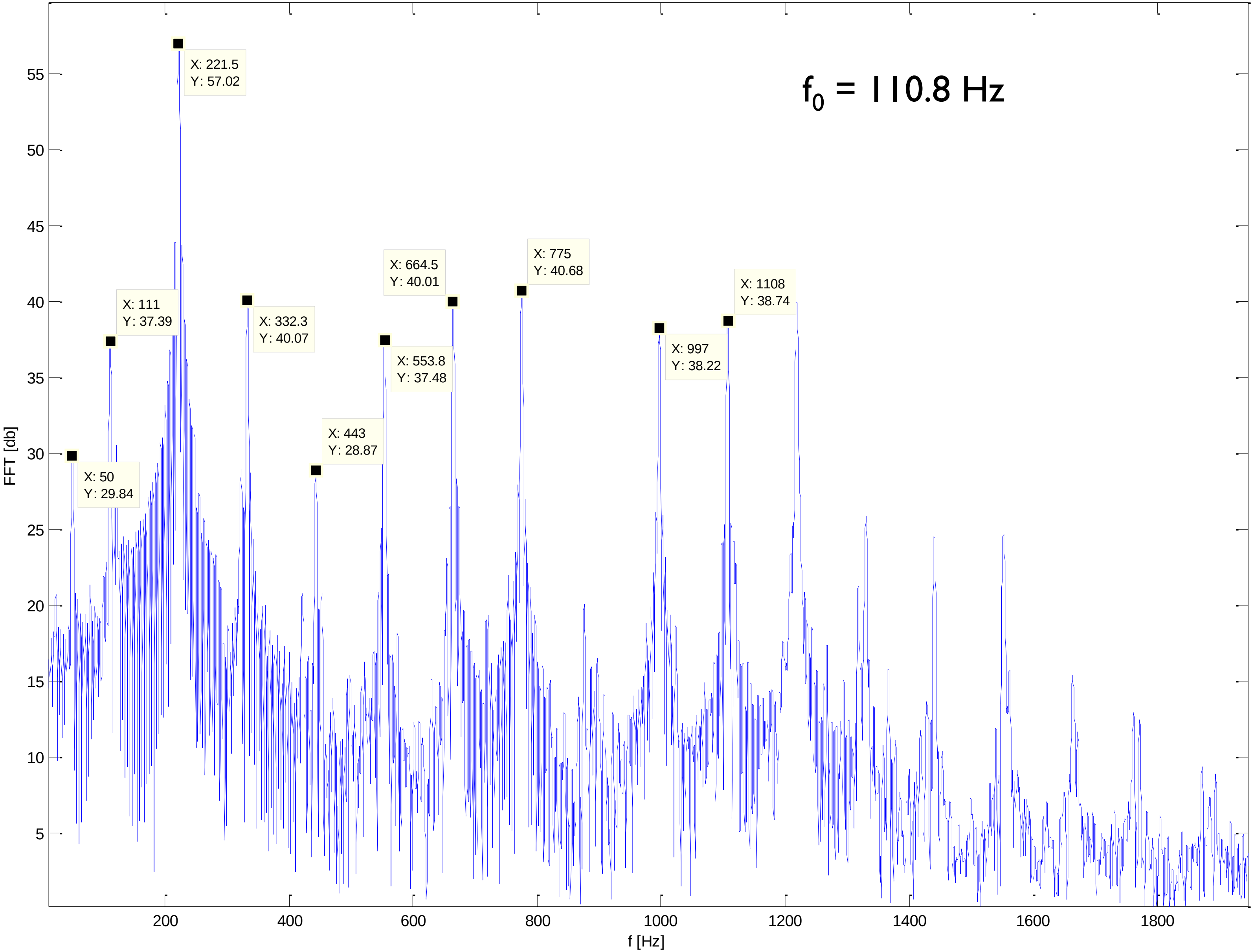
- Example : FFT algorithm makes redundant calculations

- Solution : using FFT optimized for real signals

Implementation Problems & Solutions

- Example : low frequency resolution

- Solution : use values of higher harmonies


X: 221.5
Y: 57.02
X: 111
Y: 37.39
X: 50
Y: 29.84
X: 332.3
Y: 40.07
X: 443
Y: 28.87
X: 553.8
Y: 37.48
X: 664.5
Y: 40.01
X: 775
Y: 40.68
X: 997
Y: 38.22
X: 1108
Y: 38.74
FFT [db]
f [Hz]
55
50
45
40
35
30
25
20
15
10
5
200
400
600
800
1000
1200
1400
1600
1800

f_0 = 110.8 Hz

Environment

- The program is implemented using Carbide C++ environment for Symbian

- Run on Nokia cellphone emulator, and Nokia N95 cellphone purchased for the project

Environment

- Nokia N95 specifications:
  - Operating System: Symbian OS v9.2
  - CPU Type: ARM 11 (clock rate: 332MHz)
  - Free Executable RAM Memory: 18 MB
  - Audio sampling rates: 8 – 48 KHz
  - Has on-device debug capability

Conclusions

- Project goal was achieved
- All requirements were met
- The application could be used freely
- Good feedbacks

Live Demonstration

FFT algorithm: Real signal optimization

Original signal:

$$n \in [0, N-1]$$

$$m, k \in [0, (N/2)-1]$$

$$u[n] \in \mathrm{R}$$

Define x[n]:

$$x[m] = u_e[m] + j * u_o[m]$$

$$\mathbb{F}\{x[m]\} = X[k] = U_e[k] + j * U_o[k]$$

According to classic FFT:

$$U[k] = U_e[k] + W_N^{-k} * U_o[k]$$