

Processamento de áudio em tempo real em dispositivos computacionais de alta disponibilidade e baixo custo

André J. Bianchi

21/10/2013

1 Introdução

2 Metodologia

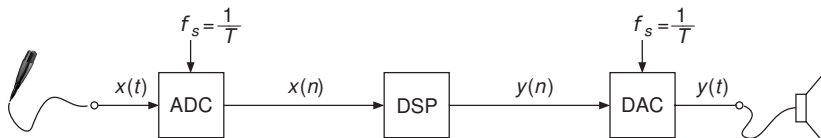
3 Arduino

4 GPU

5 Android

6 Conclusão

Proposta de pesquisa



Objetivo:

- Avaliação do desempenho ...
- ... de diferentes dispositivos computacionais ...
- ... para processamento de áudio em tempo real.

Metodologia desenvolvida

Passos preliminares:

- Identificação de modelos computacionais.
- Escolha de dispositivos.
- Escolha de algoritmos de processamento em tempo real.
- Métricas e métodos para avaliação de desempenho.

Experimentação:

- Entrada, saída e agendamento.
- Testes e resultados.
- Análise.

Identificação de modelos computacionais de interesse

Critérios gerais:

- Baixo custo e alta disponibilidade.
- Licenças de uso.
- Expressão computacional.

Critérios “musicais”:

- Entrada e saída de áudio.
- Sensores.
- Mobilidade.

Escolha de modelos computacionais e dispositivos

- Microcontroladores: Arduino.
- Processadores paralelos: Placas GPU.
- Dispositivos móveis: Sistema operacional Android.

Escolha de algoritmos de processamento em tempo real

Transformada Rápida de Fourier (FFT):

$$X(k) = \begin{cases} E_k + e^{\frac{-2\pi i}{N}k} O_k & \text{se } k < \frac{N}{2} \\ E_{k-\frac{N}{2}} + e^{\frac{-2\pi i}{N}(k-\frac{N}{2})} O_{k-\frac{N}{2}} & \text{se } k \geq \frac{N}{2} \end{cases}$$

Convolução no domínio do tempo:

$$\mathbf{w} = \mathbf{x} * \mathbf{h} \Rightarrow w_r = \sum_{m=0}^{N-1} x_m h_{r-m} \Rightarrow W_k = X_k H_k.$$

Síntese aditiva:

$$y(n) = \sum_{k=1}^K r_k(n) \sin\left(\frac{2\pi f_k n}{R}\right), \quad n \geq 0.$$

Phase Vocoder: FFT + Síntese aditiva.

Métricas e métodos para avaliação de desempenho

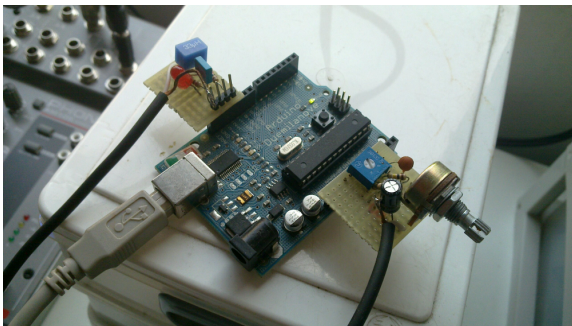
Parâmetros determinantes da complexidade computacional:

- Tamanho do bloco (período em amostras).
- Parâmetros específicos de cada algoritmo.

Métricas de avaliação:

- Tempo de processamento de um bloco de amostras.
- Instância máxima viável em tempo real.

Arduino



Microcontrolador Atmel AVR ATmega328P:

- CPU RISC, 16 MHz, 8 bits.
- 2 KB SRAM.
- Portas digitais com ADC e PWM.

Entrada e saída de áudio e agendamento (ATMega328P)

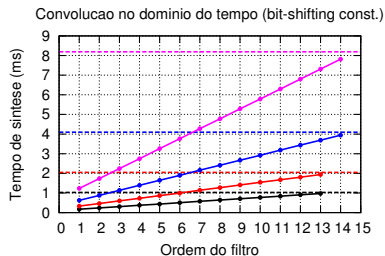
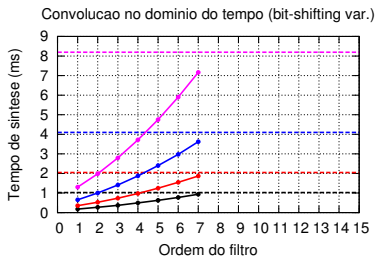
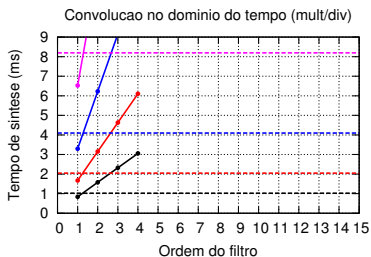
Características gerais:

- ADC de 8 e 10 bits, até ≈ 80 KHz.
- PWM com 8 ou 16 bits.
- Agendamento por interrupção.

Configurações dos testes:

- Entrada e saída de 8 bits.
- Frequência de operação: 31250 Hz.

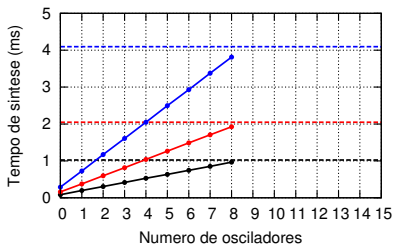
Convolução no Arduino



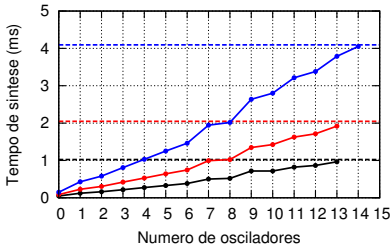
Síntese aditiva no Arduino

bloco de 32 amostras —●—
 bloco de 64 amostras —●—
 bloco de 128 amostras —●—

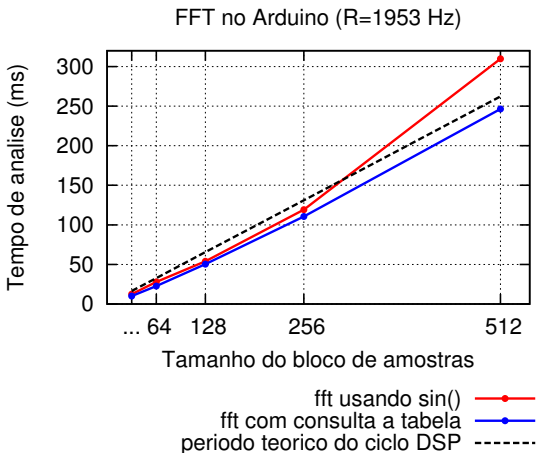
Síntese Aditiva em Arduino (usando um loop)



Síntese Aditiva em Arduino (using inline code)



FFT no Arduino (a 1953 Hz)

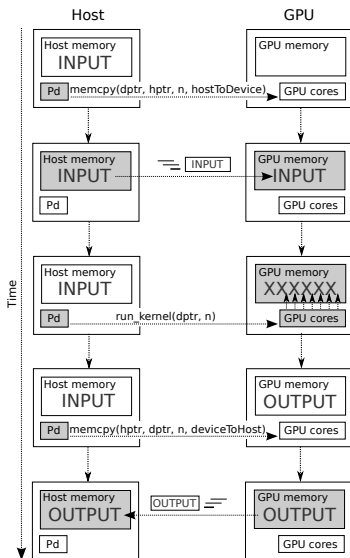


Processamento de áudio em tempo real em Arduino

Pontos importantes a serem observados:

- Configuração de relógios e pré-escaladores.
- Escolha dos tipos (byte, unsigned long, int, float, etc).
- Uso de laços e condicionais.
- Consulta a variáveis e vetores.
- Restrição das instâncias dos algoritmos.

Entrada, saída e agendamento na GPU



Entrada, saída e agendamento:

- Pure Data.

Medição dos tempos de:

- Transferência de memória.
- Execução do programa.
- Total de ida e volta.

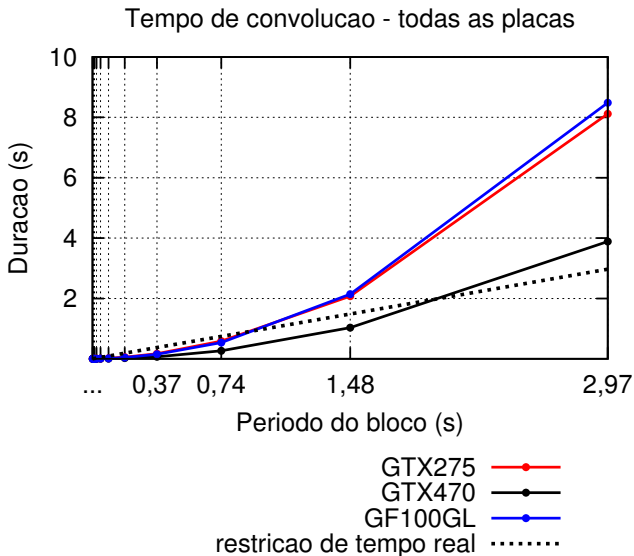
Modelos de placa e implementações

	GF100GL	GTX275	GTX400
CUDA cores	256	240	448
Memória RAM (MB)	2000	896	1280
Banda de memória (GB/s)	89.6	127.0	133.9

Implementações paralelas:

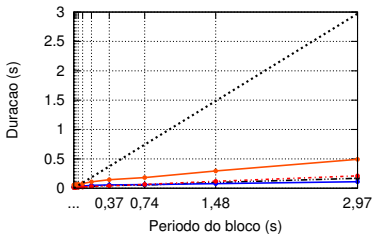
- Convolução.
- FFT (da API).
- Phase Vocoder (FFT + Síntese Aditiva).

Convolução na GPU



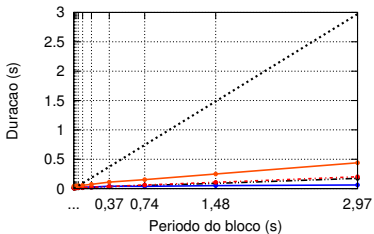
FFT na GPU

Tempo de roundtrip da FFT - GTX275

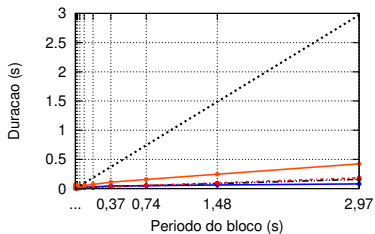


hospedeiro para dispositivo - - - ● - - -
 tempo de kernel — ● —
 dispositivo para hospedeiro - - - ● - - -
 roundtrip — ● —
 restricao de tempo real - - - ● - - -

Tempo de roundtrip da FFT - GTX470

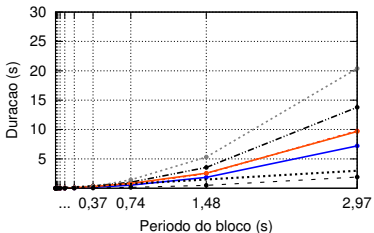


Tempo de roundtrip da FFT - GF100GL



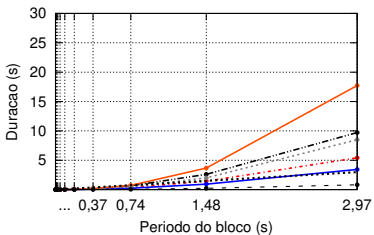
Síntese aditiva do Phase Vocoder na GPU

Tempo de síntese do PV - GTX275

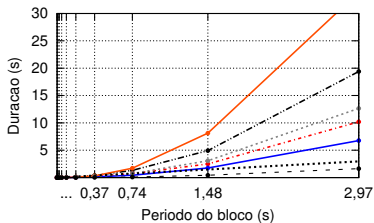


1. interpolação cúbica
 2. interpolação linear
 3. consulta truncada
 4. função seno
 5. interpolação de textura
- sem cálculo
restricao de tempo real

Tempo de síntese do PV - GTX470



Tempo de síntese do PV - GF100GL



Processamento de áudio em tempo real em GPU

Algumas conclusões:

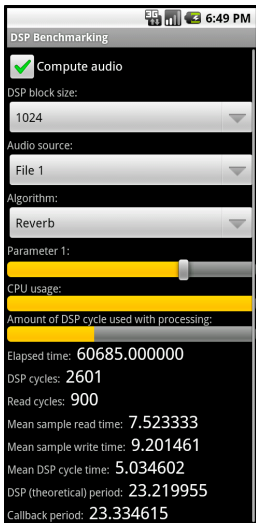
- Modelos não profissionais podem valer a pena.
- Tempo de transferência e da FFT são similares.
- Cuidado com o uso de operações da GPU.

Processamento em tempo real em Android

- Kernel do Linux.
- Entrada de áudio: microfone, chamada, arquivo, rede, etc.
- Saída de áudio: 8 e 16 bits.
- Agendamento da API.



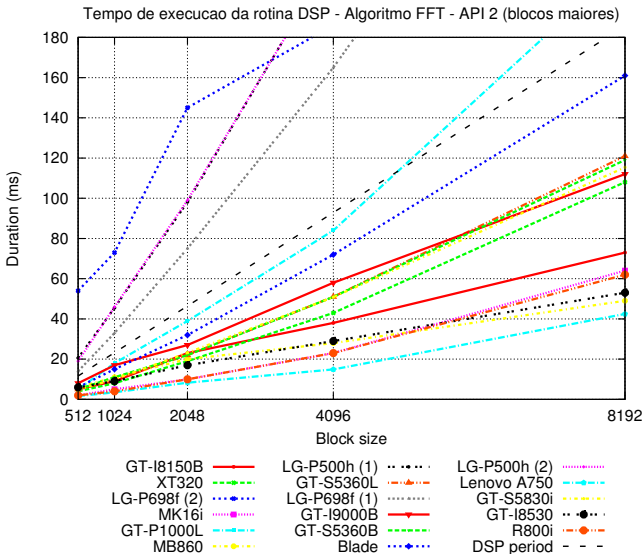
Aplicativo para Android e testes



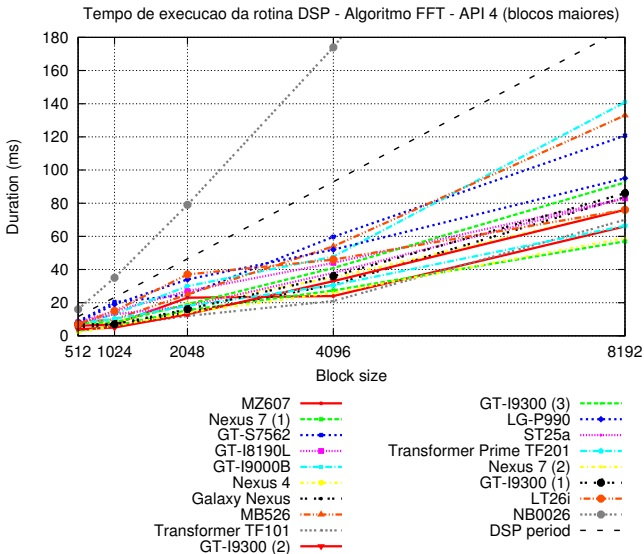
Cenário dos testes:

- Chamado com instruções.
- Envio dos resultados por email.
- 35 aparelhos testados.
- 14 algoritmos em 2 fases.

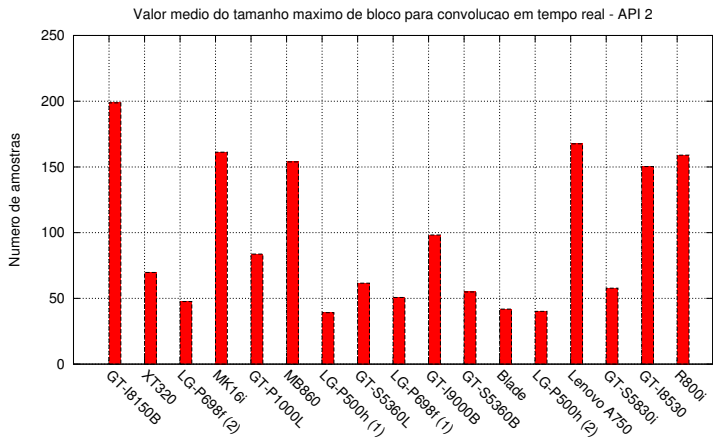
FFT no Android (API 2)



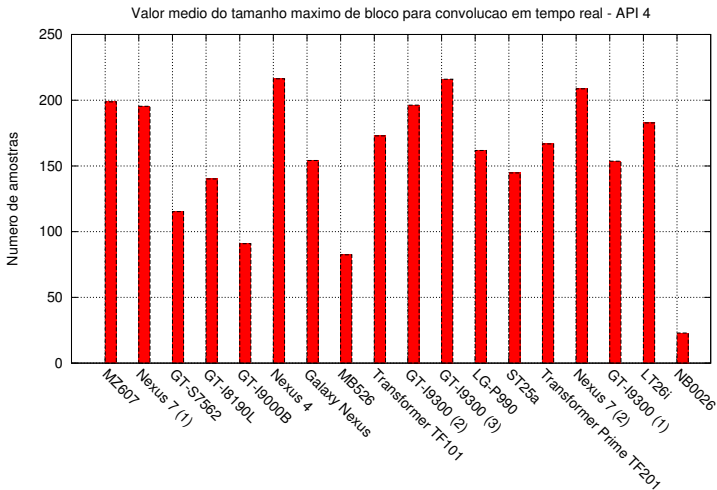
FFT no Android (API 4)



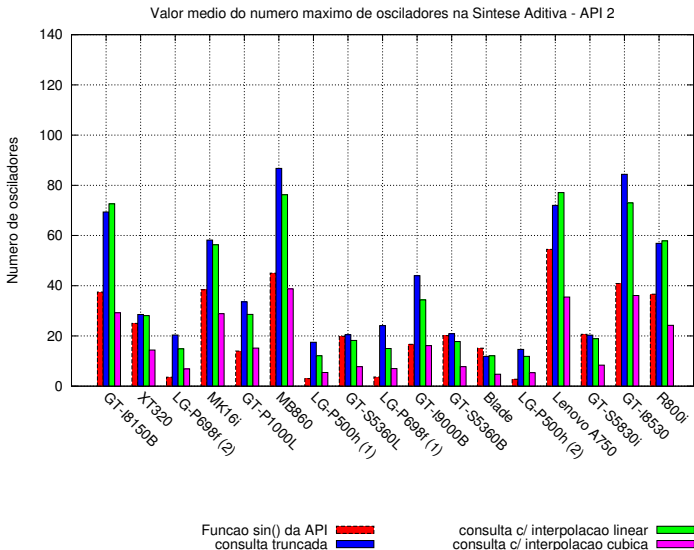
Convolução no Android (API 2)



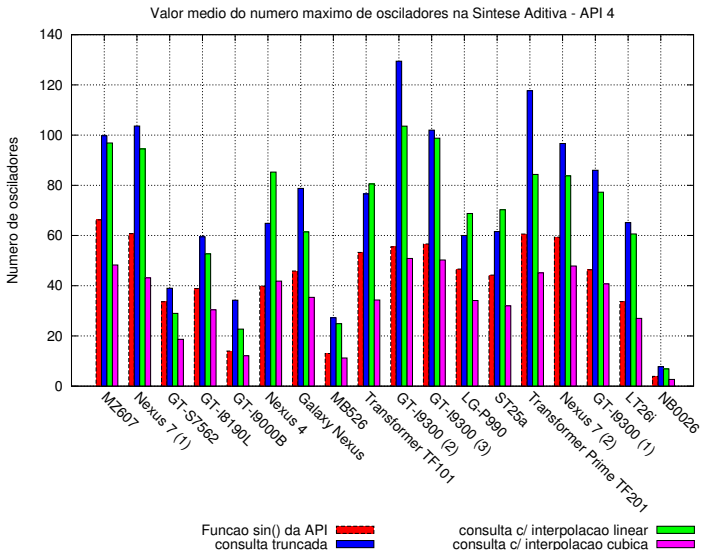
Convolução no Android (API 4)



Síntese aditiva no Android (API 2)



Síntese aditiva no Android (API 4)



Processamento de áudio em tempo real em Android

Pontos importantes a serem observados:

- Heterogeneidade de capacidade computacional.
- Versão da API influencia o desempenho.

Considerações finais

- A metodologia apresentada pode ser usada para avaliação de qualquer dispositivo computacional.
- Foram estudados modelos com possibilidades e restrições distintas.
- Foram evidenciadas as características e o desempenho de cada um para o processamento de áudio em tempo real.

Publicações e apresentações em congressos

2012:

- Bianchi (2012). Processamento de áudio em tempo real em sistemas Android. Terceiro Workshop em Música Ubíqua.
- Bianchi e Queiroz (2012a). Measuring the performance of realtime DSP using pure data and gpu. Proceedings of the International Computer Music Conference 2012, pp. 124-127.
- Bianchi e Queiroz (2012b). On the performance of real-time DSP on Android devices. Proceedings of the 9th Sound and Music Computing Conference, pp. 113-120.

2013:

- Bianchi e Queiroz (2013). Real time digital audio processing using Arduino. Proceedings of the Sound and Music Computing Conference 2013, pp. 538-545.
- Carvalho Jr, Rosan, Bianchi e Queiroz (2013). FFT benchmark on Android devices: Java versus JNI. Proceedings of the 14th Brazilian Symposium on Computer Music. (a ser publicado)

Fim!

Obrigado pela atenção!

- Contato: ajb@ime.usp.br
- Grupo de Computação Musical do IME/USP:
<http://compmus.ime.usp.br/>
- Esta apresentação: <http://www.ime.usp.br/~ajb/>