GPU TECHNOLOGY CONFERENCE 2010

Real-time Multichannel Audio Convolution

J.A. Belloch, A. M. Vidal, F. Martinez-Zaldivar and A. Gonzalez

Presented by: Jose A. Belloch

Supported by PROMETEO/2009/013 (Generalitat Valenciana)

and TEC2009-13741 projects (Science and Innovation Department, Spain)

Contents

1. Introduction to multichannel audio
2. Convolution Algorithm
3. One channel implementation on GPU
4. Scalability to Multichannel: Massive convolution
5. GPU Applications with multichannel signals
6. GPU Crosstalk canceller application
7. DEMO: Multichannel GPU applications
8. Future Research

1. Introduction

• Multichannel acoustic signal processing has experienced a great development in recent years.

STEREO SYSTEM

1. Introduction

• Multichannel acoustic signal processing has experienced a great development in recent years.

4.1 SYSTEM

1. Introduction

• Multichannel acoustic signal processing has experienced a great development in recent years.

5.1 SYSTEM

1. Introduction

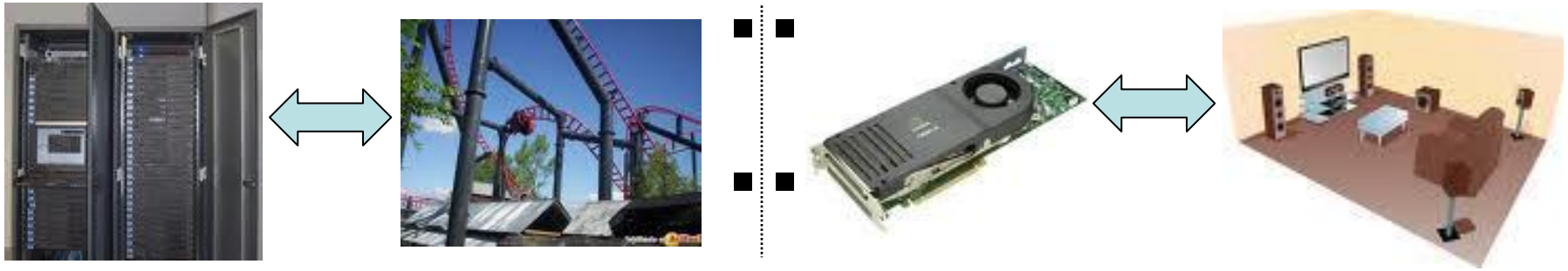
• Multichannel acoustic signal processing has experienced a great development in recent years.

CINEMA SYSTEM

1. Introduction

- The growing need to incorporate new effects and to improve the experience of hearing has increased the number of loudspeakers.

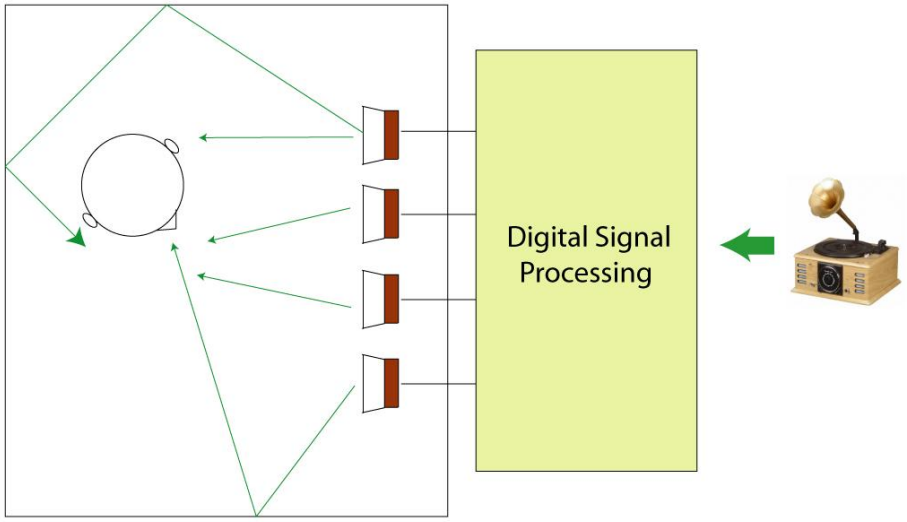
- Achieving the desired effect requires high capacity computing.



1. Introduction

• Basic operation in multichannel acoustic signal processing: Massive Convolution.

• Carrying out different convolutions of different channels in a parallel way.



2. Convolution Algorithm

- Definition:

$$y[n] = \sum_{j=0}^{N-1} x[j]h[n-j]$$

Where x has M elements h has N elements and y has L elements

$$L \geq M + N - 1 \qquad M \gg N$$

- Convolution theorem states:

If both signals x and h are padded with zeros to the length L

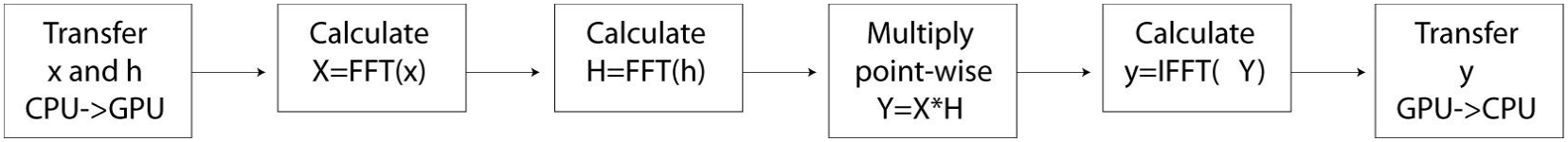
$$DFT[y] = DFT[h]DFT[x]$$
$$y = DFT^{-1}[Y]$$

3. One channel Implemmentation on GPU

• The use of GPU with its computational cores opens the chance of parallelizing the computational load.

•NVIDIA has its own FFT library over a GPU: CUFFT.

•Basic Algorithm scheme in GPU for a convolution:



•Problems:

- Latency in transfer data CPU<->GPU.
- Signal x is extremely long.
- Size of signal is not known, such as a real-time application.

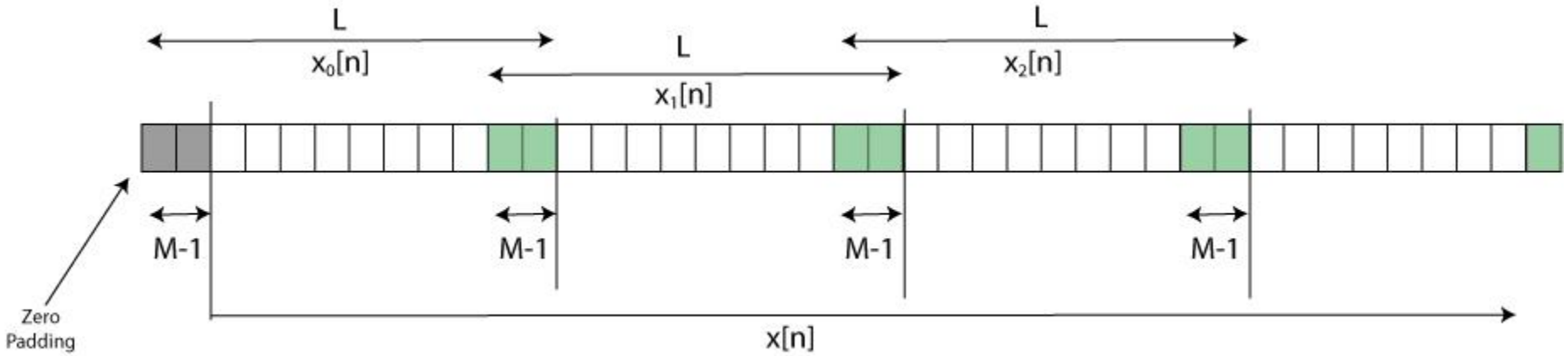
3.1 Long Signals

•In order to carry out a convolution, the signal is divided into several smaller parts.

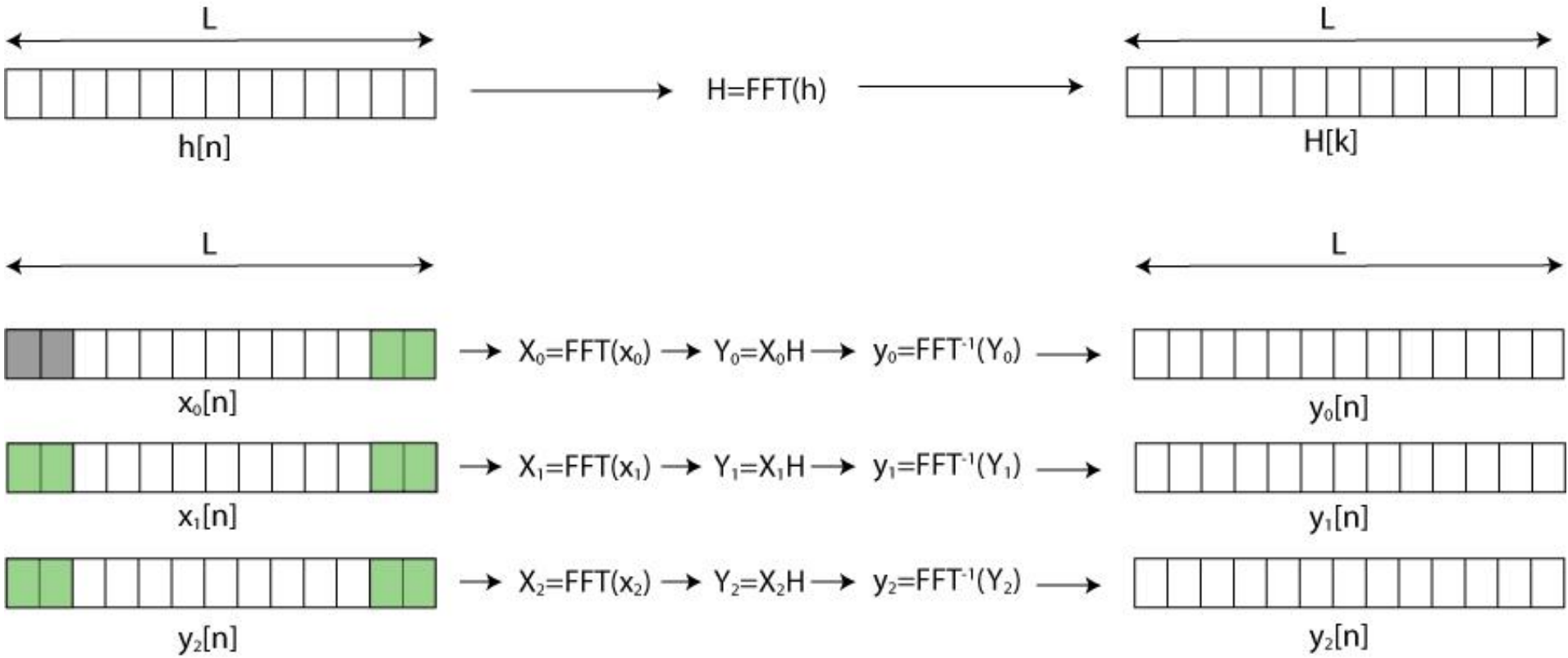
•*Overlap-save* technique is used to obtain the convolution of long signal.

•Signal parts of L samples are taken.

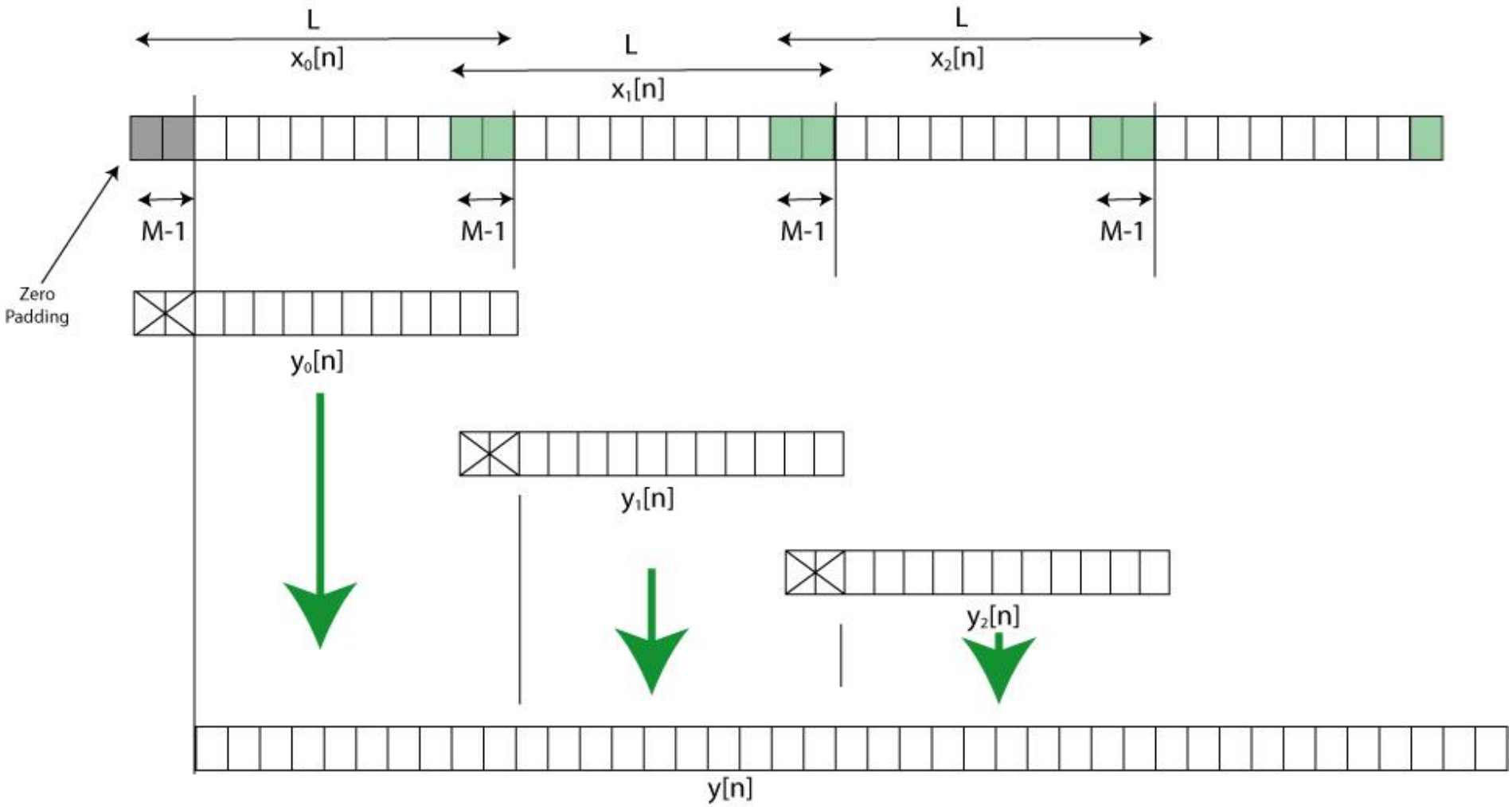
•Each part has an overlap of M-1 samples with the previous part. (M=number of coeficients of filter h)



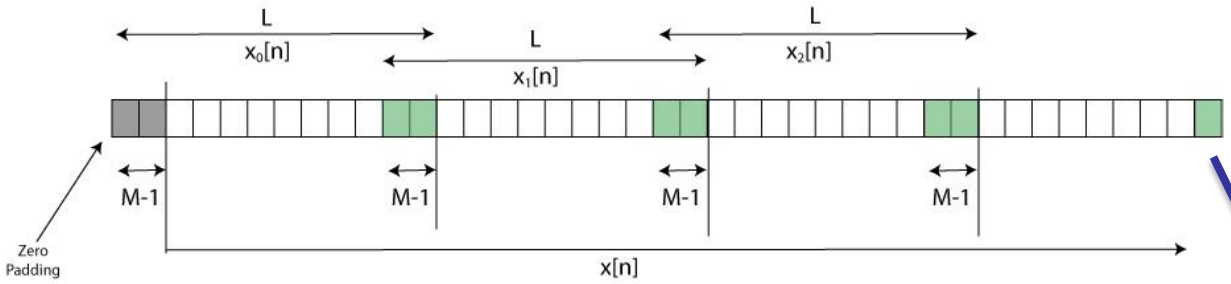
3.1 Long Signal

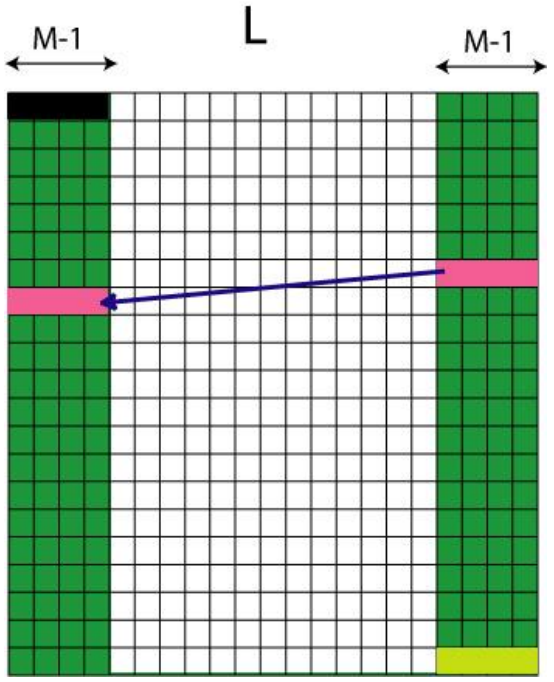


3.1 Long Signal



3.2 Parallelism





•Over each fragment, the same operation is being carried out.

• NVIDIA FFT library, CUFFT, allows carrying out multiple FFT 1D at the same time.

•A matrix could be configured with all the signal parts in order to carry out as many FFT as rows of this signal matrix has simultaneously.

3.2 Parallelism

• Last CUDA releases includes a new property: *Concurrent Copy and Execution*, which allows GPU computing while data is being transferred from CPU to GPU and vice-versa.

•Therefore, the latency in transfering data can be overlapped with the computation:

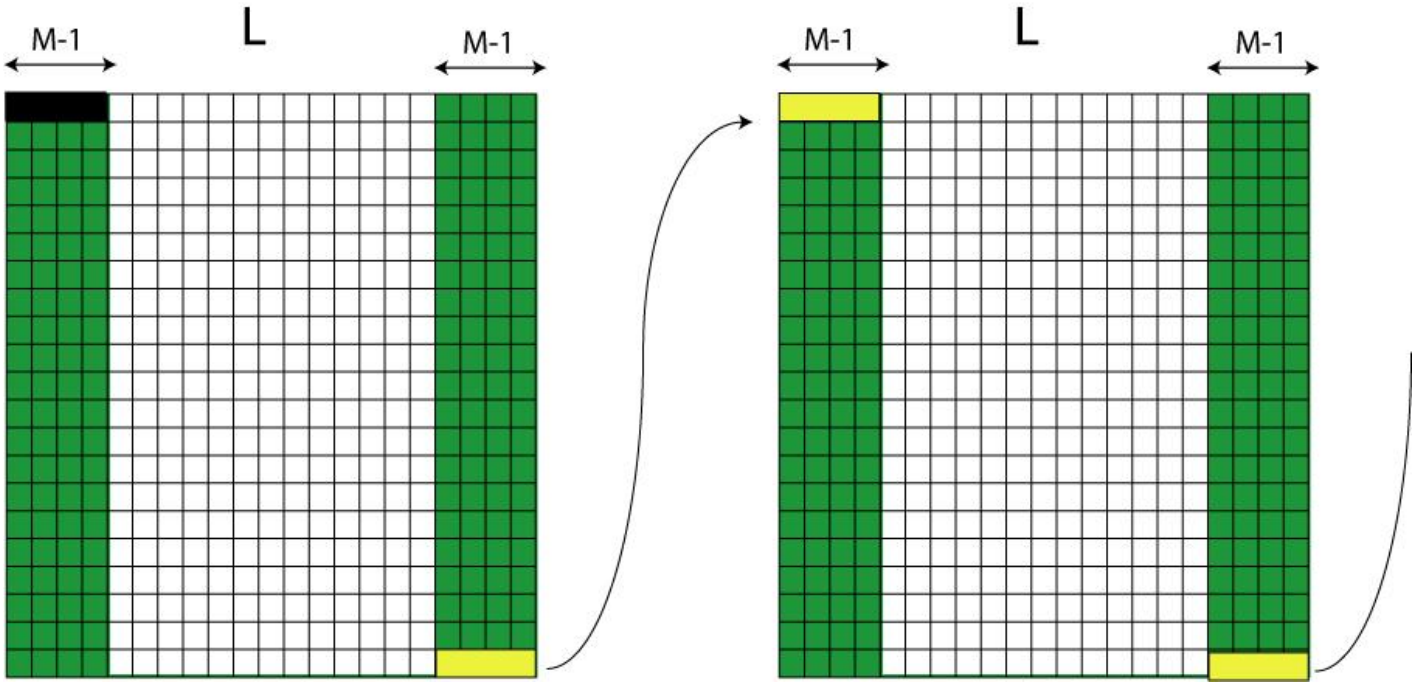
- Achieving high speedup of the convolution

•Making possible real-time applications.

- The Signal matrix containing the signal fragments could be considered as a buffer, filled by the incoming samples.

3.2 Parallelism

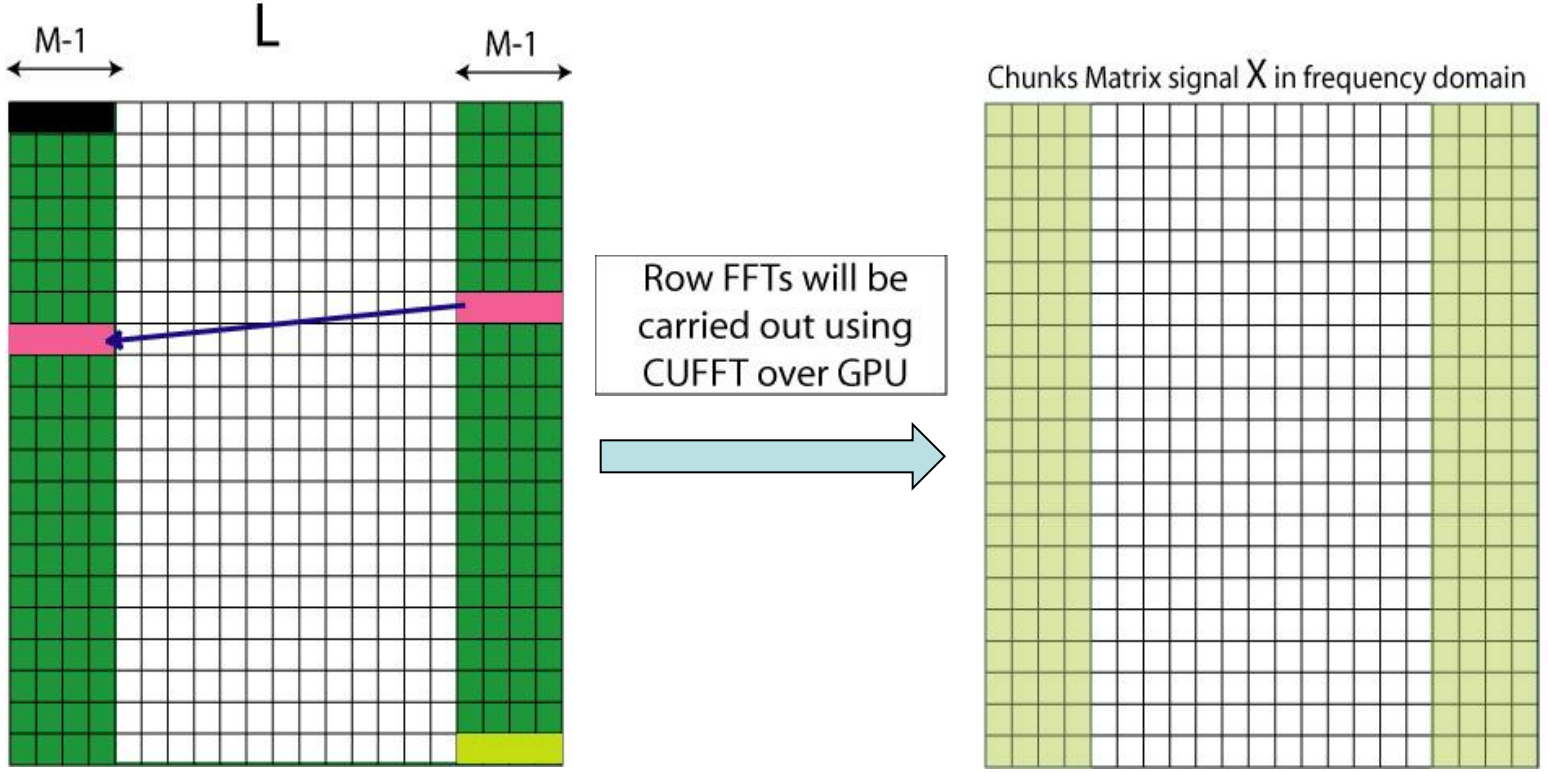
•In that sense, the property *concurrent copy and execution* while audio samples are coming can be used. A matrix-buffer gets filled.



•Once Matrix *i* is filled, it is sent to GPU. Incoming samples begin to fill Matrix *i+1*.

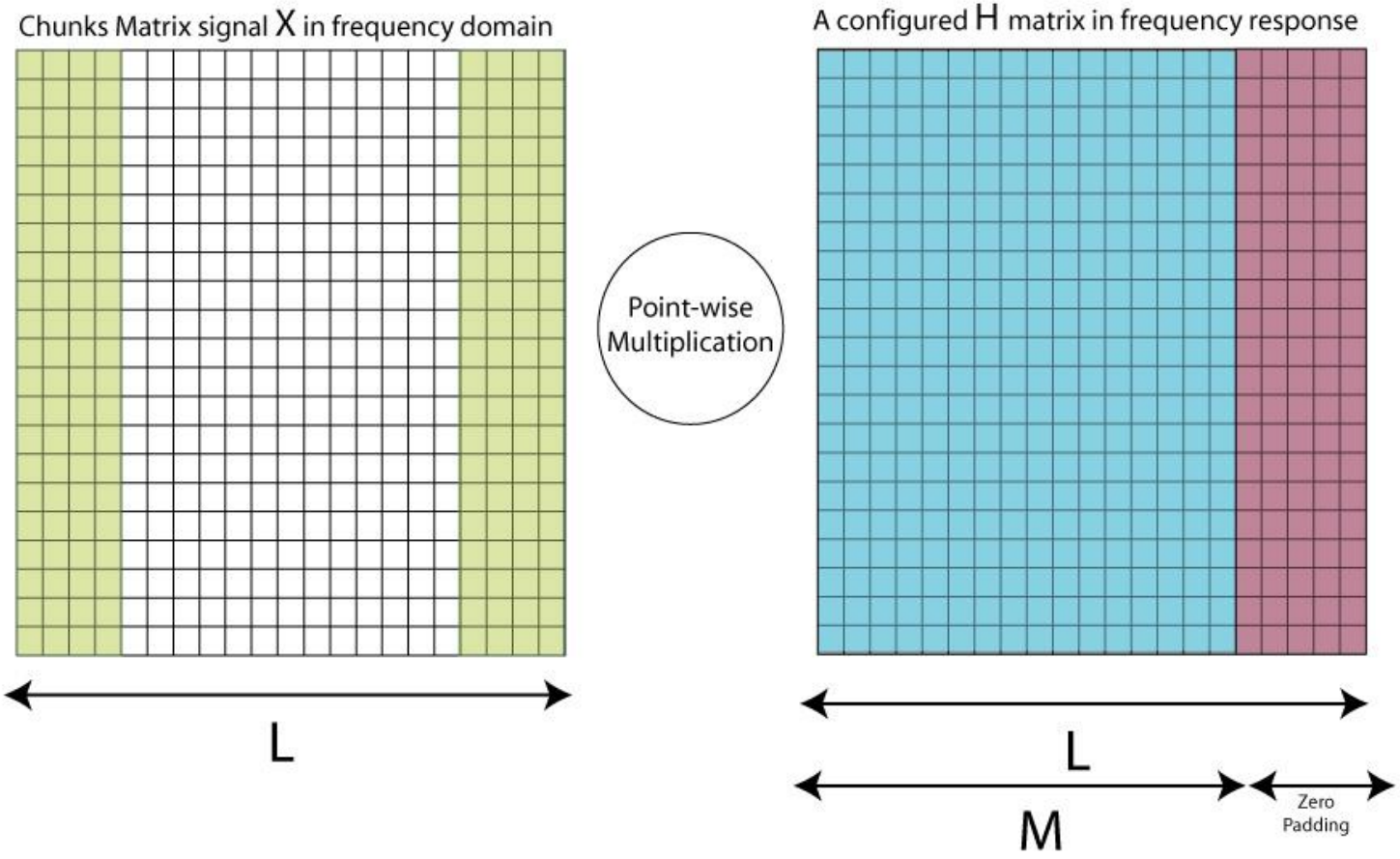
3.2 Parallelism

• Once the signal matrix buffer is full, it is sent to the GPU where CUFFT is applied: Row FFTs will be carried out over GPU



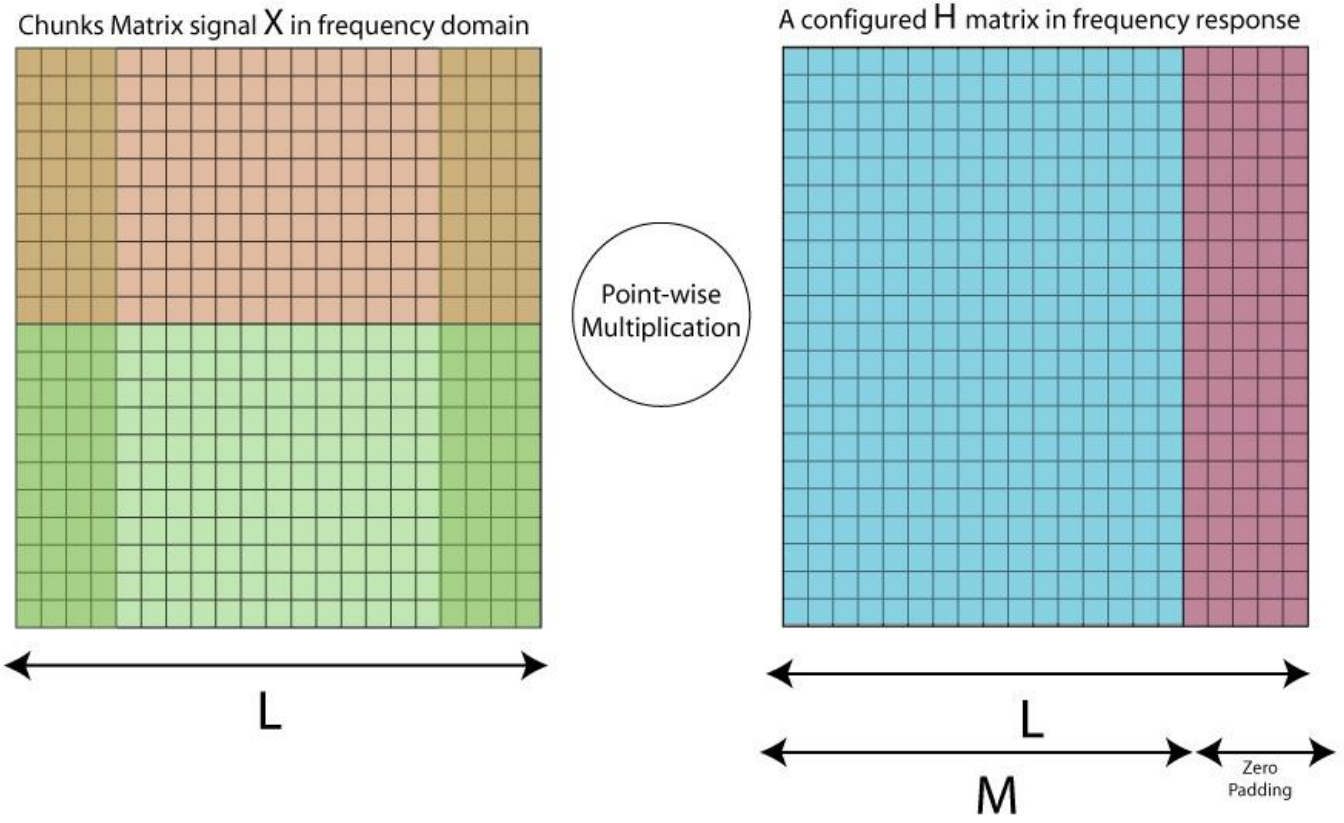
3.2 Parallelism

• A frequency response matrix will be formed in the GPUs, in order to later multiply this two matrices in a point-wise manner.



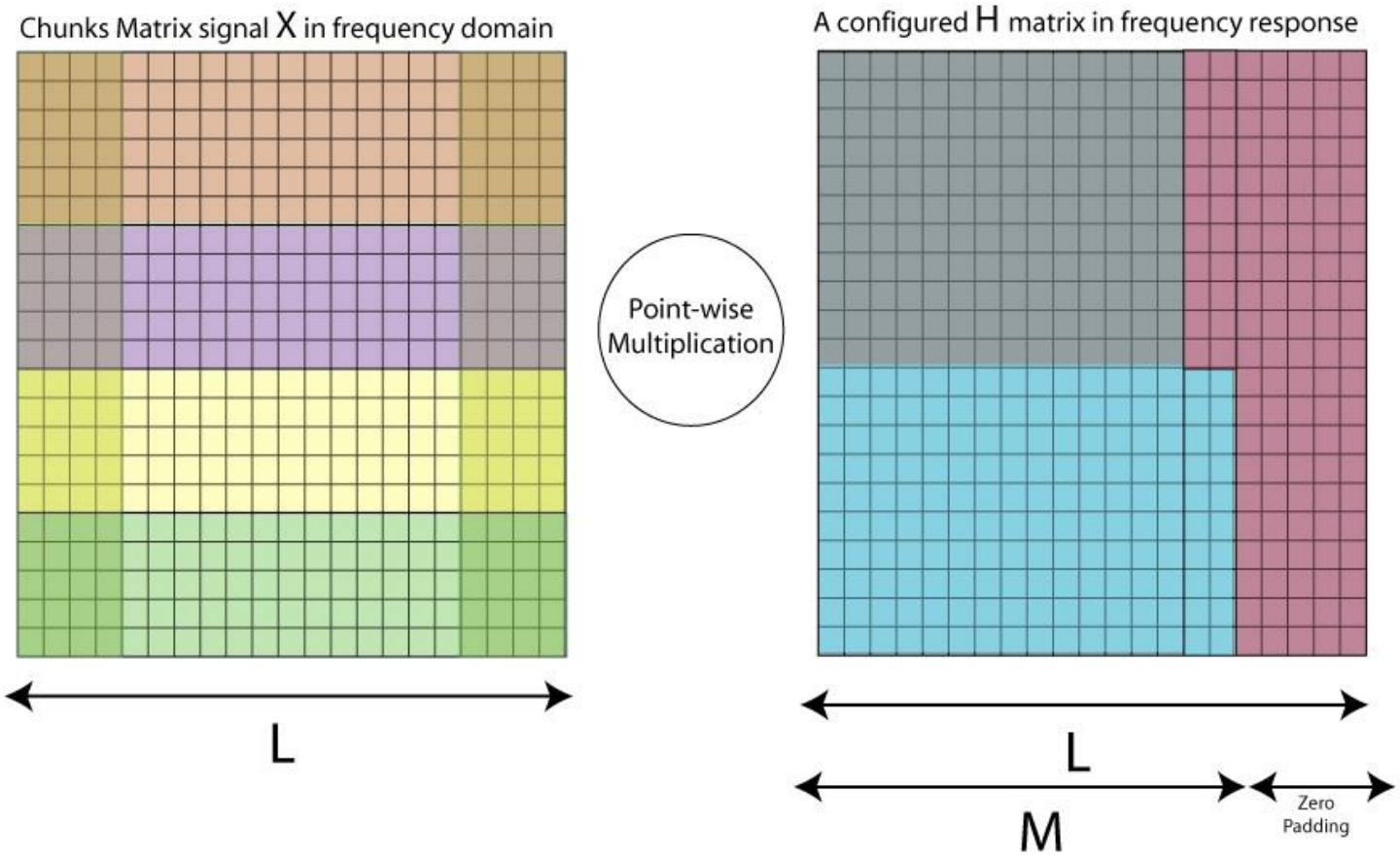
4. Scalability to Multichannel: Massive Convolution

• It is obvious that the hearing effects explained in the introduction cannot be represented by a simple filter nor with an only signal. So, what would it happen if there were 2 signals, but one filter? Resources Sharing!

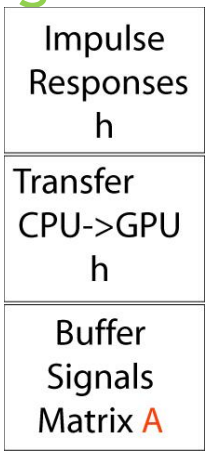


4.1 Massive convolution

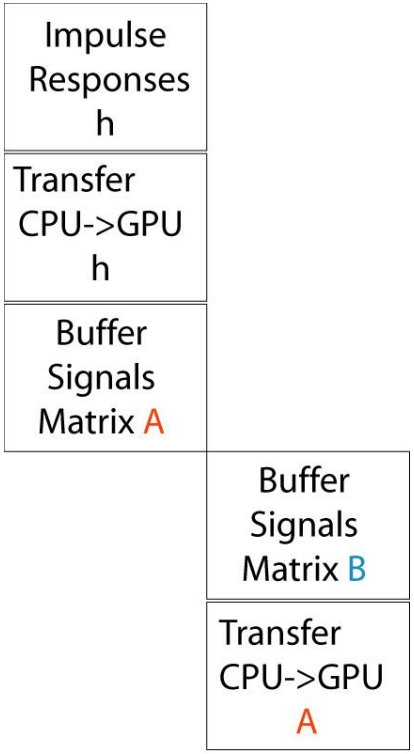
- Or maybe the signal has 4 channels and two different filters



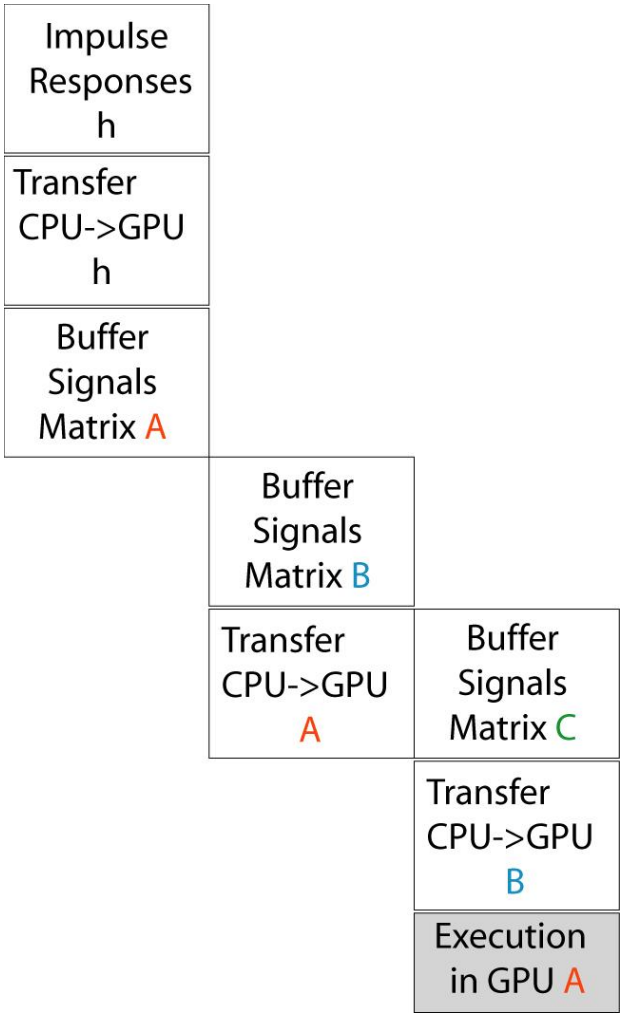
4.2. Pipelined algorithm

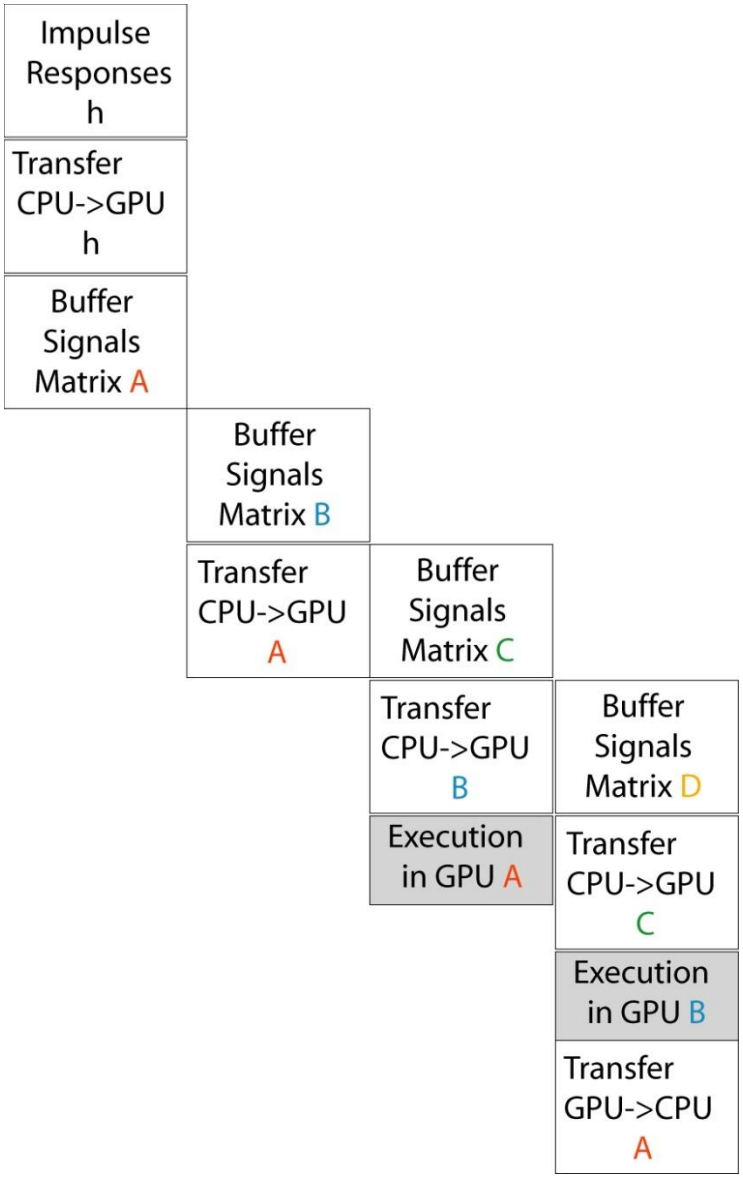


4.2. Pipelined algorithm



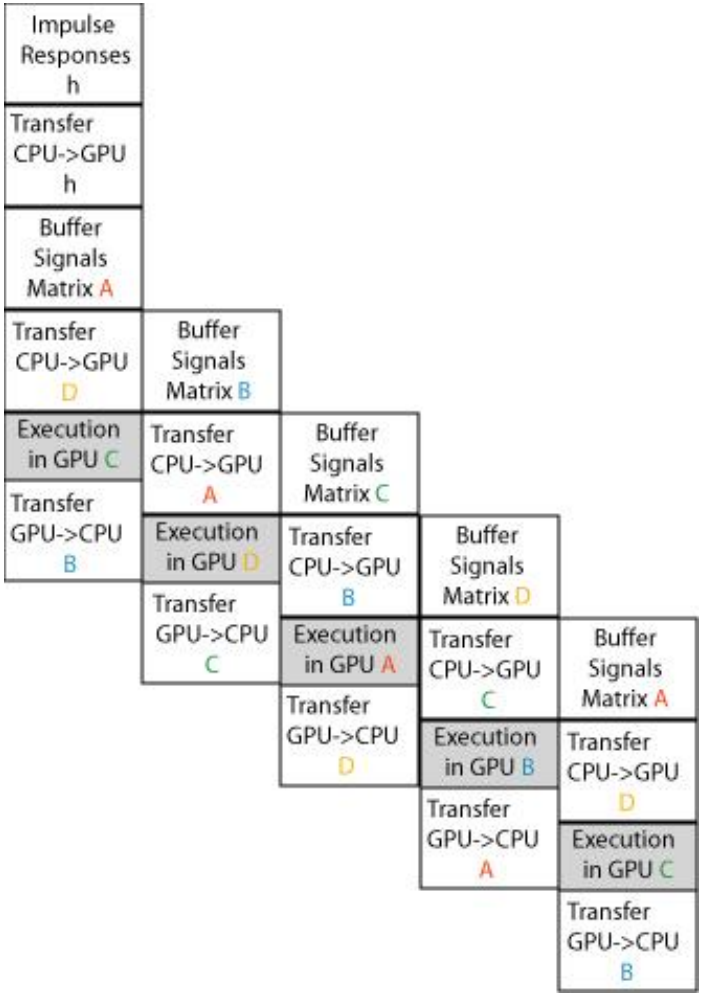
4.2. Pipelined algorithm




Impulse Responses h
Transfer CPU->GPU h
Buffer Signals Matrix A
Buffer Signals Matrix B
Transfer CPU->GPU A
Buffer Signals Matrix C
Transfer CPU->GPU B
Buffer Signals Matrix D
Execution in GPU A
Transfer CPU->GPU C
Execution in GPU B
Transfer GPU->CPU A

4.2. Pipelined algorithm

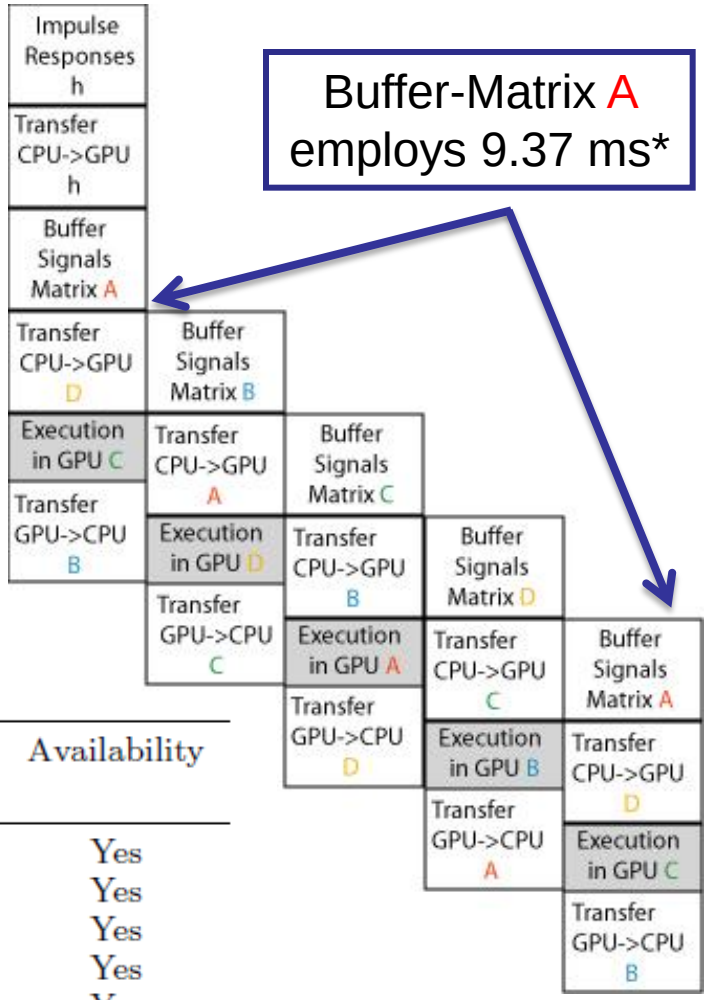
- After organizing the parts of a signal inside a Matrix, 4 different matrices play a role in this algorithm simultaneously:
  - Matrix A is filled by the incoming samples.
  - Matrix B is sent from CPU to GPU
  - Matrix C is being used by the GPU to obtain the convolution.
  - Matrix D is sent from GPU to CPU



4.2. Pipelined algorithm

• In a real time audio application: Transfer and Computation on GPU must spend less time than filling the samples in buffer.

•This time depends on the rate of the incoming samples. (Audio Sample Frequency is 44,1 Khz).

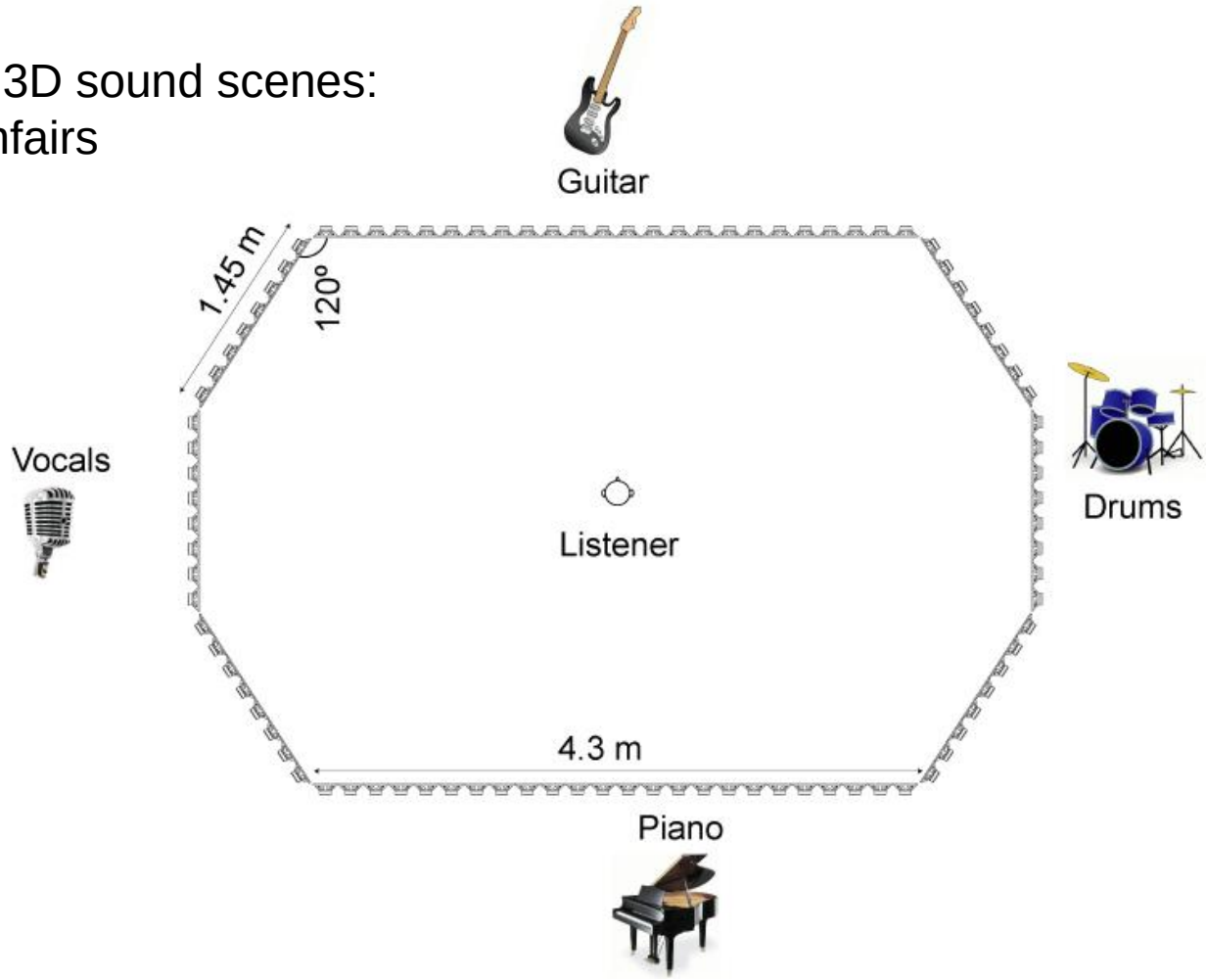


Number of channels	Occupacy of rows per channel	Time employed filling buffer	Use of GPU (%)	Availability
1	32	212.6ms	4.4%	Yes
2	16	106.3ms	8.8%	Yes
4	8	53.15ms	17.6%	Yes
8	4	26.9ms	35.2%	Yes
16	2	13.2ms	70.5%	Yes
32	1	6.6ms	141%	No

*Data mesured with a TESLA C1060

5. GPU Applications with multichannel signals

•Synthesis of 3D sound scenes:
Theaters, Funfairs



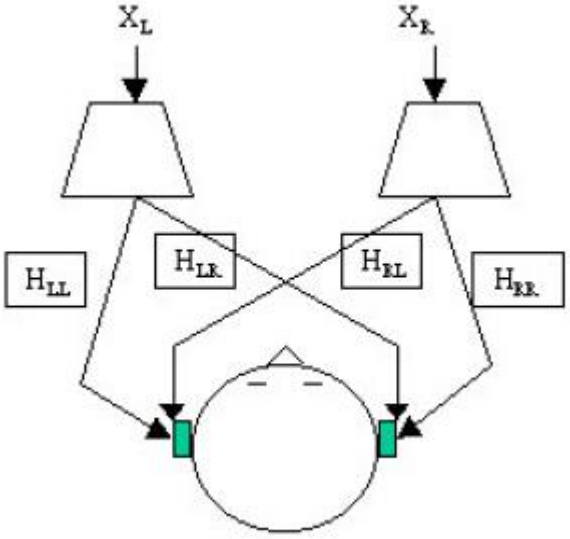
5. GPU Applications with multichannel signals

• **Crosstalk canceller**.

• Crosstalk is a important problem for binaural reproduction.

•Calculation of fields from loudspeaker to ear and process the inverse of the field.

Without earphones



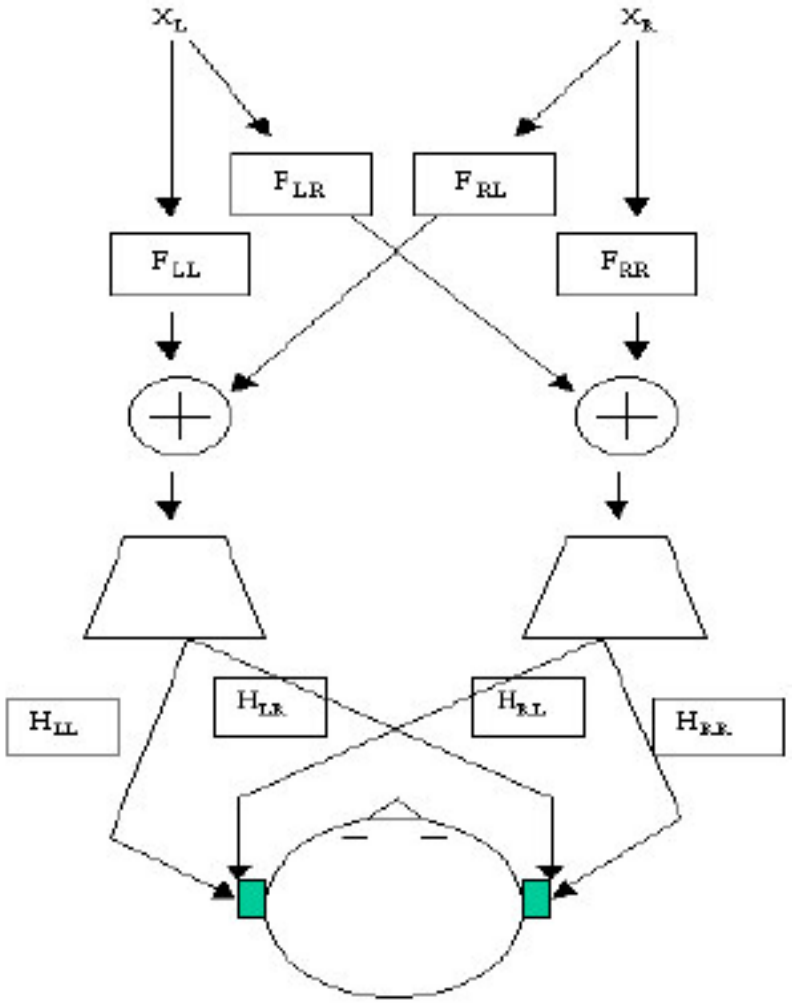
6. GPU Crosstalk canceller application

- Left ear:

$$L = (x_L * F_{LL} + x_R * F_{RL}) * H_{LL} + \\ + (x_L * F_{LR} + x_R * F_{RR}) * H_{RL}$$

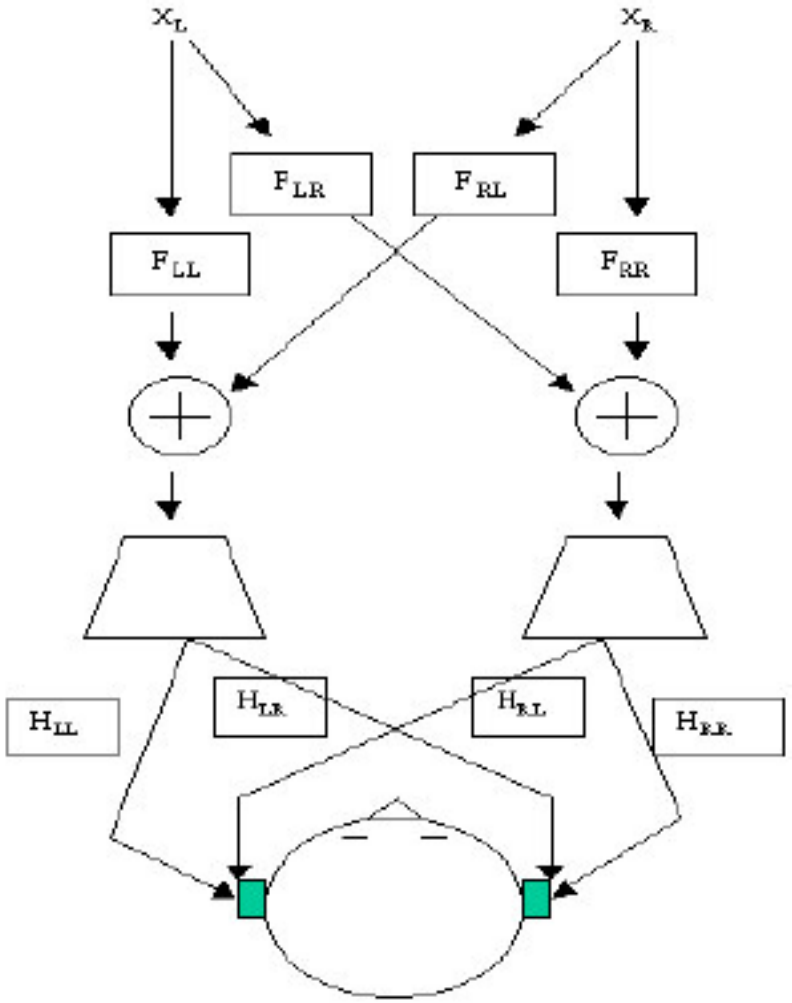
$$L = x_L * (F_{LL} * H_{LL} + F_{LR} * H_{RL}) + \\ + x_R * (F_{RL} * H_{LL} + F_{RR} * H_{RL})$$

$$F_{LL} * H_{LL} + F_{LR} * H_{RL} = 1$$
$$F_{RL} * H_{LL} + F_{RR} * H_{RL} = 0$$



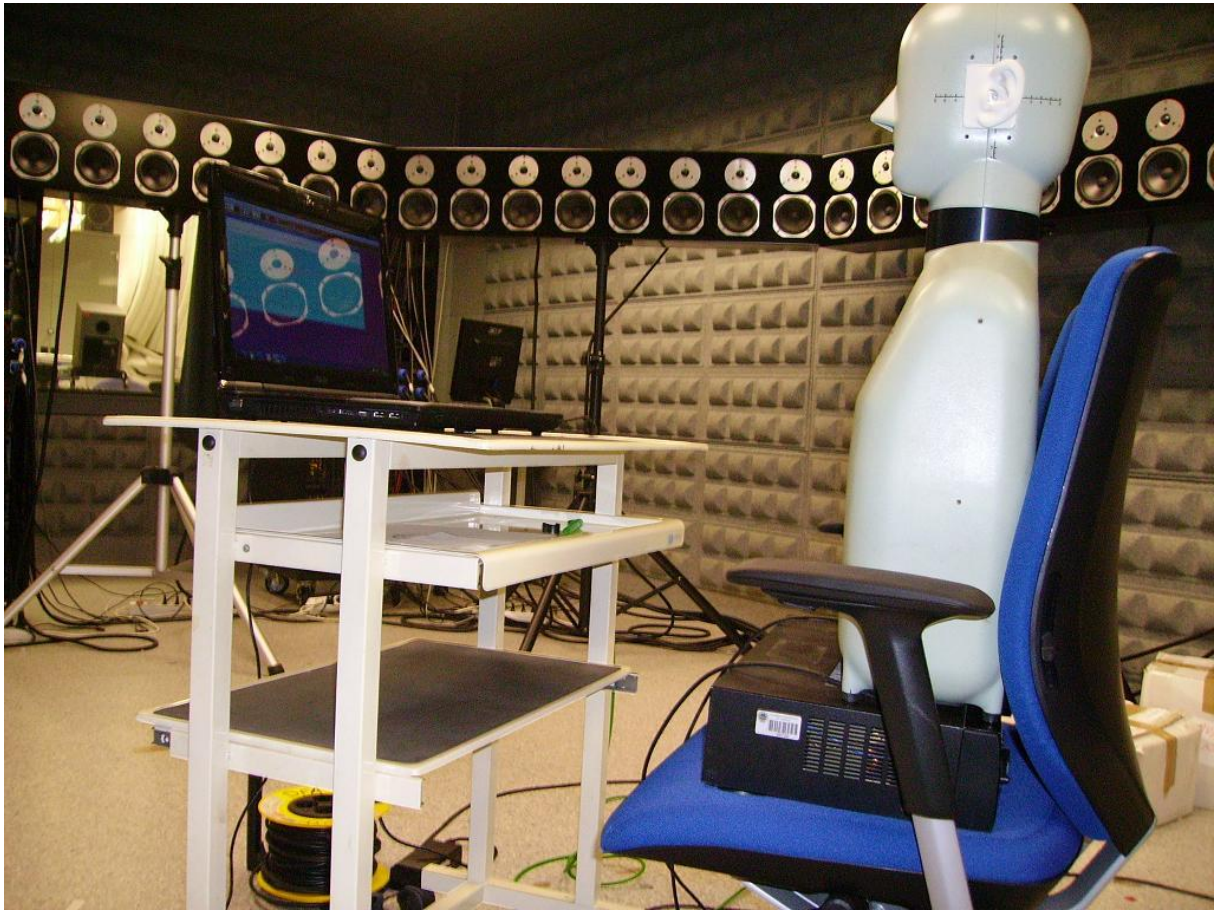
6. GPU Crosstalk canceller application

• We propose to implement a crosstalk canceller bank carrying out all the processing through GPU netbook GTS 360M freeing up CPU resources.



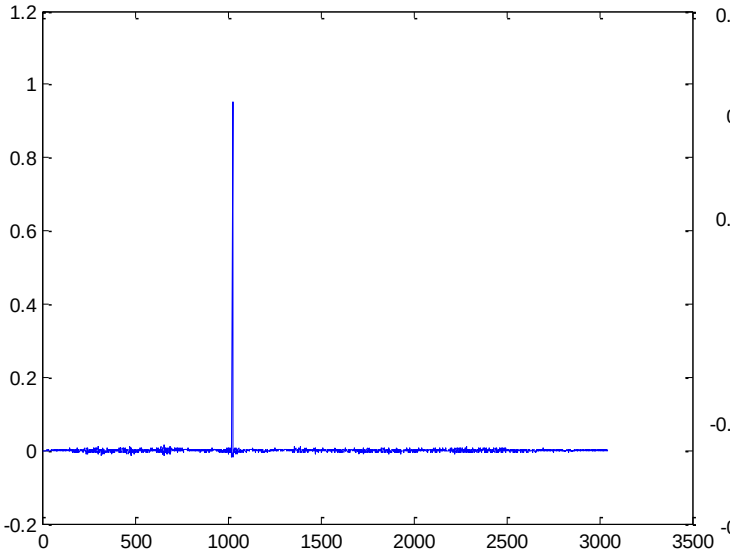
6. GPU Crosstalk canceller application

- Measures from $H_{LL}, H_{LR}, H_{RR}, H_{RL}$ were taken at the ITEAM laboratory:

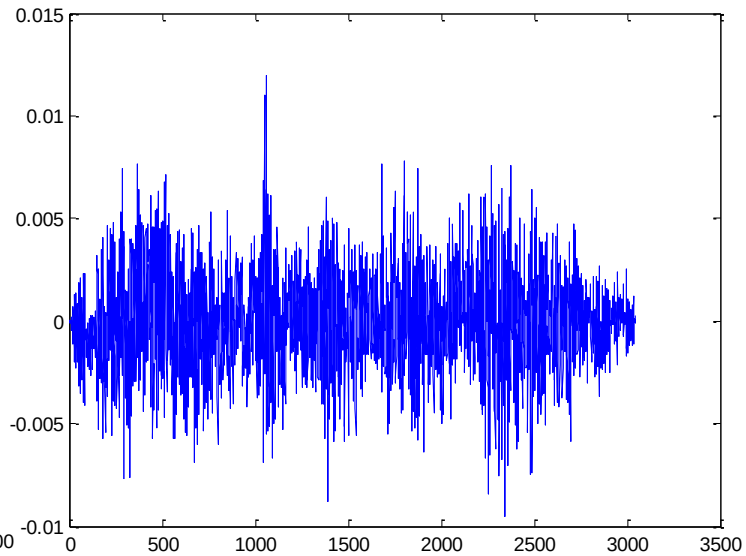


6. GPU Crosstalk canceller application

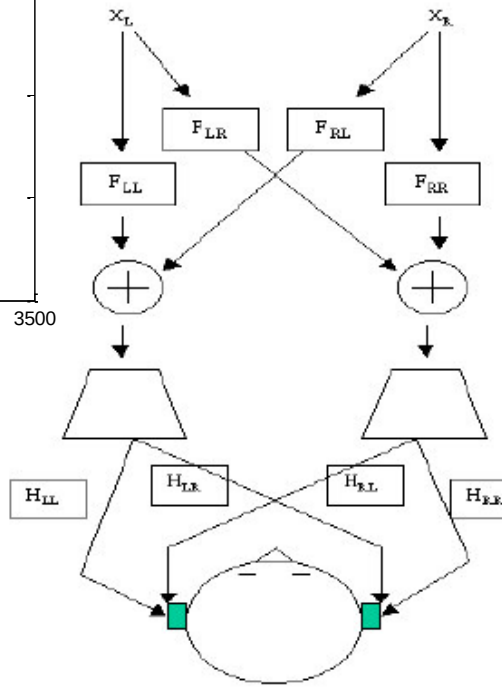
- Left ear:



Contributions from left signal to left ear



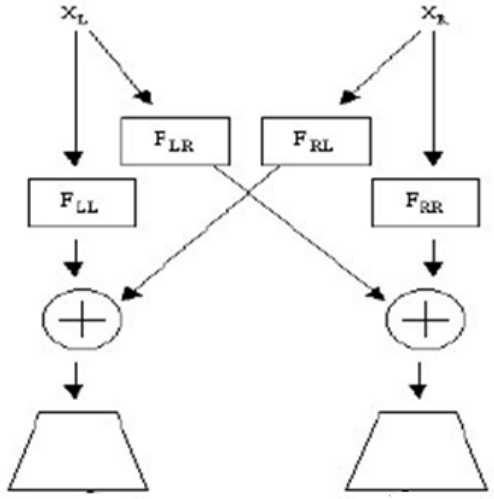
Contributions from right signal to left ear



7. DEMO: Multichannel GPU applications

4) Crosstalk Application canceller with GPU.

Each signal before reaching the loudspeaker goes through two filters and then is summed to another signal



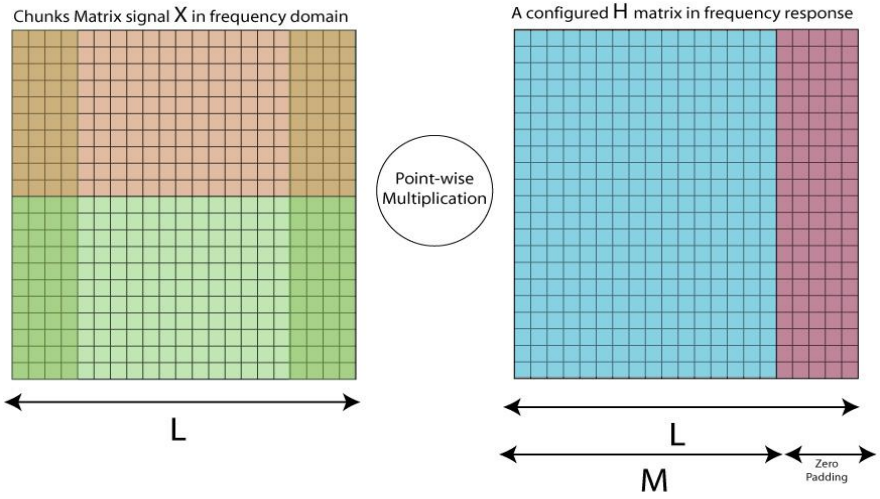
7. DEMO: Multichannel GPU applications

- Let's try a demo application where all Signal Processing goes through GPU

1) Two individual sounds:
   - Voice
   - Piano

2) Low passfilter in real-time simultaneuosly:

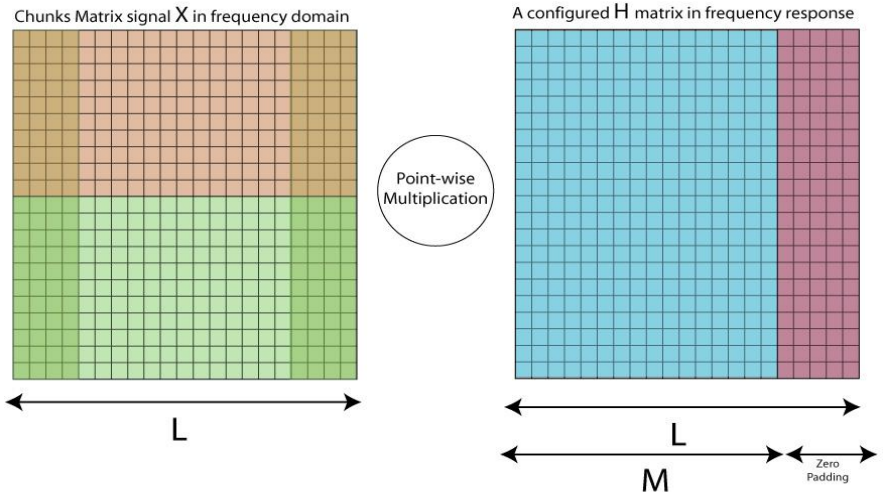
   - Voice -> Right Loudspeaker
   - Piano-> Left Loudspeaker



7. DEMO: Multichannel GPU applications

3) Lowpass filter in a real-time using CPU. (Task Manager Comparing)

- Voice -> Right Laudspeaker
- Piano-> Left Laudspeaker



7. DEMO: Multichannel GPU applications

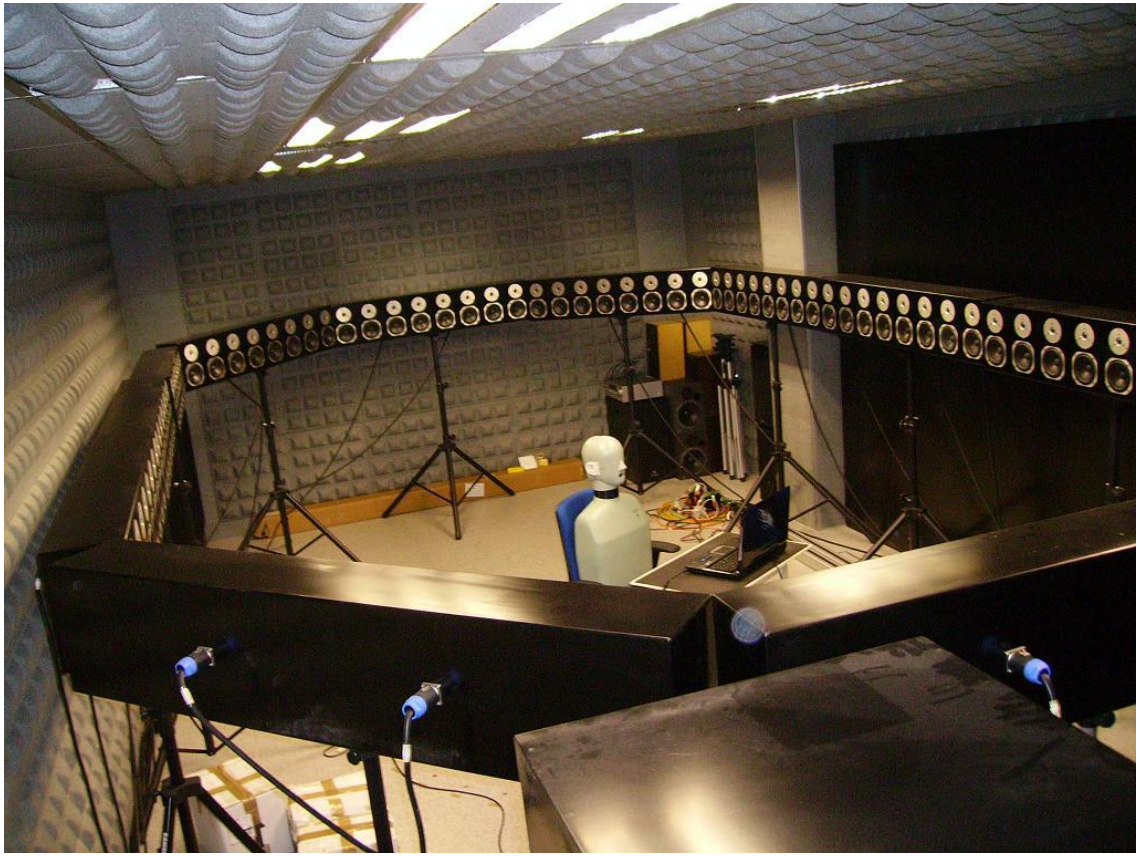
- Testing with only two channels:
    - CPU goes arround 10% 14%.
    - GPU goes arround 6% 10%
    - Save 4% of use of GPU.
- Making an Interpolation:
    - 4 Channels -> 8%.
    - 16 Channels -> 32%.
    - 32 Channels -> 64%.

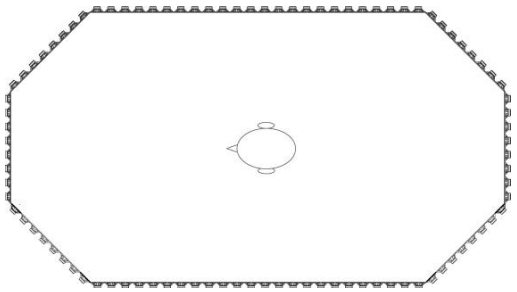
8. Summary

- Massive Convolution on a GPU.
- Software that allows carrying out different audio operations: Signal filter, Room ecualization, signal delay, phase changes, amplitude changes…
- Scalability software adaptable to any number of channels or filters.
- Free up cpu resources and let carry out other task in cpu while processing happens on GPU. Saving energy and improve the environment.

8. Future Research

- Deal with a big number of channels.

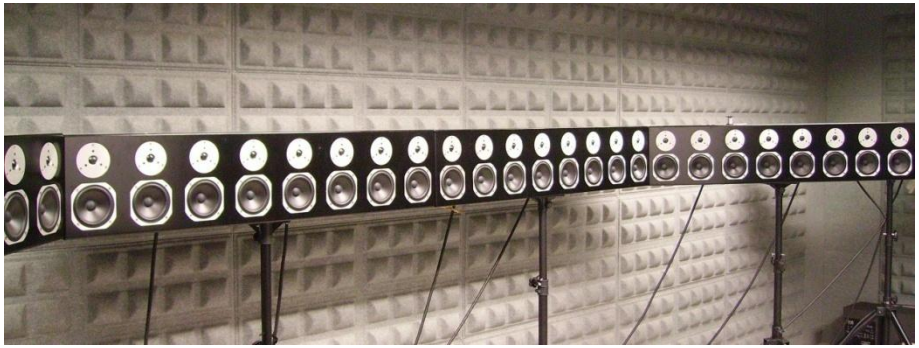




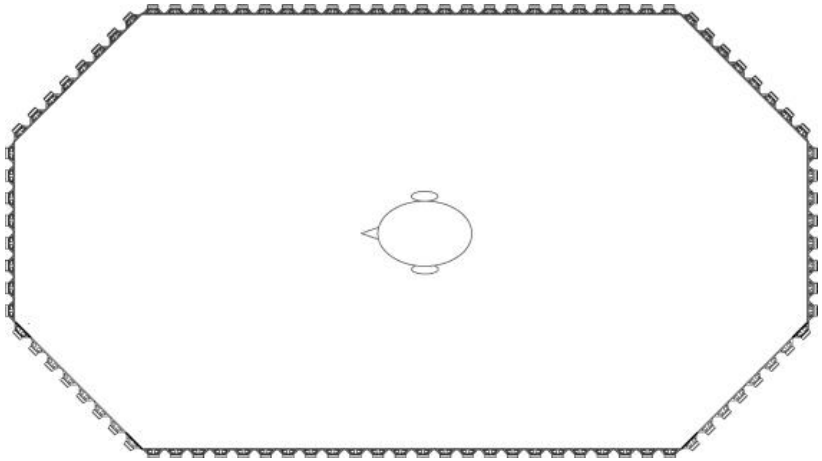
•Our rendering system is composed by 96 loudspeakers

8. Future Research

- Our objective: Manage the whole audio signal processing using GPU.

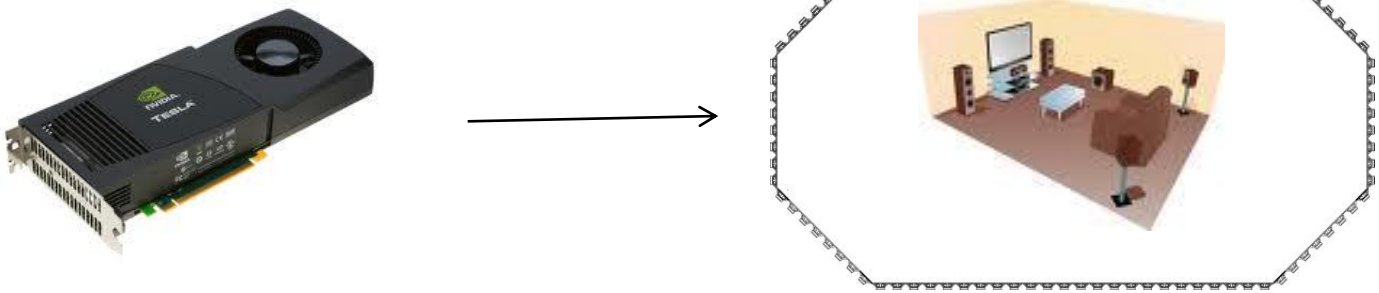


- Freeing CPU resources.
- Saving energy
- Improve the environment



8. Future Research

- Achieving the most incredible audio effects in order to bring them home!



- Without caring about the source of the audio signal:
  - Audio files: wav, mp3,…
  - Cds
  - Streaming music over internet.

Thanks for
your attention