Real Time Processing of Microphone Array Information Applying GPU Unit and CUDA Platform

SNEJANA PLESHKOVA
Faculty of Telecommunications
Technical University – Sofia
1000 Sofia St. Kl. Ohridski 8
BULGARIA
snegpl@tu-sofia.bg

VIKTOR HRISTOV
Faculty of Telecommunications
Technical University – Sofia
1000 Sofia St. Kl. Ohridski 8
BULGARIA
vhh@tu-sofia.bg

ALEXANDER BEKIARSKI
Faculty of Telecommunications
Technical University - Sofia
1000 Sofia St. Kl. Ohridski 8
BULGARIA
aabbv@tu-sofia.bg

Abstract: - Microphone arrays are the basic audio sensors delivering the appropriate information from which is possible to determine the direction of sound source arrival. There are a lot of methods and algorithms proposed for effective microphone array information processing like time delay estimation, cross correlation, beamforming, etc. All of these methods need the extensive calculations difficult to realize in real time for practical applications in area of mobile robot speakers tracking, noise cancellation, video conferencing, etc. Some implementations with digital signal processors (DSP), field programmable gate arrays (FPGA) and other high speed circuits are proposed to decrease time of microphone array information processing. The goal in this article is to apply popular in computers GPU (Graphic Processing Units) and their appropriate programming platform CUDA (Computer Unified Device Architecture) for the purposes of real time microphone array information processing.

Key-Words: - Microphone Arrays, Sound Source Localization, Real Time Processing, Parralell Programming, GPU, CUDA

1 Introduction

Microphone arrays are very popular devices suitable to receive sound waves arriving from different directions [1]. They transform all of the received sound waves in appropriates output audio signals containing information necessary to calculate and determine the direction of arrival of the sound waves from the sound sources to the position of the microphone array. There are appropriate methods of processing these output microphone array signals extracting the information as angle for direction of sound arrival from sound source. The main drawback of all of these methods is the computational complexity leading to impossibility or difficulty of real time realization in practical cases like mobile robot motion control with voice commands [2], speaker's identification or recognition [3], video conferencing [4], noise cancelation [5], etc. To resolve this problem are proposed some solutions with digital signal processors (DSP) [6], field programmable gate arrays (FPGA) [7] or other usually specially designed devices. Most of them achieved real time processing of incoming from microphone array audio signals to determine direction of sound source of arrival, but at the cost of the complex and not always convenient for practical implementations. The development of power graphical processing units (GPU) [8] with corresponding facility of parallel programming platform CUDA (Computer Unified Device Architecture) [9], primarily dedicated to speed games and multimedia applications on desktops and laptops [10], [11] is the one of the main argument of the goal in this article to propose their use in the implementation of real time processing of audio signals from the microphone arrays.

2 The Basic Principles of Microphone Array Information Processing to Define the Calculation Problems

2.1 General Structure of Microphone Arrays

The structures of microphone arrays are different according both the number of microphones and their linear, planar or spatial positions and placements. In this article for definiteness is presented on Fig. 1 a general structure of microphone array. It is seen from Fig. 1 that the direction of sound waves arriving to the microphones $M_1, M_2 ....M_{N-1}, M_N$ is θ, if it is assumed that the sound source generate a plane sound wave and the distance between microphones is equal to d .

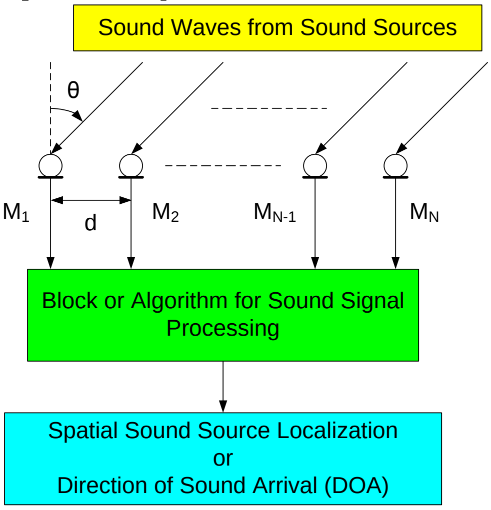

Fig.1. General structure of microphone array.

Therefore, the processing of all audio signals from microphones $M_1, M_2 ....M_{N-1}, M_N$ is presented on Fig. 1 as “Block or Algorithm for Sound Signal Processing”. The result from this processing is presented on Fig. 1 as input in the block named “Spatial Source Localization or Direction of Sound Arrival (DOA)” and must be consider as calculated direction of sound arrival θ. There are different approaches to realize calculation of direction of sound arrival θ: time delay estimation [12], cross correlation [13], convolution [14], beamforming [15], etc. In this article for definiteness is chosen to use a concrete binaural microphone array model [16] to demonstrate GPU unit and CUDA platform ability for real time microphone array information processing for determination direction of sound arrival θ.

2.2 Binaural Model of Microphone Array Using GPU Unit and CUDA Platform in Calculation of Direction of Sound Arrival

On Fig. 2 is shown the model of binaural microphone array proposed to use calculation of sound of arrival by means of GPU Unit and CUDA platform using right ***M1*** and the left ***M2*** microphones with the appropriate cones added to them [17]. The right ***M1*** and the left ***M2*** microphones received coming from sound source **Sound Waves** and transforming them in corresponding **Audio Signals.** The next block proposed to use the hardware (**GPU**) unit and software (**CUDA**) platform for calculation of sound arrival as the angle θ (the output of this block). The last block on Fig. 2 shows the possible ideas to use the calculated angle θ as necessary and useful information in most popular applications (for example: mobile robot motion control with voice commands, noise cancelation, videoconferencing, etc.) using direction of sound source arrival.

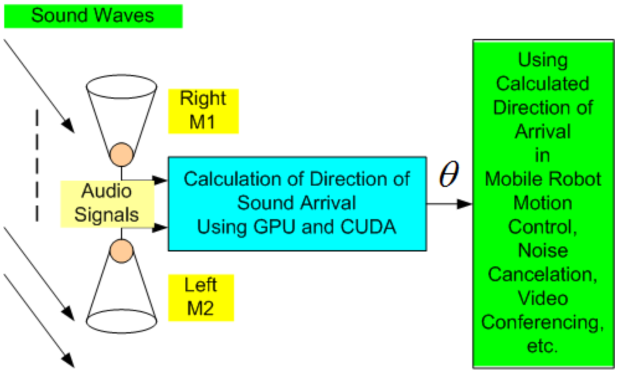

Fig.2. Binaural microphone array model in which is proposed to use calculation of sound of arrival by means of GPU Unit and CUDA platform.

The proposed on Fig. 2 calculation of sound arrival as the angle θ, applying as hardware (**GPU**) unit and as software (**CUDA**) platform, can be realized using different methods and algorithms. To test and demonstrate the ability of GPU unit and CUDA platform to real time microphone array information processing and for definiteness in the binaural microphone array model (Fig.2) is chosen to use a concrete computational algorithms (briefly presented in next section) in which the most time consuming operations in determination direction of sound arrival θ are prepared using GPU unit and CUDA platform.

3 Algorithm for direction of arrival calculation using GPU unit and CUDA platform

The proposed algorithm for direction of arrival calculation using GPU unit and CUDA platform is shown on Fig. 3. Each two current frames of audio signals s_1^{fr} and s_2^{fr} from the right ***M1*** and the left ***M2*** microphones are input in GPU unit memory (Fig. 3), where in the next block they are processed applying the basic operations like cross correlation [13], convolution [14] or other chosen operations to extract the necessary information for determination direction of sound arrival θ. The results from these operations are used in the next block (Fig. 3) to determine time difference $\tau_{1,2}^{fr}$ between the audio signals from two current frames of received by the right ***M1*** and the left ***M2*** microphones. Based on determined time difference $\tau_{1,2}^{fr}$ is calculated the desired angle θ^{fr} of sound source arrival, which can be output from GPU memory to be used in other applications like mobile robot motion control with voice commands from a speaking person, speaker's identification or recognition, video conferencing, noise cancelation, etc.

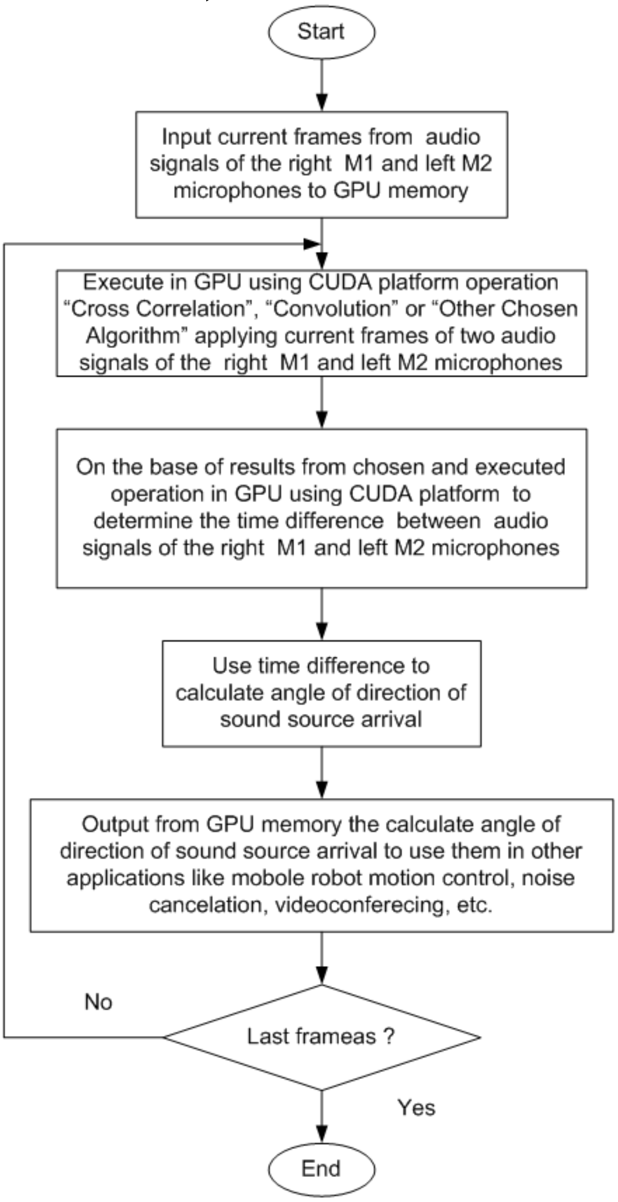

Fig.3. The proposed algorithm for direction of arrival calculation using GPU unit and CUDA platform.

The steps of proposed algorithm allow to determine the total calculation time T_{da}^{fr} of sound source arrival as follow:

$$T_{da}^{fr} = T_{in}^{fr} + T_{proc}^{fr} + T_{diff}^{fr} + T_{\theta}^{fr} + T_{res}^{fr}, \qquad (1)$$

where

$T_{in}^{fr}, T_{proc}^{fr}, T_{diff}^{fr}, T_{\theta}^{fr}$ and T_{res}^{fr} are respectively: the time for input current frames of audio signals s_1^{fr} and s_2^{fr}; the time for processing cross correlation, convolution or other chosen operations in GPU unit memory; the time difference $\tau_{1,2}^{fr}$ determination; time for calculation of angle θ^{fr} and the time for output the calculated value of angle θ^{fr} of sound source arrival. In equation (1) most of the values ($T_{in}^{fr}, T_{diff}^{fr}, T_{\theta}^{fr}, T_{res}^{fr}$) are small and can be ignored. Therefore, the total calculation time T_{da}^{fr} of sound source arrival greatly depend from time of processing T_{proc}^{fr}:

$$T_{proc}^{fr} >> T_{in}^{fr} + +T_{diff}^{fr} + T_{\theta}^{fr} + T_{res}^{fr} \qquad (2)$$

$$T_{da}^{fr} \approx T_{proc}^{fr} \qquad (3)$$

Equations (2) and (3) shown the importance of the execution time of processing T_{proc}^{fr}. Therefore it is proposed to calculate and use only this time of processing T_{proc}^{fr} as estimation of the whole execution time T_{da}^{fr}, when applying different methods like cross correlation, convolution or other chosen operations, in the algorithm on Fig.3 for determination direction of sound arrival θ. For example, if it is chosen to use method of cross correlation $s_{corr}^{fr}(k)$ between audio signals s_1^{fr} and s_2^{fr} of two current frames of received by the right ***M1*** and the left ***M2*** microphones for determination direction of sound arrival θ, then the following equation describe cross correlation in time domain:

$$s_{corr}^{fr}(k) = \sum_j s_1^{fr}(j) s_2^{fr}(k - j + 1), \qquad (4)$$

where

$k = 1,2,.......m + n + 1$;

$j = \max(1, k + 1 - n),......, \min(k, m)$;

m and n are the number of samples of audio signals s_1^{fr} and s_2^{fr}, respectively (usually $m = n$).

For $m = n$ equation (4) can be modified using the following iterative procedure:

$$s_{corr}^{fr}(1) = s_1^{fr}(1) s_2^{fr}(1)$$

$$s_{corr}^{fr}(2) = s_1^{fr}(1) s_2^{fr}(2) + s_1^{fr}(2) s_2^{fr}(1)$$

$$s_{corr}^{fr}(3) = s_1^{fr}(1) s_2^{fr}(3) + s_1^{fr}(2) s_2^{fr}(2) + s_1^{fr}(3) s_2^{fr}(1)..$$

$$s_{corr}^{fr}(n) = s_1^{fr}(1) s_2^{fr}(n) + s_1^{fr}(2) s_2^{fr}(n-1) + ...$$

$$.........s_1^{fr}(n) s_2^{fr}(1).......$$

$$s_{corr}^{fr}(2n-1) = s_1^{fr}(n) s_2^{fr}(n) \qquad (5)$$

From equations (4) and (5) follow the computation complexity, described as the number of operations O_T^{fr} for calculating in time domain the cross correlation $s_{corr}^{fr}(k)$ between two current frames of audio signals s_1^{fr} and s_2^{fr} from the right ***M1*** and the left ***M2*** microphones:

$O_T^{fr} = n^2$ (For the case $m = n$) (6)

The other way of calculate cross correlation between audio signals s_1^{fr} and s_2^{fr} in two current frames of received by the right ***M1*** and the left ***M2*** microphones is to calculate cross correlation in frequency domain. This possibility is based on the assumption to express the cross correlation in frequency domain of two sequences audio signals s_1^{fr} and s_2^{fr} as the multiplication of their Fourier transforms. Therefore, this is shown in Fig. 4 for the case of calculating cross correlation between the audio signals s_1^{fr} and s_2^{fr} with GPU unit and CUDA platform in frequency domain. The computational complexity for calculating cross correlation between the audio signals s_1^{fr} and s_2^{fr} with GPU unit and CUDA platform in frequency domain is the following:

$O_F^{fr} = n\log(n)$ (For the case $m = n$) (7)

The comparison of equations (6) and (7) lead the conclusion of smaller computational complexity calculating of cross correlation between audio signals s_1^{fr} and s_2^{fr} with GPU unit and CUDA platform in frequency domain. Therefore, it is chosen to use calculation of cross correlation in frequency domain for the purpose of real time microphone array information processing with GPU unit and CUDA programming platform. In the same way is possible to define the appropriate equations (not shown in this article for briefness) for number of operations for calculation time of processing T_{proc}^{fr} not only for the case of cross correlation, but also for the cases applying convolution or other chosen operations, in the algorithm on Fig.3 for determination direction of sound arrival θ. The practical determinations of time of processing T_{proc}^{fr}, executing different methods like cross correlation, convolution or other chosen operations, in the algorithm on Fig.3 for determination direction of sound arrival θ, are caried out in the next part of this article. The achieved experimental result for the tested different methods in determination direction of sound arrival θ are analysed and compared to confirm the ability of GPU unit and CUDA platform for real time microphone array information processing for determination direction of sound arrival θ.

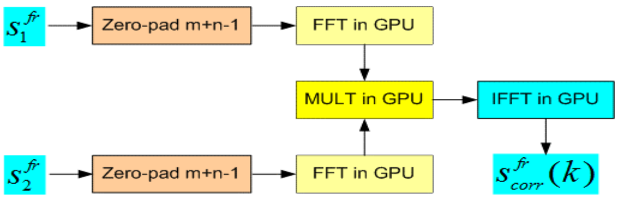

Fig.4. The case of calculating cross correlation between the audio signals s_1^{fr} and s_2^{fr} with GPU unit and CUDA platform in frequency domain.

4 Experimental Results

In the experiments are used the proposed on Fig. 2 binaural microphone array model in which the calculation of sound of arrival is made by means of GPU Unit and CUDA platform. As microphone array is used MEMS microphone evaluation board STEVAL-MKI126V2 based on STA321MPL and MP34DB01 [18]. The experiments to test the ability of different models of GPU units of real time audio signals processing from the microphone arrays are carried out with two popular for desktops and laptops GPU: NVIDIA GeForce GTX560 [19] and NVIDIA GeForce GTX920M [20], respectively. The experiments are carried out in the following way according to the algorithm steps on Fig. 3:

- continuous (audio steaming) input in computer the audio signals, received from the right ***M1*** and the left ***M2*** microphones of binaural microphone array (parts of these audio signals are shown as time diagrams on Fig. 5);
- arrangement to input also continuous (real time) in GPU memory each two current frames s_1^{fr} and s_2^{fr} from received audio signals, corresponding to the right ***M1*** and the left ***M2*** microphones;
- calculating the time T_{sf1} and T_{sf1} of current frames s_1^{fr} and s_2^{fr} using the following equation:

$T_{sf1} = N_{sf1} * f_{sf1}$ (7)

$T_{sf2} = N_{sf2} * f_{sf2}$, (8)

where

T_{sf1} and T_{sf2} are respectively the times T_{sf1} and T_{sf1} of current frames s_1^{fr} and s_2^{fr};

N_{sf1} and N_{sf2} - respectively the number of samples in current frames s_1^{fr} and s_2^{fr};

f_{sf1} and f_{sf2} - respectively the frequency of samples for the current frames s_1^{fr} and s_2^{fr};

it is assuming that the following conditions are satisfied:

$T_{sf1} = T_{sf2} = T_{sf}$, $N_{sf1} = N_{sf2} = N_{sf}$ and

$$f_{sf1} = f_{sf2} = f_{sf} \quad (9)$$

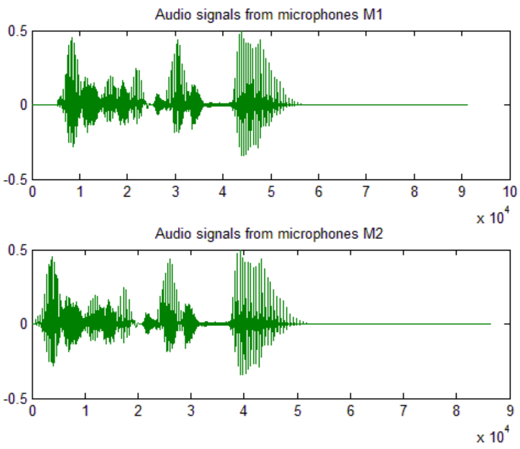

Fig.5. Time diagrams of audio signals as two parts, received continuously as auduo streams from the right ***M1*** and the left ***M2*** microphones of binaural microphone array.

- using CUDA Toolkit [9] to execute the operations of algorithm shown in Fig. 4 for the cases of calculating cross correlation $s^{fr}_{corr}(k)$ or convolution $s^{fr}_{conv}(k)$ with GPU unit and CUDA platform in frequency domain;
- calculating angle θ^{fr} of sound source arrival using previous determined cross correlation $s^{fr}_{corr}(k)$ or convolution $s^{fr}_{conv}(k)$;
- calculating execution time T^{GPU}_{corr} or T^{GPU}_{conv} (equal of processing time T^{fr}_{proc} in equation 3) for cross correlation $s^{fr}_{corr}(k)$ or convolution $s^{fr}_{conv}(k)$ applying GPU unit and CUDA Toolkit;
- repeat all of above mentioned steps until the end of audio signals s^{fr}_1 and s^{fr}_2.

The same steps listed above are carried out also in a Matlab program [21] and Simulink [22]. For this case also is calculating the execution time T^{Matlab}_{corr} or T^{Matlab}_{conv} (equal of processing time T^{fr}_{proc} in equation 3) of cross correlation $s^{fr}_{corr}(k)$ or convolution $s^{fr}_{conv}(k)$ between each two frames of the audio signals s^{fr}_1 and s^{fr}_2, but applying Matlab. The execution times T^{GPU}_{corr}, T^{GPU}_{conv} and T^{Matlab}_{corr}, T^{Matlab}_{conv} achieved both applying GPU with CUDA Toolkit and Matlab program are presented in Table 1.

Table1. Comparison of execution times T^{GPU}_{corr}, T^{GPU}_{conv} and T^{Matlab}_{corr}, T^{Matlab}_{conv} achieved both applying GPU with CUDA Toolkit and Matlab program

Calculating angle θ^{fr} of sound source arrival using:	Time T^{fr}_{proc} in seconds for different methods of execution:	
	Cross correlation	Convolution
NVIDIA GPU GeForce GTX560 with CUDA	T^{GPU}_{corr} 0.0167	T^{GPU}_{conv} 0.0123
NVIDIA GPU GeForce GTX920M with CUDA	T^{GPU}_{corr} 0.0092	T^{GPU}_{conv} 0.0106
Program in MATLAB	T^{Matlab}_{corr} 0.239	T^{Matlab}_{conv} 0.205

5 Conclusion

Comparing the values of different execution times T^{GPU}_{corr}, T^{GPU}_{conv} and T^{Matlab}_{corr}, T^{Matlab}_{conv} in Table 1 achieved applying GPU with CUDA Toolkit and Matlab program lead to the following results:
- execution times T^{Matlab}_{corr}, T^{Matlab}_{conv} using Matlab program are significantly greater than the execution times T^{GPU}_{corr}, T^{GPU}_{conv} achieved applying GPU with CUDA Toolkit;
- execution times T^{GPU}_{corr}, T^{GPU}_{conv} using NVIDIA GeForce GTX560 [19] are a little greater than the execution times T^{GPU}_{corr}, T^{GPU}_{conv} applying NVIDIA GeForce GTX920M [20];
- execution times T^{GPU}_{corr}, T^{GPU}_{conv} using NVIDIA GeForce GTX560 [19] and NVIDIA GeForce GTX920M [20] satisfy the condition of real time microphone array information processing because they are smaller in comparison of $T_{sf1} = T_{sf2} = T_{sf}$;

$$T^{GPU}_{corr} \text{ and } T^{GPU}_{conv} << T_{sf1} = T_{sf2} = T_{sf}\,, \quad (10)$$

For example if $N_{sf1} = N_{sf2} = N_{sf} = 256$ samples in the current frames s^{fr}_1 and s^{fr}_2 and the sample frequency of audio signals in the current frames s^{fr}_1 and s^{fr}_2 is $f_{sf1} = f_{sf2} = f_{sf} = 8000Hz$, therefore: $T_{sf1} = T_{sf2} = T_{sf} = 0.032s$, which is greater than the values of T^{GPU}_{corr} and T^{GPU}_{conv} presented in Table1, when using NVIDIA GeForce GTX560 [19] and NVIDIA GeForce GTX920M [20] in calculation of angle θ^{fr} of sound source arrival.

This verify the ability of NVIDIA GeForce GTX560 or GTX920M GPU (Graphic Processing Units) with their appropriate programming platform CUDA (Computer Unified Device Architecture) for real time microphone array information processing and calculating the angle θ^{fr} of sound source arrival, which is the goal of this article.

6 Acknowledgment

This paper was supported by Technical University – Sofia inner program for research projects under Contract № 161ZF0003-07: "Methods and tools for the study of audio signals".

References:

[1] M. Brandstein, D. Ward, *Microphone Arrays Signal Processing Techniques and Applications*, Springer-Verlag, 2001.

[2] S. Pleshkova, Al. Bekiarski, Sh. Sehati Dehkharghani, K. Peeva, *Perception of Audio Visual Information for Mobile Robot Motion Control Systems*. Chapter 6, Computer Vision in Control Systems-2, Springer Series: Intelligent Systems Reference Library, Edited by M. Favorskaya, J. Lakhmi, Vol. 75, 2015, pp. 135-167.

[3] J. Stokes, Speaker Identification Using a Microphone Array and a Joint HMM with Speech Spectrum and Angle of Arrival. *IEEE International Conference on Multimedia and Expo*, Toronto, 9-12 July 2006, pp.1381-1384.

[4] HD Video Conferencing Solutions. Polycom Microphone Array. *Polycom, Inc.*,

[5] Iv. Tashev, Al. Acero, Microphone Array Post-Processor Using Instantaneous Direction of Arrival. *Proceedings of International Workshop on Acoustic, Echo and Noise Control IWAENC*, Paris, 12-14 September 2006 pp. 1-4.

[6] X. Zhang, J. Huang, E. Song, H. Liu, B. Li, X. Yuan, Design of Small MEMS Microphone Array Systems for Direction Finding of Outdoors Moving Vehicles, *Sensors*, 2014, Vol.14, pp. 4384-4398.

[7] J. Tiete, F. Dominguez, B. Silva, L. Segers, Kr. Steenhaut, A. Touhafi, Sound Compass: A Distributed MEMS Microphone Array-Based Sensor for Sound Source Localization, *Sensors*, 2014, Vol.14, pp. 1918-1949.

[8] http://www.geforce.com/hardware/gpus

[9] https://developer.nvidia.com/cuda-toolkit

[10] Sz. Sandor, Segmentation of colon tissue sample images using multiple graphics accelerators. *Computers in Biology and Medicine* (2014), Vol. 51, pp. 93-103

[11] G. Kertész, S. Szénási, Z. Vámossy, Parallelization Methods of the Template Matching Method on Graphics Accelerators. *16th IEEE International Symposium on Computational Intelligence and Informatics*, Budapest, 2015, pp. 161-164.

[12] J. Chen, Y. Huang, J. Benesty, *Time Delay Estimation*. Chapter 8, Audio Signal Processing for Next-Generation Multimedia Communication Systems, Springer, Edited by Y. Huang, J. Benesty, 2004, pp. 197-227.

[13] U. Klein, Tr. Quốc Võ, Direction-of-arrival estimation using a microphone array with the multichannel cross-correlation method. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2-15 December 2012, pp. 251 – 256.

[14] M. Hossain, M. Rashid, M. Islam, M. Ali, Convolution constrained robust beamforming techniques for broadband microphone array. *International Conference on Electrical & Electronic Engineering (ICEEE)*, 4 - 6 November 2015, pp. 277 – 280.

[15] J. Bitzer, K. Kammeyer, K. Simmer, An alternative implementation of the superdirective beamformer. *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, October 1999, pp. 991–994.

[16] I. Merks, B. Xu, T.Zhang, Design of a high order binaural microphone array for hearing aids using a rigid spherical model. *International Conference on Acoustics, Speech, and Signal Processing ICASSP*, 2014, pp. 3678 - 3682.

[17] T. Dau, D. Püschel, A. Kohlrausch, A quantitative model of the effective signal processing in the auditory system. *J. Acoust. Soc. Am.*, vol. 99, no. 6, 1996, pp. 3615-3622.

[18] http://www.st.com/content/st_com/en/products/evaluation-tools/audio-ic-eval-boards/steval-mki126v2.html

[19] http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-560

[20] http://www.notebookcheck.net/NVIDIA-GeForce-920M.html

[21] http://www.mathworks.com/products/new_products/release2013a.html

[22] http://www.mathworks.com/products/simulink/