Parallel Signal Processing Based-On Graphics Processing Units

1st Ilyos Khujayorov
Computer Systems
Tashkent University of Information technologies named after Muhammad Al-Khwarizmi
Tashkent, Uzbekistan
i.khujayorov@tuit.uz

2th Manon Ochilov
Computer Systems
Tashkent University of Information technologies named after Muhammad Al-Khwarizmi
Tashkent, Uzbekistan
ochilov.mannon@mail.ru

***Abstract*— This article analyzes parallel implementation of GPU in the process of generating speech spectrogram images, which is most commonly used as a parameter in speech recognition. The convenience of generating and recognizing spectrogram images from speech signals is that they are rich in character sets compared to other speech recognition methods. Therefore, most modern speech recognition systems use spectrograms. Based on the results of the experiment, it can be seen that the production of spectrogram images from speech signals in the graphics processor was faster than the central processor. This will improve the quality of speech recognition and identification in real time.**

Keywords—speech, recognition, frequency, spectrums, spectrogram, GPU, CUDA, acceleration, FFT, DCT, Wavelet transform.

I. Introduction

Speech processing tasks relate to classical digital signal processing algorithms, including noise removal, extraction of useful components, segmentation and subsequent spectral analysis in the selected base system. The following are algorithms for the parametric representation of a speech signal in the form of the fundamental frequency, spectral formants, spectrograms, the allocation of vowels and consonants. The basic procedures exist for the acoustic treatment path. At the final stage, intelligent identification and recognition algorithms are used.

In real systems of registration, processing, parametric representation and voice transmission, an increase in processing speed may be required firstly, for an acoustic stage of processing. Specifically, in the first stage, to the translation of the speech signal into a phonogram of recording from a microphone. With generally accepted speech sampling frequencies of 16/22 KHz, the amount of processed information of only one word reaches several tens of thousands of bytes.

Algorithms used for the parametric representations reduce repeatedly the images of speech and prepare them for recognition and decision making as well. The signal is conversed into the parametric representation more often realized spectral methods: Fourier transform cepstral and wavelet analysis.

The properties of spectral analysis algorithms make it possible to effectively use parallel computing methods [1,2].

At the final stage of processing, mining algorithms based on probabilistic and recursive processing methods lend themselves well to parallelization in the GPU [3-7] and give the expected gain in processing speed.

At present, taking into account the most of modern sound recognition systems work in real time and performance-demanding many algorithms used in them put high demands data processing and transmission speed in these systems [3].

Thus, the greatest effect of accelerating calculations in real-time systems that implement the tasks of digital processing of speech signals is achieved at all the above stages.

It is desirable to introduce hybrid architectures (CPU + GPU) parallel computing systems that quickly implement data processing and data transfer algorithms to effectively implement in such requirements exist in these systems. Stream processing in the tasks of the analysis of speech signals [1,2,6,7] on graphic processors should be considered as a new technology, including the following advantages in parallel calculations:

– GPU memory is optimized for maximum throughput, which speeds up the loading of a data stream;
– most of the GPU transistors are designed for computing, not for controlling program execution;
– During querying memory, due to pipelining data processing, no suspension of calculations are considered.

This article analyzes parallel implementation of GPU in the process of generating speech spectrogram images, which is most commonly used as a parameter in speech recognition.

II. The Spectrogram

Parameterization plays an important role in solving speech-processing problems owing to the separation of the character vector from the speech signal. The character vector supports to distinguish one speech from another. This process makes the unprocessed speech signal much more compact than the incoming signal, stable and reversible. It is obvious that in most modern speech recognition systems today, spectral images generated from speech signals are widely used [8-11].

Spectrogram is a time-dependent indicative image of the signal strength and density. It is presented in a two-dimensional graph. The horizontal axis (X-axis) represents the time (Time (ms)) and the vertical axis (Y-axis) indicates the frequency (Frequency). The real-time amplitude frequency characteristics are determined by color intensity. The process of generating the spectogram is shown in Figure 1.

978-1-7281-2564-0/19/$31.00 ©2019 IEEE

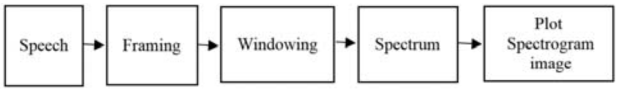

Fig. 1. The process of creating the spectrogram from speech signal

Looking at Figure 1 above, the incoming signal is transformed into a spectrogram image after several steps.

In the first step, the signal is divided into frames. The frame size depends on the discrete frequency of the signal. Typical processing of speech signal in 10–30 ms incision is accepted.

In the second step, each frame is passed through the filter window. When the incoming signal is divided into frames, interruptions occur within the frames. When calculating the spectrum of such a signal, the amplitude-frequency characteristic can have an infinite form, convergence will be unsatisfactory. To eliminate this effect, special weighing windows are used. They smoothly reduce to a function near the edges of the analyzed area. The signal section selected for analysis is multiplied by a weight window, which eliminates function discontinuities when smoothing this section of the speech signal. There are many weight windows, all of them have a very similar shape and largely eliminate the considered spectrum distortions.

The third step uses one of the spectral transform algorithms for each frame in the window. This process requires significant computing resources, as the amount of incoming data rises as the timing of the computations grows non-linearly. Each of the spectral transform algorithms has its own characteristics, which differentiate the visual state of the signal. The most commonly used spectral transform algorithms in the production of spectrogram images are Fast Fourier Transform (FFT) [19], Discrete Cosine Transform (DCT) [20], and Wavelet transform [21]. We describe the features of these algebras in generating spectrogram images.

A. *Using FFT*

FFT is widely used to produce spectrogram images. Because spectrum acquisition using FFT is fast and provides compact and accurate information about speech signal.

B. *Using DCT*

Discrete cosine transform (DCT) expresses the sequence of a finite number of data points as a sum of cosine functions that oscillate at different frequencies. It turns out that the cosine functions are much more efficient, since for approximation the signal [12]. In particular, a DCT is a Fourier transform, similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs about twice as long and work on real data with uniform symmetry. For the efficient analysis of speech signals based on spectrogram images, the image quality plays an important role. Spectrogram images generated using DCT have a higher accuracy than other substitution alloys, in particular DFT. Spectrograms generated using DCT provide more information about the sound (about the magnitude of sound) than the spectrograms generated by DFT.

C. *Using Wavelet transform.*

Wavelets were proposed by mathematicians and are essentially new mathematical concepts and objects [13]. Especially importance of the possibility in principle of wavelets to representation are unsteady signals. All many specialists in digital signal processing are convinced that the Fourier transforms in the classical form do not provide the necessary accuracy for the representation of non-stationary signals, in particular, to which speech signals belong. The function graph looks like wave-like oscillations with an amplitude decreasing to zero far from the origin. However, this is a particular definition - in the general case, the signal analysis is performed in the plane of wavelet coefficients (scale - time - level) (Scale-Time-Amplitude). Wavelet coefficients are determined by the integral transformation of the signal. The obtained wavelet spectrograms fundamentally differ from ordinary Fourier spectra in that they give a clear reference of the spectrum of various signal features to time [26].

In the last step based on spectrum coefficients from frames generated spectrogram image.

The advantage of the spectrogram is that it allows you to specify high and low signal frequencies over time. This provides a good pattern template for training them on the neural network. Another advantage of the spectrogram is that it is rich in character sets so that we can use image recognition techniques to isolate these characters and combine two areas of research. As we know, nowadays neural networks are the basis of modern speech recognition systems. Neural networks form a knowledge base based on large-scale data analysis and present the end result of recognition.

The one of the most effective approaches are the creating a character set based on spectrogram images from speech as well as the creating known system based on them. Many studies have been carried out by scientific researchers on this approach [14 - 16]. Creating a knowledge base based on spectrogram images requires a lot of resources from the computing system. Because, if the larger the number of character sets per class (in our case, the spectrogram images) the more the performance of the familiar system increases. Therefore, generating thousands of spectrogram images requires large computational resources.

III. SPECIFICITIES OF PARALLEL PROGRAMMING ON CUDA

The use of modern GPUs allows you to get multiple productivity growth in solving a number of scientific and technical problems. However, to achieve the best results, it is necessary to take into account a number of specificities [17–18]:

- The graphics processor (GPU) consists of several multiprocessors, which, in turn, consist of cores. Each core simultaneously executes 32 threads (warp). For example, the NVidia GeForce GTX 480 consists of 15 $\times$ 32 = 480 cores and can simultaneously execute up to 15,360 lightweight threads. Streams are combined into blocks and block grids. Each stream has its identifying coordinates;
- maximum performance can be achieved when performing the same type of actions on a large number of processed data units;
- the memory architecture has a complex organization: global memory (voluminous, but slow), local memory, shared memory (fast), constant memory, etc.

- memory access specificities: for maximum throughput, all memory requests should be aligned.

A detailed description and application specificities of graphics processors can be found in the respective manuals [17].

IV. APPLICATION OF CUDA TECHNOLOGY FOR SPEECH PROCESSING TASKS

The choice of the discrete frequency is important in speech recognition. If the sampling frequency is small, the critical parameters will be lost, and the larger ones will result in unnecessary parameters. Usually speech recognition is used signals generated at the discretization frequency at 8 kHz, 11,025 KHz, 16 KHz, and 22.05 KHz. If the signal's frame length is 25 ms, the length of the frame will be 200 units in 8 kHz the discretionary frequency, 275 units in 11 kHz, 400 units in 16 kHz, 550 units in 16 kHz. One of the above frame lengths is selected to generate a spectrogram image from the speech signal. It is often based on the level of signal length 2 in spectral switching algorithms for qualitative generation of spectrogram images. In most cases, frames with 256, 512, and 1024 values are used to produce spectral images qualitatively and for the signal length in spectral transform algorithms are based on the power of 2 (this is a condition of most applied spectral analysis programs in Fourier bases).

In order to generate spectrogram images in the GPU, we first need to know the structure of the program execution in CUDA. Due to the versatility of the CUDA architecture, the algorithms executed by this platform can have various rather complex structures. The simplest scheme, reflecting the optimal sequence of loading computational data streams from memory, is as follows.

1. The CPU loads the necessary data into the video memory, GPU registers and into the memory for constants.
2. The central processor runs the computing core executable on the GPU. The sizes of the grids into which the “bundles” and computational flows will be organized as the programmer indicates them.
3. At the beginning of the execution of the computational core, each thread determines in its local and (or) global number, according to which it will process its part of the initial data prescribed by the programmer.
4. Computational streams in parallel mode copy the data from video memory to shared memory and to registers by that this data can be processed faster.
5. If the data copied to shared memory will then be shared different computing threads, then synchronization is required (operator the __sync threads ()).
6. In parallel mode, the calculations provided by the algorithm are performed, the results are written to shared memory or immediately to video memory, if the results are final.
7. It may be necessary for all computational threads to complete the current stage of the calculation before moving any of them to the next stage (for example, loading new data into shared memory). To do this, synchronize the computational threads with (operator the __sync threads).
8. Next, the transition to the next stage of the algorithm or the completion of the computing core.

Based on the above, we perform the following sequence to generate speech spectrograms of speech in the GPU:

1) The signal read (.wav) is executed in the CPU.
2) The number of blocks and streams to be used is set (the kernel function parameters are set) and the signal is copied to the global memory of the GPU.
3) A kernel that performs instantaneous generation of spectrogram images is launched. (the core is a function of the graphics processor).
4) When the computation is complete, the results are copied from the graphics processor memory to the CPU memory. (for further work on the results obtained).
5) The graphics processor will free up space for calculations.

V. TESTING

To test the algorithm, a computer of the following configuration was used:

1. The central processor – Intel Core i5-2310 2,90 ГГц.
2. The number of core – 4.
3. RAM –8 Gb RAM DDR3.
4. Video card – NVidia GeForce GTX 555 Ti (2 Gb, 192 CUDA core).
5. CUDA version – 7.5.18
6. The compiler Microsoft Visual Studio 2010 (vectorization SSE2).
7. Data Type – double (64 bits).

In the experiments, we examined the generation of DCT, FFT, and Haar Wavelet spectrogram images in the GPU. The results of the application of DCT-based spectrogram images in sequential and parallel variants are presented in Table 1.

TABLE I. TEST RESULTS OF CONSTRUCTION FOR SPECTROGRAMS BASED ON DCT IN CONSISTENT AND PARALLEL VERSIONS OF THE PROGRAM.

Frame size	Sequential option on CPU (ms)	GPU time (ms)	Acceleration
8	29.11	3.29	8.8
16	56.76	4.39	12.92
32	120.3	8.44	14.25
64	231.87	16.98	13.66
128	467.59	34.53	13.54
256	**961.36**	**69.03**	**13.92**
512	**2108.58**	**143.35**	**14.7**
1024	**6991.33**	**299.32**	**23.36**
2048	23090.21	555.75	41.56
4096	46922.48	1159.54	40.47

As can be seen from Table 1, the frame size increased by 14 times at 256, 15 times at 512, and 23 times by 1024. The transfer time to the GPU memory is included in the total time. Acceleration performance in the FFT and Haar Wavelet displacement spectrogram yielded almost the same results as the DCT. This is illustrated by the graph in Figure 2.

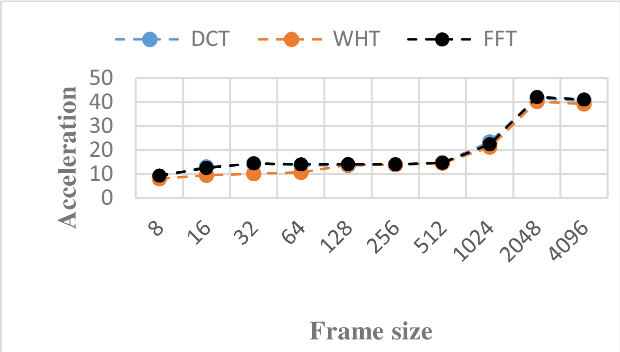

Fig 2. Speed graphs for generating FFT, DCT, and Haar Wavelet displacement spectrogram images in GPUs.

CONCLUSION

The convenience of generating and recognizing spectrogram images from speech signals is that they are rich in character sets compared to other speech recognition methods. Therefore, most modern speech recognition systems use spectrograms. Based on the results of the experiment, it can be seen that the production of spectrogram images from speech signals in the graphics processor was faster than the central processor. This will improve the quality of speech recognition and identification in real time.

The resource-required stage in generating a spectrogram image is to calculate the spectral values from the signal frames. As we know, spectral switching algorithms are built on the basis of matrix and vector operations. Therefore, the parallel processing of these algorithms in the GPU is successful.

REFERENCES

[1] M.Musaev, U.Berdanov, M.Rahimov, Shukurov K, "Parallel algorithms for acoustic processing of speech signals" International Conference on Signal and Image Processing (ICSIP 2016). China during August 13-15. DOI: 10.1109/SIPROCESS.2016.7888297

[2] M.M.Musaev, M.S.Kardashev, "Spectrum anall'sis of signals on multicore processors", Scientific Joumal of Digital Signal Processing, no.2, pp.82-86, Moscow, 2014.

[3] Pavel Zatsepin, Denis Gefke. NVIDIA CUDA application to train and decode the Hidden Markov Models. Bulletin of the Tomsk Polytechnic University.Vol 324 No 5 (2014). pp.115-121.

[4] Gallo, E.; Tsingos, N.; "Efficient 3D Audio Processing with the GPU", 2004, .In GP2, ACM Workshop on General Purpose Computing on Graphics Processors.

[5] Panteleev A. Digital signal processing on modern graphics processors. 15th International Conference "Digital Signal Processing and its Application - DSPA'2013". pp. 68-75.

[6] J.B. SrivastavaR. PandeyJ. Jain. Implementation of digital signal processing algorithm in general purpose graphics processing unit (GPGPU). International Journal of Innovative Research in Computer and Communication Engineering. Vol. 1, Issue 4, June 2013.

[7] Jake Gunther, Hyrum Gunther, T.K. Moon. GPU Acceleration of DSP for Communication Receivers. Proceedings of the GNU Radio Conference, 2(1), 2017, Pmc:pmc5695887.

[8] D. Polap, M. Woźniak. Image approach to voice recognition. 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings. 2018. pp.1-7.

[9] Lonce Wyse. Audio Spectrogram Representations for Processing with Convolutional Neural Networks. Published 2017 in ArXiv.org.

[10] M Ahmadi, N J Bailey, B S Hoyle. Phoneme recognition using speech image (spectrogram). Published in IEEE: Proceedings of Third International Conference on Signal Processing (ICSP'96). DOI: 10.1109/ICSIGP.1996.567353.

[11] Q. T. Nguyen et al., "Speech classification using sift features on spectrogram images," Vietnam Journal of Computer Science, vol. 3, no. 4, pp. 247–257, 2016.

[12] Rohini R. Mergu and Shantanu K.Dixit. A new paradigm for plotting spectrogram. Journal of Information Systems and Communication.Volume 3, Issue 1, 2012, pp.-158-161.

[13] Chui, Charles K. (1992). An Introduction to Wavelets. San Diego: Academic Press. ISBN 0-12-174584-8.

[14] Longhao Yuan, Jianting Cao. Patients' EEG Data Analysis via Spectrogram image with a Convolution Neural Network. Conference: International Conference on Intelligent Decision Technologies. DOI: 10.1007/978-3-319-59421-7_2.

[15] Venkatesh Boddapati, Andrej Petef, Jim Rasmusson, Lars Lundberg. Classifying environmental sounds using image recognition networks. December 2017. Procedia Computer Science 112:2048-2056. DOI: 10.1016/j.procs.2017.08.250.

[16] Dennis, J., Tran, H. D., & Li, H. Spectrogram Image Feature for Sound Event Classification in Mismatched Conditions. IEEE Signal Processing Letters, 18(2), 130–133. doi:10.1109/lsp.2010.2100380.

[17] A.V. Boreskov , A.A. Harlamov. The basics of working with CUDA technology (Russian Edition) (Russian). ISBN-10: 5519561575. Paperback – March 10, 2015.

[18] Jason Sanders, Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming. Publisher: Addison-Wesley Professional; 1 edition (July 29, 2010). 320 pages.