The Technology of Parallel Processing on Multicore Processors

Musaev Muhammadjon Mahmudovich and Berdanov Ulug'bek Abdumurodovich
Computer Engineering Faculty, TUIT, Tashkent, Uzbekistan
Email: mm.musaev@rambler.ru, ulug-907@mail.ru

***Abstract*—This paper discusses the technologies of parallel processing of signals and images on dual-core and quad-core processors. The issues of accelerating the computation by the parallel vector-matrix organization procedures when performing spectral analysis algorithms in different basic systems Fourier and wavelet transform are consider. The techniques used for parallel computation on multi-core processors at the level of the algorithms are considered and highlighted features implementation of a parallel version of the FFT algorithm on sites that consume at the implementation of algorithms for the highest amount of CPU time. In the paper presents the tools that were used by the authors during the experiments are given. The use of these funds provided to simplify writing multi-threaded applications and provided an opportunity to objectively assess the resulting acceleration. For the implementation of the pilot studies, the authors created a software package which includes direct and inverse spectral transforms used in all basic systems. The results showed that multi-core processors in the processing of signals and images partition in too large parts do not allow the processors to load evenly and to achieve a minimum computation time and splitting into smaller parts increases the share of unproductive expenditure.**

***Index Terms*—signal, images, multicore processors, parallel programming, spectral transformation**

I. Introduction

Expanding the scope of telecommunication services, the increase of amount transmission/reception and processing of audio, voice and image signals require a further increase in computing speed. One way is to use processors with multi-core cores processing, with this approach the parallel processing at the application level requires a major effort. Successful implementation of the algorithm, a significant increase in performance multi-core processors can only be achieved with consideration of possibilities of numerical methods, the successful distribution of subtasks, the professional use of concurrent constructs in programming languages and the effective use of tools. Using of computing potential of multicore processors can only be performed by division of performing calculation on independent parts and organization of implementation of each part on the different cores [1].

Manuscript received March 3, 2015; revised July 15, 2015.

Currently, multi-core processors are not used efficiently enough. Virtually all n-core processors do not make calculations in n times faster than single-core, although the increase in speed is significant. The magnitude of the acceleration computing significantly depends on the type of application.

One of such approaches is a digital signal and image processing, especially spectral analysis methods Modern technology of voice, audio and video data processing and transmission is based largely on the spectral analysis methods. Numerical methods based on the spectral analysis, at use of multi-core processors should be designed as systems of parallel and interacting processes that allow execution on the independent computing cores. Used algorithmic languages and system software must ensure the creation of parallel programs, synchronization and mutual exclusion organize asynchronous processes.

II. Parallel Processing Technology

Consider the process technology of preparation and execution of programs of parallel processing for multi-core processors. At the initial stage the numerical method which is the basis of the algorithm and allows for the possibility of separation processing tasks on parallel functioning subtasks, is analyzed. Numerical methods used in the problems of compression and speech synthesis, recognition and identification of images, widely used various two-dimensional unitary transformations: Fourier transform, discrete cosine transform, Haar functions of the system, Hadamard functions of the system, wavelet functions. On the basis of these transformations are widely used procedures such signal and image processing, as the convolution of the sequences, interpolation/decimation, the calculation of the spectrum, non-linear processing, linear prediction. More complex processing tasks are a combination of these simple procedures with the addition of traditional arithmetic-logic operations. In turn, the basis of these procedures and techniques are algorithms processing of long sequences: the accumulation of sums of pairwise multiplications, multiplication of vectors and matrixes, a cyclic rearrangement.

At the second stage of implementation the development of algorithms based on numerical methods is performed. From the application developer is required to focus on the distribution of subtasks with information communication between processor cores. One of the most

doi: 10.18178/ijsps.4.3.252-257

difficult problems is to transfer data (initial, intermediate and final) between tasks, and thus to each processor core. Time delays in the communication between the cores can affect the speed of parallel computing. The process of creating a parallel algorithm can be divided into the following steps.

1) Decomposition - analysis of the original task with a view to its division into separate subtasks and fragments of data.

2) Communication engineering required for sending the initial, intermediate and final data, release of the information dependencies between tasks.

3) The distribution of subtasks to each processor core, which means planning calculations.

At the third stage of realization, after development of algorithm it is necessary to write the parallel programs. Many algorithms for signal and image processing can be expressed in terms of vector operations, so the including of vector operations in the language tools allows much easier to write many fragments of software. For example, the addition of two vectors is not necessary to increase the index commands, checking the condition out of the cycle, the transition at the beginning of the cycle body.

Thus, the stream processing in signal and image processing tasks should be considered as a technology, which includes the following elements of the preparation and execution of programs [2]:

- Analysis of the numerical method and corresponding algorithm for creating opportunities for independent processing threads;
- Choice of effective language means writing programs;
- Identification and analysis fragments in algorithms with a maximum consumption of CPU resources;
- Development of technologies forming flows using modern means of parallelization;
- Using effective tools in the software implementation parallel versions of algorithms;
- Evaluation of the effectiveness of the developed parallel solutions using special performance analyzer software.

Algorithms applications consist of both parallel and series of sold units. From the well-known law Amdahl follows that increasing the number of processor cores affects only the parallel part of the program. It follows the first consequence Amdahl's Law: reducing the sequential part by increasing the parallel part is more important than increasing the number of processor cores. Particularly noteworthy are the overhead of the actual cost of the operating system and thread synchronization.

Gustafson law gives the formula for acceleration scalable parallel programs:

$$acceleration = N + (1 - N) \cdot K$$

In this formula, N - number of processor cores, K - the ratio of the execution time consecutive part of the program to the total time of execution of the program. This law models the linear acceleration, which depends on the number of processor elements, and represents more real benefits from multicore processors.

When linear processing for various one-dimensional (signals) and two-dimensional (image) of unitary transformations are the main vector multiplication input samples by the matrix of basic functions. In the spectral processing tasks are dominated by the same types of actions on the input portions of data: accumulation procedures sums of pairwise multiplications, cyclic rearrangement operations, the minimum information dependence between the portion of data and the same size of the portions. These features make it possible to conclude that in general this spectral analysis algorithms in parallel procedures prevail, and the algorithms are included in the number of highly scalable and efficiently implemented multicore processors. Scalability is a property of parallel programs with an increasing number of processor cores to show acceleration of calculations.

Next will be considered algorithms of spectral transformations in various basic systems allow the use of different approaches to the formation of independent execution threads for parallel execution of them. When checking accelerate calculations in this work to perform computational processes used first one processor (time-sharing) and then compared with the data obtained correctness and speed of the separation of subtasks on multiple processors. As a hardware platform were used dual-core (for work in two flows) and quad-core (for work in four flows) processors Intel Core.

Convenient option programs are to create a parallel version of the sequential algorithm. The conversion of the sequential algorithm in to the parallel can be as finding a large number of repetitive operations on independent data, as well as changes in the algorithmic structure, the search for other approaches to the problem. Techniques used for parallel computing problems in signal and image processing, include:

- Parallel cycles, iteration space which is divided by the number of processing threads;
- Reduction algorithms, that means, execution of a set of transactions with the accumulation of total sum, work or other functions;
- Allocation processes (tasks) that can be executed simultaneously;
- Recursive calls a method within itself the appointment of different computing threads to carry them out;
- The use of parallel patterns of matrix-vector operations.

The effectiveness of the developed parallel algorithm depends on the software implementation of the medium: the parameters of CPU, mechanisms for creating execution threads in the operating system, the number of threads. The maximum acceleration of parallel algorithms always achieved when execution is the computing threads, the number of which is provided by the specification of the processor.

Parallelization of computational procedures in spectral methods described in this paper, we consider the example of the algorithm fast Fourier transform (FFT). Similar threading methodology used for the other five spectral methods: discrete cosine transform (DCT), Haar transform and Hadamard-Walsh, Haar and wavelet Daubechies transform.

This paper considers the implementation of the algorithm Cooley-Tukey [3], where the number of samples of a signal is a degree of two. Original vector signal sampling is divided into two equal halves vector, each again divided into two parts and on each of the newly acquired sets FFT is performed. Calculations in the implementation of the parallel version performed independently.

This algorithm is parallelized based on the following features:

- Input and output vectors are viewed as two-dimensional tables, rows and columns are processed independently in different streams;
- Within a single stream FFT computes the local data lines and columns.

In this case, the real source of the vector display in the two-dimensional array does not happen: this operation is replaced by the corresponding indices are recomputed using their bit-reversal [4]. FFT calculation is performed repeatedly on the individual elements of a vector. At each stage is repeatedly performed the same basic operation on the two input variables. The analysis revealed several major characteristic regions that consume the greatest amount of CPU time:

- Allocation of memory for dynamic arrays of data and its release;
- Initialization initial values;
- Basic computing operations in cycles: matrix-vector multiplication, the application of the basic operations and the formation of transformation matrixes.

A parallel algorithm for computing the FFT must be providing uniform distribution between the original vectors between processing threads. Calculating's of one step use independent data and can run in parallel, where each thread uses the basic elements of a vector operation only once. The steps are executed sequentially, the next steps are used elements of the vector calculated in the previous steps. After each step to synchronize execution threads, the waiting time for synchronization should be minimal.

This concept of parallelism can be realized by two algorithms, the first of which implements parallel computing cycles at each FFT stage, and the second uses a recursive call tasks are executed in parallel.

The analysis has shown that more obvious realization is a recursive method. In a recursive version of the parallel algorithm on each step of the task is divided into two equal sub-tasks of calculations of FFT. These two subtasks are created in the field of parallel computing and executed in parallel. Thus, after completion of these tasks it is necessary to perform one step of the FFT sequence.

Other basic Fourier systems, discussed below, are different form the basis functions and the rules of calculation of the spectrum. The basic ideas of parallelism investigated in this section are fully applicable to these basic systems. The validity of a statement can only be verified experimentally. On the example of the tasks developed only a general approach to the creation of parallel algorithms and programs. The actual results of established algorithms will be discussed later.

III. Estimating the Acceleration of Computational Algorithms

For the implementation of computational algorithms and evaluating the acceleration were selected hardware and software tools and experiments:

- Dual-Core processor Intel Core i3-2120, quad-core Intel Core i5-2310;
- Parallel programming technology Open MP (Open Multi-Processing), for processors with shared memory. This is a set of compiler directives, library routines and environment variables intended for programming multi-threaded applications on multi-processor and multi-core systems. The implementation of this technology is available for many platforms - Unix, Windows NT;
- A package of Intel Threading Building Blocks (TBB) [5]. Is a library that simplifies writing multi-threaded applications, offers the possibilities of creating additional parallel applications to the C++ users;
- A package of Intel V Tune Amplifier XE, which helps to analyze the features of the algorithm and identify fragments of the application, where it can use the available resources more efficiently.

Also the opportunities of multithreading technology were used in the Windows API (library Win32 Native Threads). In the experiments, software system was created. The complex is a software package signal and image processing, which includes direct and inverse-dimensional and two-dimensional spectral transformations within the above-mentioned basic systems and created as a single user program.

The structure of the complex includes numerous basic and auxiliary functions for working with memory, subroutine calls the implementations of numerical methods, experimental histograms and test cases.

As a result of selection in the individual functions of sequential and parallel algorithms for all types of spectral transformations were obtained serial and parallel methods for implementing the direct and reverse transformations and matrix calculations for each of the transformations. An important part for the optimization of the software complex is memory allocation of algorithms and initialization values of the original signal. For an accurate assessment of the run-time function or a portion of the program were placed from 1000 to 10000 iterations in cycles, depending on the values of processed fragments. Moments of beginning and end of the experiment were measured prior to and just after the all current cycle, and then the resulting time difference is divided by the number of the executed iterations. Due to the large volume of data on the execution time of some transformations with different inputs results are presented mainly in graphical form.

The following is represented the results of the analysis of the execution time for a few basic systems of functions that give detailed information about the effectiveness of these decisions.

IV. Evaluating Efficiency in Processing Signals

The number of decompositions of the original signal is selected in the range from 16 (n = 4) up 16384 (n = 14). The maximum value of acceleration of corresponds to the number of processing threads fragments of signal. The acceleration is effective if it is more than 1.0. In this case, the parallel algorithm is faster than the serial [6]-[8].

In Fig. 1 is shown the graph of acceleration for the three basic systems: FFT, DCT and Haar transforms. Effective acceleration starts from the point n = 6 for the Haar basis and point DCT and point n = 9 for the FFT. To the point n = 13, 14 the values of acceleration in all three bases give maximum acceleration, which is close to the number of processing threads.

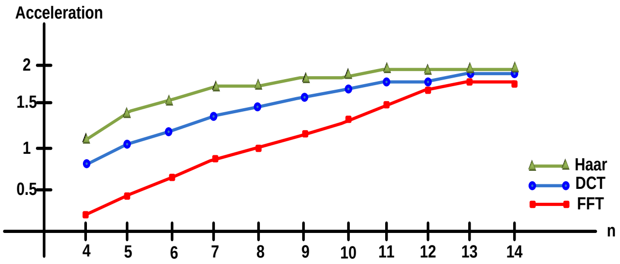

Figure 1. Analysis of the acceleration at two threads.

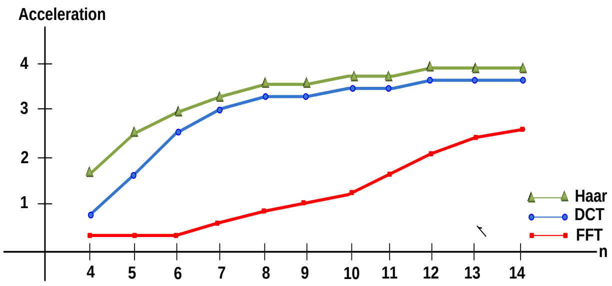

Figure 2. Analysis of the acceleration at four threads.

Fig. 2 shows graphs of the acceleration to four processing threads. Almost Haar and DCT give good acceleration rate from the first points of the graph algorithms.

The FFT algorithm has got less efficient and at the last points n = 14 does not reach the maximum value, equal to the number of threads.

The studies were also conducted to evaluate the effectiveness of training tools of preparation algorithms for parallelization at four processing threads fragments of signal. The opportunities packages were estimated Open MP, TBB and Win32. Fig. 3 shows graphs of acceleration for three packages at implementing parallel algorithm DCT, and Fig. 4 indicates graphs for parallel algorithm "Haar wavelet." Analysis of the graphs shows that the most effective package is package Open MP, which acceleration from the first points (n=4) and rapidly reaches maximum efficiency.

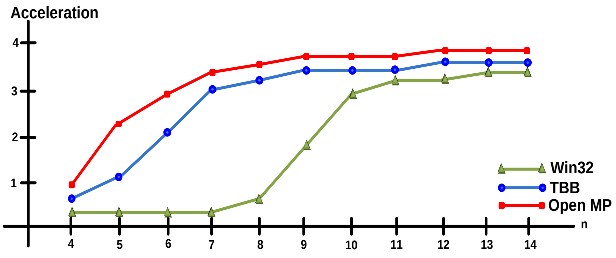

Figure 3. Efficiency of packages for DCT algorithm

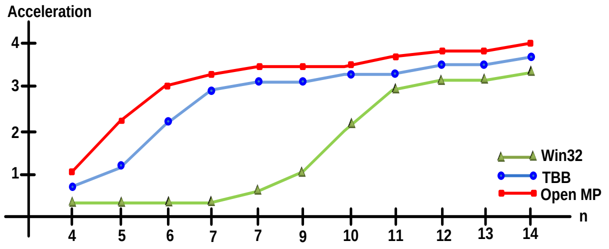

Figure 4. Efficiency of packages for the algorithm "Haar wavelet"

There are a variety of methods just to decompose of the image space into fragments for subsequent processing each fragment. To preserve the integrity of images, the image is divided into blocks of small size. In areas of the image with a smooth change of brightness in such a partition handicaps are not noticeable. Areas of the image with a high spatial frequency (in contrast contours and boundaries of the image) are appeared handicaps. In this case, the block sizes of the partition of image should be enlarged.

With the account of the diversity of the types of images and different types of decomposition, the values of fragments are selected in the range from 8×8 to 1024×1024 pixels. In the figures, number of samples to simplify is given by one figure, although the block has the dimension n×n.

The first description of the considered - general method of partitioning of the general image field into blocks of various sizes is considered in the Fig. 5, inside which there are elementary fragments by the size 8×8.

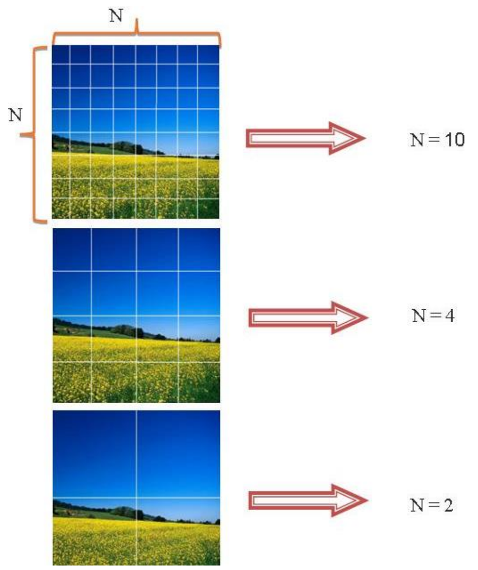

Figure 5. Decomposition of the image for streaming representation

The total size of the image field is 2048×2048 pixels, standard size processing units 8×8. For N = 10 the whole field is obtained in a standardized 65536 units, each 8x8. Next steps which is partitioning allow receiving blocks of the larger size, but within each block partitioning into fragments standard size of 8×8 remains. Large blocks (threads) are distributed across multiple cores of the processor. For example, when N = 2 is obtained on 4 blocks of 1024×1024 pixels, they are made as a single thread.

Software package implements all the previously mentioned spectral transforms applied to the image as a function of two variables. As a tool for the implementation of the parallel version of the algorithm of spectral analysis was used quad-core processor and a package of Open MP.

Fig. 6 shows the combined graphs of growth acceleration for the first method of partitioning the image field. The results of growth acceleration for 5 methods of spectral analysis: DCT algorithms Haar, Walsh-Hadamard, methods for “Haar wavelet” and “Daubechies wavelet.” All graphs are combined in one figure. Analysis of the graph shows the acceleration characteristics of affinity in dependence on the initial matrix partitioning fragments.

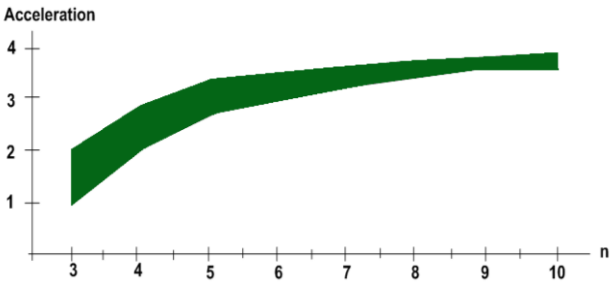

Figure 6. Combinations of graphics acceleration for streaming representation matrices

Fig. 7 shows a second way of decomposition in the total field image into fragments of different sizes.

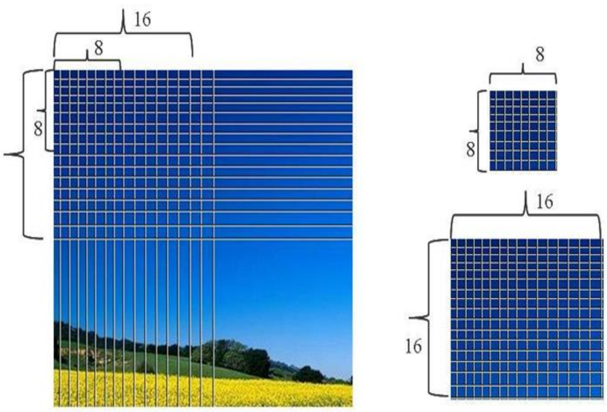

Figure 7. Decomposition the total image into various sizes

The whole image is divided into equal parts (fragments) in a different size in each time. First, a set of identical units of size 8×8 pixels, then the measurement is carried out to accelerate the matrix divided by the same cell size of 16×16. In this case, the two-dimensional spectral transformation changes the dimension from 8 to 16. The process is repeated with larger elements as long as the overall image matrix is not divided into 4 large fragments of 1024×1024 pixels.

Program which implements the second method of partitioning the original image matrix, is characterized by the introduction of two additional functions. These functions are enclosed in the initial matrix partitioning rules and conversion in the formation of fragments to be processed in parallel and serial modes.

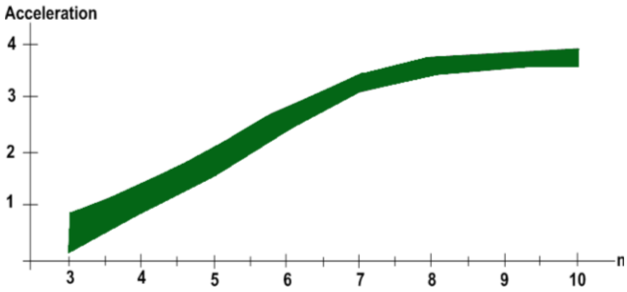

Figure 8. The combination of graphics of acceleration at partitioning the image in the different size

Fig. 8 illustrates the combinations of graphics of acceleration of growth for the second method of partitioning the image matrix. Tasks of parallelization of algorithms in the second method partitioning images in small values of processing fragments do not give a significant acceleration. At maximal levels of acceleration, unlike the first method of division, the graphics are located only at points n = 9 and n = 10.

V. Conclusion

1) To improve the efficiency of parallel computations, it is necessary to analyze the computational schemes, to carry out their decomposition into subtasks, to determine the order of their subtasks of information exchange, to share is subtasks to each processor core.

2) For the vector-matrix calculations in the algorithms of digital signal and image processing characteristic is repeating the same action for different elements of vectors and matrixes, indicating the presence of data parallelism, so the parallelization of such operations is reduced, as a rule, to the separation vectors and matrix elements between processor cores.

3) The main parameter that affects on scalability of algorithms of spectral transformation and acceleration of applications on multi-core platform - is the size of the fragments of the signal, which determine the computational complexity of numerical algorithms. The acceleration dependence from the fragment size has in the nature an increasing function, followed by saturation. The maximum value of acceleration at the optimal solution tends to the number of processors in the system. The minimum threshold of acceleration depends on the type of spectral analysis and the parallelization tools used.

4) The monotonous character of acceleration growth graphs with increasing fragment size indicates that the system cost for a low value signal or image fragments inhibit the growth acceleration. In multi-core processors

partition on too large parts cannot load evenly processors and don't allow to achieve a minimum computation time, and too fine granularity means the growth of unproductive expenditure synchronization.

REFERENCES

[1] S. Akher and J. Roberts, *Multi-Core Programming, Increasing Perfomance through Software Multi-threading*, Intel Press, 2006, pp. 75-98.

[2] M. M. Musaev, "Parallel computing in digital signal processing," *Scientific-Technical and Information-Analytical Journal TUIT*, no. 3, pp. 5-10, 2013.

[3] S. G. Johnson and M. Frigo, *Implementing FFTs in Practice*, Rice University, Houston TX: Connexions, 2008, pp. 65-73.

[4] R. Gonsales and R. Vuds, *Digital Image Processing*, Moscow: Technosfera, 2006, pp. 576-612.

[5] J. Reinders, *Intel Threading Building Blocks: Outfitting C++ for Multi-Core Processor Parallelism*, Sebastopol, California: O'Reilly Media, Inc., 2007. pp. 334.

[6] M. M. Musaev and M. S. Kardashev, "Spectrum analysis of signals on multicore processors," *Scientific Journal of Digital Signal Processors*, no. 2, pp. 82-86, 2014.

[7] S. Haykin and B. Veen, *Signal and Systems*, USA: Wiley International Edition, 2003, pp. 641-413.

[8] H. G. Stark, *Wavelets and Signal Processing*, Springer Berlin Heidelberg, 2005, pp. 22-28.

Musaev Muhammadjon Mahmudovich, from Uzbekistan, was born on October 12, 1942. He graduated from the Tashkent Polytechnic Institute in the area of Mathematical and computational complexes and systems in 1966. Doctor of Science. He is a current member of the IEEE. In 1990, he joined the Tashkent State Technical University. He had been working in the Academy of Sciences of the Republic of Uzbekistan. Since 2005, he works at the department of Computer Systems in Tashkent University of Information Technologies (TUIT). His current interest is in the field of digital signal processing, parallel information processing and real time systems. He is currently researching digital processing of speech and digital image processing.

Berdanov Ulug'bek Abdumurodovich, from Uzbekistan, was born on July 22, 1987. He obtained Bachelor of Applied Informatics from the National University of Uzbekistan (NUU) in 2009, and Master in Applied Informatics from Tashkent University of Information Technologies (TUIT) in 2012. He published many papers in different journals. He is junior researcher and assistant-teacher at the department Computer systems, Tashkent University of Information Technologies (TUIT), Uzbekistan since 2012. His current interest is in the field of digital signal processing, parallel information processing and real time systems. He is currently researching digital processing of speech and digital image processing.