International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 335 - 339

Parallel Solving Tasks of Digital Image Processing

Khujayarov Ilyos Shiraliyevich, Senior teacher, department of Information Technologies, Samarkand branch of Tashkent university of information technologies, Samarkand, Uzbekistan, *shurcha_2@mail.ru*
Islomov Shahboz Zokir ugli, assistant teacher, department of Information security, Tashkent university of information technologies, Tashkent, Uzbekistan, *shox_gold91@mail.ru*
Nuriev Sirojiddin Abduraxmonovich, assistant teacher, department of Software Engineering, Samarkand branch of Tashkent universityof information technologies, Samarkand, Uzbekistan, *nuriyev14051989@gmail.com*
Ochilov Mannon Musinovich, student of computer engineering department, Samarkand branch of Tashkent university of information technologies, Samarkand, Uzbekistan, *ochilov.mannon@mail.ru*

Abstract**—** In this paper is given methods of establishment parallel computing on the several calculating stream of image spectrum processes on two dimensional FFT, DCT and Walsh-Hadamard basic systems. Creating parallel algorithms of spectrum analysis as a used technologies OpenMP, Intel TBB and Intel Cilk Plus and their library opportunities are provide.

Keywords- *Parallel solution, OpenMP, TBB, Intel Cilk Plus, Image processing, multi-core, 2D FFT, 2D DCT, 2D Walsh-Hadamard.*

*****

I. INTRODUCTION

The main trend in the development of modern computer technology is the development of multi-core and multi-processor computer systems. This is due to the constant growth of computational needs, on the one hand, and the achievement of the maximum clock frequency on which microprocessors can work with existing technologies of their production, on the other. The widespread distribution of such computer systems makes it necessary to develop a new, radically different from the existing one, the software that can run in parallel. As much rapidly are being developed technology and software to writing those programmes. The development of such areas of science as image processing, pattern recognition, speech processing, and many other possibilities depends on the use of parallel computer systems.

One of the main directions of development of information technology is the development of methods and image processing and analysis algorithms, due to the tendency to use natural human forms of information exchange, which include visual display of reality.

Progress in the development of algorithms image processing it leads to an increase in the volume of information processed, as well as the complexity of the operations. Use of traditional computing means for processing in real time is difficult and requires the use of special means of parallel processing. To solve this problem are often used parallel and multiprocessor systems.

Systems for digital signal processing (DSP) algorithms are major processing audio and video images, is widely used in multimedia applications. Currently, the most complex algorithms signal and image processing are performed in the spectral domain, by decomposition of the original signal spectrum. Signal processing in the spectral domain is a labor-intensive tasks, with an increase in the volume of the input data at run time computing grows nonlinearly.

Image digital processing algorithms are used for increasing computing efficiency until nowadays reducing number of current algorithms computing operations by optimizations or simplifying number operations of algorithms. Examples can include multiple options of the Fourier-bases which recursive calculation in the fast Fourier transform, wavelet - transformation. However, multi-core processors bring a new dimension to the process of optimization algorithms, allowing you to manage data flow, split computation across multiple computing devices through the use of branching algorithms and execution flow calculations.

Stream processing in digital image processing tasks should be considered as a new technology, including the following training programs and performance elements:

- Analysing and detecting probably part for parallel computing in the program;
- Specifying number methods and tools for ensuring parallel computing;
- Selecting programming language tools which used for efficiency writing solution;
- Studying organisation technologies of parallel stream based on modern special tools
- Detecting and correcting errors which appears at the time parallel computing;
- Evaluating efficiency of the created program with helping current special tools.

II. ALGORITHMS AND HARDWARE

We have to solve two main functions for improving efficiency of digital image processing:

- Constructing more performance computing systems;

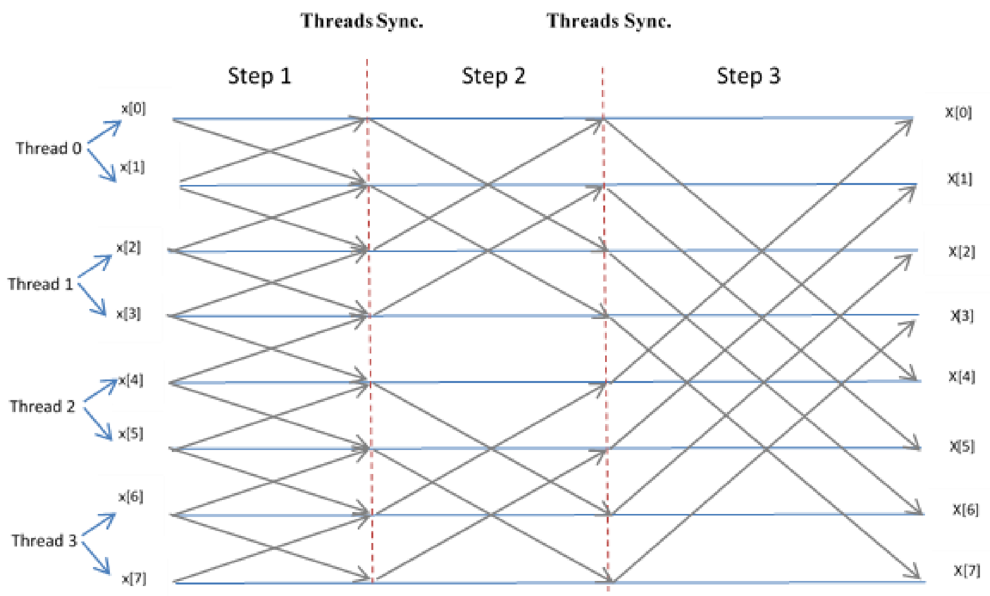

Fig.1. Parallel implementation of the FFT algorithm

- Creating efficiency software based on parallel
- processing algorithms;

Hardware platforms of the multi-core processors are differenced mainly by two indexes, they are:

- Separated for core, general memory (cache, RAM) and distributed memory (a core has autonomous cache or RAM);
- Method exchange data between cores (distributed scope – bus or communication relations).

Hardware paralleling has speedily technological character and it depends opportunity of electron components and association with the increase of the execution units.

Paralleling software level consists of very wide functions. Difficulties optimization of the computing digital image pre-processing problems on the high level depend a lot of factors.

Changing sequence to parallel process offers to find iteration operations over no-related data each to other, to change algorithm structure and to research solving method other type of the problems. Methods of parallel computing consist of following tasks:

- parallel cycles, iteration fields which divided number of computing stream;
- reduction algorithms, computing same type of from a lot of operations which collected as a result;
- allocation functions or processes which performance at same time;
- recursive calls are a method within itself with the purpose of the different computing threads to their implementation;
- using parallel templates of matrix operations;
- libraries and compilers which avtomamatization of high level parallelism;

Creating efficiency parallel algorithm related to above given programming performance tools, such as, processor parameters, mechanisms of creating computing stream on the operation system and number of the streams.

Maximal intensity of parallel algorithms usually achieves performance by number of computing streams which given processor specifications. In following table 1. is given technical specifications of the Intel Core i3-2120 and Intel Intel Core i5-4200M processors which used to determine level of intensity.

Table 1. Technical specifications of Intel Core i3-2120 and Core i5-4200M processors

Name	Cores	The number of threads	Core clock, GHz	Hyper-threading
Intel Core i3-2120	2	4	3.30	Yes
Intel Core i5-4200M	4	4	3.20	No

In first solution we will review to create parallel version of the two dimensional immediately fast Fourier transform (FFT) algorithm which widely used on the image spectrum processing. Also, we know that two dimensional immediately Fourier substitution is computed by two times performance by rows and columns of the one dimensional FFT. Accordingly, we can parallel compute of two dimensional FFT by parallel computing one dimensional FFT.

Fast Fourier Transform (FFT) proposed by Cooley and Tukey [2]. Operation "Butterfly" (Figure 1) is a function, at the entrance of which there are two numbers: *A* and *B*. The output is the number of *A* + *BW* and *A* - *BW*. The number of the *W* - the turn coefficient - calculated by this formula:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, k = 0,1,\dots,N-1 \quad (1)$$

$$W_N^{n(k)} = e^{-i\frac{2\pi}{N}n(k)}$$

where N - the dimension of the FFT, $n(k)$ - an integer, depending on the number k of the operation "butterfly" algorithm FFT. It can be partitioned into two parts with odd and even indices recursively until 2-point DFTs are obtained. By recursive operation and applying the symmetric property and periodicity of W_N, the time complexity of this DFT is reduced from O(N^2) to O(N logN) and this resulting method is called FFT.

We will obtain following expressions by divided FFT into pair and odd indexes $X[k] = X_e(k) + W_N^k X_0(k)$ and $X\left(k + \frac{N}{2}\right) = X_e(k) - W_N^k X_0(k)$. According to recursive computing of FFT N changes from N to $N/2$ and these two points are different at every level of FFT.

For example, for F(2) in the first step, where N = 2, F(3) is partner, in the second step, where N = 4, F(4) is its partner, and so on, until N is FFT_{length}. These two complex numbers depend to each other and they establish two complex numbers together for next level.

These two dependent data are called pair numbers in this paper. For these pair complex numbers, $N/2$ as the distance between these pair samples is called the pair distance, which varies from step to step and will be doubled in the next step. These pairs are the source of data dependencies on FFT. In fact, in the FFT each pair of complex data points has a partner with a $2^n / 2$ pair distance in step n.

Therefore, to compute the further value in the next step, another complex number in the same pair distance must be known. Figure 2 indicates the radix-2 FFT method. After the bit-reversal process, the complex data stream passes through *logN* steps to form the output stream.

In the parallelized 1-D FFT with lowest data dependency, after the bit-reversal process, the data samples are located in their sequence.

Bit reversal - the transformation of a binary number by a reordering of the bits in it is reversed. Bit-reversal applies only to the index of the array elements and is intended to change the order of these elements, the values of the elements do not change themselves.

The serial data stream is divided into p parts with each having (N/p) points and distributed among all processors, where N is the FFT length and p is the number of processors. To decrease data dependency, all N/p data blocks in each processor must be in sequence; namely, N/p samples must be sequential samples in the reversed-bit sample data stream. That is, the first (N/p) samples contain 1st, 2nd,...,(N/p)th data samples, and so on, and in the first log(N/p) levels, pair samples are all located in one processor. Hence, the first $\log(N/p)$ levels of FFT can be performed by each processor individually and the other $logN - log\left(\frac{N}{p}\right) = \log p$ levels are then done by data exchange between processors [4].

Two-dimensional discrete cosine transform (DCT) is one of the orthogonal transformations, which is used in data compression algorithms with losses, such as MPEG and JPEG. Direct two-dimensional DCT is described as the following formula:

$$A_{u,v} = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1} F_{j,k}\cos\left[\frac{\pi}{N}(j+\frac{1}{2})\right]\cos\left[\frac{\pi}{N}v(k+\frac{1}{2})\right] \quad (2)$$

Inverse transformation:

$$F_{j,k} = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1} C(u)C(v)A_{u,v}\cos\left[\frac{\pi}{N}(j+\frac{1}{2})\right]\cos\left[\frac{\pi}{N}v(k+\frac{1}{2})\right] \quad (3)$$

Here $C(0) = 2^{-\frac{1}{2}}$; $C(w) = 1$, $w = 1,2,3.....N-1.$, $F_{i,j}$ - image matrix, $A_{u,v}$ - spectral coefficients.

An orthogonal system of Walsh functions has a special place among the linear orthogonal transformation to effectively reduce redundancy of computing operations. Walsh - Hadamard is relatively simple implementation for conversion in real time and is characterized by the ability to use fast algorithms for computing, implemented on a multi-core processor. Walsh functions is a family of functions that form an orthogonal system, taking only values 1 and -1 in the whole domain. The group of the Walsh functions form a matrix. Direct conversion Walsh - the product of the matrix of the original image on the matrix coefficients HW:

Direct conversion:

$$C_{k,i} = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{i=0}^{N-1} F_{k,i}W_k(k,i) \quad (4)$$

Inverse transformation:

$$F_{k,i} = \sum_{k=0}^{N-1}\sum_{i=0}^{N-1} C_{k,i}W_k(k,i) \quad (5)$$

where $F_{k,i}$ - image samples, $C_{k,i}$ - spectral coefficients, $W_k(k,i)$ - elements of the transformation matrix [1].

III. RESULTS OF THE ANALYSIS OF TWO-DIMENSIONAL SPECTRAL TRANSFORMATION

Two-dimensional spectral transformation in many respects similar to the one-dimensional, the difference is that the images and the transformation matrix and have the same dimension. The basic operation of the spectral analysis - decomposition in the spectrum, obtained by multiplying the matrices and the calculation of the partial sums of the sequence of numeric values.

Features of spectral transformations that can be used in the creation of parallel processing algorithms on multicore processors:

- vectors and matrix signals, images, direct and inverse transformations are binary dimension and effectively interpreted by the hardware-software platforms of processors;
- dominated by three major types of mathematical operations: multiplication of a vector by matrix multiplication of the matrix by the matrix calculation of partial sums of numerical sequences;
- Analysis of algorithmic complexity shows the possibility of the effective implementation of these models on multicore processors with more cores that are multiples of two.

IV. METHODS OF PARALLELIZATION ALGORITHMS

Above given image digital processing spectrum methods express itself direct and inverse substitution formulas, multiplication input values to matrix of spectrum substitution basis functions are expressed. Above we have seen several views of such as, basis function algorithms. It seems that image spectrum process consists of matrix operations. Paralleling two dimensional spectrum substitution algorithms are used parallel multiplication matrix to matrix.

Consider a parallel matrix multiplication algorithm, the basis of which will be based on the partition of the matrix A in the continuous sequence of strings.

Definition of subtasks. From the definition of matrix multiplication, it follows that the calculation of all elements of the matrix C can be performed, independent of one another. As a result, a possible approach to parallel computing is to use as a base subtasks procedures for determining one element of the resulting matrix C. To carry out all the necessary calculations, each sub-task needs to perform calculations on the elements of the one row of the matrix A and one column of the matrix B. The total number obtained with this approach subtasks is equal to n^2.

In carrying out practical calculations amount formed under tasks usually exceeds the number of available processing elements (CPU and / or core), and thus, it is inevitable coarsening stage basic tasks. In this plan, it may be useful aggregation of computing already in step selection of basic subtasks. A possible solution may consist to merge into a single sub-tasks all the calculations associated with not one, but several elements of the resulting matrix C. For further consideration under this section define the basic task as a procedure for calculating all the elements of one of the rows of the matrix C. This approach reduces the total number of sub-tasks to the value n. To perform all the necessary calculations basic subtasks should be available to one of the rows of the matrix A and all columns of the matrix B. This problem can be solved simply - duplication of the matrix and all subtasks. It should be noted that this approach does not lead to duplication of real data as developed algorithm is focused the use of computer systems for a total shared memory, to which there is access from all the processing elements used.

Analysis of information dependencies. To calculate the one row of the matrix C is necessary that in each subtask of the It contained a row of matrix A and had access to all columns of matrix B. We will implement this one by parallel computing technologies.

Zooming and distributing of part issues. Separated main part issues are characterized of transferring same computing difficulties and same value of data. Main part issues leads to enlarge, whenever matrix size n will be large than number of computing elements (processors or core) p. This problem can be solved by combining several the neighboring matrix lines. In this case inputting matrix A and resulting matrix C is divided into one line horizontal ribbon. Size of ribbon must be selected in view $k=n/p$. This in turn is supplied computing workload to the same way computing elements.

V. PARALLEL PROGRAMMING MODEL

At the stage of program realization of parallel algorithm the developer has a choice: create computational streams manually, directly referring to tools the operating system, or use a special programming library, extension language, application programming interface or other means, automates the creation low-level code and allows you to design parallel program areas on a more high abstract level[3].

Nowadays, many parallel algorithm programmers use parallel programming models which work in C/C++ and Fortran programming language, they are:

- Intel Cilk Plus (C/C++ Language extensions to simplify parallelism)
- Intel Threading Building Blocks (Widely used C++ template library for parallelism)
- Domain specific libraries – Intel Integrated Performance Primitives (IPP), Intel Math Kernel Library (MKL)
- Established Standards – Message Passing Interface (MPI), OpenMP, Coarray Fortran, OpenCL
- Research Development – Intel Concurrent Collections, Offload Extensions, Intel Array Building Blocks, Intel SPMD Parallel Compiler.

At that time MKL and IPP library contains a set ready functions in different areas computing(including digital signal processing algorithms), other libraries and language extensions contain definitions for the types of tasks, functions and operators of the language, allowing to build their own parallel algorithms.

By using above given parallel programming models NxN size of image over separated parts use parallel which takes spectrum coefficients.

Fig. 2, Fig. 3, Fig. 4, shows a comparison of growth acceleration algorithm two-dimensional DCT, two-dimensional Hadamard and two-dimensional FFT. Implementation using parallel versions of such popular tools like OpenMP compiler directives, library Intel Threading Building Blocks and Intel Cilk Plus.

In two dimensional FFT, DCT and Walsh-Hadamard basic systems experiments results of spectrum substitution parallel algorithms analysis, namely, readout signal values which intensifying achieved parallelism is presented at table 2.

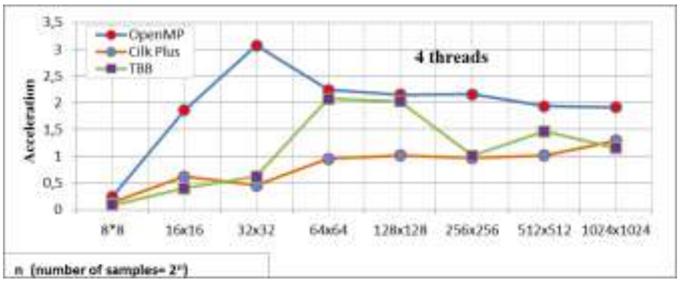

Fig.2. Comparison of acceleration algorithm DCT implemented using libraries TBB, OpenMP, Intel Cilk Plus.

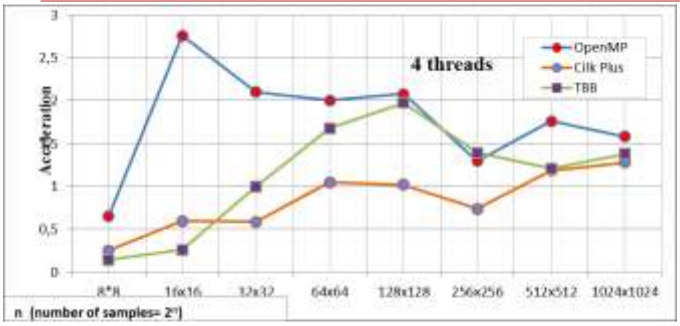

Fig.3. Comparison of acceleration algorithm Hadamard implemented using libraries TBB, OpenMP, Intel Cilk Plus.

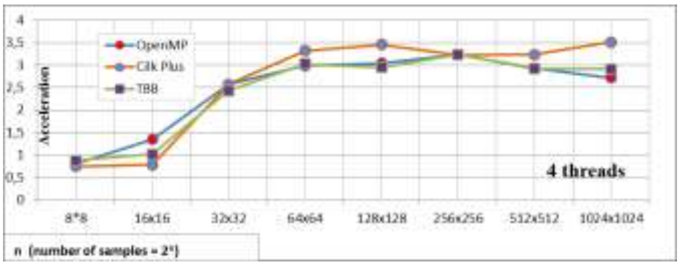

Fig.4. Comparison of acceleration algorithm FFT implemented using libraries TBB, OpenMP, Intel Cilk Plus.

Table. 2. Effective value of the number of counts for various multi-threaded tool

Tools	2D FFT	2D DCT	2D Walsh-Hadamard
OpenMP	16×16	16×16	16×16
TBB	16×16	64×64	64×64
Intel Cilk Plus	32×32	1024×1024	512×512

It seems, for efficiency computing of parallel algorithms need supply that size of calculation uses more than in table 2 given efficiency border.

VI. CONCLUSION

1. We can say by graphics of result analysis, Open MP technology was achieved efficiency results which above was used parallelism of spectrum substitution algorithms. Open MP technology gives good results on the establishing parallel regions itself for creating parallel algorithms, signal and image spectrum processing algorithms by special directives which they are used for easy synchronous. Two dimensional Wash-Hadamard and FFT algorithms of TBB library are for paralleling, Intel Cilk Plus technology is for paralleling of FFT recursive computing algorithm, these methods give more efficiency.

2. The optimum number of threads for parallel processing must be equal to the number of processing threads provided by the processor specification. At the same time, two physical cores may be four logical and capable of operating four processing threads at a level close to that of real quad-core processor.

3. Comparison of programming libraries for the implementation of in-line parallelism shows a different character to accelerate growth with an increase in the number readout N. With a small number of more efficient readout of signal is the use of OpenMP compiler directives (acceleration occurs when the number readout from the 16×16). If the number readout greater than 128×128 TBB algorithms behave more stable. If the number readout more than 16×16 Intel Cilk Plus technology gives a good result in the recursive computation FFT.

REFERENCES

[1] Мусаев М.М., Кардашев М.С., “Спектральный анализ сигналов на многоядерных процессорах//” Цифровая обработка сигналов. 2014. №2. С. 82-86.

[2] M. Musaev, I. Khujayarov, A. Buriboev, Accelerate the Solution of Problems of Digital Signal Processing Technology Based INTEL CILK PLUS, Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658) Volume 03– Issue 02, April 2015.

[3] Blair-Chappell S., Stokes A. Parallel Programming with Intel Parallel Studio XE. – Indianapolis, Indiana: «John Wiley & Sons, Inc.», 2012. – 556 p.

[4] Mehrdad Fallahpour, Chang-Hong Lin, Ming-Bo Lin, Chin-Yu Chang . Parallel One-and Two-Dimensional FFTs on GPGPUs. Conference: Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference.

[5] Reinders J. Intel Threading Building Blocks. Outfitting C++ for Multi-Core Processor Parallelism. – Sebastopol, California: «O’Reilly Media, Inc.», 2007. – 334 p.

[6] Greg Slabaugh, Richard Boyes. Multicore Image Processing with OpenMP. IEEE Signal Processing Magazine (Volume:27 , Issue: 2). March 2010.