Utilization of Multicore Processors Using Parallel Image Enhancement Algorithms

Preeti Kaur[1], Anil Kumar[2]

Guru Nanak Dev University, Department Of Computer Science,

Amritsar, Punjab, India

[1] Preeti19111990@gmail.com

[2] kumanil007@gmail,com

Abstract: ***This paper presents performance results on testing Matlab applications using the parallel computing and the distributed computing toolboxes in multicore environment. The results show that the leveraging of multicore platforms can speed up considerably the processing of images through the use of parallel computing tools in Matlab. The application used as a testbed is classical in the context of high performance computing: image processing. In any case, all processing units should be effectively used in order to optimize the performance of parallel applications. Systems is statistical significance on the factors represented by the number of workers utilized and the number of images processed, yielding more than a 500% performance increase by using 8 Matlab workers on Intel(R)Core(TM)i5 machine.***

Keywords: Image Processing, Parallel Computing, Processors, Parameters.

1. INTRODUCTION

Image enhancement [3] is basically improving the opinion of information in images for human viewers and providing `better' input for other automated image processing techniques. The principal objective of image enhancement is to improve the quality of an image using parallel computing and to make it more suitable for a given task and a specific observer.

During this process, images are equally distributed among the workers. The choice of attributes and the way they are modified are specific to a given task. The main objective of use of the parallel computers is to speed up computations and improve efficiency and response time by using multiple CPUs, or to perform larger computations which are not possible on single processor system.

In general, we can divide parallel computing into two major classes with respect to memory. The first is the class of distributed memory and another is shared memory.

In the real time environment, there are huge tasks that require large amount of processing power and time for computer vision applications. The perception of having more than one computation resources for solving certain time consuming problems seems more interesting and valid.

Coordination among multiple computation resources and work distribution among them place major challenges on system designer.

Our objectives in this paper are as following. 1) Explore the parallel and distributed image processing in simple way. 2) Present the study of parallel and distributed image processing with tools and technology used application domains and ongoing research work. 3. Results are evaluated after running the application using various parameters.

The approach presented for processing the digital images using parallel computing order to reduce execution times [1]. This approach is tested over images of several resolutions using grayscale, brightening and contrast of images.

1.1 Image Processing

In this section the application to image processing is focused. Image processing [6] has become an applied research area that goes from professional photography to several different fields such as astronomy, computer vision; medical imaging etc.

On comparing ability of human and computer for processing of image, we find that human brain can process images very fast because of its ability to generalize and infer knowledge from image data and large number of neurons working in parallel, and on the other hand the computerized image processing has many limitations compared to human brain.

However the advancement in Artificial intelligence, parallel computing and network based computing, the solution of above is possible with intelligent image processing with parallel and distributed system. The emphasis in algorithm design has shifted from sequential algorithms to parallel algorithms as more computers have incorporated some form of parallelism [4].

The aim of digital image processing is to improve the pictorial information in order to perform other tasks such as feature extraction or pattern recognition. Image processing is usually an expensive and time consuming task for example in serial processing, a grey scale image of 1024*1024 pixels, will require more CPU time to make more than one million operations.

According to Jain [4], digital image processing consists of the application of function that transform a two dimensional image using a computer. Crane [5] define this task as a science that manipulates digital images that covers an extend set of techniques to enhance or distort them. Image processing operations are done at following levels:

Low level image processing
Intermediate image processing
High level image processing

Low level image processing [7] operators operate at pixel level. The input to low level image processing operators is an image and output is image or data. Few examples of low level image processing are contrast enhancement, noise reduction and noise removal in image.

Intermediate image processing operations operate on abstractions derived from the pixels of the image. These processing operations derive abstractions from the image pixels so that it can help in further decision making about image. Examples are region labeling, object tracking.

High level image processing operations operates in order to generate higher abstractions. They work on abstractions derived from immediate level image processing operators. They are used to interpret the image content. These operations work on graphs, lists, relations among regions.

1.2 Parallel Computing

Parallel computing [8] is an alternative to solve problems that require large times of processing or handling amount of information in acceptable time .In the parallel processing program is able to create multiple tasks that work together to solve problem.

The main idea is to divide the problem into simple tasks and solve them concurrently so that time is divided. Depending upon the requirement of the application and available budget, the selection of architecture is done. The parallelism can be applied in image processing applications by three main ways: 1) Data Parallel 2) Task Parallel 3) Pipeline Parallel

1.2.1 Data Parallel: In data parallel approach [7], the data is divided and distributed among the computing units. The main challenge is efficient data decomposition and result composition. The main issue must be considered for efficient parallel execution is load balancing.

Image data should be distributed among computing units in such a way that there will be less unnecessary communication among computing units and each unit gets approximately same load. The data parallelism to image data can be applied using one of three basic ways: i) Pixel Parallel ii) Row or Column Parallel iii) Block Parallel. At present, the most of the parallel image processing applications use row/column parallel or block parallel

1.2.2 Task Parallel: In task parallel approach [7], image processing instructions/ low level operations are grouped into tasks and each task is assigned to a different computing unit. An image processing application consists of many different operations. The main challenge in task parallel approach is efficient data decomposition and result composition.

1.2.3 Pipeline Parallel: If image processing application requires multiple images to be processed, then pipeline processing of images can be done. In pipeline processing, images will be in different stages at same time.

Time	Phase1	Phase 2	Phase 3	Phase 4	Phase 5
T1	Image 5	Image 4	Image 3	Image 2	Image 1
T2	Image 6	Image 5	Image 4	Image 3	Image 2
T3	Image 7	Image 6	Image 5	Image 4	Image 3

A parallel program must have some features for a correct and efficient operation. These features include the following:

1. Granularity: It is defined as the number of basic units and it is classified as :

I. **Coarse – grained:** Few tasks of more intense computing.
II. **Fine – grained:** A large number of small parts and less intense computing.

2. Types of parallel processing:

I. **Explicit -** The algorithm includes instructions to specify which processes are built and executed in parallel way

II. **Implicit -** The compiler has task of inserting the necessary instructions to run the program on a parallel computer.

3. Synchronization: This prevents the overlap of two or more processes.

4. Latency: This is the time transition of information from request to receipt

5. Scalability: It is defined as the ability of an algorithm to maintain its efficiency by increasing the number of processors and the size of the problem in same proportion.

6. Speedup and efficiency are metrics to assess the quality of parallel implementation.

1.3 Introduction of Muticores:

As multicore [9] architectures overtake single-core architectures in todays and future computer systems, traditional applications with sequential algorithms can no longer rely on technology scaling to improve performance. Instead, applications must switch to parallel algorithms to take advantage of multicore system performance.

Image processing applications exhibit a high degree of parallelism and are excellent candidates for multicore systems. Signal and image processing programmers can benefit dramatically from these advances in hardware, by modifying single-threaded code to exploit parallelism to run on multiple cores. One of the major challenges of multicore architectures is not only to aim toward high performance but also to efficiently harness the computing power of these systems. This increases core utilizations among all applications executed on the chip and thus, chip efficiency.

1.4 Distributed System:

There are two main architecture of distributed system i) The Master Slave ii) Peer to Peer. These are discussed below.

1.4.1 Master Slave: The master slave architecture approach uses the "Distribute Compute and Gather" philosophy for parallel image processing. In this architecture approach, the master processing unit divides and distributes the image data to the slave processing units. All slave processing units work in parallel to achieve assigned task. Then master processing unit gathers and assembles the image back.

1.4.2 Peer to Peer: In peer to peer architecture, each participating entity has same capabilities and either entity can initiate a communication. The participating entities make a portion of their resources directly available to other networked participating entities, without the need for central coordination.

2. SCOPE OF THE PAPER

The overall goal of the work is to measure the performance of the proposed strategy over the existing image enhancement algorithms.

1. This research work deals with reducing the amount of required time to represent the digital images.
2. This research work does not deal with other operations that can be applied on the digital images.
3. This research work also deals with the process to collect, analyse, and evaluate the digital images to prove the effectiveness and efficiency of the proposed strategy.
4. Different type of tests will be implemented using proposed algorithm to test various aspects of the image enhancement algorithms in parallel computing.
5. Visualization of the experimental results and drawing appropriate performance analysis.
6. Appropriate conclusion will be made based upon performance analysis.
7. For future work suitable future directions will be drawn considering limitations of existing work.

Throughout the research work emphasis has been on the use of either open source tools & technologies or licensed software.

3. LITERATURE SURVEY:

Benoit Gennart et al. [10] presents a tutorial description of the CAP Computer Aided Parallelization Tool. CAP has designed with goal of letting the parallel application programmer having the complete control about how his application is parallelized. Authors discuss the issues of flow control and load balancing and show the solutions offered by CAP and also show how CAP can be used to generate relatively.

Y.Krishnakumar et al. [11] says that typical real time computer vision tasks require huge amount of processing power and time for handling real time computer vision applications. Parallel processing founds to be the only solution to obtain the require processing speed for handling high-speed image processing applications. Generally SIMD architecture is suitable under low level processing while MIMD architecture is suitable for high-level processing. This paper works on realization of a parallel operating SIMD/ MIMD architecture for image processing applications.

Cristina Nicolescu et al. [12] presents a data and task parallel low-level image processing environment for distributed memory systems. Image processing operators are parallelized by data decomposition using algorithmic skeletons. Image processing applications are parallelized by task decomposition, based on the image application task graph. In this way, an image processing application can be parallelized both by data and task decomposition, and thus better speed-ups can be obtained. Author validates method on the multi-baseline stereo vision application.

Namiko Ikeda et al. [13] proposed a new algorithm for fingerprint image enhancement by pixel-parallel processing. This algorithm makes use of morphological filter and can be mapped on a compact pixel-parallel

ISSN 2320 – 4486

architecture, such as a fingerprint identification chip. The algorithm enhances ridges by extracting center lines of ridges and removing white noises using the center line image, and enhances valleys by dilating only thin and disconnected valleys. The accuracy of fingerprint identification becomes much better, about four times, than that of the conventional method. The execution time is also sufficiently small, 0.14 msec.

F.J. Seinstra et al. [14] described an infrastructure that allows an image processing researcher to develop parallel applications transparently. Author had discussed the abstract parallel image processing machine (APIPM), and the performance models based on it. These results are highly accurate and strongly suggest that the core of proposed infrastructure forms a powerful basis for automatic parallelization and optimization of complete image processing applications.

Qingming Yao et al.[15] presents proposed double-parallel scheme can dramatically speed up image processing methods even on a low-cost FPGA platform with low frequency and limited resources, which is very meaningful for practical applications. Due to FPGA's flexibility and parallelism, it is popular for accelerating image processing. The double-parallel scheme includes an image-level parallel and an operation-level parallel. The image-level parallel is a high-level parallel which divides one image into different parts and processes them concurrently. The operation-level parallel, which is embedded in each image level parallel thread, fully explores every parallel part inside the concrete algorithms.

A Fakhri A Nasir et al.[16] in this paper reviewed the steps of image analysis done in some image processing focusing on agriculture application and also the details analysis of parallel and distributed image processing. The memory architecture in parallel and distributed image processing and some suitable application programming interface (API) in parallel and distributed image processing are examined. In general, this study provides basic understanding of parallel and distributed image processing for agriculture application.

Axel S. Trajano et al. [17] focuses on implementing different image processing routines integrated into a system that will execute it on single and distributed processors. It will deal with routines such as threshold, brightness, contrast, invert, gamma, smoothing, edge detection and enhancement, erosion and dilation, opening and closing. The execution time of every routine done through sequential and parallel will be presented to monitor the statistics that will help evaluate the performance between single and distributed computing.

Rosas, R.L et al.[18] present a SIMD architecture implemented on FPGA devices, this architecture is based on parallel processing units with internal pipeline and uses Sobel gradient operators for edge detection and also show an application for electronic boards images with surface mount devices. This architecture takes advantage of the FPGA's capabilities for parallel processing in order to reduce the time needed using a sequential machine. The proposed architecture is able to segment up to 43 images of 640×480 pixels in one second with 40 MHz clock frequency.

4. OBJECTIVES OF THIS RESEARCH WORK

4.1. The primary objective is to reduce **MEAN SQUARE ERROR (MSE)** of digital images using proposed technique than existing technique. The MSE represents the cumulative squared error between the reconstructed image and the original image. The MSE can be evaluate using the following equation:

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

In the previous equation, M and N are the number of rows and columns in the input images, respectively. The lower the value of MSE, the lower the error

4.2. Another objective is to maximize the **PEAK SIGNAL-TO-NOISE RATIO (PSNR)** between images. The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a enhanced image. The higher the value of PSNR, the better the quality of the reconstructed image. The MSE and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image quality. PSNR represents a measure of the peak error. To compute the PSNR, the block first calculates the MSE. The block computes the PSNR using the following equation

$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right)$$

R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1.

4.3. MD (Maximum Difference): Maximum difference (MD) is calculated using Eq. and it has a good correlation for all tested image enhancement techniques.

$$\text{Maximum Difference (MD)} = Max(|f(i,j) - f'(i,j)|)$$

4.4. SC (Structural Content): Correlation, a familiar concept in image processing, estimates the similarity of the structure of two signals. This measure effectively compares the total weight of an original signal to that of a coded or given. It is therefore a global metric; localized distortions are missed. This measure is also called as structural content. The Structural content is given and if it is spread at 1, then the image is of better quality and large value of SC means that the image is of poor quality.

$$\text{Structural Correlation/Content (SC)} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j)]^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}f'(i,j)^2}$$

4.5. NAE (Normalized Absolute Error): Normalized absolute error computed by Eq. is a measure of how far is the reconstructed image from the original image with the value of zero being the perfect fit. Large value of NAE indicates poor quality of the image.

Normalized Absolute error (NAE) =

$$\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left|\ [f(i,j).f'(i,j)]\ \right|}{\sum^{M}\sum^{N}\left|\ f(i,j)\ \right|}$$

4.6. Speed up: It is the ratio between sequential execution time and parallel execution time where the sequential time execution time is sum of total computation time of each task and parallel time execution is the scheduling length on limited number of processors.

$$S_P = (\textstyle\sum_{i=1}^{n} T_i) / T_p$$

4.7. Efficiency: The efficiency of a parallel program is a measure of processor utilization where Sp = Speedup , Np = Number of processors.

$$EFF = S_P \,/\, N_P$$

5. PROBLEM DEFINITION:

This paper deals with the utilization of the multicores of the systems. The main objective is to reduce the execution time and waiting time of the users. To achieve this we have used fork and join policy of parallel algorithms which will enable given job to run in parallel so that we can achieve the maximum speed of the cores of the multicore systems. To achieve the objectives of this research work, we have used image enhancement algorithm. We have used MATLAB [19] simulator to simulate the desired behavior. To do performance analysis we have also run the enhancement algorithm in sequential manner. To implement it in parallel we have used Matlabpool. In order to evaluate the results in efficient manner we have considered various images with different sizes and what we have found that is when size increases we gain more speed and efficiency. Also different image metrics for quality measurement will be used to evaluate the performance of proposed algorithm.

Proposed algorithm will be implemented in Matlab and uses the feature of image processing tool box.

6. RESEARCH METHODOLOGY:

To attain the objective, step-by-step methodology is used in this research work. Subsequent are the different phases which are used to accomplish this work.

6.1. Orientation

This research work starts with the orientation in the area of parallel computing. By consulting journals of various applications in parallel computing, reading news articles, participating in seminars and discussing with the experts. This research employs a structured method to obtain high quality information, called a related work.

6.2. Literature survey

To explore the available knowledge on the area of digital image processing, parallel computing, literature survey will be conducted using a systematic approach. High quality papers are selected to explore the existing techniques.

6.3. Proposed algorithm implementation

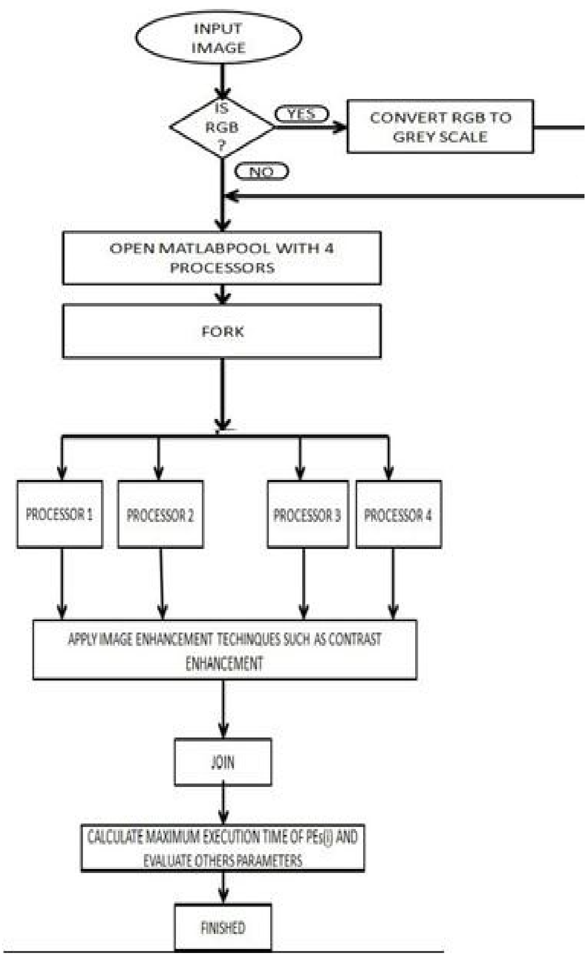
We have gone through following sequential image processing algorithms and developed its parallel versions with different approaches in core i5 processor and found good results with variable numbers of threads passed as input parameters. We focus on the most promising and well supported techniques with an emphasis in image processing application

6.4Load Distribution between Processors:

Suppose we are taking the following image (1) as an example. The main step of these algorithms is to determine the number of tiles to be generated. The number of tiles corresponds to the amount of threads. If only a thread exists, the computation is just sequential computation. Otherwise if there are two or more threads than the image is divided into distinct areas, as shown in Figure 2. Each thread is responsible for processing the pixels included in its tile and to execute different tasks but considering to maintain synchronization between all the processor otherwise there will be the situation of deadlock between processors.

Fig1.

Fig2.



6.4. Performance analysis

In order to do performance analysis, comparisons will be made by taking images with different image size. Comparisons table and diagrams will be made based upon the outcomes of the experimental results. Different image quality metrics and other parallel computing parameters such as speedup, efficiency, serial time, parallel time, response time and resource utilization will be considered to evaluate the performance of proposed algorithm.

6.5.Basic introduction of Matlab

Matlab[19] is a general algorithm development environment with powerful image processing and other supporting toolboxes. With the rapid development of multicore CPU technology, using multicore computer and Matlab is an intuitive and simple way to speed up the computing for image enhancement algorithm. In this paper, we try to have a view for using the Matlab parallel toolbox to accelerate the image enhancement algorithm by two schemes of task-parallelism and data-parallelism modal. The results show that the parallel versions of former sequential algorithm with simple modifications achieve the speedup up to 6.6 times.

6.6. Matlabpool

Matlabpool enables the parallel language features in the MATLAB language (e.g., parfor) by starting a parallel job that connects this MATLAB client with a number of labs. Matlabpool or matlabpool open starts a worker pool using the default parallel configuration, with the pool size specified by that configuration. You can also specify the pool size using matlabpool open poolsize, but most schedulers have a maximum number of processes that they can start (8 for a local scheduler). If the configuration specifies a job manager as the scheduler, matlabpool reserves its workers from among those already running and available under that job manager. If the configuration specifies a third-party scheduler, matlabpool instructs the scheduler to start the workers.

7. Discussions

We consider a grey scale image represented by two – dimensional integer valued array im (H, W), where H and W are the height and the width of the image. The first coordinate (x) denotes the number of row and (y) denotes the number of column. Grey scale images are discriminations of real black and white photographs. Then the grey scale image is passed to matlab pool consists of 4 processors using fork policy. Apply the image enhancement algorithms among the 4 processors. Then take the result such as maximum completion time of the processing elements. Compute the performance parameters and image processing parameters at the end.

8. Performance Results:

Fig 3.1 is showing the result of serial time and parallel time computed by running the image enhancement algorithms by taking the different images at different size.

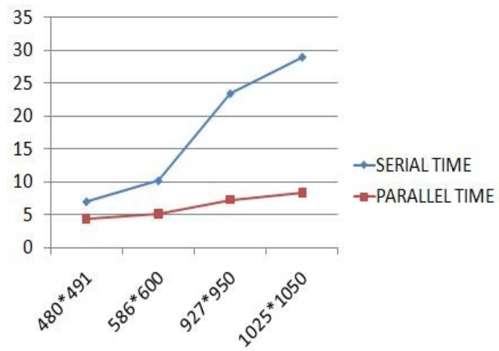


Fig 3.2 Results of speedup obtained from different images.

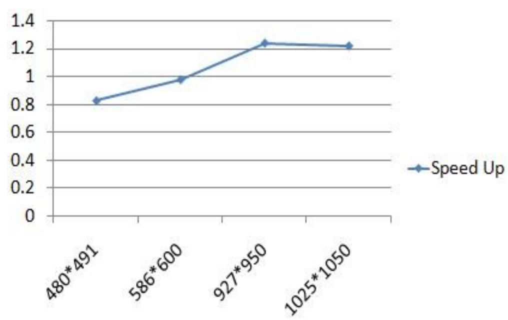


Fig 3.3 Results of efficiency obtained from different images.

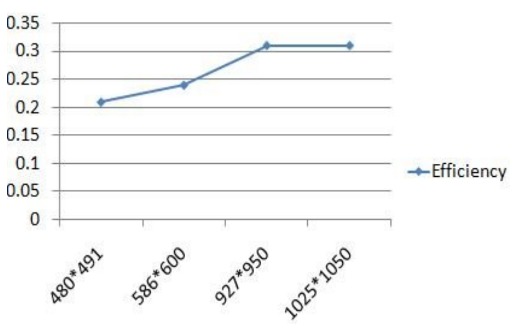


Fig 3.4 Results of MSE (Mean Square Error) obtained from different images.

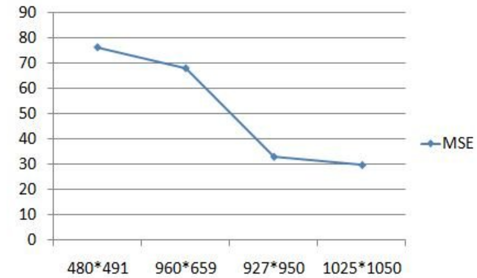


Fig3.5 Results of PSNR (Peak Signal to Noise Ratio) obtained from different images

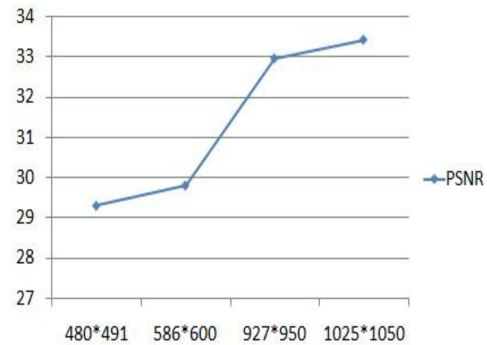


Fig3.6 Results of NAE (Normalized Absolute Error) obtained from different images

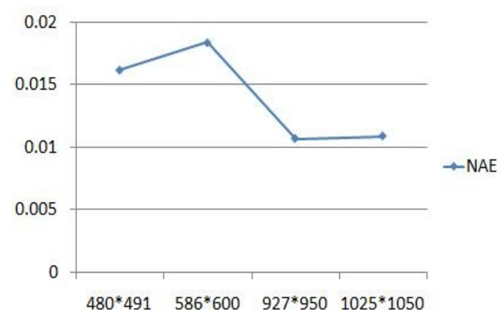


Fig3.7 Results of MD (Maximum Difference) obtained from different images

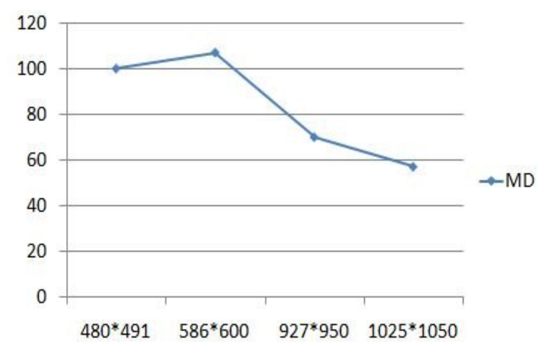
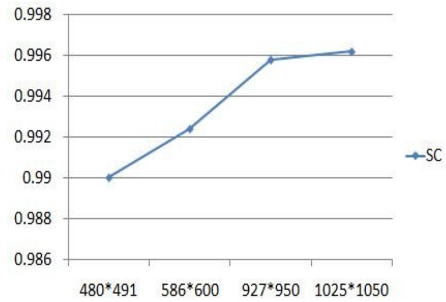


Fig3.8 Results of SC (Structural Content) obtained from different images



9. CONCLUSION & FUTURE SCOPE

This work presents parallel implementation of sequential image processing algorithm i.e. contrast algorithm. Parallel implementation of algorithm was developed using matlab threads in order to leverage the parallel processing capability of current processors with multiple cores and we can see that the speed up, efficiency, parallel time that is computed is good. We also focus on other image parameters and results are evaluated.

The focus of this implementation was to improve the performance of image processing algorithm and maximum utilization of the multicores. In terms of performance, parallel implementation was about two and a half times faster than the sequential processing. This is a very promising result since it allows the exploitation of the vast processing power of current processors with multiple cores. In the future, the intention is to use the same principle of division of work in tiles to calculate more and more parameters and proposed the ways to utilize resources better.

10. REFERENCES

[1]. Thomas Bräunl, "Tutorial in Data Parallel Image Processing," Australian Journal of Intelligent Information processing Systems (AJIIPS),vol. 6, no. 3, 2001, pp. 164–174

[2]
MatlabMPI: http://www.ll.mit.edu/MatlbMPI/

[3] Adlin Sharo T, Kumudha Raimond, " A Survey on Color Image Enhancement Techniques"

[4].A. K. Jain, Digital Image Processing. Prentice Hall, 1989.

[5]. R. Crane, A simplified approach to image processing: classical and modern techniques. Prentice Hall,1997.

[6] R.C. Gonzalez and R.E. Woods, Digital Image processing (3rd Edition).Upper Saddle River, NJ, USA: Prentice Hall, Inc 2006.

[7] C. Soviany, "Embedding data and task parallelism in image processing applications," Ph.D. dissertation, Technische Univ. Delft, 2003.

[8]http://en.wikipedia.org/wiki/Parallel_computing.

[9]http://en.wikipedia.org/wiki/Multicore processor

[10] Benoit Gennart , Roger D.Hersch , Computer- Aided Synthesis of parallel image processing applications .

[11]Y.Krishnakumar,T.D.Prasad, P.Raju, K.V.S.Kumar, B.Kiranmai, "Realization of a parallel operating SIMD-MIMD architecture for image processing application" International Conference on Computer, Communication and Electrical Technology – ICCCET 2011, 18th & 19th March, 2011.

[12] Cristina Nicolescu, Pieter Jonker , "A data and task parallel image processing environment" Faculty of Applied Physics, Pattern Recognition Group, Delft University of Technology, Netherlands

[13] Namiko Ikeda, Mamoru Nakanishi, Koji Fujii, Takahiro Hatano, Satoshi Shigematsu, Takuya Adachi, Yukio Okazaki, Hakaru Kyuragi, "Fingerprint Image Enhancement by Pixel-Parallel Processing" NTT Lifestyle and Environmental Technology Laboratories ,Japan.

[14] F.J. Seinstra and D. Koelma, "Transparent Parallel Image Processing by way of a Familiar Sequential API''Intelligent Sensory Information Systems, Faculty of Science, University of Amsterdam ,Amsterdam, The Netherlands.

[15] Ye Li, Qingming Yao†, Bin Tian‡, and Wencong Xu, "Fast Double-Parallel image processing based on FPGA" ,State Key Laboratory of Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

[16] A Fakhri A Nasir, M Nordin A Rahman and A Rasid Mamat, "A Study of Image Processing in Agriculture Application under High Performance Computing Environment" **I**nternational **J**ournal of **C**omputer **S**cience and **T**elecommunications [Volume 3, Issue 8, August 2012]

[17] Axel S. Trajano and Joseph Anthony C. Hermocilla, "Implementation of Selected Image Processing Routines for Single and Distributed Processors" CMSC 190 SPECIAL PROBLEM, INSTITUTE OF COMPUTER SCIENCE.

[18] Rosas, R.L. , de Luca. ,Santillan, "SIMD architecture for image segmentation using Sobel operators implemented in FPGA technology" Dept. of Electrical. Eng., CINVESTAV-IPN, Mexico City, Mexico

[19] Gaurav Sharma, Jos Martin "MATLAB: A Language for Parallel Computing"