ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

Image Processing Tasks using Parallel Computing in Multi core Architecture and its Applications in Medical Imaging

Sanjay Saxena[1], Neeraj Sharma[2], Shiru Sharma[3]

Research Scholar, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India[1]

Associate Professor, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India[2]

Assistant Professor, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India[3]

Abstract: To find accurate & reliable result in image analysis, it is important that image is processed and analyzed using image processing suitable AI technique further at the same time it is highly desired that processing time must be minimum. Preprocessing of the image makes it more clear and visible, while parallelizing of the algorithm optimizes the speed at which the image is processed. This paper explores current multi-core architectures available in commercial processors in order to speed up the image processing tasks. Parallel Implementation of Many sequential algorithms of Image processing was examined and analyzed in test and achieved good result if all the recourses are efficiently used. Main objective of this paper is to design some parallel image processing algorithms like segmentation, noise reduction, features calculation, histogram equalization etc by using Multi Core architecture and comparative study with some sequential image processing algorithm. These parallel algorithms are able to work with different number of thread, so as to take all the benefits of the upcoming processors having any number of cores. As medical imaging refers to view the human body in order to diagnose, monitor and treatment planning. This paper also describes the application of parallel computing applied in different Medical Imaging techniques like CT, PET scans etc.

Keywords: Parallel Computing, Image Processing, Histogram, Multi Core, CPU, Latency, CT, PET.

I. INTRODUCTION

Image is the 2 dimensional distributions of small image points called as pixels. It can be considered as a function of two real variables, for example, f(x,y) with f as the amplitude (e.g. brightness) of the image at position (x,y). Image Processing is the process of enhancing the image and extraction of meaningful information from an image. Image processing is gaining larger importance in a variety of application areas. Active vision, e.g. for autonomous requires substantial computational power, in order to be able to operate in real time[1]. According to Jain [3], digital image processing consists of the application of functions that transform a two dimensional image using a computer. Others authors as Crane [4] define this task as a science that manipulates digital images that covers an extend set of techniques to enhance or distort them.
Medical imaging is the technique and process used to create images of the human body (or parts and function thereof) for clinical purposes or medical purpose [5]. There are so many different medical image modality are present like CT, PET, MRI etc. These Modalities are having different characteristics and used as per requirements. Because size of these images are very large so for analysing these modalities take so much time to process sequentially and give result after some time. So if we divide this sequentially processing to efficient parallel processing than we can find good result in very reasonable time or if we are able to process basic steps like Image Enhancement, Morphological operation, feature calculation quickly than it will be beneficial for the medical practitioner. So by the parallel computing we can save time and/or money, we can solve larger problems in very short time periods. Parallel computing provides concurrency and by this we can use non – local recourses very efficiently. It also removes the limit of serial computing. Matlab 2010a onwards finally enables the 'Parallel Computation Toolbox' for student use. It's not part of the core Matlab student package, but it is now available as

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

an add-on toolbox. Of course, it's been available for commercial users for some time.[6] There is an increasing recognition that High-Level Languages, and in particular scripting languages such as MATLAB, provide enormous productivity gains in developing technical and scientific code [11].

II. IMAGE PROCESSING WITH PARALLEL COMPUTING

Image Processing with Parallel computing is an alternative way to solve image processing problems that require large times of processing or handling large amounts of information in "acceptable time" (according to each criterion). As we can see that medical imaging requires lots of memory space and time to process so by parallelizing we can find efficient and fast result. In parallel processing, a program is able to create multiple tasks that work together to solve a problem [7]. The main idea of parallel image processing is to divide the problem into simple tasks and solve them concurrently, in such a way the total time can be divided between the total tasks (in the best case). Parallel Image processing cannot be applied to all problems, in other words we can say that not all the problems can be coded in a parallel form. A parallel program should must have some features for a correct and efficient operation; otherwise, it is possible that runtime or operation does not have the expected performance. These features include the following [8]:

Granularity.- It is defined as the number of basic units and it is classified as:
Coarse-grained.- Few tasks of more intense computing.
Fine grain.- A large number of small parts and less intense computing.

Type of parallel processing:
Explicit.- The algorithm includes instructions to specify which processes are built and executed in parallel way.
Implicit.- The compiler has the task of inserting the necessary instructions to run the program on a parallel computer.

Synchronization.- This prevents the overlap of two or more processes.

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

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 the same proportion [9].

III. PROPOSED ALGORITHMS FOR PARALLEL IMAGE PROCESSING TASKS

We have gone through following sequential image processing algorithms and developed its parallel versions with different approaches in core i3 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 SIP application examples, namely: MatlabMPI [2], bcMPI [12], pMatlab [13], Star-P [14] and the MATLAB Distributed Computing Toolbox (DCT) [15].

Load 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, 3, 4. 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.

Fig. 1

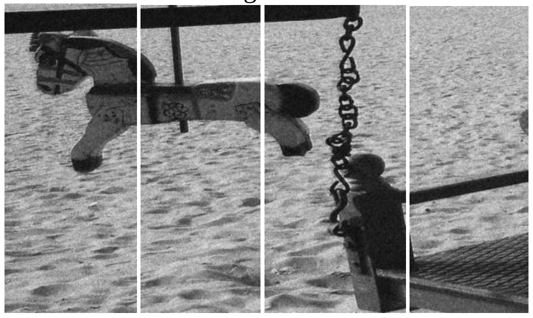

Fig. 2

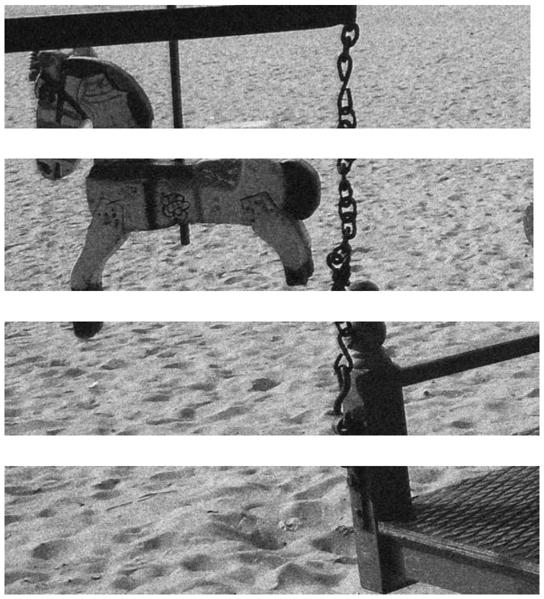

Fig. 3.

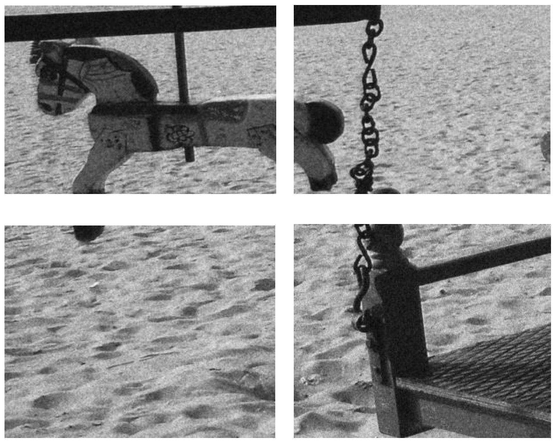

Fig. 4.

According to the requirement we can divide an image and send it different processor.

a. <u>Parallel Segmentation by Region Growing Technique and Calculation of different Features of Segmented Regions</u>

Region growing is one of the very famous techniques of segmentation. Main drawback of the region growing is that it is time consuming. This algorithm is designed to reduce timing of this algorithm. These are the steps involved in it.

1. Take Input image on client processor.

2. Selection of the Seed pixels(It is based on some user criterion like pixels in a certain gray level range, pixels evenly spaced on a grid, etc.) for different regions on client processor.[5]

3. Count number of seed points and activate same number of kernels.

4. Send copy of image and respective seed pixels on each kernel processor.

5. Regions are then grown from these seed pixels to adjacent depending on pixel intensity, gray level texture, or color, etc on each kernel processor for this image information is also important because region grown is based on the membership criteria.

6. Calculation of different features of different ROI (region of interest) on individual kernel.

7. Send segmented regions and information of ROIs to the client processor and deactivation of worker kernels.

8. Reconstruct the image on client processor and Display of all the information.

b. <u>Parallel Segmentation by Global Thresholding and Calculation of Fourier and Regional Descriptors of an Image</u>

This algorithm consists of the following steps.

1. Divide the image into Quad Tree Structure in client processor.

2. Send a flag from client processor to different worker processor and activation of four worker processors.

3. Send all the parts of the image on different kernels or different labs or different worker processors.

4. Chose Initial Thresholds T0, T1, T2, T3 in different labs individually.

5. Calculate the mean value of each lab µ of the pixel below the threshold and the pixel above the threshold value.

6. Compute a new threshold as

$$T = (\mu 1 + \mu 2)/2 \text{ in each lab.}$$

7. Repeat steps 5 and 6 until there is no change in threshold in each lab or worker.

8. Send segmented region to client from the different labs.

9. Deactivate worker processors.

10. Reconstruct the Segmented Image.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

11. Calculate the Regional and Fourier descriptors of Segmented Image on client processor.

c. Complex Noise reduction using Parallel Computing:

This algorithm consists of the following steps:

1. Take input Image on Input Image on client processor.

2. Generate Copies of the Image on client processor.

3. Activate Number of Kernels or Workers or Labs as per requirement.

4. Send Copy of the Image into different Kernels.

5. All the labs consist of filters for e.g.

Lab 1 – Median filter
Lab 2 - Wiener Filter
Lab 3 – Order statistical filter
Lab 4 – Min – Max filter etc.

6. Perform Convolution and find PSNR and MSE of the image on individual kernels.

7. Send back Values of MSE and PSNR value to the client

8. Compare the result of received data on client processor.

9. Display the image with Highest PSNR and lowest MSE Value.

10. Deactivation of kernels

d. Histogram Equalization of an Image by Parallel Computing

1. Divide the image in quad tree format or in row wise or in column wise format and activate required number of kernel processors.
2. Form the cumulative histogram on each part of image on each kernel processor.
3. Normalize the value by dividing it by total number of pixels.
4. Multiply these value by the maximum gray level value and round off the value.
5. Map the original value to the result of step 3 by a one to one correspondence.
6. Send all equalized histogram to the client processor.

IV. RESULT AND DISCUSSION

This section describes the results obtained with the parallel implementation of above algorithms. This also deals with the environment used in the experiments, the test images, and results, along with the evaluation of the performance obtained with the parallelization.

Test Environment
The experiments were all performed on an Intel Core i3 – 2350M Processor 2.30 GHz, 3 GB of RAM, Hard Disk Drive 320 GB
Software used MATLAB R2011a and JAVA JDK 1.6.0_21, 64 bit operating system.

Parallelization Scheme in MATLAB
Multithreaded Parallelism
Explicit Parallelism

Following Table 1 illustrates the performance of parallel image processing algorithm on different size images.

Images	Size	Algorithms	Time(T1) Taken by Sequential Algorithm(in sec)	Time(T2) Taken by Parallel Algorithm(in sec)	Observed Speed Up µ = T1/T2
Cameraman.tif	256 x 256	a	0.3267	.15691	**2.082085**
		b	0.268	0.0956	**2.803347**
		c	0.9132	0.9005	1.014103
		d	0.259	0.0823	**3.1567**
Bluredtext.jpg	256 x 768	a	1.0896	.00783	1.081134
		b	4.6117	.10005	2.196
		c	0.1394	.06983	1.9965
		d	0.9857	0.3279	**3.006**
Colorleaf.jpg	1281 x 843	a	3.98154	2.99105	1.331149
		b	4.4722	3.9612	1.129
		c	.319738	3.665	1.9972
		d	1.793	0.8932	**2.007**

Table 1: Performance Evaluation Table

V. CONCLUSION & FUTURE SCOPE

This work presents parallel implementation of different sequential image processing algorithm. It was developed an algorithm using OpenMP threads in order to leverage the parallel processing capability of current processors with multiple cores and we can see that the speed up is good.
The focus of this implementation was to improve the performance of segmentation, de noising, histogram processing keeping the reproducibility of results. In terms of performance, parallel implementation was about two and a half times faster than the sequential segmentation. 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 write an out-of-core version of the segmentation algorithm. This version would allow the segmentation of images that do not fit in main memory. Thus, it is expected that the image segmentation can handle extremely large data efficiently and without requiring special hardware. It is also expected to propose others parallel versions for different hardware like clusters and GPUs (Graphics Processing Units).

VI. APPLICATION IN MEDICAL IMAGING

Medical imaging require more computing power than a traditional sequential computer can do and we also know that for medical imaging, it is important that the image is clear and be obtained as quickly as possible. We can achieve through the process of parallelizing. Parallelizing optimizes the speed at which the image is produced. Normal a CT scan, PET scan, etc take much time to produce the result so by parallelizing our code we can find good result in very reasonable time.

REFERENCES

[1]. Thomas Bräunl, “Tutorial in Data Parallel Image Processing,” Australian Journal of Intelligent Information rocessing Systems (AJIIPS), vol. 6, no. 3, 2001, pp. 164–174
[2]. MatlabMPI: http://www.ll.mit.edu/MatlabMPI/
[3]. A. K. Jain, Digital Image Processing. Prentice Hall, 1989.
[4]. R. Crane, A simplified approach to image processing: classical and modern techniques. Prentice Hall, 1997.
[5]. http://en.wikipedia.org/wiki/Medical_imaging
[6]. J. Fung and S. Mann, “Using graphics devices in reverse: Gpu-based image processing and computer vision," in 2008 IEEE International Conference on Multimedia and Expo. IEEE, June 2008, pp. 9-12.
[7]. P. Pacheco, An Introduction to Parallel Programming. Morgan Kaufmann, 2011.
[8]. A. G. López, J. Delgado, and S. Castañeda, “Metodologías de paralelización en la supercomputadora cicese2000,” Departamento de Cómputo Dirección de Telemática Centro de Investigación Científica y de Educación Superior de Ensenada, Tech. Rep., February 2000.
[9]. R. T. Rasúa, “Algoritmos paralelos para la solución deproblemas de optimización discretos aplicados a ladecodificación de señales,” Ph.D. dissertation, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia, España, 2009.
[10]. Thomas Bräunl,” Tutorial in Data Parallel Image Processing,” in Australian Journal of Intelligent Information Processing Systems (AJIIPS), vol. 6, no. 3, 2001, pp. 164–174 .
[11]. A. Edelman, P. Husbands, S. Leibman, “Interactive Supercomputing’s Star-P Platform: Parallel MATLAB and MPI Homework Classroom Study on High Level Language Productivity,” *HPEC*, 2006.
[12].bcMPI:http://www.osc.edu/hpc/software/apps/bcmpi.shtml
[13] pMatlab: http://www.ll.mit.edu/pMatlab/
[14] Star-P: http://www.interactivesupercomputing.com/
[15]DCT: http://www.mathworks.com/products/distribtb/

BIOGRAPHY

Sanjay Saxena is a research scholar at shool of Biomedical Engineering, IIT(BHU) Varanasi, Uttar Pradesh, India. His research interests are Medical Image Processing, Parallel Computing, Computer Vision.

Dr Neeraj Sharma is associate professor in school of Biomedical Engineering, IIT(BHU). His specialization area is Biomedical Signal & Image Processing & Bio instrumentation.

Dr Shiru Sharma is assistant professor in school of Biomedical Engineering, IIT(BHU). His specialization area is Biomedical Control System Analysis.