REGION WISE PROCESSING OF AN IMAGE USING MULTITHREADING IN MULTI CORE ENVIRONMENT & ITS APPLICATION IN MEDICAL IMAGING

Sanjay Saxena¹, Neeraj Sharma², Shiru Sharma³

Research Scholar, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India¹
Associate Professor, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India²
Assistant Professor, School of Biomedical Engineering, IIT (BHU), Varanasi, UP, India³

ABSTRACT

Fast processing of an image is a major requirement in many of image processing applications especially in the case of medical imaging. In today’s scenario we can see that real time image processing applications need an enormous amount of processing power, computing proficiency and vast resources to perform the image processing applications. The limitations appear on image processing systems due to the dimension and nature of the image to be processed. It seems that parallel image processing would be the best solution to obtain higher speed of operation at real time resources constraints & fully deployment of available resources. This paper represents region wise parallel processing of the image instead of implementing mainly existing parallel algorithms such as grid wise or blocks wise processing to achieve excellent and speedy result. Although the existing parallel computing algorithm provide to some extent parallelism for image processing but fails to provide image processing operations varying at a huge rate. Parallel processing by image’s region of interest allows a larger image to be sub-divided into a set of different regions & each region is handled by a particular core present in the processor, where each core performing their individual task. This type of processing gives significantly efficient result rather than the existing parallel implementation of image processing application. Our experiments illustrate that the parallel implementation of the algorithm using results in a speed-up a propos 200% compared with sequential implementation on a core i3 processor, while a speed-up a propos 600% on Intel Xeon Processor. This paper also explains its application in medical imaging.

Keywords: Core, Image Processing, Medical Imaging, Multithreading, Parallel Computing, Region.
I. INTRODUCTION

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas [1]. As we know that it is the process to evaluate and manipulate the image in order to improve the superiority and finding several meaningful information from the image. Parallel processing by image region allows a larger imaging task to be assigned to a set of parallel pipelines, each performing some task but on individual regions [2]. Basically there are three types of image processing operations low level operations like smoothing, convolution, histogram, generation etc middle level image processing operations like region labeling, motion analysis etc and high level image processing operations object recognition [3]. Main problem arises in the image processing applications when we work on large images like medical images (CT, PET, and MRI) and image data set. Due to size of the image and large data set it takes more time to compute. To compute the entire image processing algorithm in reasonable time it is necessary to compute parallel instead of implementing sequential.

The main goal of this research is to improve efficiency of parallel processing in multi core architecture which is easily available and extracting the statistical features of particular region of interests (ROIs) as per requirement.

II. THEORY

Statistical features of the image plays a very vital role in image understanding. The various statistical measures [4, 10] are mean, mode, median, variance, standard deviations, covariance, skewness and kurtosis. All of these measures are used in a wide range of scientific and social research, including: biostatistics, computational biology, computational sociology, network biology, social science, sociology and social research etc. In this research we are able to find features like standard deviation, variance, covariance, mean, median, mode, auto correlation coefficient, kurtosis, skewness etc of individual distinct region. Some of them are described as below

**Mean:** The arithmetic mean filter [5], also known as averaging filter, operates on an sliding ‘m×n’ window by calculating the average of all pixel values within the window and replacing the center pixel value in the destination image with the result. Its mathematical formulation is given as follows

\[
f(x, y) = \frac{1}{mn} \sum_{(r,c) \in W} g(r, c)
\]

Where ‘g’ is the noisy image, f(x,y) is the restored image, and ‘r’ and ‘c’ are the row and column coordinates respectively, within a window ‘W’ of size ‘m×n’ where the operation takes place[11].

**Standard Deviation:** In terms of image processing it shows how much variation or "dispersion" exists from the average (mean, or expected value). Mathematically standard deviation is given by

\[
f(x, y) = \sqrt{\frac{1}{mn-1} \sum_{(r,c) \in W} [g(r, c) - \frac{1}{mn} \sum_{(r,c) \in W} g(r, c)]^2}
\]
Variance: The variance \[8\] is a measure of how far a set of numbers is spread out.

\[
f(x, y) = \frac{1}{mn-1} \sum_{(r,c)\in W} \left[ g(r, c) - \frac{1}{mn-1} \sum_{(r,c)\in W} g(r, c) \right]² \tag{3}
\]

Mode: In statistics \[4\], the mode is the value that occurs most frequently. In image processing filter will each pixel value by its most common neighbor. Mathematically,

\[
\text{Mean} - \text{Mode} = 3(\text{Mean} - \text{Median}) \tag{4}
\]

These above & many more statistical features are explained in \[11\].

If we are able to find these features by individual region present in the image than it could be very beneficial for the images having more than one regions with different characteristics. In this paper we are able to find statistical features individually for particular region. It also deals with developing a tool which efficiently parallelizes our sequential process to find ROI present in Medical Images in very minimum of time as per requirement by the medical practitioner. It is necessary to check when the parallel approach is precious or when it is not. Another objective of this paper is to produce efficient ideas of paralleling. Steven J Simske \[2\] explained the aspect of region wise parallel image processing. Parallel processing by image region allows a larger imaging task to be assigned to a set of parallel pipelines, each performing the same task but on a different data set. Image analysis of this type is readily amenable to a familiar cloud computing approach, map-reduce, when for example a large set of images is being analyzed to find a single item. Examples of parallel processing by image region include (multi-page) document skew detection and multiple face detection and tracking. Following there is a figure which is explained by Steven J Simske the region wise processing.

![Fig. 1: Example of a parallel processing by image region implementation. In this example, the slides captured from 13 different slide adapter-equipped scanners are shown. 13 parallel processors can be used for analysis of the slides (and parallel processing by image)[2]](image1.jpg)
To measure the speed up of parallel execution we are using the following relation which is defined in [12] and Image processing with multi core architecture is explained in [14].

**Observed Speedup:**
Observed speedup of a code which has been parallelized, defined as:

\[
\text{Observed Speedup} = \frac{\text{wall-clock time of serial execution}}{\text{wall-clock time of parallel execution}}
\]

And many more speed up measurement techniques are explained by Enrique Alba [13].

Parallel computing can also be applied by MATLAB. As we know that MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. It also provide efficient parallel computing tool box for multithreading.

**Parallel MATLAB:**
MATLAB is widely used for developing/prototyping algorithms. The High Level Language and Integrated Development/Visualization environment leads to productive code development by parallelizing MATLAB code

1. Algorithm can be run with different/larger data sets.
2. Algorithm can be run with larger parameter sweeps.
3. Compute times may be reduced.

So in this paper we are using MATLAB to do multithreading.

**III. IMPLEMENTATION**

Our implementation mainly consist of three phases:

1) Preprocessing & Segmentation of the image on client core or processor
2) Activation of threads in the cores and assignment of ROIs to different threads
3) Calculation of distinctive features of Regions as per requirements, Produces the result on client processor & De allocation of threads.

Main steps of our implementation are given in the following Flow Chart:
IV. RESULT AND DISCUSSION

This section describes the results obtained with the region wise parallel implementation. 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.
Evaluation

Two experiments are given to show the speed difference between the sequential computing and the parallel computing. We have tested above algorithm on 12 different sized cell images. And the efficiencies of the parallel computing implemented by these algorithms analyzed and compared respectively. In this paper of we are showing two cell image on different cores.

Experiment 1:
Hardware Configuration:
CPU: Inter (R) Pentium (R) Core i3 CPU 2.30 GHz.
RAM: 2GB.
Hard Disk Drive: 320 GB
Software Environment:
Operating System: Windows 7 Home Basic 64 bit.

<table>
<thead>
<tr>
<th>Images</th>
<th>Size</th>
<th>Active Threads in Cores</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>1</td>
<td>0.786</td>
</tr>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>2</td>
<td>2.15</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>1</td>
<td>0.98</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>2</td>
<td>1.68</td>
</tr>
</tbody>
</table>

Table 1: Calculation of speed up in Core i3 Processor

Experiment 2:
Hardware Configuration:
CPU: Intel(R) Xeon(R) E5640 @ 2.67 GHz
RAM: 12 GB
Software Environment:
Operating System: Windows 7 Home Basic 64 bit.
Development Tools: MATLAB R2012a

<table>
<thead>
<tr>
<th>Images</th>
<th>Size</th>
<th>Active Threads in Cores</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>1</td>
<td>0.87</td>
</tr>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>4</td>
<td>3.8</td>
</tr>
<tr>
<td>Cell1.tif</td>
<td>256 X 256</td>
<td>8</td>
<td>5.65</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>1</td>
<td>0.95</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>2</td>
<td>1.2</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>4</td>
<td>3.42</td>
</tr>
<tr>
<td>Cell2.tif</td>
<td>3172 X 1024</td>
<td>8</td>
<td>5.91</td>
</tr>
</tbody>
</table>

Table 2: Comparative Study of Sequential and Parallel Version
Result of segmentation and some statistical features calculation of different cell’s regions present in the image on client processor

**Figure 2:** Original Image

**Figure 3:** Segmented Image

**Figure 4:** Segmented Binary Image of a single cell

**Figure 5:** Centroids of different segmented image regions

**Figure 6:** Histogram of region1 in Thread of a activated core

**Figure 7:** Histogram of region2 Thread of a activated core
Fig. 8: Histogram of Region 3

Fig. 9: Histogram of Region 4

Fig. 10: Histogram of Region 5

Fig. 11: Histogram of Region 6

Fig. 12: Standard Deviations of Regions

Fig. 13: Mean of Regions
V. CONCLUSION & ITS APPLICATION IN MEDICAL IMAGING

We can see from the above experiment that processor having more cores produces good speed up if the algorithm is efficiently organized. This work presents region wise parallel processing of the image. By implementing these above we can see that all of the activated cores are able to handle their individual region. Histogram of different cell’s region is calculated by different threads activated in different cores in the same way we calculated different statistical features of different cell’s region. We can observe that this is a very promising result since it allows the exploitation of the enormous processing power of existing processors with multiple cores. The main focus of this implementation to improve performance of parallel image processing. In the future, the purpose is to use the same principle of division of regions present in the image to Medical Dicom images like CT, PET image of abdomen and assignments of different regions like liver, kidney, spleen etc to different activated kernels so that information gain is efficient and execution time is fast. This version would allow the segmentation of large dicom 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). 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


AUTHOR’S PROFILE

1. **Sanjay Saxena**, currently Pursuing PhD in School of Biomedical Engineering, IIT(BHU), Varanasi, UP, India. His research interests are Medical Image Processing, Parallel Computing, Computer Vision.

2. **Dr. Neeraj Sharma** is an Associate Professor at the School of Biomedical Engineering, IIT(BHU), Varanasi, UP, India. His research interests are Biomedical Image & Signal Processing, Bio Instrumentation.

3. **Dr. Shiru Sharma** is an Assistant Professor at the School of Biomedical Engineering, IIT(BHU), Varanasi, UP, India. Her research interest is Biological Control system,