# Image Processing: A Review

[1]Hardeep Singh, [2]Sonia Sharma
[1,2]Dept. of Computer Engineering & Technology, G.N.D.U, Amritsar, Punjab, India

**Abstract**
This paper presents an approach to the point to point processing of digital images using parallel computing, particularly for grayscale, brightening, darkening, and thresh-holding and contrast change. The point to point technique applies a transformation to each pixel on image concurrently rather than sequentially. This approach used CUDA as parallel programming tool on a GPU in order to take advantage of all available cores. Preliminary results show that CUDA obtains better results in most of the used filters. Except in the negative filter with lower resolutions images Open CV obtained better ones, but using images in high resolutions CUDA performance is better.

**Keywords**
Image Processing, Parallel Computing, CUDA, OpenCV, Taverna

## I. Introduction

Nowadays, Image processing plays a very essential role in numerous fields, for example, optics, computer science, mathematics, surface physics and visual psychophysics.In case of computer vision, its applications include remote sensing, feature extraction, meteorology, face detection, finger-print detection, optical sorting, astronomy, argument reality, microscope imaging, lane departure warning system. So point-to-point image processing is to do some changes in picture. It is basically the modification of image according to our requirement digitally. Image Processing with parallel computing is an alternative way to solve image processing problems that require largetimesof processing or handling large amounts of information in "acceptable time". The main idea of parallel image processing is to divide the problem into simple tasks and solve them concurrently, in such a way that 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 thatnot all theproblems can be coded in a parallel form.

## II. Previous Work

Sanjay Saxena, Neeraj Sharma, and Shiru Sharma, in 2013, presented their work on parallel implementation of different sequential algorithms. The main focus was on to improve the performance of segmentation, de-noising and histogram processing. They used multi-core architecture for designing some parallel processing algorithms like noise reduction, features calculation etc. RafalPetryniak, in 2008, examined an algorithm for edge finding which was then analyzed on different tests. He gave the strength and weakness of each parallel approach. Medical images were the main focus in the research. The conclusion drawn was that every algorithm is not dedicated to parallel computing and also parallel solutions can improve the efficiency of image processing. PreetiKaur, in 2013, calculated the various parameters such as fork time, serial time, join time, parallel time, and overheads. Her work deals with reducing the amount of time required to represent the digital images. This work helps to improve the performance of image processing algorithm and allows maximum utilization of the multi-core.

## III. System Design

Basic flow diagram of the steps of image processing is as follows:

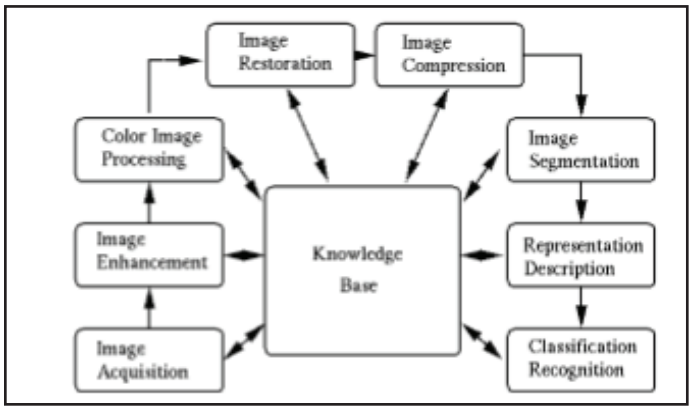


Fig. 1

## IV. Parallel Computation

### A. Features of a Parallel Program

A parallel program must have some features for a correct and efficient operation otherwise; it is possible that runtime or operation does not have the expectedperformance. These features include the following:–

1. **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:** An enormous number of small parts and less intense computing.
2. **Synchronization:** This prevents the overlap of two or more processes.
3. **Latency:** This is the time transition of information from request to receipt.
4. **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.
5. **Speedup and efficiency:** Theseare metrics to assess the quality of parallelimplementation.
6. **Overheads:** Extra time needed for the computation.

### B. CUDA (Computed Unified Device Architecture)

It is scalable parallel programming model and a software environment specifically used for parallel computing. CUDA is a parallel programming standard which is released in NVIDIA. Generally, it is used to develop software that are used for graphics processors and is used to build up a diversity of general purpose applications for GPUs that are tremendously parallel and run on hundreds of GPU's processors or cores. It uses a language that is very analogous to C language and has a high learning curve. It has some extensions to that language to use the GPU-specific features that include new API calls and some new type qualifiers that apply to functions and variables. It has some definite functions, which is called as kernels. It can be a function or a full program invoked by the Central Processing Unit. It also provides common memory and synchronization among threads.

### C. OpenCV

OpenCV (Open Source Computer Vision) is an open source library originally developed by Intel, which provides functions for creating real time applications of computer vision and machine learning. One for image processing and computer vision, another one for automatic learning and the last one that provides functions for handling image and video and graphic user interface for presentation. This library is written in C and C++ and can be run in environments such as Linux, Windows and Mac OS X. It is possible to obtain optimized codes using the "Integrated Performance Primitives" (IPP) library that has low-level optimized routines used in several algorithms.

## IV. Types of Parallel Processing

### A. Pipeline Parallelism

In this sort of processing, long sequences of procedures, or tasks, are parallel, but additionally, there are overlappingsuccessive processes all through which numbers of parallel tasks are possible. The relational model matches in to thatmodel really well. The result of some relational operatorsbecomes the input for other operators; thus, some waitingtime is involved. There is a considerable amount of timesaved in the completion of an activity through theappropriate use of direction parallelism.

### B. Independent or Natural Parallelism

In this sort of parallelism, the tasks don't rely on othertasks. As a result, complete delivery time is considerablyreduced.

### C. Task Parallelism

In parallel job strategy, image processingrecommendations of low level procedures are collected in to tasks and each job is given to another research unit. Picture processing software consists of several various operations. The key concern in job parallel strategy is Successful knowledge decomposition and effect composition.

## V. Algorithms For Parallel Image Processing Tasks

The key stage of these formulas is to determine how many tiles to be generated. How many tiles fit to the total amount of threads? If just a thread exists, the computation is simply successive computation. Otherwise if you can find two or more strings then the image is divided into distinct areas, as revealed in Determine 5, 6. Each thread isin charge of processing the pixels included in its tile and to accompish various tasks but considering the maintenance of synchronization between all the processors usually there will be the situation of deadlock between processors. Assume we're getting these images for instance

Fig. 1: An Image

Fig. 2: Division of Image into Vertical Threads

Fig. 3: Division of Image Into Horizontal Threads

## VI. Design Approach For Taverna-Based Parallelization of Imaging

The essential methods of Taverna–based application development are decomposition of computer software techniques into components. Two types of decomposition need to be differentiated: element decomposition and knowledge decomposition. Portion decomposition decomposes the application into elementary units, which may be viewed as a dark package between identified feedback knowledge and production data. How you can decompose the application usually follows these steps:

- Decomposition of the application into theoretically different components for easier reusability of these building blocks;
- Decomposition of large but independent components into smaller components, wherever possible, creating check pointing data that can be used if the task needs to be restarted.
- Decomposition of each component by differentiating the input and output parts to facilitate the data structure modification.
- Finally, a decomposition of each acquired component into objects with a log strategy to enable debugging and error tracking procedures.
- Data decomposition is easy to understand and is related to the decomposed components. Acomponent executed with a part of the data forms a task.

## VII. Conclusion

This paper centers on parallel computing in digital image processing tasks. Different options that come with parallel program and the requirements for better performance are given. The forms of parallel processing in image processing receive i.e., data, task and pipeline parallelism. We've also presented the algorithms for parallel image processing tasks.

Parallel Segmentation by Region Growing Technique and Calculation of different Top features of Segmented Regions, Parallel Segmentation by Global Thresholding of the image and Histogram Equalization of an Image by Parallel Computing would be the three algorithms given in detail.An application of parallel computing in medical imaging is discussed in detail. For imaging development, a workflow engine called Taverna is discussed and its design approach is also given.

## References

[1] Sanjay Saxena, Neeraj Sharma, Shiru Sharma,"Image processing tasks using parallel computing in multi core architecture and its applications in medical imaging", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 4, 2013

[2] E. G. Farrugia J.-P., Horain P., Y. Alusse,"Gpucv: A framework for image processing acceleration with graphics processors," In 2006 IEEE International Conference on Multimedia and Expo, pp. 585–588, 2006.

[3] J. Fung, 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.

[4] Z. Yang, Y. Zhu, Y. Pu,"Parallel image processing based on CUDA," In Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Vol. 03 (CSSE '08) , 2008, Vol. 3, pp. 198-201.

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

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