Parallel Image Processing Concepts

Sukhbeer Singh
Sri Sai College Of Engineering And Technology Badhani Pathankot -Punjab, ss.sukhbeer@gmail.com

Sarbjeet Singh
Sri Sai College Of Engineering And Technology Badhani Pathankot -Punjab, sarbaish@gmail.com

Sukhvinder Singh
Sri Sai College Of Engineering And Technology Badhani Pathankot -Punjab, sukhaish@gmail.com

Mandeep Kour
DAVIET Jalandhar -Punjab, er_mandeepdhaiwal@yahoo.in

Parallel Image Processing Concepts
by

Sukhbeer Singh- M Tech (CSE)[1], Sarbjeet Singh- M Tech (CSE)[2],

Sukhvinder Singh- M Tech (CSE)[3]. Mandeep Kour- M Tech (CSE)[4]

(Sri Sai College Of Engineering And Technology Badhani Pathankot -Punjab)[1,2,3]

(DAVIET Jalandhar -Punjab)[4]

Email- ss.sukhbeer@gmail.com[1], sarbaish@gmail.com[2], sukhaish@gmail.com3,
er_mandeepdhaiwal@yahoo.in[4]

Abstract:- *Image processing is a task of analysing the image and produces a resultant output in linear way. Image processing tasks are widely used in many applications domains, including medical imaging, industrial manufacturing, entertainment and security systems. Often the size of the image is very large, the processing time has to be very small and usually real-time constraints have to be met. The image analysis requires a large amount of memory and cpu performance, to cope this problem image processing task is parallelized. Parallelism of image analysis task becomes a key factor for processing a huge raw image data. Parallelization allows a scalable and flexible resource management and reduces a time for developing image analysis program. This paper presenting, the automatic parallelization of image processing task in a distributed system, in which suitable subtasks for parallel processing are extracted and mapped with the components of distributed system. This paper presents different design issues of parallel image processing in distributed system. Which helps the image analysis tasks that how to post processing the image in parallel. This technique is quite interactive especially when developing parallel program, as this requires little efforts for finding a suitable distribution of program module and data.*

Keywords:- Parallel Image Processing, Distributed system, objects, Basic Concepts, Design Issues, Parallel Processing Methods.

1. Introduction:-
The use of parallel libraries for image processing[1] is a common practice in the implementation of monolithic applications. Nowadays there is an increasing interest in moving towards distributed and heterogeneous applications also in the image processing community. The computational Grid may represent an adequate middleware support and infrastructure at this purpose. To improve the parallelism in a distributed system[2] for image processing, a multi based agent[4] is required. An agent is concept of artificial intelligent, a multi agent system is acommunity of agents which cooperate to solve common problems in parallel using image analysis tasks. Agents are grouped into five classes based on their degree of perceived intelligence and capability:

1. simple reflex agents
2. model-based reflex agents
3. goal-based agents
4. utility-based agents
5. learning agents

To analyse the image tasks in distributed system, the subtasks for parallel peocessing are extracted and mapped to the components of distributed system. This concept enables the multi agents to analyse the tasks in parallelization.

A distributed system consists of multiple autonomous computers[2][5] that communicate through a computer network. The computers interact with each other in order to achieve a common goal. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs.[3]

2. The parallel image processing objects:-
There are set of five objects[7] in parallel image processing which are obtained by grouping together the library objects:

- **Point Operations** Operations, that take in input only one IMAGE object, i.e. square root, absolute value, sine, or a

collective value, i.e. the maximum, minimum pixel value of an IMAGE;

- **Image Arithmetic** Operations, that take two input IMAGE objects, i.e. addition, product, subtraction etc;
- **Geometric Operations** Operations, that take one IMAGE object and one MATRIX object as input, i.e. rotation, translation and scaling;
- **Convolution Operations** Operations, that take one IMAGE object and one KERNEL object as input, for example involving a neighborhood of each pixel: percentile, median. or combined together by two binary functions;
- **Differential Operations** Operations, that take one IMAGE as input and perform differential operators, i.e. Hessian, gradient, Laplacian.

3. Basic concepts:-

The basic concepts of the models of parallel image processing, the level of digital image processing and the concepts of a multi-agent system[4][7] and its features.

3.1 Parallel architecture:-The models of parallel processing systems includes pipeline processing, asynchronous parallel processing and synchronous or data parallel processing.

Pipelining[8] is a technique of decomposing a sequential task into subtasks, with each subtask being executed in a sequential dedicated stage that operates concurrently with all other stages. Such architecture provides parallel vector processing by sequentially streaming the output results of one stage into the pipeline of another as input In Asynchronous parallel systems, each processor has its own control flow and so executes its own program. In data-parallel systems on the other hand, all processors or PEs (Processor Elements) receive their command at the same time on their potentially different local or data or are inactive (Vector or data parallel processing). Data parallel image processing operations map particularly well on linear SIMD arrays.

Hence in a data-parallel system there is only one sequential control flow and no independent asynchronous processes. A data-parallel system corresponds to the synchronous model of parallelism, an organization that includes many parallel processing units PPUs (or PEs) under the control of a common control unit.

3.2 Digital Image Processing Levels:- Digital Image Processing distinguishes two levels: low level and high level processing, the lower levels are also called image pre-processing whereas the higher ones are called image recognition, image understanding or computer vision, and belong to the area of artificial intelligence (AI)[2][9].

Lower level image processing usually converts image data into image data, (e.g. contrast enhancement, noise reduction or calculates simple characteristics of the input image, e.g. contours, histograms, transformations, etc…). In lower level processing interpretation of image content is irrelevant; it is only in image recognition that the data of pre-processed image is interpreted to attempt object recognition.

3.2.1 Laplacian Edge Detector:- The Laplace local image operator is one of the simplest edge detection algorithms in the field of image processing. An edge is made visible by using a neighbor's pixel value to suppress its own one. Each processor reads the neighbor's pixel value and subtracts it from the local one, which is multiplied by the number of the neighbors.

3.2.2 The Sobel Edge Detector :- The Sobel edge detector uses a simple convolution kernel to create a series of gradient magnitudes. For those you of mathematically inclined, applying convolution K to pixel group p can be represented as:

$$N(x,y) = \sum_{k=-1}^{1} \sum_{j=-1}^{1} K(j,k)p(x-j,y-k)$$

So the Sobel Edge Detector uses two convolution kernels, one to detect changes in vertical contrast (h_x) and another to detect horizontal contrast (h_y)

3.3 Agent-based Computing:- As the computing landscape moves from a focus on the individual standalone computer system to a situation in which the real power of computers is realized through distributed, open and dynamic systems, we are faced with new technological challenges and new opportunities. The characteristics of dynamic and open environments in which, for example, heterogeneous systems must interact, span organizational boundaries[10], and operate effectively within rapidly changing circumstances and with dramatically increasing quantities of available information, suggest that improvements on the traditional computing models and paradigms are required. In particular, the need for some degree of autonomy, to enable components to respond dynamically to changing circumstances while trying to achieve over-arching objectives, is seen by many as fundamental. While this notion is not intended to suggest an absence of

control, some application contexts offer no alternative to autonomous software.

A multi-agent system is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. Advantages of Reactive Agents are Speed and flexibility. The Reactive agents can act extremely fast in dynamic environments ("fast, cheap and out of control"). Surprisingly complex behaviours can be obtained using teams of simple reactive agents, Robustness due to simplicity and redundancy (in multi-agent system**s).**

Using agent technology for parallel image processing has got some evident advantages, especially in our case: The concurrent, cooperative working technique of a multi-agent system consists of several individual modules running on a distributed system and cooperating to solve common problems[11]. Agents are suitable for efficient, distributed planning [thus we will use them for planning the parallel processing of image analysis tasks. Agent can be used for management of distributed resources. Therefore, we will use agent for efficiently distributing data gathering result. Agents are suitable for parallel image processing. So the agents not only plan, but also perform the parallel processing.

4. Design issues:-

A distributed system consist of concurrent processes accessing distributed resources. Major Design Issues[12]
 Of distributed system in image processing are:-

- Object Models and Naming Schemes:- this issue helps in accessing the image in distributed environment using different naming scheme.
- Distributed Coordination:- this issue coordinates the different parts of a image amoung different agents.

- Interprocess Communication:- this helps in cooperating different items of a image.

- Distributed Resources:- to access resources by multi agents, resource distribution is required.

- Fault Tolerance and Security: helps in recovery of a image.

6. Quality:-

Their quality can be assessed using a well defined set of requirements that permits to verify the library effectiveness and efficiency. These requirements include:

- easy of use, the most compelling need for the users is a simple and natural tool to exploit the operations allowed, therefore a good point is represented by a well designed interface;
- transparency to the parallelism and the optimization policy, it is essential to provide tools that shield their users from the intrinsic complexities of parallel computations;
- efficiency, the users should be capable of obtaining significant performance gains in the most common image processing operations;
- portability, it is essential to ensure executions on different target machines, especially with the new emerging technologies and architecture;
- robustness, it is essential to provide computations that are insensitive to the variations of the data and ensure correct results;
- scalability, a well-designed software architecture has to increase its performance under the different used technologies, especially when resources are added;
- completeness, the users have not the necessity of using other packages related with the environment or the parallelism.

6. Methods of parallel process:-

A system and method for processing images includes a plurality of image providers configured to transmit images. A plurality of destination processors receives the transmitted images and transforms the transmitted images to internally useable image data. A plurality of feature object engines find and identify in the internally useable image data a plurality of objects. A plurality of object classifier engines index and classify the plurality of objects found by the feature object engines.

There are three types of parallelism[6][13]: task parallelism, data parallelism and pipelining as shown in figure-1.
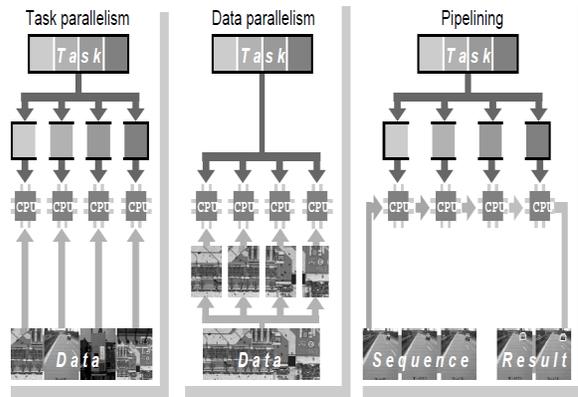

Figure-1- Methods of parallel image processing

Conclusion:-

This paper presents, a parallel image processing system based on the concept of reactive agents in distributed environment.

Refrences:-

1. J. Ferber, Multi-Agent System: An Introduction to Distributed Artifcial Intelligence, Addison-Wesley, Reading, MA, 1999.

2. F. Harrouet, P. Reignier, J. Tisseau, Multiagent systems and virtual reality for interactive prototyping, Proceedings of ISAS'99, Orlando, USA, Vol. 3, 1999, pp. 50–57.

3. J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 679–698.

5. R.M. Haralick, Ridges and valleys on digital images, Comput. Vision Graphics Image Process. 22 (1983) 28–38.

6. Maximilian LTuckenhaus, A multi-agent system for parallelizing image analysis tasks, Proceedings of IAS'98, Sapporo, Japan, 1998, pp. 579–586.

7. J. Liu, Y.Y. Tang, Adaptative image segmentation with distributed behavior based agents, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1999) 544–551.

8. P. Ballet, V. Rodin, and J. Tisseau, "A multiagent system for detecting concentric strias," in Applications of Digital Image Processing XX, vol. 3164 of Proceedings of SPIE, 30 July–1 August 1997pp. 659–666.

9. LITWINOWICZ P.: Processing Images and Video for an Impressionist Effect. In Proceedings of ACM SIGGRAPH 97 (New York, 1997), Whitted T., (Ed.), Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH, pp. 407–414.

10. HAEBERLI P.: Paint By Numbers: Abstract Image Representations. Computer Graphics (Proceedings of ACM SIGGRAPH 90) 24, 4 (1990), 207–214

11. G. Weiss. Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.

12. M. Wooldridge, N.R. Jennings, D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems, , 2000, 3(3) 285-312.

13. K. S. Decker, J. Garvey, A. Humphrey, and R. Lesser. "A real-time control architecture for an approximate processing blackboard system". International Journal of Pattern Recognition and Artificial Intelligence, 7(2): , 1993 265—284.