

A Parallel Video Frame Segmentation Approach on Multicore Processors

Basavaraj Bagewadi¹, Chaitra Jatti², Richa Koregaonkar³, Soumyashree Somasagara⁴,
Satyadhyan Chickerur⁵ and Pramit Mehta⁶

^{1,2,3,4}*Department of Information Science and Engineering*

⁵*Centre for High Performance Computing*

⁶*Collaborator, Centre for High Performance Computing
B V Bhoomaraddi College of Engineering and Technology*

Hubli, Karnataka, India

chickerursr@bvb.edu

Abstract

Image Segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Pre-processing 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 presents the performance analysis of parallelizing the video frame segmentation using region growing technique on varying of processors. Region growing Technique is also classified as pixel-based image segmentation method since it involves the selection of initial seed points where determining of whether the pixel neighbors should be added to the region or not is done. The process is iteratively carried out and the region is correspondingly grown for each image. Video is divided into frames, which are passed to different cores. The processing of frames is carried out simultaneously in cores, hence parallelizing the technique. Time analysis for serial and parallel implementation is done. Analysis of time required for processing the frames on different number of cores is carried out.

Keywords: *Multicore Processing, Segmentation, Parallelizing, Region growing*

1. Introduction

Parallel Computing is all about performance [1,4]. Parallel Computing uses more than one core simultaneously to execute a program in order to achieve faster processing. Serial implementation of a program is done using a single core where as parallel implementation of the same uses more than one core where the workload is balanced among different processors. Hence efficient hardware utilization is achieved. Serial implementation is not the appealing approach in cases where large amount of information has to be processed. A sequential program will only run on one of the processor cores, which will not become significantly faster than those in use today. In parallel programs, concurrent execution of cores cooperates in faster completion of work. There is a growing interest for video processing in multimedia applications. The processing of frames in such videos plays a vital role, which has to be done in acceptable time.

Video is accepted as input, which is divided into frames. These frames are divided among different cores. These frames are subjected to segmentation to extract region of interest. This is done using region growing technique. The processing begins with the specification of initial seed points. The regions are iteratively grown by comparing all unallocated

neighboring pixels to the regions. The difference between a pixel's intensity value and the region's mean is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region. Ultimately, concurrent processing of many frames using different cores results in the division of large task. These large tasks are further divided into smaller tasks facilitating faster execution. There are so many different medical image modalities present like CT, PET, MRI etc. These modalities are having different characteristics and used as per requirements. In such medical applications, the size of the images is very large, which is a concern. Accuracy plays a major role here. If executed serially, as the size of the image increases, processing time correspondingly increases. In order to get the accurate results, the image is segmented into blocks and given to different processors parallelizing the process [5]. Thus for different sized images, time taken for processing is less in parallel compared to serial implementation. But in case of traffic applications, series of frames are to be considered where accuracy is not a concern but speed of processing plays a major role. Thus instead of segmenting a single image into blocks and giving it to different processors, series of frames are distributed accordingly to different processors. This makes the process efficient by reducing the processing time. In the work done as proposed in this paper, importance is given to the processing speed.

2. Methodology

2.1 Region Growing Technique

Region growing is a simple image based segmentation method [2, 3]. It is also classified as pixel-based image segmentation method since it involves selection of initial seed points. Neighboring pixels of initial seed points are examined. Determine whether these pixels should be added to the region or not. This process is iteratively carried out until the regions are grown.

2.2 Implementation

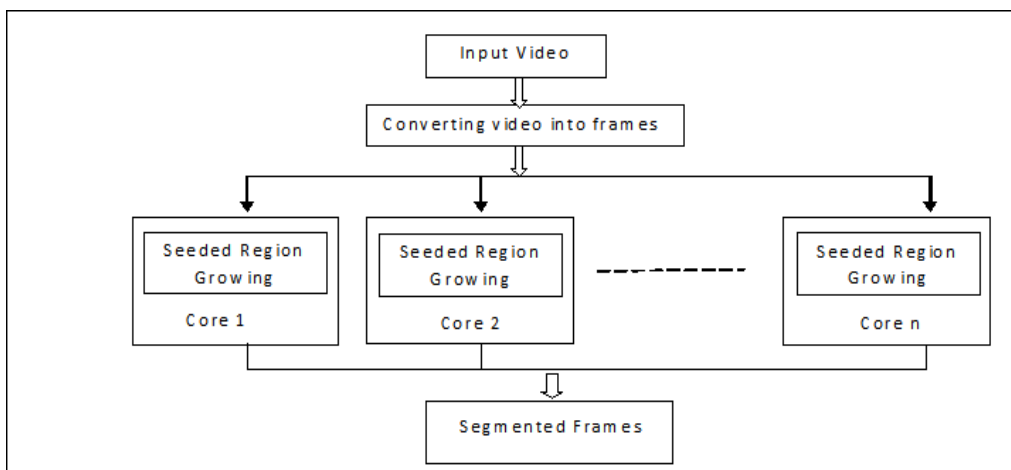


Figure 1. Distribution of Work among Cores

Video is divided into frames. Different lengths of videos are taken and corresponding frames are extracted in order to analyze the results. The video frames are distributed among different cores. For *e.g.*, if 6 cores are considered, 6 images are processed at a time in 6 cores correspondingly as shown in the Figure 3. Each frame is subjected correspondingly to region growing technique. In the region growing technique, the initial seed pixels are determined as shown in Figure 2. The initial region begins at this location of seed points. The region is hence grown from these seed points to adjacent points. The region is iteratively grown by comparing all unallocated neighboring pixels to the region. The difference between a pixel's intensity value and the region's mean is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process stops when the intensity difference between region mean and new pixel become larger than a certain threshold (t). The processed images are stored. Figure 3 shows the segmented images for the corresponding given video frames.

Region based segmentation is a technique for determining the region of interest directly. The basic formulation for region-based segmentation as in [1] is:

- (a) $\bigcup_{i=1}^n R_i = R$
Means that the segmentation must be complete, that is, every pixel must be in a region
- (b) R_i is a connected region, $i=1,2,\dots,n$
Requires that points in a region must be connected in some predefined sense
- (c) $R_i \cap R_j = \emptyset$ for all $i=1,2,\dots,n$
Indicates that the regions must be disjoint
- (d) $P(R_i) = TRUE$ for $i=1,2,\dots,n$
Deals with the properties that must be satisfied by the pixels in a segmented region
- (e) $P(R_i \cup R_j) = FALSE$ for any adjacent region R_i and R_j
Indicates that region R_i and R_j are different in the sense of predicate P

$P(R_i)$ is a logical predicate defined over the points in set R_i and \emptyset is the null set



Figure 2. The Determination of Seed Pixels for the Given Image

Both serial and parallel implementations are carried out. Serial implementation of region growing segmentation uses only one core and single frame is processed at a time. Hence the time consumption increases when whole video is processed. As the video size increases, the number of frames increases leading to increase in the serial processing time. Parallel implementation makes use of all the cores. Cores divide the frames among themselves and processing takes place in parallel. Thus time taken for parallel processing is lesser than serial processing when the number of processing frames increases. Time taken for both serial and parallel implementation is recorded.



Figure 3. Shows the Original Frame in the First Column and Corresponding processed image using Region Growing Technique in the Second Column where Number of Cores Considered is 6

3. Results and Analysis

Images and video frames are snapshots of a physical world where different parts of a picture capture simultaneous, independent physical events. Such events where large amount of data has to be processed incur long execution times. The proposed algorithm is aimed at reducing the processing time of video frame segmentation in parallel manner and is carried out by dividing the work among different cores. Considering videos of different sizes i.e. varying the number of frames, serial and parallel execution times are recorded as shown in Table 1.

Table 1. The Serial and Parallel Processing Time for Different Number of Video Frames

Video size (no of frames)	Parallel processing time (Minutes)	Serial processing time (Minutes)
100	7	12
200	17	24
300	28	35
500	50	58
600	48	77
900	97	106

The performance variation in serial and parallel execution is depicted graphically in Figure 4. As per the graph, parallel execution takes less processing time as compared to serial execution.

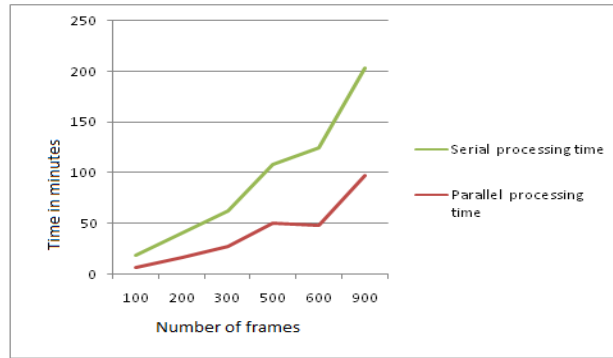


Figure 4. Comparison of Serial and Parallel Processing time with Respect to Number of Frames

Analysis of performance of parallel computing is carried out by variation in the number of cores and the corresponding execution times in varied cores are recorded as shown in Table 2. Graph is plotted for the same as shown in Figure 5. The graph shows that processing time decreases with the increase in the number of cores.

Table 2. Shows the Time Taken for Processing when Different Cores are Used

Number of cores	Time (in minutes)
2	145
4	119
6	101
8	93
10	77
12	73

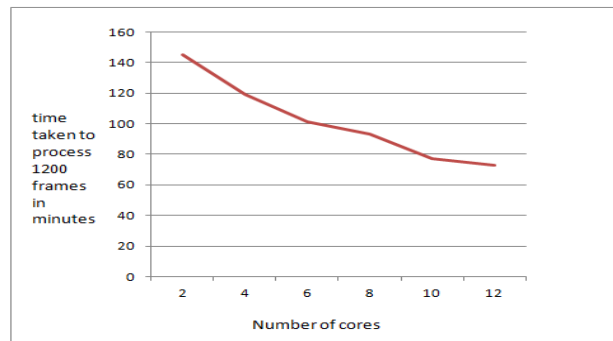


Figure 5. Graph Showing Comparison of Processing Time Taken for Different Cores in Processing 1200 Images

4. Conclusion

This work presents the comparative study of serial as well as parallel implementation of video frame segmentation using Region Growing Technique. Different lengths of videos are taken. Processing of frames is done and the resultant segmented images are stored. Serial as well as parallel implementation time is recorded. As the video size increases *i.e.*, the number of frames increases, processing time decreases in parallel implementation. The focus of this implementation is to increase the performance. Due to load balancing among different

processors, the effective results are obtained in parallel implementation. This process of parallelizing the region growing technique is distributed among different cores. The work is carried out by executing the program in 12 cores, 10 cores so on up till 2 cores. The number of cores used implies the number of images processed at a time. It is observed that the processing time decreases as the number of cores increases.

Acknowledgments

The authors would like to thank the Management, the Principal, Vice Principal, Dean R & D and Head of the department ISE for support and encouragement. The authors would also like to thank members of the Crest Division of KPIT, Pune The authors appreciate the anonymous referees for suggesting improvements in the paper.

References

- [1] T. Bräunl, “Tutorial in Data Parallel Image Processing”, Australian Journal of Intelligent Information processing Systems (AJIIPS), vol. 6, no. 3, (2001), pp. 164–174.
- [2] H. Zhang, J. E. Fritts, S. A. Goldman, “Image Segmentation Evaluation: A Survey of Unsupervised Methods”, Computer Vision and Image Understanding, vol.110, no.2, (2008), pp. 260-280.
- [3] A. K. Jain, “Digital Image Processing”, (1989), Prentice Hall.
- [4] P. Pacheco, “An Introduction to Parallel Programming”, Morgan Kaufmann, (2011).
- [5] P. Bertels and D. Stroobandt, “Java and the Power of Multi-Core Processing”, the Second International Conference on Complex, Intelligent and Software Intensive Systems, CISIS (2008), pp. 627 – 631.

Appendix: Results of various frames from real time video and the results using the proposed approach

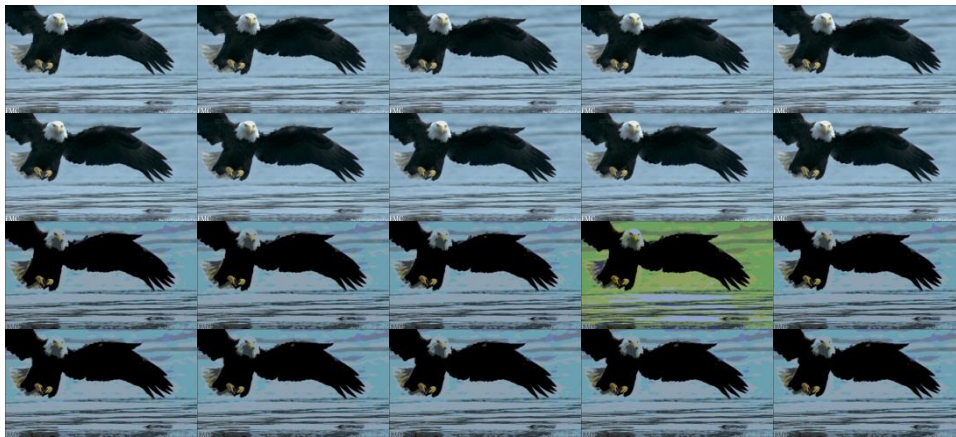


Figure A-1. First Two Rows Show the Input Video Frames and the Next Two Rows are Results of the Proposed Algorithm on the Input Frames



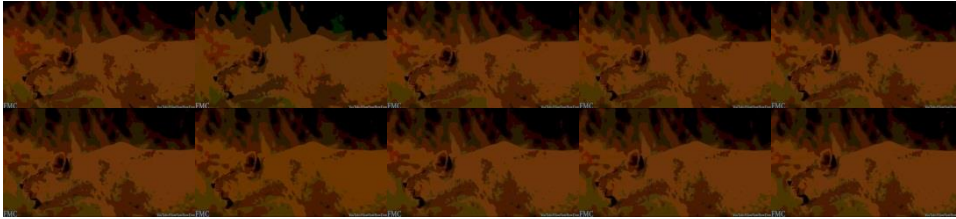


Figure A-2. First Two Rows Show the Input Video Frames and the Next Two Rows are Results of the Proposed Algorithm on the Input Frames



Figure A-3. First Two Rows Show the Input Video Frames and the Next Two Rows are Results of the Proposed Algorithm on the Input Frames

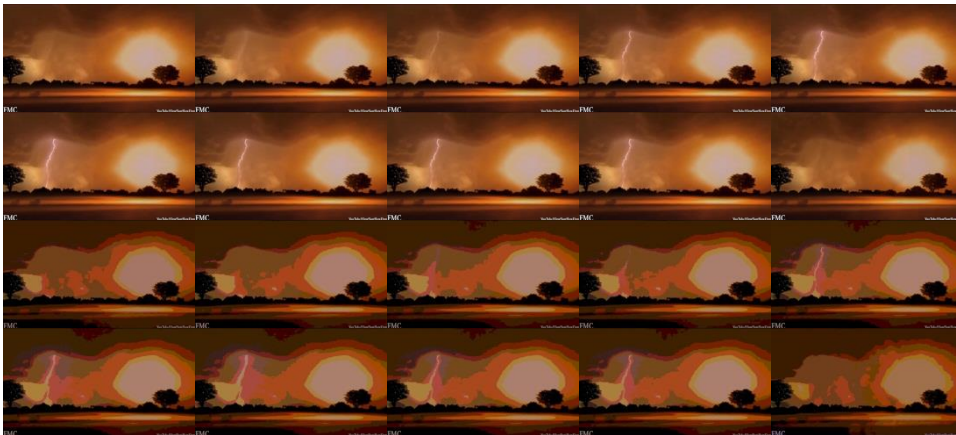


Figure A-4. First Two Rows Show the Input Video Frames and the Next Two Rows are Results of the Proposed Algorithm on the Input Frames

