MPI based Edge Detection of Coloured Image using Laplacian of Gaussian Filter

Prakash K. Aithal
Manipal Institute of Technology,
Manipal University, Manipal

U. Dinesh Acharya
Manipal Institute of Technology,
Manipal University, Manipal

Rajesh G
Manipal Institute of Technology,
Manipal University, Manipal

ABSTRACT

Edge detection is the process of identifying points in a digital image at which image brightness changes sharply. Edge detection is one of the fundamental tasks in Image Processing. Image processing tasks such as object identification, segmentation and robot vision requires high quality edge detection. In this paper an edge detection using Message Passing Interface (MPI) has been implemented with 2, 4 and 8 processes. The paper compares the result with sequential implementation of edge detection. There is an improvement in performance as we increase the number of processes. Depending on the image size we can increase the number of processes. MPI is more suitable for a large size images like geospatial image. The edge detection using MPI is number of processes time faster than its sequential counterpart and speed up is approximately equal to number of processes used. The performance improved by a factor of 5 with 5 processes for colored image and by a factor of 4 for grey scaled image.

General Terms

Image processing, High performance computing.

Keywords

Edge detection, MPI, Laplacian of Gaussian Filter.

1. INTRODUCTION

Image edge detection is crucial for image analysis applications such as image representation, segmentation and robot vision [1]. The purpose of edge detection is to discover the information about shape and reflectance in an image. [2]. Many sequential implementation of edge detection can be found in the literature. A comparative study of different edge detection algorithms can be found in [3]. As many image processing applications edge detection also can be parallelized. Edge detection using MPI is first of its kind. The remaining paper is organized as follows. Section 2 discusses sequential edge detection approach using Laplacian of Gaussian filter. Section 3 explains edge detection using Laplacian of Gaussian filter with the help of n processes in MPI. Section 4 elaborates image reconstruction. Section 5 gives the results analysis.

2. SEQUENTIAL EDGE DETECTION USING LAPLACIAN OF GAUSSIAN FILTER

Image matrix is considered and convoluted with Laplacian of Gaussian filter to get Convolution matrix. Entries in the Laplacian of Gaussian filter are as follows.

$$\begin{matrix} 0 & 3 & 0 \\ 3 & -9 & 3 \\ 0 & 3 & 0 \end{matrix}$$

The Laplacian of Gaussian filter has advantage that only one filter is sufficient. When you convolute the image with this matrix you get the convolution matrix. The resultant matrix is the edge detected image.

The following algorithm explains how to find the edge of an image sequentially.

```
1.  Read the jpeg image.
2.  Convert the image into a  matrixof type double.
3.  Sum=0;
4.  For y= 1 to image_width-1 in step of 1 do
5.  For x=1 to image_height-1 in step of 1 do
6.  For k=-1 to 1 in step of 1 do
7.  For j=-1 to 1 in step of 1 do
8.  sum = sum + LaplacianOfGaussian Filter
    [j+1][k+1]*image[y - j][ x - k];
9.  Assign sum to  imgeedge[y][x].
10. End
11. End
12. End
13. End
14. Write imageedge to a jpeg file.
```

The imageedge matrix corresponds to the edge detected image. Instead of 3X3 filter if we use a 5X5 filter then quality of edge detection will be high. A 5X5 Laplacian of Gaussian filter is depicted below.

$$\begin{matrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{matrix}$$

To encorporate the 5X5 filter one need to change the lines 4-7 in the above algorithm as follows

```
4.  For y= 2 to image_width-2 in step of 1 do
5.  For x=2 to image_height-2 in step of 1 do
6.  For k=-2 to 2 in step of 1 do
7.  For j=-2 to 2 in step of 1 do
```

Time complexity of the algorithm is O(image_width_of_image * image_height_of_image * image_width_of_filter * image_height_of_filter)

3. EDGE DETECTION THROUGH MPI

Image is taken from Sample Imagery List of GeoEye Inc. (formerly Orbital Imaging Corporation or ORBIMAGE) which is an American commercial satellite imagery company. Image taken is used for testing the MPI based edge detection algorithm. The Image is divided among the processes horizontally. The following algorithm will describe the process of finding the edge through MPI.

1. Find the size of the processes. Size is number of processes in the group.
2. Find the rank of processes. Rank of a process is task id of the process.
3. Read the jpeg image.
4. Clone the image.
5. Scatter the image among n processes.
6. Calculate the convolution matrix as mentioned in section 2 for each of the sub images which are scattered.
7. Now once the convolution matrix is calculated in each of the process store the resultant edge detected sub images as jpeg images.
8. For N processes you will get N edge detected sub images.

After edge detected sub images are generated pass it onto the image reconstruction phase. Image reconstruction phase combines the edge detected sub images into one edge detected complete image.

Time complexity of the algorithm is O(image_width_of_image * image_height_of_image * image_width_of_filter * image_height_of_filter)/number of processes+overhead.

4. IMAGE RECONSTRUCTION

Steps to reconstruct image is as follows

1. Read the each sub image one by one in order.
2. Add only those pixels whose value is below certain threshold to the image under construction.
3. If any pixel value is less than the threshold value in a row then add the entire row to the image under construction. This guarantees that the sub images will never overlap neither in horizontal nor in vertical direction.

Time complexity of the algorithm is O(image_width_of_image * image_height_of_image)

5. RESULTS

Sequential algorithm of finding edges in an image is slower and quality of the image will be low. We cannot apply sequential algorithm in case of large geo spatial images. A comparative study of sequential and parallel version of edge detection is tabulated in Table1. Figure 1 displays the original image. Figure 2 shows the edge detected image using sequential algorithm. Figure 3-5 depicts edge detected sub images from multiple processes. Reconstructed image is shown in figure 6.If edge detection algorithm is applied on grey scale image then it will be even faster since there algorithm has to process only one dimension. Though we get performance improvement we will lose information in the process. A comparative study of time taken by sequential and parallel implementation on grey scale image is given in Table2. If number of processes are more it reduces the rounding of errors which in turn increases accuracy and quality of results[4].

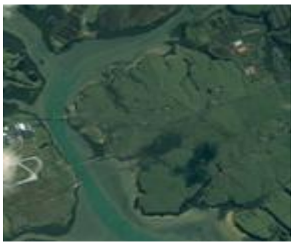

Fig 1: Original geospatial image (Image is taken from Sample Imagery List of GeoEye Inc. (formerly Orbital Imaging Corporation or ORBIMAGE) which is an American commercial satellite imagery company.)

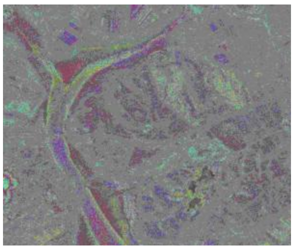

Fig 2: edge detected through sequential algorithm

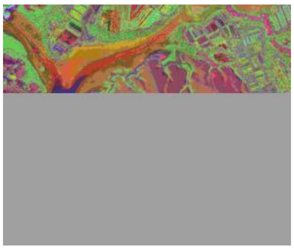

Fig 3: Sub image generated by process 1

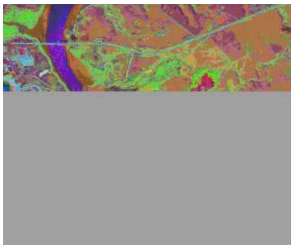

Fig 4: Sub image generated by process 2

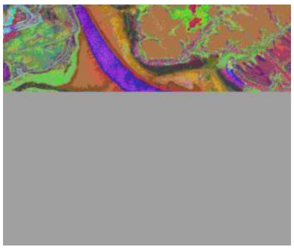

Fig 5: Sub image generated by process 3

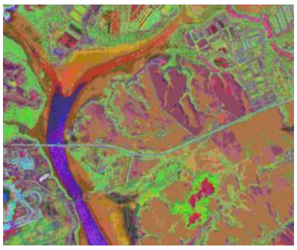

Fig 6: Reconstructed image

Table 1.Comparison of time taken by processes (including image reading and writing for colored image of size 8.65MB and 3000X3000pixel dimension)

Number of Processes	Time taken to detect the Edge	Time taken to reconstruct the image	Total time taken
1	20minutes	0minutes	20minutes
2	10minutes	1minute	11minutes
3	7minutes	1minute	8minutes
4	5minutes	1minutes	6minutes
5	4minutes	1minute	5minutes
6	4minutes	1minute	5minutes
7	4minutes	1minute	5minute
8	3minutes	1minute	4minutes

From the above table it is clear that as the number of processes increases the time reduces by a factor of number of processes for edge detection. The number of processes to be used is upper bound by size of the image. Image reconstruction time is just one minute and does not depend upon the number of processes involved rather on size of the image. For image of size 8.65MB and 3000X3000 pixels dimension there is no performance improvement if we increase the processor number more than 8.

Table 2.Comparison of time taken by processes (including image reading and writing for grey scale image of size 8.65MB and 3000X3000 pixel dimension)

Number of Processes	Time taken to detect the Edge	Time taken to reconstruct the image	Total time taken
1	8minutes	0minutes	8minutes
2	5minutes	1minute	6minutes
3	4minutes	1minute	5minutes
4	3minutes	1minutes	4minutes
5	2minutes	1minute	3minutes
6	2minutes	1minute	3minutes
7	2minutes	1minute	3minute
8	1minutes	1minute	2minutes

6. CONCLUSION

The paper compared edge detection done sequentially to that done in parallel. The results got clearly shows that MPI based edge detection algorithm out performs sequential version both in time and image quality. Further work can be taken to identify the structures in images such as rivers, cultivation land, home, buildings, bridges etc. in a geospatial image and implement it using clusters.Further improvement in terms of quality can be improved by increaing the filter size to 5X5 or 7X7.

7. REFERENCES

[1] ArpitaGopal, SonaliPatil, AmreshNikamInternational Journal of Computer Science and Application Issue 2010. A Parallel Algorithm for Image Edge Detection using difference chain encoding.

[2] Lei Zhai, Shouping Dong, Honglian Ma. 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing.Recent Methods and Applications on Image Edge Detection

[3] Raman Maini, HimanshuAggarwalStudy and Comparison of Various Image Edge Detection Techniques. International Journal of Image Processing (IJIP), Volume (3) : Issue (1).

[4] Rajiv Chopra, Advanced computer architecture- A practical approach. S chand company ltd, first edition 2009.

[5] Image is taken from Sample Imagery List of GeoEye Inc. (formerly Orbital Imaging Corporation or ORBIMAGE) which is an American commercial satellite imagery company.