I J C T A, 9(15), 2016, pp. 7503-7510

An Improved Approach with Iterative Feature Extraction for Image Processing

Kamaljeet Dash*, Manjusha Pandey* and Siddharth S. Rautaray*

ABSTRACT

Due to the increasing popularity of cheap digital photography equipment, personal computing devices with easy to use cameras, and an overall improvement of image capture technology with regard to quality, the amount of data generated by people each day shows trends of growing faster than the processing capabilities of single devices. For oth'11""er tasks related to large-scale data, humans have already turned towards distributed computing as a way to side-step impending physical limitations to processing hardware by combining the resources of many computers and providing programmers various different interfaces to the resulting construct, relieving them from having to account for the intricacies stemming from its physical structure With the aid of freely available implementations of this model and cheap computing infrastructure offered by cloud providers, having access to expensive purpose-built hardware or in-depth understanding of parallel programming are no longer required of anyone who wishes to work with large-scale image data the issues of processing two kinds of such data - large data-sets of regular images and single large images using Map Reduce Finally, the application of distributed image processing on two example cases: a 265GiB data-set of photographs and a 6.99 Giga pixel image. Both preliminary analysis and practical results indicate that the Map Reduce model is well suited for distributed image processing in the first case, whereas in the second case, this is true for only local non-iterative algorithms, and further work is necessary in order to provide a conclusive decision.

The power of parallel computing on the GPU against the massive computations needed in image processing of large images. The GPU has long been used to accelerate 3D applications. With the advent of high level programmable interfaces, programming to the GPU is simplified and is being used to accelerate a wider class of applications. More specially, focuses on CUDA as its parallel programming platform. This thesis explores on the possible performance gains that can be achieved by using CUDA on image processing. Two well-known algorithms for image blurring and edge detection is used in the experiment. Benchmarks are done between the parallel implementation and the sequential implementation.

1. INTRODUCTION

Along with the development of information technology, a problems has also appeared. As we possess more computing power, we can tackle more and more resource intensive problems such as DNA sequencing, seismic imaging and weather Simulations. When looking at these subjects, a common theme emerges all of these involve either analysis or generation of large amounts of data. While personal computers have gone through a staggering increase in power during the last 20 years, and the processing power even within everyday accessories such as smartphones is very capable of solving problems that were unfeasible for supercomputers only a couple of decades ago, analysing the amount of data generated by newest generation scientific equipment is still out of reach in some areas.

Moreover, as processor architectures are reaching their physical limitations with regard to how small individual logic gates and components can get, using distributed computing technologies has become a popular way to solve problems which do not fit the confines of a single computer. Supercomputers, GRID-based systems and computing clouds are an example of this approach. Since the fields of distributed

* School of Computer Engineering KIIT University Bhubaneswar, India, *Emails: kamaljeetdash@gmail.com, manjushafcs @kiit.ac.in, siddharthfcs@kiit.ac.in*

computing and image processing are too broad to fully cover in this thesis, this work will focus on the latter of the three with regard to image processing. Due to the increasing popularity of personal computers, smart televisions, smartphones, tablets and other devices carrying a full-fledged operating system such as Android, iOS or Windows 8, and due to the capability of these devices to act as producers of many kinds of content instead of being passive receivers (like radio and television, for example), there is a need to be able to process that content. Photos need to be resized, cropped and cleaned up, and recorded sound and video need to be shaped into a coherent whole with the aid of editing software.

These procedures however may not be something that is best tackled on the same device that was used for recording, because of limiting factors in processing power, storage space and in some cases battery life. However, with the widespread availability of wireless internet or high-throughput cell phone networks, any of the aforementioned devices can simply upload their data to a more capable computer in order to do necessary processing. In many cases the recorded media will be consumed using a different device (for example, viewing holiday photos taken with your smartphone on your computer or smart TV). Therefore, it can be argued that both the steps of transferring media from the recording device and processing it are inevitable anyway. Facebook and YouTube both provide a good example of this scenario: the user can upload their media in more or less in an unprocessed format and the frameworks take care of resizing and re-encoding the media so that it can be consumed by users. However, since these services are very popular, as a consequence the amounts of data that is needed to process are also huge. For example, 72 hours of video data is uploaded to YouTube every minutes.

Even without going into details of video compression or the processing pipelines involved, it is easy to see how even a day’s worth of uploads (103nfeasible to 680 compute hours) without resorting quickly to distributed computing becomes. u For solving processing tasks involving data of this scale, engineers at Google (the parent company of YouTube) designed the map Reduce model of distributed computing, of which Apache Hadoop is the most popular open source implementation. In this, it’s implementation in the form of Hadoop, mageand explo processing.

1.1. Image processing pipeline

The main aim of the image processing pipeline is to provide a proof of concept solution to the tasks of recognising certain objects and extracting information by way of optical character recognition (OCR). As mentioned before, this is by no means a working solution that is ready to be applied in real world situations. However, a superficial analysis of the results suggests that, with some optimisation and tuning, it is suitable for extracting some information out of the data-set. The following is a rough description of each of the steps in the process:

1. The object script performs object recognition and returns a list of matching pixel coordinates in the target image. In case of several pixels, the average is calculated. If no pixels were returned, halt processing.
2. Thresholding and labelling on the target image in order to convert it to a list of regions.
3. Erosion to eliminate regions that are too small.
4. Dilatation to bring the regions that remain back to their original size.
5. Calculate bounding boxes for all the remaining regions.
6. Based on the pixel coordinates from step 1, select the region that is located at these coordinates. If there is no region, halt processing.
7. The top left and bottom right pixel coordinate of the selected region specifies the area to extract from the target image.

1.2. Processing a large image using a local non-iterative algorithm

A practical application of distributed image processing in the scenario of applying a local non-iterative algorithm on a image with large spatial dimensions. The rest of this section is structured as follows first, a description of the image itself and the motivation behind the task. Further on, I will outline the divide-and-conquer approach used in splitting the image into manageable pieces, briefly describe the specifics of the MapReduce implementation of the logarithm a was measured and compared with its non-distributed (sequential) counterpart

1.3. Gaussian blur

One of the simplest local algorithms in image processing is Gaussian blur (also known as Gaussian smoothing). It's most common application is noise reduction. The bilateral filter algorithm described later on in this thesis is an improvement of Gaussian blur with regard to edge-preservation capability.

1.4. Image Processing and CUDA

Image processing is a type of signals processing in which the input is an image, and the output can be an image or anything else that undergoes some meaningful processing. Converting a coloured image to its grayscale representation is an example of image processing. Enhancing a dull and worn fingerprint image is another example of image processing. More often than not, image processing happens on the entire image, and the same steps are repeatedly applied to every pixel of the image. This programming paradigm is a perfect candidate to fully leverage CUDAs massive compute capabilities.

This section will compare the performance differences between software that are run on a sequential processor (CPU) and a parallel processor (GPU). It will consist of performing various image processing algorithms on a set of images. Image processing is ideal for running on the GPU because each pixel can be directly mapped to a separate thread. The experiment will involve a series of image convolution algorithms. Convolutions are commonly used in a wide array of engineering and mathematical applications. A simple high-level explanation is basically taking one matrix (the image) and passing it through another matrix (the convolution matrix). The result is your convoluted image. The matrix can also be called the filter.

2. PROPOSED WORK

Image segmentation is a fundamental problem in image processing and computer vision. Extensive study has been made and many techniques have been proposed among which the active contour. The basic idea of active contour model is to evolve a curve under some constraints to extract the desired object. According to the nature of constraints, the existing active contour models can be categorized into two types: edge-based models.

The foremost essential goal of the segmentation process is to partition an image into regions that are homogeneous (uniform) with respect to one or more self- characteristics and features. Active contour methods are applied in a wide range of problems including visual tracking and image segmentation.

Algorithm-1: Edge Detection Algorithm-

Step1: Create kernel h indexed from 0 to m-1 horizontally and 0 to n-1 vertically and populate it with Kernel Coefficients.

2) Compute kernel half width, m2 = oor

(m/2) Compute kernel half height, n2 = oor

(n/2) 3) sum = 0

for k = -n2 to n2 loop

for j = -m2 to m2 loop

sum = sum + h(j + m2, k + n2) f(x - j, y - k)

end loop

end loop g(x, y) = sum

For a single pixel

Convolution of an image ignoring the borders

1) Create kernel h indexed from 0 to m-1 horizontally and 0 to n-1 vertically and populate it with kernel Coefficients.

2) Compute kernel half width, m2 = o or (m/2)

Compute kernel half height, n2 = o or (n/2)

3) Create an M x N output image, g

4) for all pixel co-ordinates, x and y, loop g(x, y) = 0 end loop

5) for y = n2 to N-n2-1 loop

or x = m2 to M-m2-1 loop

Compute g(x, y) using previous algorithm

end loop

end loop

The detection of the abnormal nuclei is based on a morphological image reconstruction process.

The segmentation of the nuclear boundaries is accomplished with the application of the watershed transform in the morphological colour gradient image, using the nuclear markers extracted in the detection step. An unsupervised nucleus segmentation method based on a water immersion algorithm was presented in [2]. In addition, methods based on Hough transform which take advantage of the expected similarity in nuclei shapes, have also been introduced [3]. A combination of the generalized Hough transform and deformable models is used in [4,5], in order to and a set of templates specific to nuclear shape. Furthermore, a fuzzy logic engine has been proposed for the cell nuclei segmentation [6,7].

Algorithm formulated is as follows:

For i = l: N

For j = l: M

If Q(x, y) = 255

Then

P(x, y) = P(x, y)

Else

P(x, y) = 255

End

Here P represents the original cell image of size xN in the grey scale format and Q represents the segmented binary image. Q has the value of 255 for all pixels inside the nucleus and 0 in other regions.

In order to obtain the image for grey value calculation the algorithm formulated

Then we obtain P(x, y) which will contain nucleus pixels as in the original grey scale image and all nonnuclear pixels replaced by 255.

For the process of extracting features of sequential image frames after multiple image frames are input to slave servers that execute a Map function, only the Map function is used to output a significant number of features for each image. First, when a pixel of image frame contains a significant amount of features, such as corners and edges, a new pixel is generated in the output image, and the pixel value is set to be a feature value. Otherwise, a black pixel is generated in the output image. Only the Map function is used to output a significant number of features for each image

Input Split a chunk of the input that is processed by a single map.

Record Reader An input split is divided into records, and the map processes each record (key-value pair) in turn. A Record Reader is used to generate record key-value pairs.

Splitting Logic Assumption: Size of an Image Split will not be greater than HDFS block size. Consider image of dimensions 20 x 35 pixels. Based on numerical Splits argument, the image is divided

The above figure 1 represents counter model specimen of cancer cell with noise which is combined with the blood platelets and lumens. This is the raw image taken directly with electronic microscope and a high definition cameras (SLR cameras, DSLR cameras). Under the day light conditions for better resolution.

The above figure 2 here represents the same cell image taken in the figure 1 after a single iteration and filling, where the noise is removed. By noise here we refer to the blood platelets and lumen that are not required for the analysis and diagnosis of cancerous cells. The noise removal after iteration and filing helps to achieve the enhanced and concentrative input for the next level of RGB content removal for further image iteration.

After iterations and one time filing the image is further input for next level filing in order to sharpen the RGB contents. The RGB content filing here is the removal of RGB content which makes the image more

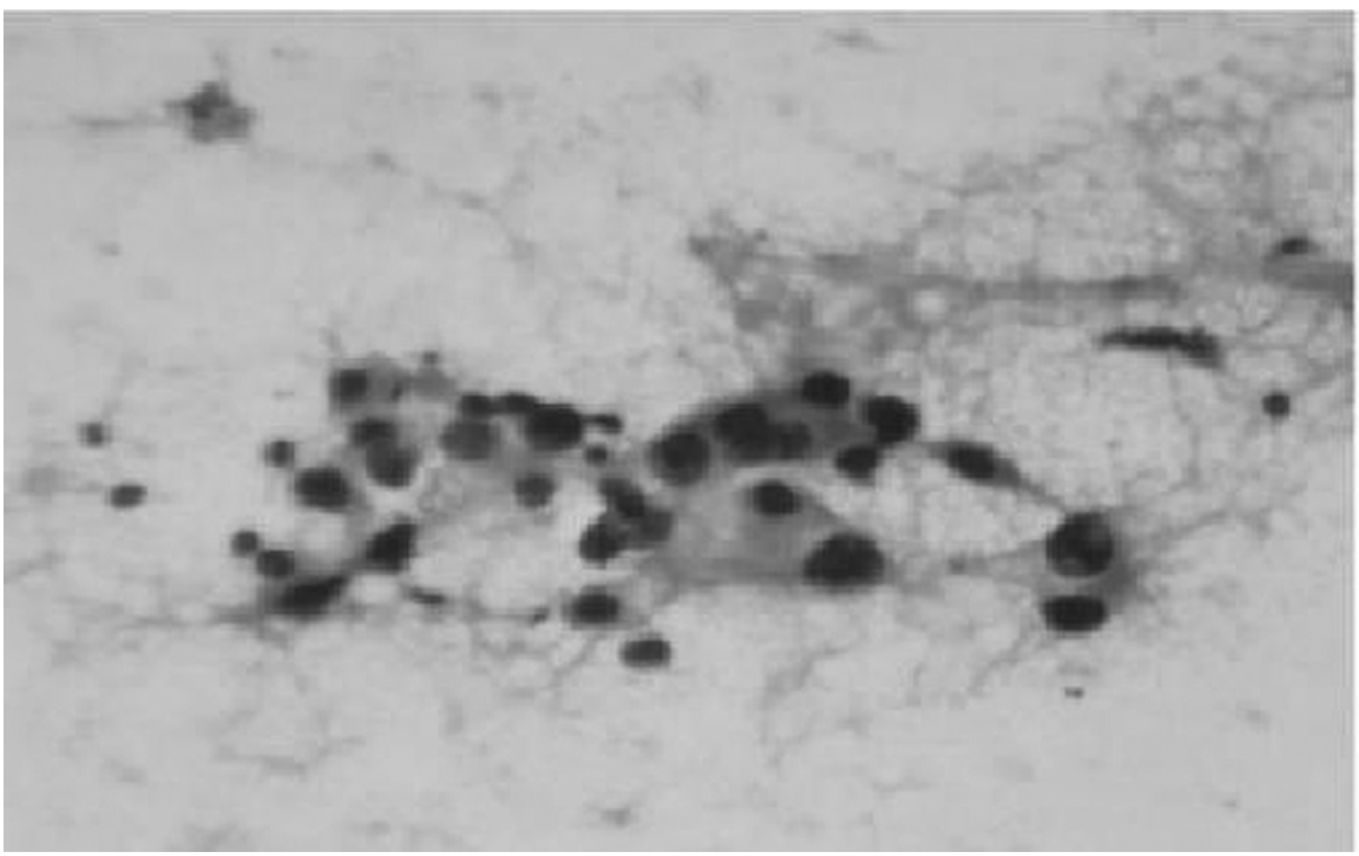

Figure 1: cell specimen noisy image

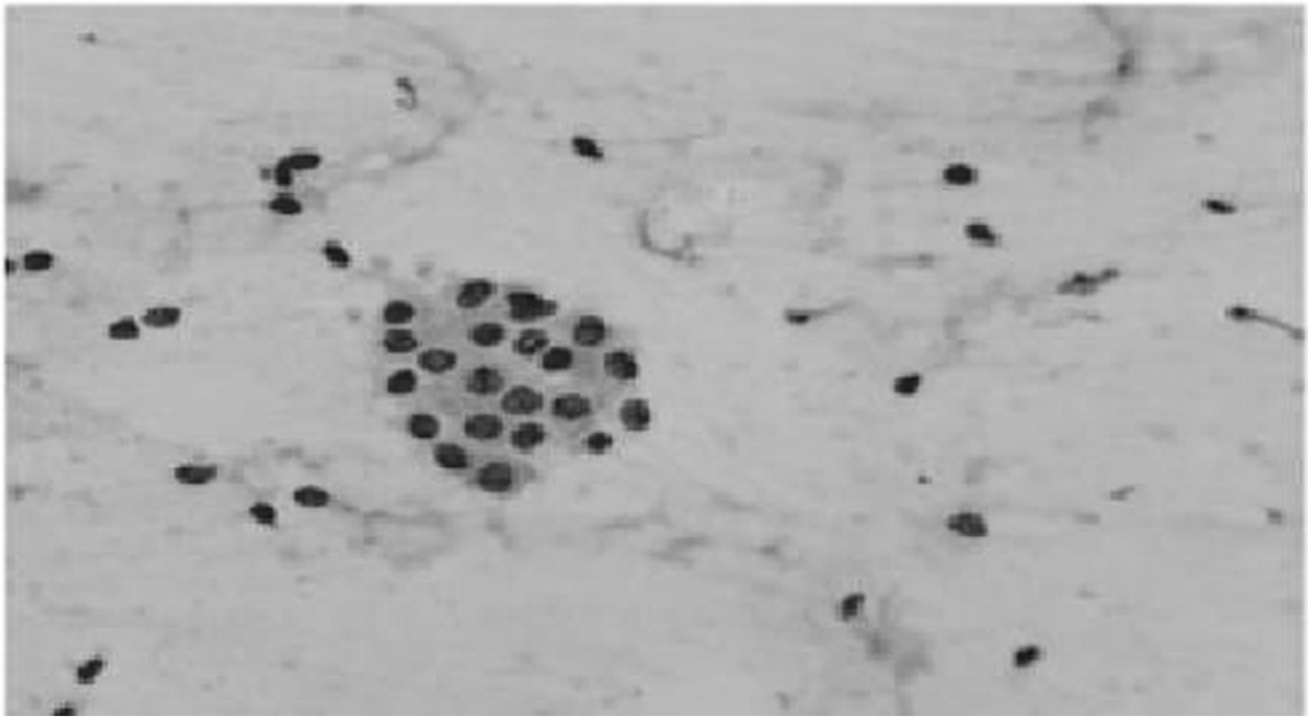

Figure 2: cell image after iteration

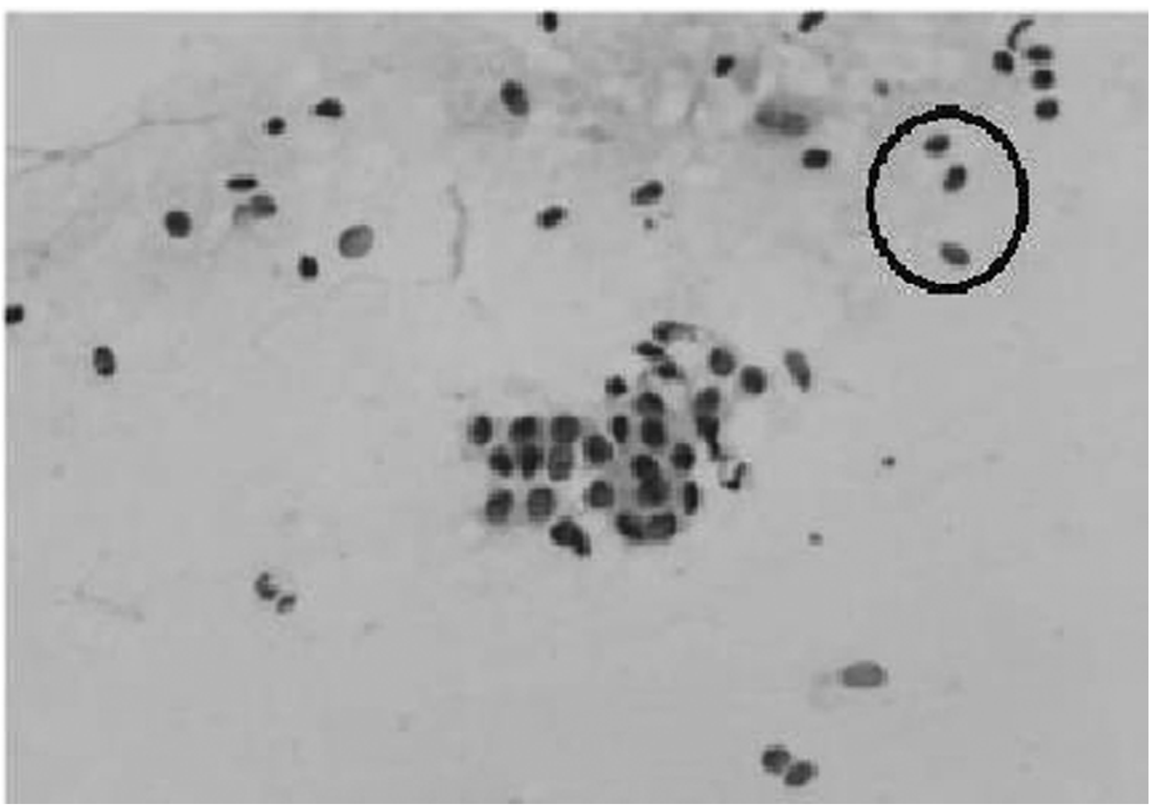

Figure 3: Image shows the exact content of RGB after filing

sharper and specific to the feature content of the image that was to be extracted. This filling is repeated. The filing is repeated extraction of more accurate feature which results in the gradual decrease if resolution, makes the image more sharper and accurate feature extracted. During the the experiment many inter-formatted image is resulted, But we have presented the final filed result.

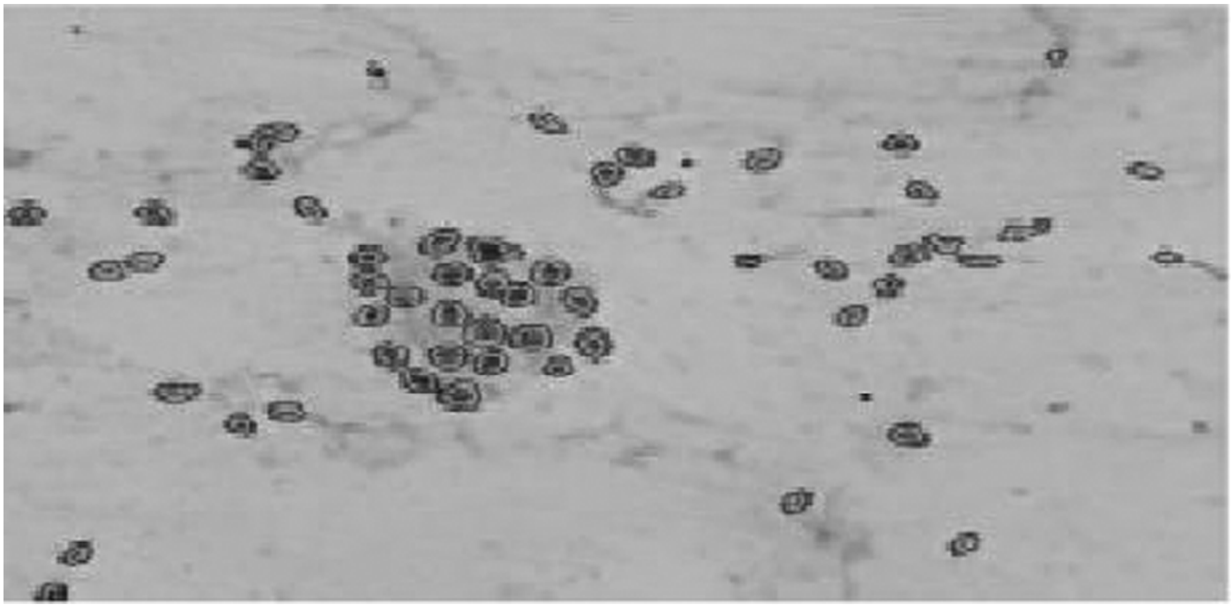

Figure 4: Resultant image

Here the resultant image is iterated and filed for concrete result where the required feature is being perfectly visible. The feature at in the resultant image will more accurate if we increase the gradient value. This increases in the resultant feature extraction which is the strength of proposed image iteration and filing mechanism. This image could be made company ready for analysis and diagnosis by the expert resulting in more accurate diagnosis.

3. RESULTS

When cells in a tissue begin to show abnormal growth, the cells deviates from the normal morphological characteristics one by one. Size of the cell nuclei and depth of staining are two important features that increase during abnormal or cancerous growth. After obtaining the segmented image in the grey scale format we can count the intensity values for each nucleus. After calculating the sum of grey scale intensity for each nucleus, it can be plotted in the forms of DNA ploidy histograms. Fig. shows the cell lll1age and corresponding histograms for benign cells.

3.1. Future work

Commercially available cameras are taking bigger and better pictures, but researchers are also working with images. Here are two approaches to very high resolution. First is to push the camera and lens to the limit. The Giga pixel project shows what can be done with a lens and camera. They are currently taking 4 Giga pixel pictures and pushing for more. To put that in perspective, in a photo of a football field, a blade of grass would be about 100 pixels wide. You can see examples of extreme zooming that makes possible here. The Photosynth project demonstrates the stitching together of many images into a very large composite image. This project is interesting in at least two ways. For one, it shows how we may one day navigate and zoom around on a large document like a newspaper page. This user interface may end up replacing scroll bars and zoom buttons. Second, it demonstrates collaborative image making, since the photos may be taken by many people.

REFERENCES

[1] ApacheHadoop,http://en.wikipedia.org/wiki/Apache/Hadoop.

[2] Cuda on hadoop, http://wiki.apache.org/hadoop/CUDAn//

[3] Nvidia fermi architecture for high performance computing, http://www.nvidia.com/object/fermiarchitecture.html.

[4] Nvidia cuda c programming guide, http://developer.download.nvidia.com/compute/cuda/32/toolkit/ docs/CUDA [5]A Cn Programming Guide.pdf.Nvidia cuda c programming guide, http://developer.download.nvidia.com/compute/cuda/32/t oolkit/ docs/CUDA Cn Programming Guide.pdf.

[6] Nvidia fermi architecture for high performance computing, http://www.nvidia.com/object/fermiarchitecture.html.

[7] Y. Lu et al., Study of content-based image retrieval using parallel computing technique, Proceedings of the ATIPs, pp. 186 191, 2007

[8] O. Kao et al., Scheduling aspects for image retrieval in cluster-based image databases, Proceedings of First IEEE/ACM Cluster Computing and the Grid, pp. 329 - 336, 2001

[9] Y. Ling, Y. Ouyang, "ImageParallelSemanticComputing,"Information-CCCM,Ret 259, 2008.

[10] G Yuli, F Jianping, S Shinichi. A novel Approach for Filtering Junk Images from Google Search Results. Springer LNCS 4903, pp. 1 12, 2008.

[11] T. White, Hadoop: de (1sted.). Sebastopol,the definitive CA: OReilly Media; 2009 gui.

[12] J. Venner, Pro Hadoop, Springer Berlin, 2009. pp.

[13] P. Wichian and R. Walisa, Optimizing and Tuning MapReduce Jobs to Improve the Large-Scale Data Analysis Process, International journal of Intelligent Systems, Vol. 00, 1-16, 2012