International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 5, Issue 3, Apr - May 2016; 37-44

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

PROTOTYPING ALGORITHM FOR PERFORMANCE ENHANCEMENT FOR CBIR USING NVIDIA CUDA 7.5

KASHINATH CHANDELKAR

Department of Computer Science & Engineering, Birla Institute of Technology, Mesra

BISR Campus, MIA Jaipur, India

ABSTRACT

The needs for CBIR (Content Based Information Retrieval) have increased exponentially due to rise in electronic gadgets. Text, Image, Audio and video are some of the forms of content across the network used for IR (information retrieval). The leading service provider faces everyday challenge to manage CMLC (Content Management Life Cycle). Extracted features from the data corpus are matched and indexed using TDI (Term Document Index) with the query futures for Information Retrieval. One of the neglected areas in the designed application for the specified cause is its adaptability to work in cloud and grid computing environment. The paper is an attempt that confirms improved efficiency for a given algorithm with efficient resource management using MATLAB R2015a and NVIDIA CUDA 7.5.

KEYWORDS: CBIR, CUDA, CMLC, IR, TDI

INTRODUCTION

The evolutions for performance, quality application and solutions have come a long way, keeping aside traditional information systems. These systems carry limited space, processing units, shared memory and not forgetting designed applications meant to suite existing architecture. To overcome these challenges cloud Computing has emerged in support of grid and parallel computing. As the number of internet users have increased across the globe need for information have also increased exponentially. As a result of which CBIR (Content Based Information Retrieval) system has evolved considering text, image, audio, Video and other supported file formats.

The Literature Survey conducted for CBIR has helped to explore different dimensions of how CMLC (Content Management Life Cycle) maintains diverse file formats. The data corpus is assumed to require content in a repository. The user sends queries to the data corpus through secured data stream, if the required content is well arranged as per the keywords, extracted features, color and shape, search engine have a better chance to extract content in minimum time and space. If the traditional system uses CPU (Central Processing Unit) in the serial it takes specified time. The performance increases if the data stream is processed parallel. The Algorithm designed for CBIR on a given platform has to run on CPU followed by GPU if available both series and parallel. The paper demonstrates how performance can be improved for a given algorithm by customizing the CUDA kernel.

TRADITIONAL CONTENT BASED INFORMATION RETRIEVAL (CBIR)

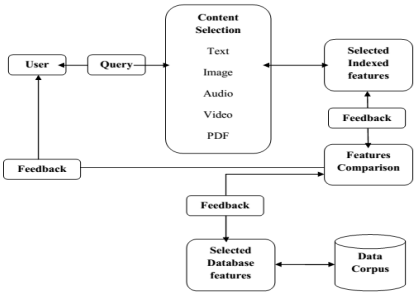


Figure 1: Traditional Content Based Information Retrieval System

The above system explains the process for content extraction from the existing data corpus assuming that the data is managed to its best state. The user sends a request for the content using search engine, in turn lists of options are given to the user for further selection. Required features are decided by the user giving feature list. These lists of features are compared with existing available data sources in the distributed system. If the features are matched with the query selection than feedback in the form of the results are sent to the user. The details are discussed in existing methodologies and Literature Survey section.

EXISTING METHODS AND LITERATURE SURVEY

Figure 2: Word Cloud Summarizing Methods for CBIR

As shown in the Word Cloud, methodologies involved in Information retrieval are categorized into following methods.

Cloud Based

Due to existing limitations in traditional Information Retrieval systems, the applications are migrated into the cloud environment. CBIR is made possible by matching content from the query image that with data corpus from the cloud [4] [5]. This reduces computation complexity for large databases. Image segmentation for feature extraction [11] using windows azure and other tools like rack space also yields better results. Partition Clustering [12] is considered under content pre-processing which is used by KNN (K nearest neighbor) algorithm for content retrieval. Map Reduce algorithm [16] when combined with Divide and Conquer increased performance in the cloud environment.

Texture Based

While extracting features in the traditional IR (Information Retrieval) system, wavelet and Gabor feature [2] [5] added with SVD (Single Volume Decomposition) technique [3] helps to design better index their by improving search. TDI (Term Document Index) [6] helps to fill semantic gap [8]. This improves content representation techniques like Histograms [9]. RGB combination has special features identified with system hardware which shall be enhanced using proper customisation.

Color Based

As shown in the CBIR Life Cycle, features are extracted from the data corpus [3] [5], indexed and represented in a specified manner like histogram and graph [7] which definitely improves performance and quality of the results. Using a hybrid approach like text and color has added advantage over single feature [9]. Color feature can also be extracted using an average mean technique [10] in addition with partition clustering [12] and Euclidean Distance method [15] which has an edge over single method.

Genetic Algorithm Based

Genes have a special role in artificial Intelligence and Semantic Data Mining. Interactive genetic algorithm [13] is used for feature extraction to cross over and Mutation. The resultant chromosome becomes more efficient and intelligent due to gain knowledge.

Histogram Based

In the Content Management Life Cycle (CMLC), information representation is more important than content extraction and other indexing. Euclidean Distance method [15] helps to represent content in the form of histogram with other available methods.

Support Vector Machine (SVM) Based

This method is used when a large amount of data needs to be extracted in a continues manner. MMI (Modified Moment Invariant) [14] is combined with ELM (Exact Legendry Moment) to give better results with more training data.

PROPOSED METHOD

The traditional CBIR system focuses on different dimensions. The proposed system focuses on Image Information Retrieval (IIR) considering all the traditional feature extraction methodologies. All the designed algorithms for the said cause show potential results. So, considering the fact that available algorithm is designed to solve the given problem, taking into account its design, security, resource management, adaptability factors for migration in a cloud based environment and tested for parallel execution to make best use of available CPU cycle.

A prototype algorithm for image whitening is taken as input. The algorithm was tested on MATLAB R2015a for execution and resource management with and without using the CUDA Kernel file. The same algorithm was executed on CUDA 7.5 in addition with kernel giving better results.

A Pseudo Code for an Algorithm is Given as Under:

Select data corpus

For each image

Run algorithm for said cause (image whitening in our case)

Customize CUDA kernel for CBIR

Employ CPU and GPU in parallel

Collect results from GPU and pass on to the CPU

Display results

Display resource utilization

End for

EXPERIMENTAL RESULTS

An algorithm for image whitening was run on a system having MATLAB R2015a having 4GB RAM windows 7 service pack 1 and windows server 2008 R2. 4 processors were utilized each having a clock cycle of 2195 MHz. A page size was 4096 bytes. An employed code has functions that does image whitening with CPU and using GPU. The collected results are as below:

Table 1: Time Taken for Function Execution
Time Taken for Each Function Execution Using the CPU (Image Whitening)

Function Name	Call	Total Time (Seconds)	Self Time (Seconds)
Run Script	1	0.930	0.334
White balance	1	0.453	0.284
Mean	2	0.169	0.155
Title	1	0.119	0.113
New plot	1	0.024	0.018
Intmean	1	0.014	0.014
New plot>observeaxisNextplot	1	0.006	0.001
Graph2d\private\lebelcheck	1	0.006	0.006
Cla	1	0.005	0.000
Graphics\private\clo	1	0.004	0.004
Grapjics\private\cloNotify	1	0.001	0.001
Usejava	1	0.001	0.001
Newsplot>ObserveFigureNextPlot	1	0.00	0.00

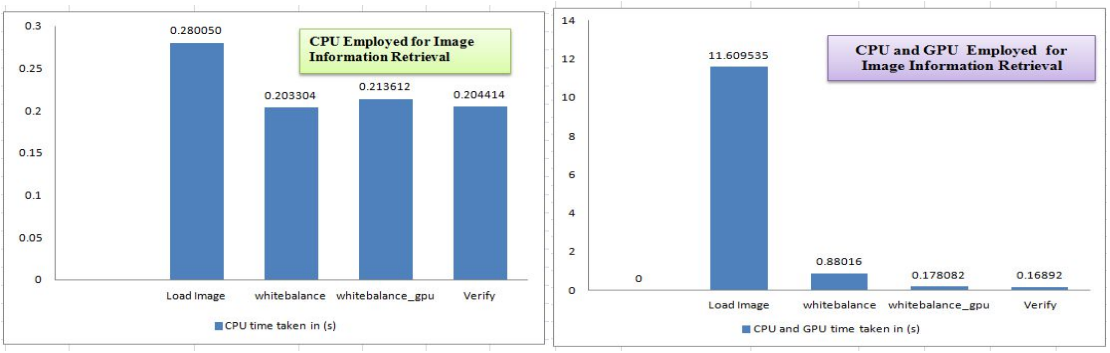
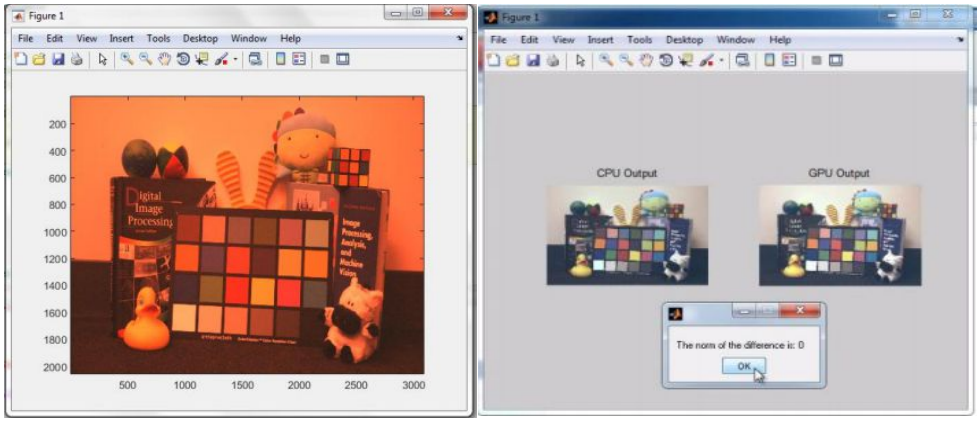


Figure 3: Comparative Analysis for IIR for Given Algorithm

Processed image using MATLAB, both showing the same output when executed on the CPU alone and with GPU with change in time spans as shown in figure 3. A processed image with original is shown below in figure 4.



Original Image **Processed image**

Figure 4: Actual Image with Processed Image Using Algorithm

As discussed in the proposed methodology and results collected in MATLAB confirms that GPU utilization enhances results for execution of an algorithm in parallel and the code shall be deployed in a cloud based environment. GPU processing on CUDA 7.5 helps to understand resource management in best possible ways.

Experimental Setup for CUDA 7.5

The above mentioned setup for Image Information retrieval having the same set of functions are taken as an input. Since MATLAB could not run kernel file, NVIDIA CUDA 7.5 having a size of 962.35 MB was customized and synched with Matlab R2015a and Visual Studio 2012 professional. A traditional whitebalance_gpu. m file was tested against modified whitebalance_gpu. Cu having CUDA 7.5 kernel that runs on Geforce 610M GPU from a system. NVIDIA Nsight was used to connect to local host with a visual profile to collect results.

One of the challenges was converting a file format from.M (Matlab script file) to.Cu (code source). The results were tested on CUDA 7.5 having NVIDIA Geforce 610M graphics card from the system used for experimental setup.

Results from CUDA 7.5

It is observed that whitebalence. m when executed o CUDA 7.5 takes 3.6 % of API time; on the other hand modified whitebalance_gpu. m takes 7.1 % of GPU time. The average API time and average GPU time remain constant for both 105 % and 1.3% respectively. Internal shared and registry memory remains unchanged as shown in the figure 4. Since algorithm designed to utilize CPU cycles only takes 34.307 seconds due to low resource management. Time consumption is more due to serial processing. Whitebalance_gpu. m used with CUDA kernel shows processing time of 2.662 seconds. The core utilization for both is compared as shown in Figure 5 (a, b).

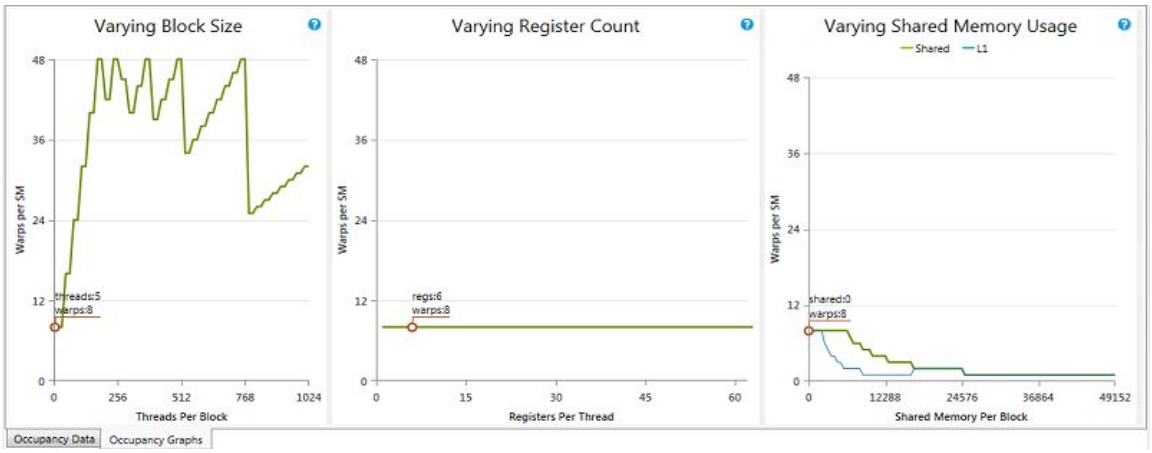


Figure 4: Internal Resource Utilization on CUDA at Runtime

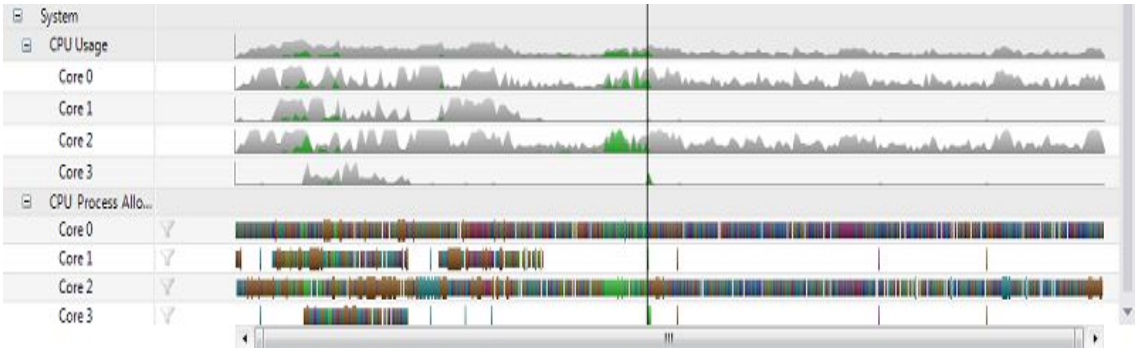


Figure 5(a): System Utilization on CPU

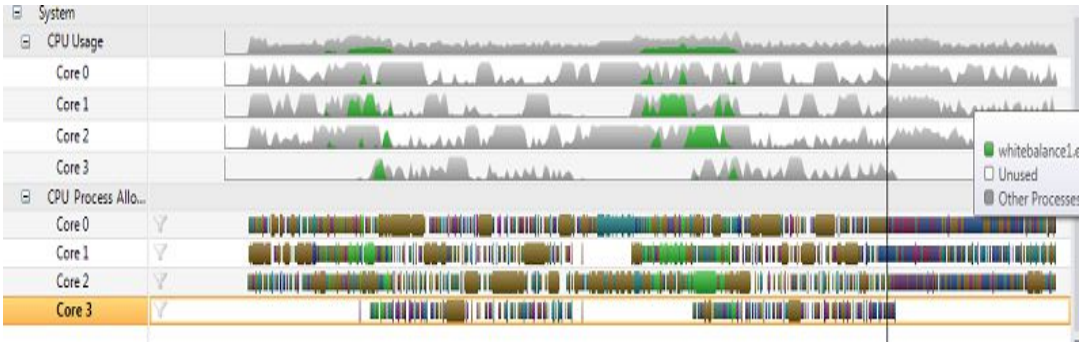


Figure 5(b): System Utilization on CPU & GPU

CONCLUSIONS

The process for image Information retrieval in the life cycle of CBIR has begun having a list of algorithms for feature extractions. Each algorithm has its own potential and yields satisfactory results for specified platforms both including traditional and cloud based. One of the basic is which was neglected that each algorithm needs CPU cycle and graphics processor to understand an image or any other feature extraction in its specified forms. The obtained results from MATLAB and CUDA 7.5 demonstrate that CBIR (Content Based Information Retrieval) becomes more efficient with better resource (CPU and GPU) management on a given platform. The experimental setup shall be extended in a cloud environment for future scope and Big Data management.

REFERENCES

1. K. Arthi and M. Vijayaraghavan, "Content based image retrieval algorithm using color models," *Adv. Res. Comput.* Vol. 2, No. 3, pp. 1343–1347, 2013.
2. P. P. Buch, "Comparative analysis of content based image retrieval using both color and texture," *Inst. Technol. Hyderabad*, pp. 8–10, 2011.
3. J. Chen and Y. Saad, "Divide and Conquer Strategies for Effective Information Retrieval," *NSF*, no. umsi-2008–18, pp. 1–9, 2007.
4. Conference, C. Science, and E. Engineering-, "a Novel Approach for Cbir Using cloud computing" *ISRASE*, pp. 101–106, 2015.
5. R. Datta, J. Li, and J. Z. Wang, "Content-Based Image Retrieval: Approaches and Trends," *Proc. ACM Int. Work. Multimedia. Inf. Retr.*, pp. 253–262, 2005.
6. J. Foley, M. Bendersky, and V. Josifovski, "Learning to Extract Local Events from the Web," *Proc. 38th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '15*, pp. 423–432, 2015.
7. N. Goel and P. Sehgal, "A Refined Hybrid Image Retrieval System using Text and Color," *Int. J. Comput. SCI.*, vol. 9, no. 4, pp. 48–56, 2012.
8. S. Kaur and R. Sharma, "Comparative Study on Content Based Image Retrieval With Various Approaches," *Int. J. Futur. Comput. Commune.*, vol. 5, no. 3, pp. 4514–4518, 2014.
9. S. Madhu, "Content based Image Retrieval : A Quantitative Comparison between Query by Color and Query by Texture," *J. Ind. Intell. Inf.*, vol. 2, no. 2, pp. 108–112, 2014.
10. R. Malini and C. Vasanthanayaki, "An Enhanced Content Based Image Retrieval System using Color, Features," *IJECSE*, vol. 2, no. 12, pp. 3465–3471, 2013.
11. R. M. Mohana and A. R. M. Reddy, "CCBIR: A Cloud Based Implementation of Content Based Image Retrieval," *WSES Trans. Comput.*, vol. 14, pp. 333–346, 2015.
12. H. Narasimhan and P. Ramraj, "Contribution-based clustering algorithm for content-based image retrieval," *2010 5th Int. Conf. Ind. Inf. Syst.*, pp. 442–447, 2010.
13. G. Raghuwanshi, N. Mishra, and S. Sharma, "Content based Image Retrieval using Implicit and Explicit Feedback with Interactive Genetic Algorithm," *Int. J. Comput. Appl.*, vol. 43, no. 16, pp. 8–14, 2012.
14. C. S. Rao, S. S. Kumar, and B. C. Mohan, "M Achine," *Int. J. Multimedia its Appl.*, vol. 2, no. 2, pp. 69–79, 2010.
15. M. Singha, "Content Based Image Retrieval using Color and Texture," *Signal Image Process. An Int. J.*, vol. 3, no. 1, pp. 39–57, 2012.
16. C. Sureshkumar, "Content Based Image Retrieval Methods Using Random Segmentation Algorithm," *Available ONLINE www.ijart.org IJART*, vol. 2, no. 4, pp. 78–84, 2012.