GPU-Parallel Implementation of Color based Medical Image Retrieval in Compressed Domain

Kuldeep Yadav  
Department of CSE,  
College of Engineering Roorkee,  
Roorkee-247667, Uttarakhand, India

Avi Srivastava  
Department of CSE,  
College of Engineering Roorkee,  
Roorkee-247667, Uttarakhand, India

M.A Ansari  
Department of Electrical,  
GBU, Greater Noida,  
Uttar Pradesh, India

ABSTRACT

In huge databases Image processing takes more time for execution on a single core processor because of slow single thread algorithms. Graphics Processing Unit (GPU) is more popular now-a-days due to their speed, programmability, low cost and more inbuilt execution cores in it. Most of the researchers started work to use GPUs as a processing unit with a single core computer system to speedup execution of algorithms. The main goal of this research work is to parallelize the process of content based image retrieval through color in compressed domain making whole process much faster than normal. In this paper, parallel implementation is focused on the well known Quadratic Distance metric approach for Color based image retrieval systems, since it is one of the most fundamental and important problems in the field of computer vision, medical image processing and content based image retrieval (CBIR). For compressed images we have taken standard JPEG format. Our work employs extensive usage of highly multithreaded architecture of multi-cored GPU. An efficient use of shared memory is required to optimize parallel reduction in Compute Unified Device Architecture (CUDA). Experimental results show that parallel implementation achieved an average speed up of 25 x over the serial implementation when running on a GPU named GeForce 9500 GT having 32 cores. Color based retrieval method of CBIR is also evaluated using Recall, Precision, F-measure, True Negative rate, and Accuracy evaluation measures.

General Terms
Content Based Image retrieval, GPU, Parallel Computing.

Keywords
Color Based Image Retrieval; CUDA; GPU; Parallelization.

1. INTRODUCTION

Graphical Processing Units (GPUs) have been proved its importance in terms of performance as hardware for computer graphics [1]. Many researchers have already been applying GPUs to implement many algorithms in various areas such as image processing, computational geometry, and scientific computation, as well as computer graphics [2-7]. GPUs play important role to speedup processing of database images matching algorithms because it has more inbuilt execution cores. The parallel implementation of image analysis algorithms using GPU encounters two problems. First, the programmer should master the fundamentals of GPU and CUDA [8]. CUDA platform is used to implement the parallel implementation of algorithms. Second, in a job it needs much process cooperation between CPU and GPU.

Presented approach of parallelization is based on the first most important phase of Image retrieval process named color based image retrieval. Color based image retrieval is used to distinguish a specific image or similar image, from a database of hundreds of images. Image database cannot be stored in uncompressed format because of limitation of space so generally these images of database are in compressed format. There are two types of retrieval process of images from the database: uncompressed domain and compressed domain. Uncompressed domain [9-14] retrieval process first decompresses the image in database for matching while compressed domain methods [15-21] match them in compressed format. The overhead cost of decompression cost too much of clock cycle so compressed format retrieval process has gained considerable amount of attention for research. Compressed domain retrieval process has only shortcoming of increasing data acquisition time and that too is negligible.

Parallel implementations on GPUs have been applied to various numerical problems [22-25] to reduce the computation time without sacrificing the degree of accuracy. Fast CBIR is one of the important problems in the field of computer vision. The decompression of images and their high computation cost are the main drawbacks of slow implementations of uncompressed CBIR systems. Computational cost reduction approaches of CBIR were proposed in [26] by Emmanuel at al. recently.

In the following sections, we present a detailed description of the proposed methodology as well as experimental results that demonstrate the efficiency of the proposed methodology.

2. INTRODUCTION TO NVIDIA CUDA ARCHITECTURE

CUDA™ is a general purpose parallel computing architecture introduced by NVIDIA. It contains the CUDA Instruction Set Architecture (ISA) and parallel compute engine in the GPU. The
CUDA architecture is programmed using C language, which can then be run with great performance on a CUDA enabled processor [27]. CUDA-enabled GPUs have hundreds of cores that can collectively run thousands of computing threads. Each core has shared resources, including registers and memory. The on-chip shared memory allows parallel tasks running on these cores to share data without sending it over the system memory bus [28]. Thread hierarchy, shared memories and barrier synchronization are the three key abstractions of CUDA. A kernel can be executed by a one dimensional or two dimensional grids of multiple equally-shaped thread blocks. A thread block is a 3, 2 or 1-dimensional group of threads. Threads within a block can cooperate among themselves by sharing data through some shared memory and synchronizing their execution to coordinate memory accesses. Threads in different blocks cannot cooperate and each block can execute in any order relative to other blocks. The number of threads per block is therefore restricted by the limited memory resources of a processor core.

CUDA kernel function is a fundamental building block of CUDA programs. When launching a CUDA kernel function, a developer specifies how many copies of it to run. We call each of these copies a task. Because of the hardware support of the GPU, each of these tasks can be small, and the developer can queue hundreds of thousands of them for execution at once. These tasks are organized in a two-level hierarchy, block and grid. Small sets of tightly coupled tasks are grouped into blocks. In a given execution of a CUDA kernel function, all blocks contain the same number of tasks. The tasks in a block run concurrently and can easily communicate with each other, which enables useful optimizations such as those of the section “Shared Memory”. GPU’s hardware keeps multiple blocks in flight at once, with no guarantees about their relative execution order. As a result, synchronization between blocks is difficult. The set of all blocks run during the execution of a CUDA kernel function is called a grid.

3. QUADRATIC DISTANCE METRIC APPROACH IN JPEG

Quadratic distance metric algorithm is a histogram matching algorithm [11]. The difference in histogram plot of the two images is called as quadratic distance metric which is calculated by formula:

$$d^2(Q,I) = (H_Q - H_I)^T A (H_Q - H_I)$$  \hspace{1cm} (1)$$

Equation contains three terms; first term gives the difference in the number of pixel in each bin. Last term is the transpose of the first, and middle gives the similarity matrix. Final answer d gives the difference between two images.

In JPEG compressed domain whole image is divided into 8x8 matrices. The histogram plot of DC coefficient [16] and pixel domain histogram are almost equal so DC coefficient that is the first term of the Fourier transform of intensity matrix of image is plotted with pixel domain histogram of query image on the histogram for distance metric calculation.

4. EVALUATION MEASURES

The method of color based image retrieval is evaluated using the six evaluation measures: Precision, Recall, F-measure, True negative rate, (Negative Rate Metric) NRM and accuracy.

Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$  \hspace{1cm} (2)$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$  \hspace{1cm} (3)$$

F-Measure:

$$F - \text{Measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$  \hspace{1cm} (4)$$

True Negative Rate:

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$  \hspace{1cm} (5)$$

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$  \hspace{1cm} (6)$$

NRM:

$$\text{NRM} = \frac{NR_{NN} + NR_{FP}}{2}$$  \hspace{1cm} (7)$$

Where

$$NR_{NN} = \frac{FN}{FN + TP}$$ and $$NR_{FP} = \frac{FP}{FP + TN}$$

5. IMPLEMENTATION

In this work, the implementation of proposed approach is based on the two set of experiments. In the first set of experiment, proposed algorithm is implemented in C language and in second set, parallel implementation is done using CUDA. The following section dictates the detailed description of the parallel implementation of the algorithm.

5.1 Parallel Implementation

In CUDA, it is assumed that both host and device maintain their own DRAM. Host memory is allocated using malloc and device memory is allocated using cudaMalloc. CUDA threads are assigned a unique thread ID that identifies its location within the thread, block and grid. This provides a natural way to invoke computation across the image, by using the thread IDs for addressing. The parallel implementation of algorithm of CBIR is shown in the form of pseudo code shown in algorithm 1.
Algorithm 1: Parallel Implementation of Color based image retrieval algorithm

Step1. Plot histogram of query image using pixel value.
Step2. Get image from repository and find DC coefficients.
Step3. Plot DC coefficients in histogram.
Step4. Parallely compute the difference in histogram pattern.
Step5. Save distance value.
Step6. If image in repository
   Goto step 2
   Else
   Goto step 7
Step7. Display in decreasing order of distance.

6. HARDWARE SPECIFICATIONS

All the experiments are carried out using the hardware specifications of GPU: GeForce 9500 GT, 1 MB DDR2, No of Processors = 4, No of core =32, RAM 1 GB, Frequency 1.35 GHz, DDR2 and CPU: Intel Core 2 Duo, 2.66 GHZ, No of cores available =2, No of thread=1, No of thread/core=1, Physical Memory =2 GB, DDR2

7. RESULTS AND DISCUSSIONS

For the testing of color based retrieval approach of CBIR, we collected a data set of MRI, CT-scan and X-ray to form database of images in compressed format of JPEG. The results of color based retrieval approach are shown in fig. 4 that demonstrates the efficiency of this approach. On the basis of visual observation, color based retrieval method of CBIR manages to find images similar to query image in database but with a drawback of a lot of time consumption. To make faster the method, we parallelized it on CUDA and achieved an average speed up of 25 x(approx) over the serial implementation when running on a GPU. The comparison of serial implementation over parallel is shown in table 1. Table 1 also shows that execution time depends on the image resolution.

Further, the performance of method is evaluated using Precision, Recall, F-measure, True Negative Rate, NRM and Accuracy measures, which show the effectiveness of method shown in table 2. Fig.2 shows the graph of execution time of GPU in seconds. Fig.3 shows the graph of speedup. Fig. 1 shows the graph of execution time of CPU in seconds. Output images of Color based retrieval approach is shown in fig. 4.

<table>
<thead>
<tr>
<th>Resolution (a X a)</th>
<th>Serial</th>
<th>Parallel</th>
<th>Speed-Up</th>
<th>Speed-Up Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>2.3</td>
<td>0.094715</td>
<td>24.283282</td>
<td>23.77983</td>
</tr>
<tr>
<td>256</td>
<td>1.83</td>
<td>0.078620</td>
<td>23.276377</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>2.217</td>
<td>0.092424</td>
<td>23.987283</td>
<td>23.114853</td>
</tr>
<tr>
<td>256</td>
<td>1.76</td>
<td>0.079128</td>
<td>22.242423</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>2.811</td>
<td>0.115262</td>
<td>24.387827</td>
<td>24.630405</td>
</tr>
<tr>
<td>256</td>
<td>2.146</td>
<td>0.086278</td>
<td>24.872983</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>3.186</td>
<td>0.126236</td>
<td>25.238382</td>
<td>25.055561</td>
</tr>
<tr>
<td>256</td>
<td>2.293</td>
<td>0.092188</td>
<td>24.872834</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>2.296</td>
<td>0.096359</td>
<td>23.827398</td>
<td>24.555416</td>
</tr>
<tr>
<td>256</td>
<td>1.912</td>
<td>0.075622</td>
<td>25.283433</td>
<td></td>
</tr>
</tbody>
</table>

Average Speed-Up 24.227212

<table>
<thead>
<tr>
<th>Image</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>TNR</th>
<th>NMR</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>33.33</td>
<td>39.9976</td>
<td>50</td>
<td>58.33</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>66.66</td>
<td>33.33</td>
<td>44.44</td>
<td>66.66</td>
<td>50</td>
<td>44.44</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>50</td>
<td>66.67</td>
<td>100</td>
<td>25</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>50</td>
<td>66.67</td>
<td>100</td>
<td>25</td>
<td>75</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>50</td>
<td>66.67</td>
<td>100</td>
<td>25</td>
<td>75</td>
</tr>
</tbody>
</table>
**Fig 1. Execution time in CPU vs. Resolution**

**Fig 2. Execution time in GPU vs. Resolution**

**Fig 3. Speed up vs. Resolution**

<table>
<thead>
<tr>
<th>Image Number</th>
<th>Query Image</th>
<th>CBIR results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="cbir_results.png" alt="CBIR Results" /></td>
</tr>
</tbody>
</table>
Fig. 4: Output images of CBIR
8. CONCLUSION

In this research work, a well known color based image retrieval algorithm of CBIR has been parallelized and analyzed with evaluation measures. The method is evaluated using Precision, Recall, F-measure, True Negative Rate, NRM and Accuracy measures. The implementation of CBIR algorithm on the graphics device is promising with large image database. However, Color based retrieval method produces images which are not similar in vision but they are not in top three results which is considerable when compared with speed up of approx 25x.

CUDA itself has been shown to be an excellent framework to accelerate computational problems in image processing, numerical solving techniques and Image Processing areas.

9. REFERENCES


