Image segmentation using CUDA K Means Algorithm

Gautham M
Department of Computer Engineering
National Institute of Technology Karnataka Surathkal
Mangalore, DK Karanataka India 575 021
Email: gauthamm1997@gmail.com

Yeshwanth R
Department of Computer Engineering
National Institute of Technology Karnataka Surathkal
Mangalore, DK Karanataka India 575 021
Email: yeshwanthr98@gmail.com

***Abstract*—Image segmentation is the classification of an image into different groups. Many researches have been done in the area of image segmentation using clustering. There are different methods and one of the most popular methods is k-means clustering algorithm. K -means clustering algorithm is an unsupervised algorithm and it is used to segment the interest area from the background. Recent rapid increase in the performance of graphic processing unit (GPU) hardware, coupled with simplified programming methods, have made GPU an efficient coprocessor for executing variety of highly parallel applications.. Parallel k-means segmentation is realized in hybrid manner i.e. proposed approach distributes computation load between Central Processing Unit (CPU) and GPU. The emphasis is placed on adaptation of the core algorithm to efficiently process datasets characteristic for image segmentation while exploiting benefits of underlying GPU hardware architecture**

***Keywords*—K-means, image segmentation,cuda.**

I. Problem Statement

Clustering is a search method for hidden patterns that may exist in datasets. It is a process of grouping data objects into disjointed clusters so that the data in each cluster are similar, yet different to the other clusters. K-rneans is one of the most famous and typical clustering algorithms and applied in many application areas such as data analyses, pattern recognition, image processing, and information retrieval . In k-means, a data point is comprised of several values, called features. By dividing a cluster of data objects into k sub-clusters, k -means represents all the data objects by the mean values or centroids of their respective sub-clusters.Since the complexity and time taken by the sequential compiler is high it is proposed to use the GPU for parallelization to the maximum extent possible.

II. Objective

The traditional sequential KMeans Algorithm is of complexity O(nk) where n is the number of points and k is the number of clusters. In the new algorithm proposed by [1] the complexity will be O(nk/p) where p is the degree of parallelism. The bjective of the project is to implement parallel version of k means image segmentation algo- rithm using CUDA parallel programming language. The project also aims to compare the efficiency between serial and parallel programs of K means image segmentation. In this case, input size refers to the number of dark points in the image .

III. Significance

Clustering can be described as grouping of data points based on similarity in the feature space . A set X of n data points is given in d dimensional feature space xjRd,i=1n, as well as number of clusters kN¡n. A cluster $CjX, j = 1, k$ with centroid $cjRd$ is consisted from all points xiX assigned to a cluster based on similarity criterion. In the k-means algorithm each cluster is represented by its centroid (mean). Minimum distance between an evaluated point and the centroids is used as a criterion for assignment of point to the related cluster

where D is defined as a distance between data point and cluster centroid, in practice this is usually Euclidean distance. Solving (1) is NP-hard problem even for two clusters. However, k-means algorithm provides non-optimal but feasible solution of the problem.

IV. Plan of action

A. Phase 1

To study and understand the kmeans image segmentation algorithm .To Implement serial code of this kmeans image segmentation algorithm in c and test with different data sets.

B. Phase 2

Implementation of parallel k-means image segmentation algorithm . using global memory and shared memory to optimize the speed of algorithm.Comparing the results timing with serial and parallel algorithms.

V. Algorithm and complexity

A. Principle

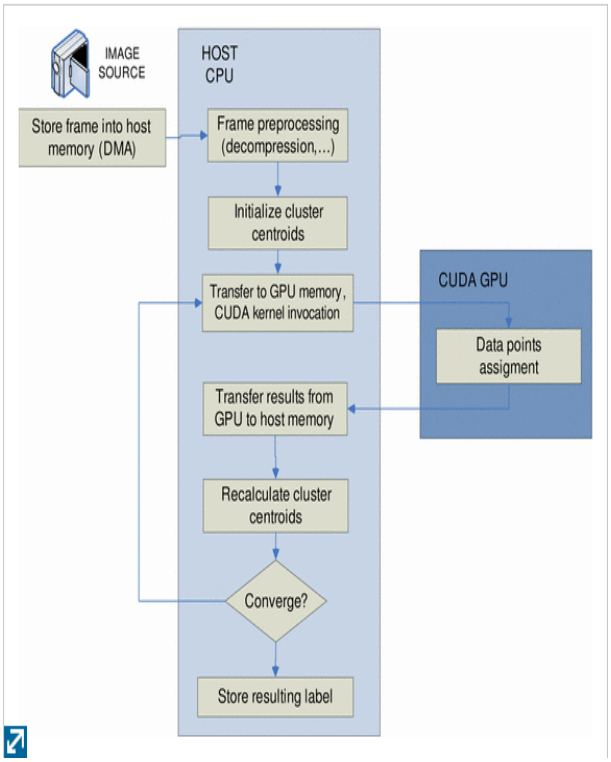
1) Initial clusters are created by choosing k random values from set X of data points. Values represent initial centroids of clusters, such initialization of cluster is known as seeding stage.
2) Next stage is known as labeling where each point xi is assigned to cluster Cj with criterion of minimal distance D satisfied. Euclidian distance is commonly used as metric for D.
3) Following the assignment of every point to the nearest cluster, centroids are recalculated in update stage. Each centroid cj is recalculated as a mean of all assigned points $xiCj$.

to calculate Euclidean distance between the pixel and the centroid, one subtraction, one multiplication and one addition per each dimension (color component) are required, consisting altogether from nine operations. Additionally, calculated distances between the pixel and the centroids have to be compared before assigning the pixel to the closest centroid.

Labeling execution time is bounded by $O(nk/p)$. Label for each point in the dataset can be calculated independent from other points, which allows simple per point workload distribution among high number of CUDA threads. This means that depending on image resolution, one or more points for labeling can be assigned to each CUDA thread. From analysis it is clear that labeling stage of k - means algorithm is suitable for parallel GPU based execution and such approach is expected to result with performance gain.

Computational bound of sequentially executed update phase is O(n+k) and this is significantly less complex compared to sequential labeling bound O(nk). But if we compare the execution cost of sequential update phase with the cost of GPU executed labeling phase, we will get the ration between the two

$$O\left(\frac{nk/p}{n+k}\right) \cong O(k/p) \ (for \ n >> k). \quad (6)$$



B. Implementation

The program takes input from the files which has x and y co-ordinates of the black pixels which is stored in file as image data. The program copies this data into two arrays. The initial clusters centroids are also stored in the file is copied to the arrays. The variable is copied to the device vectors and then kernel is launched to find distance of each point from the cluster centers. Depending on the closest distance the points are marked to the cluster groups. In each group mean is calculated to find the new cluster centroids. The process is repeated until the cluster centers doesn't change. The output of the final cluster centroids is saved into the file.The cluster centroids and all the points in the cluster are also stored in the file. The time of execution is calculated and printed in the terminal. In this implementation shared memory is used for finding the distance and grouping point to clusters.

In the kernel to find distance of each point from the cluster ,every point has a block and every block has a thread for every cluster In the kernel to group the points to cluster by marking and finding the least distance to the cluster centers, since every point is a cluster we have one thread per block to find min distance and mark the point to the group To recenter the cluster is divided into three different function kernels. First one is clear which has one block with each thread as a kernel it clears the current value of clusters. Second one is one block with each thread per cluster where each thread calculates the sum of x and y coordinates of points in the cluster. Third one has one block with each thread per cluster where each thread finds the average to assign new cluster centroids for the next iteration.

VI. Project Time Line

Following is the schedule of the project

- Oct 3 2017- Discussion of project proposal.
- Oct 4 2017- Submission of proposal report
- Oct 5-20 2017- Study of algorithm and implimentation of sequential algorithm
- Oct 23-27 2017- Mid progress discussion.
- Nov 1-10 2017- Implementation of CUDA k Means Image Segmentation Algorithm
- Nov 13-17 2017- Final demo
- Nov 24 2017- Report Submission

VII. Objectives Achieved

Following is the list of objectives achieved during the course of the project:

- Implementation of sequential k-means image segmentation algorithm.
- Implementation of parallel k-means image segmentation algorithm.
- comparing the performance of sequential and parallel algorithm.

VIII. Comparison Results

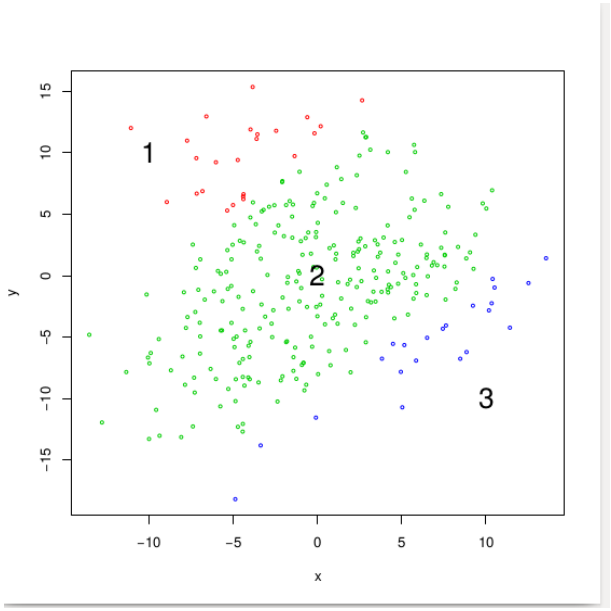
Country List			
Size of data	No of clusters	Serial time	Parallel time
100	4	105	51
100	8	113	45
500	4	154	62
500	8	172	72

IX. Conclusion

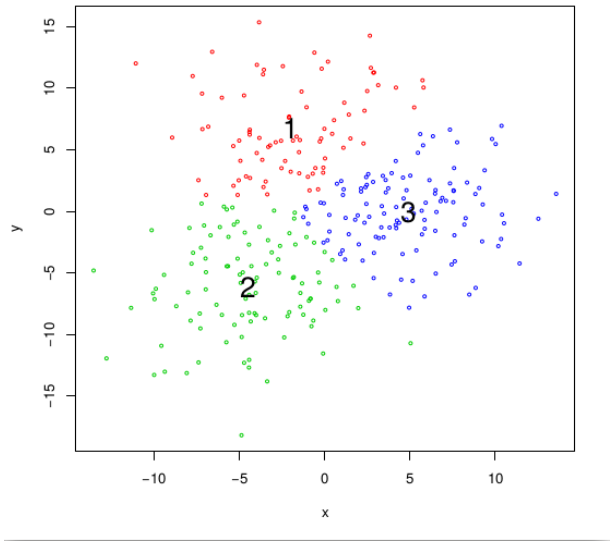
This algorithm can be used in the variety of application like grouping of same colors based on the RGB value in the image, this shows the different densities of that color used in the image Parallelism used reduces the execution time, other application include it's usage in wireless network sensor based application and used in search engine which is used to cluster the similar data and dissimilar object far from each other like better the clustering then results obtained in the front page is more accurate and popular.

X. Visual Representation

A. Points before algorithm



B. Points after algorithm



References

[1] "K-Means Image Segmentation on Massively Parallel GPU Architecture" by J. Sirotkovi, H.Dujmi and V. Papi .link:http://ieeexplore.ieee.org/document/6240695/