ADVANCED COMPUTER ARCHITECTURE

PROJECT REPORT

Color-based Image Segmentation

using Parallel K-means Clustering

*Author:*

Gabriele MIRANDO

A.Y. 2017/2018

**Abstract**

The objective of the project was to develop a C language program for color-based image segmentation using the standard k-means clustering algorithm and to speed up the execution by implementing a parallel version of the application. This report contains a complete account of how the work has been organized and what results have been obtained. It starts with a detailed analysis of the standard k-means algorithm, focusing in particular on how it can be exploited to achieve color-based segmentation through the clustering of the pixels of the image. An a-priori study of the available parallelism is then conducted, on the basis of the results obtained by profiling the serial code. The algorithm has been parallelized using OpenMP, to run on multi-core CPUs. A complete description of the parallel implementation is provided. Finally, multiple test cases for the program are taken into consideration and some observations about performance are deduced from the comparison of the execution times obtained on machines with an increasing number of cores.

Table of Contents

The Standard K-means Clustering Algorithm 4

Basic Functioning of K-means Clustering 4

Color-based Segmentation using K-means Clustering 5

The Parallelism Available in K-means 9

# The Standard K-means Clustering Algorithm

Clustering is the task that consists in organizing a given set of objects into groups, or more precisely into clusters, in such a way that objects in the same cluster are more similar to each other than to those in other clusters. The fact that is extremely easy to implement, but is also computationally very efficient, made the algorithm proposed by Stuart P. Lloyd in 1957 the most popular clustering technique used in scientific and industrial applications. Actually Lloyd’s algorithm was initially developed for partitioning an Euclidean space into a given number of well-shaped and uniformly sized convex cells. This technique is known with the name of Lloyd’s relaxation algorithm. It was then extended to address the clustering problem, giving birth to what is now referred as the k-means clustering algorithm. Although multiple variants of k-means have been later developed (Forgy, MacQueen, Hartigan-Wong), Lloyd’s version is considered the standard version of the algorithm.

## Basic Functioning of K-means Clustering

In its most general definition, the standard k-means algorithm can be used to partition a set of objects, described by a series of measurable features, into a predefined number of clusters. Let’s consider a dataset made up by N vector objects {}, each one described by M numerical attributes. Each vector object has the following form:

Using the k-means algorithm is possible to subdivide the initial set of objects into K disjoint clusters in just four steps:

1. **Initialize the algorithm**. Randomly pick K objects from the dataset and set them as initial cluster centers , with i ∈ {1, 2, …, K}.
2. **Assign each object to the nearest cluster**. For each of the object of the dataset, compute the squared Euclidean distance from all the cluster centers. Find the cluster center for which the value of the squared Euclidean distance is the lowest and assign the object to that cluster.
3. **Recompute the cluster centers**.For each cluster, compute the value of the center by calculating the mean of all the objects of the cluster. Indicating with the number of samples in cluster :

In the case of a cluster being empty, a good practice is to set its new center to the object of the dataset with the maximum distance from the center of the cluster to which the object belongs.

1. **Test for completion**. Repeat from step 2 of the algorithm until the cluster assignments do not change (this is when the algorithm converges to a solution) or until a predefined maximum number of iterations is reached.

The objective of the k-means algorithm is to find the clusters composition that minimize the Sum of Squared Errors (SSE), i.e. the sum of the squared Euclidean distances from each object in the dataset to the center of the cluster to which the object belongs. As a matter of fact, k-means aims to satisfy the following criterion of optimality:

Computing this minimum is an NP-hard problem for which no practical solution is known. K-means is an heuristic method that attempts to find iteratively and in the fastest way possible an approximation of the optimal solution for problem of the minimization of the SSE. Therefore, the algorithm allows to obtain just a local minimum, not the global minimum, and the cluster composition obtained at convergence depends on the initialization of the centers at the first step. Selecting the initial centers at random among the objects of the dataset allows to achieve acceptable results in the case of the application considered for the project, that is color-based image segmentation, but a more sophisticated initialization technique or multiple repetitions of the algorithm may be required for other applications, especially in the field of data analysis and machine learning.

## Color-based Segmentation using K-means Clustering

Image segmentation is the process of partitioning a digital image into its constituent objects. Given a digital image, performing a segmentation means to identify the main elements that make up the scene. Achieving a good quality segmentation is one of the most difficult and challenging tasks in digital image analysis, but it also represents the first step towards an effective detection of the objects present in the image. The more accurate is the segmentation phase, the more likely is the image recognition process is to succeed. For the purpose of the project, a simple color-based segmentation technique has been considered. The color-based approach relies on the fact that, in a digital image, sub-regions of pixels can be identified by looking at the pixels color. Given an initial image and the integer number K, using k-means algorithm it is possible to cluster the pixels into K groups, on the basis of the pixels color values. Pixels belonging to the same cluster are similar in terms of color. Once it is clear how the k-means algorithm works, addressing it to solve the color-based segmentation problem is a trivial task. Given a digital image of width W and height H as input, the initial dataset is then constituted by the N = W x H pixels of the image. The bigger is the size of the image considered, the higher is the number of pixels objects in the initial dataset. Since the goal is to obtain a color-based segmentation, the pixels need to be clustered taking into consideration as numerical attributes the values of their color components, i.e. the 0 to 255 intensities of their red, green and blue channels. So, each pixel object of the dataset, but also each cluster center, has the form of a 3D vector of the RGB values:

In this context, the squared Euclidean distance between one pixel and a cluster center, that is the sum of the squared differences of the values of the color components, is a measure of how much a given pixel and the pixels belonging to the cluster are similar in terms of color. At every iteration of the algorithm pixels are assigned to cluster for which the squared Euclidean distance is the lowest. The cluster center are then updated by computing the mean of all the pixel objects belonging to each cluster. A visual representation of the algorithm for color-based segmentation is shown by the flowchart of Figure 1. An additional step is required in the end after the clustering of the pixels using the standard k-means algorithm. The RGB values of each pixel of the image are replaced with those of the cluster center to which the pixel belongs. Basically, the initial digital RGB image, with millions of possible color combinations per pixel, is reduced to a K colors image.

no

Select an image

Select the value of K

**Initialize clusters centers**, randomly picking K pixels from the initial image

**Assign each pixel to the closest cluster**, by choosing the cluster for which the squared Euclidean distance is the lowest

**Update clusters centers**, computing the mean of all the pixels belonging to the cluster

**Update image**, replacing each pixel RGB values with the ones of center of the cluster to which the pixel belongs

Any pixel changed cluster?

yes

**Figure 1.** The flowchart for color-based segmentation using k-means clustering algorithm

To visually assess the results of color-based segmentation using k-means clustering, it’s useful to examine how a 720 x 480 JPEG image of a horse, which has been used as the main test image for the program, has been segmented with increasing values of K. Figure 2 shows that images with different levels of accuracy were obtained and that the choice of K was crucial for a good quality segmentation. In particular, four colors were immediately discernible in the original image. Therefore, after performing the segmentation with K = 4, the shape of the horse was clearly identifiable.

|  |  |
| --- | --- |
| **(a)** The original JPEG image | **(b)** Color-based segmentation with K = 4 |
| **(c)** Color-based segmentation with K = 8 | **(d)** Color-based segmentation with K = 16 |
| **(e)** Color-based segmentation with K = 32 | **(f)** Color-based segmentation with K = 64 |
| **(g)** Color-based segmentation with K = 128 | **(h)** Color-based segmentation with K = 256 |

**Figure 2.** The horse image segmented using different values of K

When increasing the value of K, the SSE returned by the k-means algorithm was obviously decreasing and the segmented image started to resemble the original image. It’s interesting to notice that setting K = 256 basically allowed to obtain a 256-color palette version of the initial picture.

The color-based technique adopted for the project is naive and truly effective only in particular circumstances, and it’s rarely adopted when the goal is to achieve an high quality segmentation. However, the actual purpose of the activity was to implement, parallelize and assess the performance speedup of the k-means algorithm, which has been used for the clustering of the pixels. Color-based image segmentation has just been chosen among the many applications of the k-means algorithm for its practicality and originality.

# The Parallelism Available in K-means

Before rushing into the implementation of a parallel version of the k-means clustering algorithm to speed up the color-based segmentation process, it was necessary to conduct a feasibility study to assess whether it was actually possible to parallelize the serial code and to understand how the parallelization should have been done.