Parallelization of the Algorithm *k-means* Applied in Image Segmentation

Cristian José López Del Álamo
La Salle University
National University of San Agustin
Arequipa, Peru

Lizeth Joseline Fuentes Pérez
National University of San Agustin
Arequipa, Peru

Luciano Arnaldo Romero Calla
National University of San Agustin
Arequipa, Peru

ABSTRACT

Algorithm *k-means* is useful for grouping operations; however, when is applied to large amounts of data, its computational cost is high. This research propose an optimization of *k-means* algorithm by using parallelization techniques and synchronization, which is applied to image segmentation. In the results obtained, the *parallel k-means* algorithm, improvement 50% to the algorithm *sequential k-means*.

General Terms:

speedup

Keywords:

parallelization, *k-means*, segmentation, images

1. INTRODUCTION

Several unsupervised learning algorithms have been proposed, which divide the set of objects in a number of groups according to an optimization criterion.

The grouping is defined, generally, as a process of organizing objects into groups whose members exhibit some kind of similarity [6]. The goal of clustering is to divide the set of objects, which have attributes associated to multidimensional vectors, into homogeneous groups so that patterns within each group are similars.

K-means is a algorithm widely used in clustering problems objects according to their attributes, in this regard, has been widely studied due to its applications in areas as machine learning [4], data mining [2], knowledge discovery [5], pattern recognition and classification [1], segmentation of medical images [3, 8], medical and general image [9].

Although, one biggest drawbacks of *k-means algorithm* is the high computational cost of calculating distances between all objects. For this reason, this research use parallelism and synchronization techniques in order to minimize the computational cost.

In the research, the *parallel k-means* algorithm, is applied to image segmentation, which seeks to group pixels with similar colors, improving the computational time in the segmentation, with large number of pixels, as well as, algorithm's iterations.

In the section 2 preliminary concepts are summarized, in the section 3 the methodology followed is presented and finally in the sections 4 and 5 the experiments and conclusions of the research is exposed.

2. K-MEANS ALGORITHM AND SEGMETATION

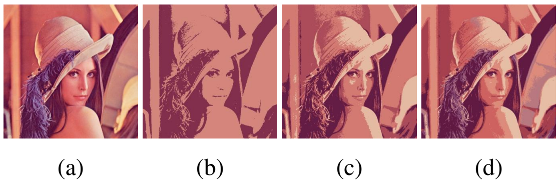
K-means is a partitional clustering algorithm, was proposed by *Stuart Lloyd* in 1957, although not published until 1982 [7]. A more efficient version was proposed and published in *Fortran* with *Hartigan and Wong 1975* [4].

The *k-means algorithm* is an unsupervised clustering algorithm that classifies the input data, represented as multidimensional points, based on their inherent distance of each other [9].

The algorithm *k-means* has computational order equal to $O(n*k*s)$ where $n =$ number Iterations, $k =$ number Groups and $s =$ number of elements to group.

The image segmentation algorithm is based on *k-means*, the pixels are clustered around k centroids C_i, $\forall i = 1 \ldots k$, which are obtained by running the algorithm 1, where N is the number of iterations, K the number of centroids and T the number of pixels.

The figure 1, show the image of Lena 1a, segmented with two centroids 1b, with four centroids 1c and with eight centroids 1d.



(a) (b) (c) (d)

Fig. 1: *K-means image segmentation*

Algorithm 1 k-means clustering algorithms

```
 1: procedure K-MEANS($k, ObjetsList, Size, Iterations$)
 2:     $Clusters[k]$
 3:     $Centers$ = list of k random centers.
 4:     $n = 0, c$
 5:     while $n \leq Iterations$ do
 6:         Clear($Clusters$)
 7:         for $i \in ObjetsList$ do
 8:             $lessDistance = INF$
 9:             for $j \in Centers$ do
10:                 $d = Distance(i, j)$
11:                 if $d < lessDistance$ then
12:                     $lessDistance = d$
13:                     $c = j$
14:                 end if
15:                 $Add(Clusters[c], i)$
16:             end for
17:             $Centers = NewCenter(Clusters)$
18:         end for
19:     end while
20: end procedure
```

In the next section, the methodology used in this research will be explained, and in the sub section 3.2, the image segmentation algorithm using the *sequential k-means* and *parallel k-means* will be exposed.

3. METHODOLOGY

In this section the methodology used is described; all steps for the image segmentation with sequential and parallel *k-means* algorithm are showed in the figure 2.

3.1 Features vector of a pixel

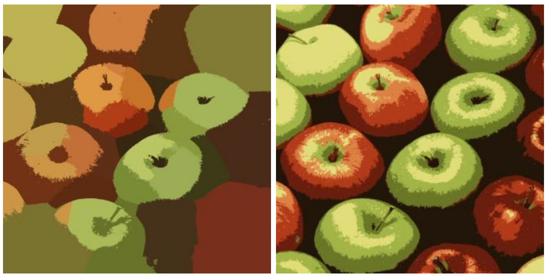
Given a set of test images, the RGB color space is used, and two types of features vector of each pixel is taken.

3.1.1 Features vector RGB. Lets p_{ij} a image pixel, then $P_{ij}(r, g, v)$ is a vector, such that, r, g and b are the colors value in the pixel (i, j) of the image and represent the descriptor from a color image. On the other hand, in the case of considerable color variation, represents a texture segmentation.

3.1.2 Features vector RGBXY. In this case, descriptor in $p_{i,j}$ is $P_{i,j}(r, g, b, x, y)$. This representation leads to a segmentation based on both, the color and position, of the pixel in the image.

3.2 Segmentation

The sequential and parallel *k-means* algorithm is applied in image segmentation process, considering the feature vectors RGBXY and RGB. With this two feature vectors, two different results are obtained, because RGBXY considers the position X,Y besides of the color, while RGB only consider the color of the pixels. Figure 3 show the segmenting of one image considering the feature vector RGB 3b and RGBXY 3a.



(a) (b)

Fig. 3: Image Segmentation with *k-means algorithm*

3.2.1 Secuential k-means. Described in the section 2.

3.2.2 Parallel k-means. The optimization process seek to parallelize the nearest centroid in line 7 of the algorithm 3; because there are no dependencies between one iteration to another. On the other hand, the line 5 remains sequentially because each iteration depends on the centers that have been calculated in previous iterations. Furthermore, it may include calculating on-site of the new centers, if done by the average, at the same time, by calculating the belonging to each group. On the other hand, it is important to use a critical section on line 15.

3.3 Experiments and results

After performing the respective tests with different images and number of threads, a result table and a graph showing the *speedup* of parallel algorithm is obtained. This process is explained in the section 4.

3.4 Comparison

Analysis of the final results, conclusions and discussion, will be discussed in section 5.

Figure 4 illustrates the input image 4a, 1600 × 1000 pixels, which is applied to the parallel algorithm with 4 threads, 20 iterations and 2, 4 and 8 centroids; segmenting into 2 groups 4b, 4 groups 4c and 8 groups 4d respectively.

4. EXPERIMENT AND RESULT

The tests were performed on an Intel (R) Core (TM) i3 2.27 GHz processor with 2.8GiB RAM. For the parallelization of the *k-means* algorithm was used the *OpenMP library* for *C++* and images for testing as shown below.

In Figures 5 and 6, the application of image segmentation is shown with the parallel *K-means* algorithm. 5a and 6a are the original images, in 5b and 6b images are segmented with two centroids, in 5c and 6c with 4 centroids. Finally, in 5d with 32 centroids (some centroids may not have grouped pixels) and 6d with 8 centroids.

After empirical testing images, is desired to obtain the optimal speedup for our algorithm, which is represented by the largest value of the inverse of the time to take a test run with N threads.

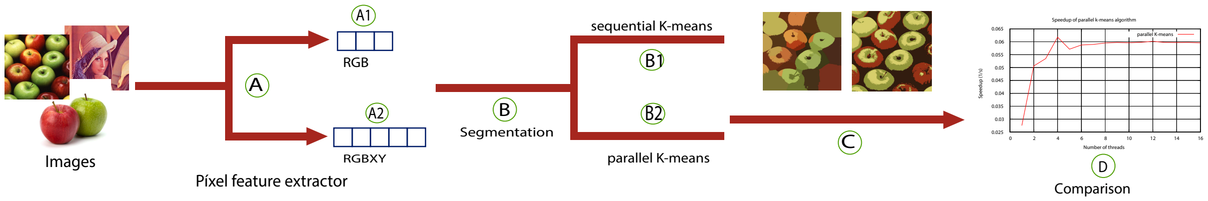


Fig. 2: *Research methodology*

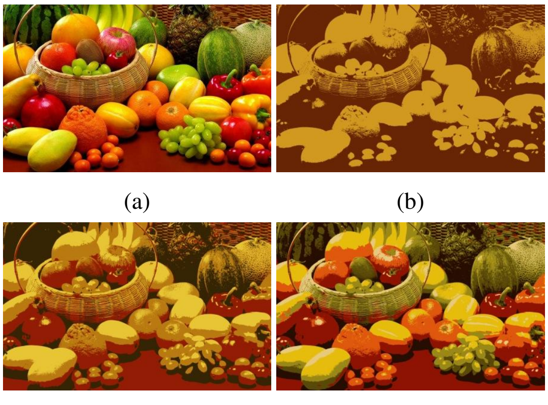

(a) (b) (c) (d)

Fig. 4: *image segmentation with parallel k-means algorithm*

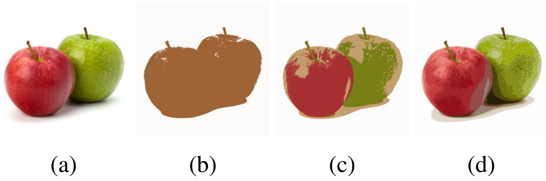

(a) (b) (c) (d)

Fig. 5: *Image segmentation with the parallel k-means algorithmo*

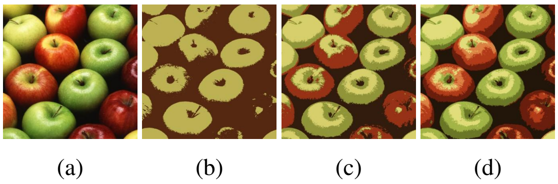
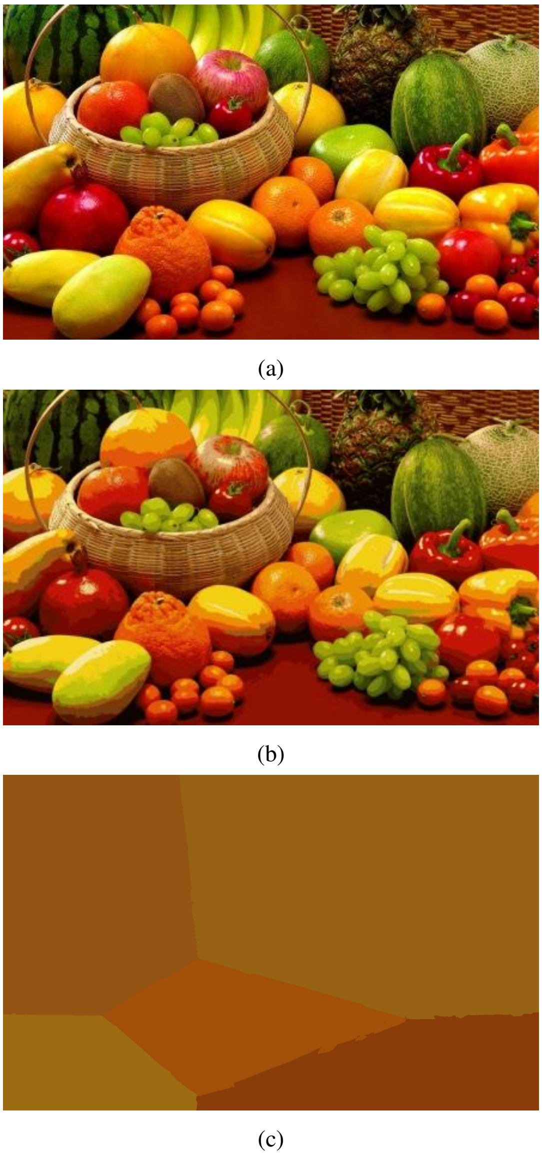

(a) (b) (c) (d)

Fig. 6: *Image segmentation with the parallel k-means algorithm*

For it, will take of the Figure 7, the image 7a, as test image, which has a dimension 1600 × 1000 pixels, the number of centroids will be 20 and the number of iterations 50.

The segmentation process with different numbers of threads will be run, which vary from 1 to 16. For each set of threads, 10 experiments were conducted and the result is taken as the average time for each of the experiments.
Figure 7b illustrated the segmentation result for one of the tests. The test had a total time of 48.8454*min*.


(a)

(b)

(c)

Fig. 7: *Image segmentation with the parallel k-means algorithm*

Table 1 show the result of the test, where you can see the average *speedup* for each number of threads, average time and improving respectively, compared to the sequential execution. The optimal *speedup* is in the test run with 4 threads ***(*)***, having improved 55% relative to the sequential algorithm and an average run-time 16.1881s. In *(*)* additional evidence that clearly show that the *speedup* decreases when the number of threads increases.

Nro de hilos	*Speedup*	Time (s)	Improvement (%)
1	0.0276079	36.2215	0.0000
2	0.0506200	19.7550	45.4606
3	0.0534600	18.7056	48.3577
(*) 4	**0.0617736**	**16.1881**	**55.3080**
5	0.0571196	17.5071	51.6666
6	0.0587511	17.0209	53.0088
7	0.0589585	16.9611	53.1739
8	0.0595100	16.8039	53.6079
9	0.0597033	16.7495	53.7581
10	0.0595853	16.7827	53.6665
11	0.0597975	16.7231	53.8310
12	0.0601453	16.6264	54.0980
13	0.0597482	16.7369	53.7929
14	0.0596862	16.7543	53.7449
15	0.0597196	16.7449	53.7708
16	0.0595534	16.7916	53.6419
100 (*)	***0.0537750***	***18.5960***	***48.6603***
1000 (*)	***0.0525497***	***19.0296***	***47.4632***
10000 (*)	***0.0374081***	***26.7322***	***26.1980***

Table 1. : *Tests results, (*) additional tests*

Figure 8 shows the pictures of average *speedup* depending on the number of threads, and in the tablet 1, a clear improvement of the parallel algorithm is observed in relation to sequential algorithm.

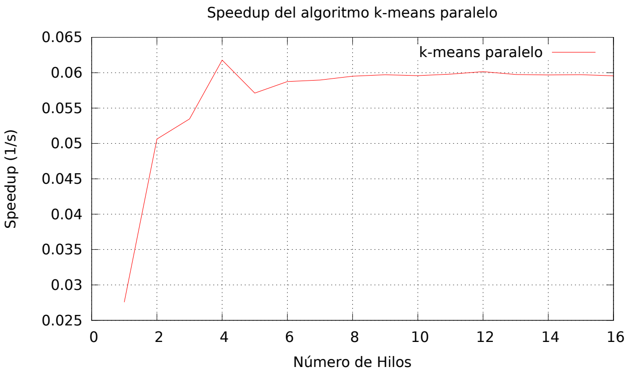


Fig. 8: *Speedup of parallel k-means algorithm in image segmentation*

5. CONCLUSIONS

From the experiments, is concluded that the parallel *k-means* algorithm with 4 threads, outperforms by 55% to the sequential *k-means* algorithm. In the test performed, in general, the the parallel *k-means* optimizes sequential *k-means* . In fact, for a small number of iterations, or small groups images is convenient to use the sequential algorithm. However, for large images where the processing of the segmentation and clustering is heavy is preferable to use parallel techniques.

From Table 1, is concluded that the optimal *speedup* is obtained with 4 threads, and to increase more threads, the performance decrease. Furthermore, is found that the parallel algorithm has better overall performance than the sequential. But also, the performance tends to decrease with the number of threads as shown in the tests.

6. FUTURE WORK

In future work, the parallel *k-means* algorithm will be applied on specific problems, such as, segmentation of satellite images. In that sense, will be developed techniques for distributed processing, in order to split an image of several mega bytes of size, on multiple machines, so that each run a part of the process of segmentation and clustering.

Acknowledgment

We thank Professor Wilber Ramos Lovón, Principal of School of Computer Science from National University of San Agustín, for his unconditional surrender to their students, to the National University of San Agustín and La Salle University for supporting this research.

7. REFERENCES

[1] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1 edition, February 1973.

[2] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. The MIT Press, February 1996.

[3] A. Ferreira, J. M R S Tavares, and F. Gentil. A review of segmentation algorithms for ear image data. In *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, pages 1–6, 2012.

[4] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[5] Mehmed Kantardzic. *Data Mining: Concepts, Models, Methods and Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[6] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm, 2001.

[7] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006.

[8] H. P. Ng, S. H. Ong, K. W C Foong, P. S. Goh, and W.L. Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65, 2006.

[9] Suman Tatiraju and Avi Mehta. Image segmentation using k-means clustering, em and normalized cuts.