New Three Methods for Improving Initialization of *k*-Means Clustering

*Abbas H. Hassin Alasadi**, *Moslem Mohsinn Khudhair*

Department of Computer Science, College of Science, University of Basra, Basra, Iraq

*Corresponding author:Email address : abbashh2002@yahoo.com (Abbas H. Hassin),
P..Box 716, Ashar, Basrah, Iraq.
Tel. 009647809835559, Fax: 0096440414811*

Abstract

The traditional *k*-means algorithm is a classical clustering method which widely used in variant application such as image processing, computer vision, pattern recognition and machine learning. It is known that, the final result depends on the initial starting points. Generally, initial cluster centers are selected randomly, so the algorithm could not lead to the unique result. In this paper, we present a new algorithm which includes three methods to compute initial centers for *k*-means clustering. First one is called geometric method which depends on equal areas of distribution. The second is called block method which segments the image into uniform areas. The last method called hybrid which combined between first and second methods. The experimental results appeared quite satisfactory.

Keywords: clustering; k-means algorithm; color Image; Image segmentation

1. Introduction

Image segmentation is the first step of the most critical tasks of image analysis, as shown in Figure (1). It is used either to distinguish objects from their background or to partition an image onto the related regions [1, 2].

The process of image segmentation is defined as: “the search of homogenous regions in an image and later the classification of these regions”. It also means the partitioning of an image into meaningful regions based on homogeneity or heterogeneity criteria. Image segmentation techniques can be differentiated into the following basic concepts: pixel oriented, Contour-oriented, region-oriented, model oriented, color oriented and hybrid. Color segmentation of image is a crucial operation in image analysis and in many computer vision, image interpretation, and pattern recognition system, with applications in scientific and industrial field(s) such as medicine, Remote Sensing, Microscopy, content based image and video retrieval, document analysis, industrial automation and quality control. The performance of color segmentation may significantly affect the quality of an image understanding system. The most common features used in image segmentation include texture, shape, grey level intensity, and color [3].

Partitional clustering algorithms such as *k*-means and Exception Maximize (EM) clustering are widely used in many applications such as data mining, compression, image segmentation, and machine learning.

Therefore, the advantage of clustering algorithms is that the classification is simple and easy to implement. Similarly, the drawbacks are of how to determine the number of clusters and decrease the numbers of iteration**.**

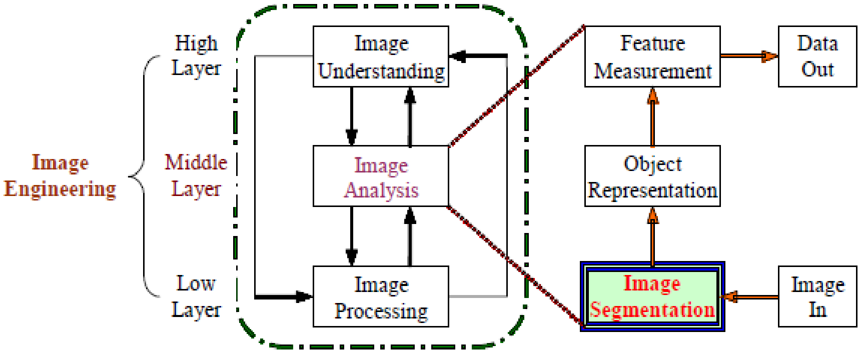


Figure (1): Image Engineering and Image Segmentation [1].

2. Related Works

Several attempts were made by researchers to improve the effectiveness and efficiency of the *k*-means algorithm. There are many researchers suggest initialized method of centroids of k-means algorithm.

Bradley and Fayyad [4] present a technique for initializing the *K*-means algorithm. They begin by randomly breaking the data into 10, or so, subsets. Then they perform a *K*-means clustering on each of the 10 subsets, all starting at the same set of initial seeds, which are chosen using Forgy’s method. The result of the 10 runs is 10*k* centre points. These 10*k* points are then used as inputs of *K*-means algorithm and the algorithm run 10 times, each of the 10 runs initialized using the *k*-final center locations (known as centroid) from one of the 10 subset runs. The resulting *K*-center locations are used to initialize the *K*-means algorithm for the entire datasets.

Douglas and Michael [5] proposed a method to select a good initial solution by partitioning dataset into blocks and applying *k*-means to each block. But the time complexity is slightly more. Though the above algorithms can help finding good initial centers for some extent, they are quite complex and some use the *k*-means algorithm as part of their algorithms, which still need to use the random method for cluster center initialization.

3. Clustering Analysis

Clustering analysis is one of the major data analysis methods widely used in many practical applications of emerging areas. Clustering is the process of finding groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups. A good clustering method will produce high quality clusters with high intra-cluster similarity and low inter-cluster similarity. The quality of a clustering result depends on both the similarity measure used by the method and its implementation and also by its ability to discover some or all of the hidden patterns [6,7].

There are two main branches of clustering: (1) hierarchical and (2) partitional [8]. In this paper, we concentrate on partitional clustering. Particularly, a popular partitional clustering method called *k*-means clustering. The problem of clustering is to partition a data set consisting of *n* points embedded in m-dimensional space into *K* distinct set of clusters such that the data points within a cluster are more similar among them than to data points in other clusters. There are a number of proximity indices that have been used as similarity measures [9]. Unfortunately, *K*-means algorithm is extremely sensitive to the initial choice of cluster centers, and a poor choice of centers may lead to a local optimum that is quite inferior to the global optimum [10,11].

4. *K*-means Clustering Algorithm

Let $X=\{x_{1,}\, x_2, \ldots, x_n\}$ be the set of *n* data in *d*-dimensional points to be clustered into a set of *k* clusters, $C=\{c_1, c_2, \ldots , c_k\}$; *K*-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized [12, 13]. Let μ_k be the mean of cluster c_k. The sum squared error (*SSE*) between μ_k and the points in cluster c_k is defined as:

$$SSE = \sum_{X_i \in c_k} \left\| X_i - \mu_k \right\|^2 \qquad (2-5)$$

The goal of *K*-means is to minimize the sum of the squared error over all *k* clusters,

$$SSE = \sum_{k=1}^{K} \sum_{X_i \in c_k} \left\| X_i - \mu_k \right\|^2 \qquad (2-6)$$

The centroid (mean) of the i^{th} cluster is defined by Equation (2) [14]:

$$\mu_i = \frac{1}{m_i} \sum_{x \in C_i} X \qquad (2-7)$$

where m_i the number of objects in the i^{th} cluster

K-means starts with an initial partition with *k* clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error always decrease with an increase in the number of clusters *k* (with *SSE*= 0 when *k*= *n*), it can be minimized only for a fixed number of clusters [15]. A pseudo-code for the *K*-means algorithm1 is shown in Figure (3).

```
Input data              // Image of size (length x width)
Input k          // Number of clusters
MSE = Large number ;
CALL Initialization ;
Do
 Old-MSE = MSE ;
 MSE1 = 0 ;
 For j = 1 to k
        $\mu_j$ = 0 ;
              $n_j$  = 0 ;
        end for j
        For i = 1 to n
              For j = 1 to k
                    Compute squared Euclidean distance $d^2(x_i, m_j)$ ;
              end for j
              Find the closest centroid $m_j$ to $x_i$ ;
              $\mu_j = \mu_j + x_i$ ;
              $n_j = n_j + 1$ ;
              $MSE1 = MSE1 + d^2(x_i, \mu_j)$ ;
        end for i
        For j = 1 to k
              $n_j = \max(n_j, 1)$ ;
              $\mu_j = \mu_j / n_j$ ;            // Re-calculation centroids
        end for j
        MSE = MSE1 ;
WHILE (MSE - MSE1< T)
```

Figure (3): A pseudo-code for the *K*-means algorithm1**.**

The specification of the points of cluster centroids is done by ***call initialization*** which is depicted in Figure (4).

```
CALL Initialization

      Select initial cluster centroids { $\mu_j$ } (j=1, 2, …, k) randomly
     from input image that has bounded with size (length x width) ;

Return
```

Figure (4): Call algorithm for initializing cluster centroids.

5. Proposed Methods and Experimental Results

We proposed an efficient algorithm which consists of three methods to compute initial centers for *k*-means clustering. First one is called geometric method which depends on equal areas of distribution. The second is called block method which segments the image into uniform areas. The last method called hybrid which combined between first and second methods.

5.1. Geometric *K*-means Algorithm

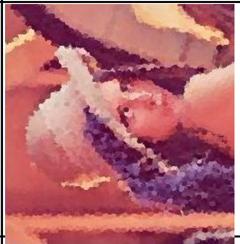
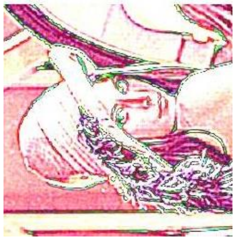
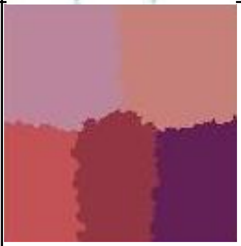
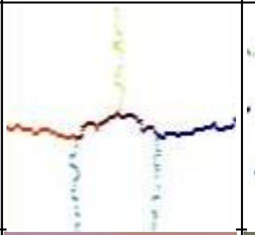
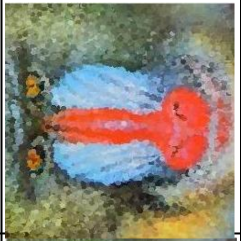
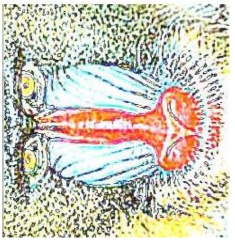
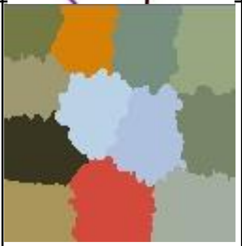
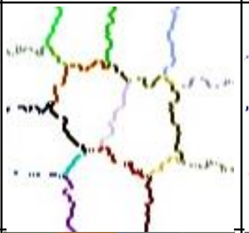
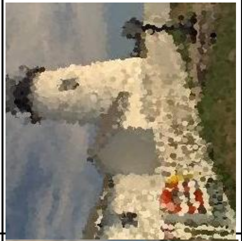
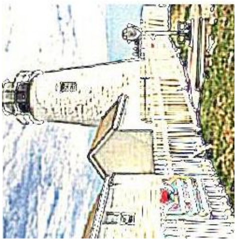
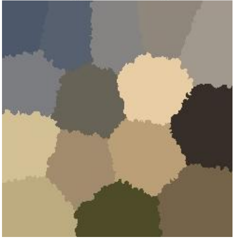
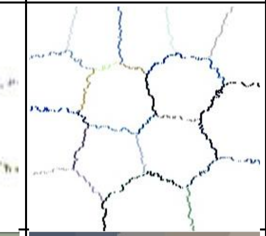
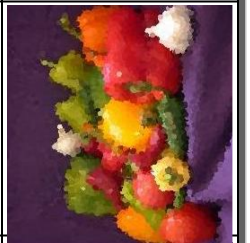
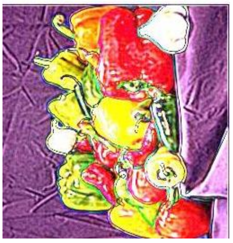
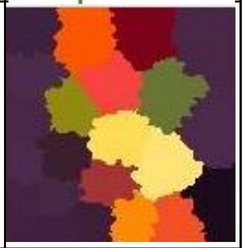
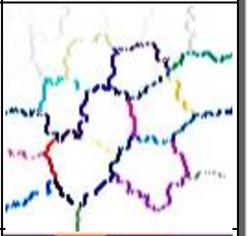
A propose algorithm is presented and includes a method to compute initial center clusters for *K*-means clustering. It is called geometric method which depends on equal areas of distribution. The *geometric algorithm2* has the same *algorithm*1 in Figure (3) but it is different in the method of generation of initial centroid of clusters. Figure (5) depicts an algorithm of initialized points of cluster centroids.

```
CALL Initialization
        $r$ = Integer ( $\sqrt{K}$ ) ;
    $q$ = Integer ($K$ / $r$) ;
    $p1$ = Integer ($width$ / $r$) ;
    $p2$ = Integer ($height$ / $q$) ;
    $xx$= Integer ($p1$ / 2) ;
    $yy$= Integer ($p2$ / 2) ;
    For $i$ = 1 to $r$
            For $j$ = 1 to $q$
                $\left.\begin{array}{l} x(1,i) = xx \\ y(1,j) = jj \end{array}\right\}$    $point\ centroid\ cluster$
                $xx = xx + p1$
            Next $j$
            $yy = yy + p2$
            xx = $Int$ ($p1$ / 2)
    Next $i$
Return
```

Figure (5): geometric Algorithm2 distribution for initializing cluster centroids.

In this method, the algorithm scans the dataset block by block. The same calculations and experiments in previous the algorithm have been implemented. Although this algorithm is more expensive in calculation than the previous algorithm but it is regarded more accurate for images that have regular distribution intensity. This matter is very considered to be very important in later section to implemented hybrid algorithm.

Figure (6) shows experimental results obtained by implementing geometric *K*-means algorithm including different operations on four images.

Original Image	Segment. result	Smooth & Lap.	Region Cluster.	Sobel Edge	K	No. Iteration
					5	20
					12	23
					15	59
					24	20

5.2. Block *K*-means Algorithm

The proposed algorithm is compresses the data into smaller dataset by producing *K*-means from each block, if the dataset contains *m* blocks, then the compressed data will contains (*k* x *m*) objects. Our algorithm scans the original dataset only one times and produces better clusters. Figure (7) summarizes the steps of the block *K*-means algorithm2.

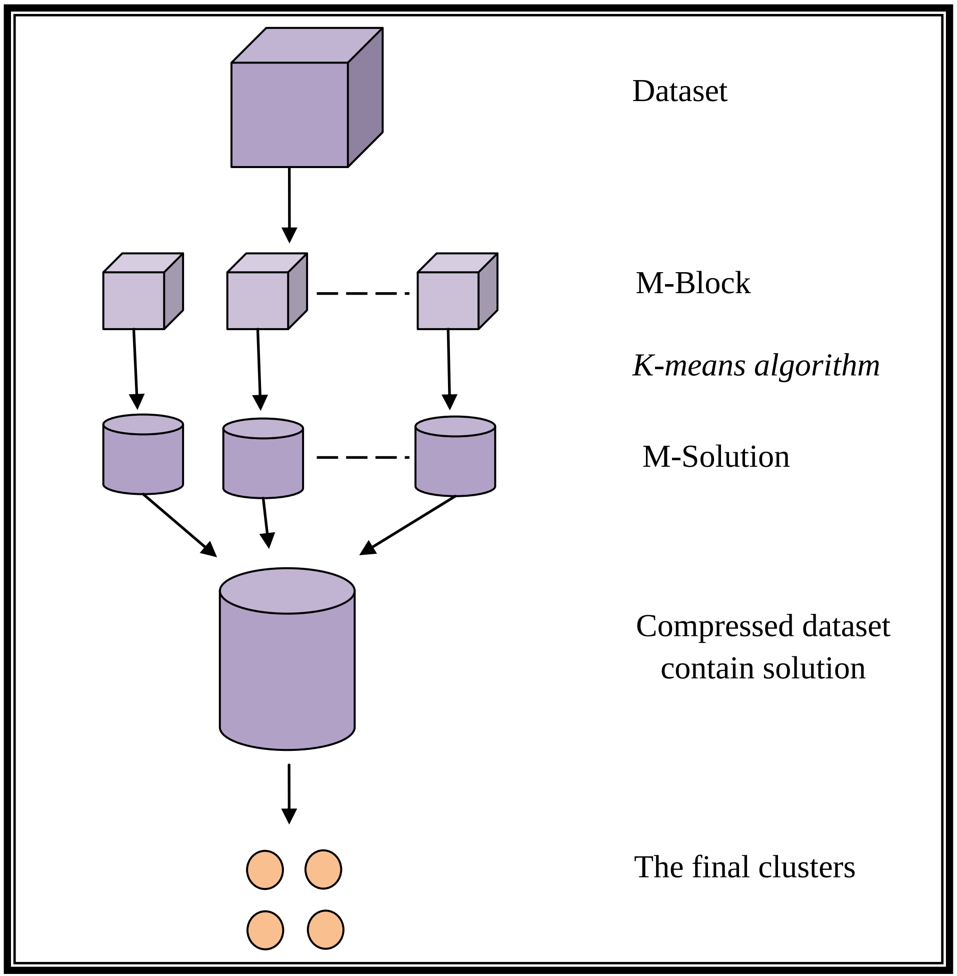


Figure (7): An overview of the Block K-means.

The main idea of this algorithm is to compress the dataset into finite number of representatives. Each representative indicates the mean value of some data points form a small cluster. This process has been done at the first phase. In the second phase we apply the *k*-means on the compressed dataset. Figure (8) exhibits the *block algorithm*.

```
Set the size of the block to s length;
row = width/s ;
col = height/s ;
For i = 1 to row
    For j = 1 to col
        read block(i, j);
      Call algorithm1 ; // With Random Initialization
        save the result ;
    end for i
end for j
```

Figure (8): Block K-means algorithm3.

Note that, in Figure (8), the method determines the size of each block and the user should determine the required number of partitions in each block. It is clear that in line 7, it is required to call implementing standard *K*-means algorithm for each block as same as *algorithm1*. From experimental results it will be better to use a small value of *k* (*2*-*9*) for each block and depends on size of image.

Figure (9) shows experimental results obtained by implementing block *K*-means algorithm including different operations on four images.

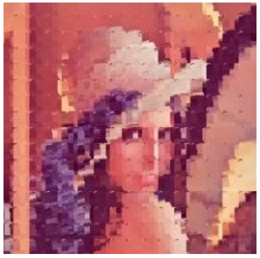
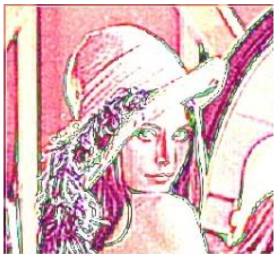
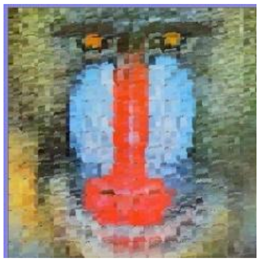
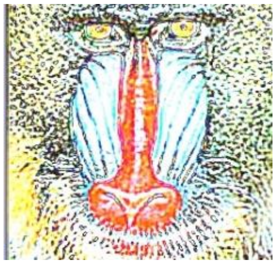
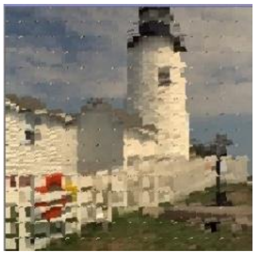
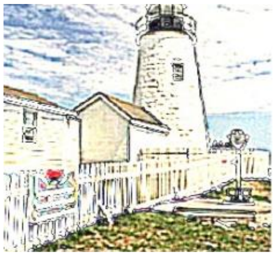
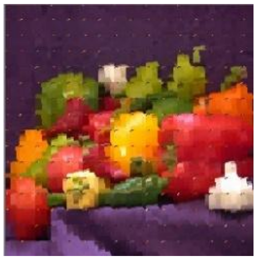
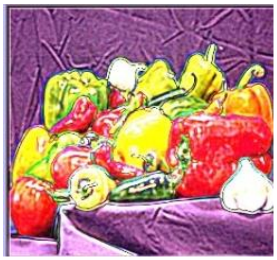
Original Images	**Segment Result**	**Smooth& Laplacian**	***K***
			5
			12
			15
			24

Figure (9): Result of execution Block K-means algorithm.

5.3. Hybrid *K*-means Algorithm

Hybrid *K*-means algorithm combined of the block *K*-means algorithm and the geometric *K*-means algorithm. It's the same block method but it has used the geometric initialized method instead of randomly distributed cluster centroids. Figure (10) exhibits the *hybrid algorithm4*.

```
1. Set the size of the block to s length ;
2. row = width/s ;
3. col = height/s ;
4. For i = 1 to row
5.      For j = 1 to col
6.           read block(i, j);
7.             Call algorithm1 ; // With Geometric Initialization
9.           save the result ;
10.     end for i
11. end for j
```

Figure (10): Hybrid K-means algorithm4.

Note that, in Figure (10), the method determines the size of each block. It is clear that in line 7, It is demanded to call implementing geometric *K*-means algorithm for each block as same as *algorithm2*.

Figure (11) shows experimental results obtained by implementing hybrid *K*-means algorithm including different operations on four images.

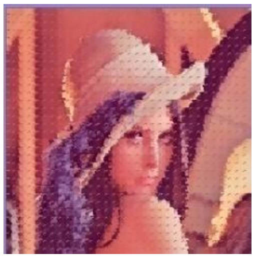
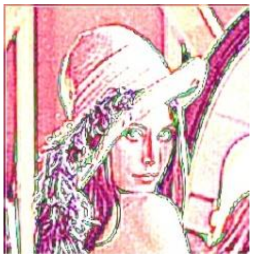
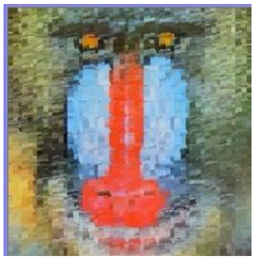
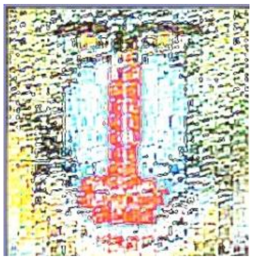
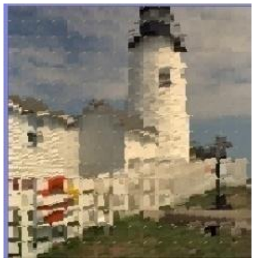
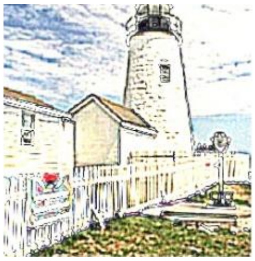
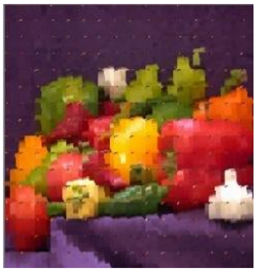
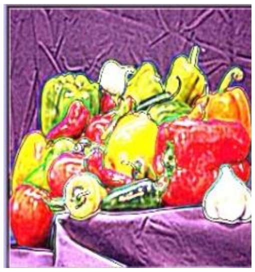
Original Images	Segment Result	Smooth & Lap.	*K*
			5
			12
			15
			24

Figure (11): Result of execution hybrid K-means algorithm.

6. Conclusion

K-means algorithm is a popular clustering algorithm applied widely, but do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Experimental results show that selecting centroids by our methods can lead to a better clustering. Moreover, the computational complexity of the standard algorithm is objectionably high owing to need reassign the data points a number of times for each iteration in the loop.

We have evaluated our methods on several different standard images. So, we have compared our results with that of *k*-means algorithm in terms of the total execution time and quality of clusters. Our experimental results are executed on PC 2.0GMHz CPU, 2.0GB RAM, 512 KB Cache.

Finally, we have compared the *CPU* time of the proposed methods (geometric, block & hybrid) with the standard *K*-means methods. The execution time of proposed methods is much lesser than the average execution time of *K*-means as shown in Figure (12).

Figure (13) demonstrates that the proposed methods (geometric, block, and hybrid) provide better cluster accuracy than the existing methods. It shows that proposed method performs much better than the random initialization algorithm. This is due to the initial cluster centers generated by proposed method which are quite closed to the optimum solution.

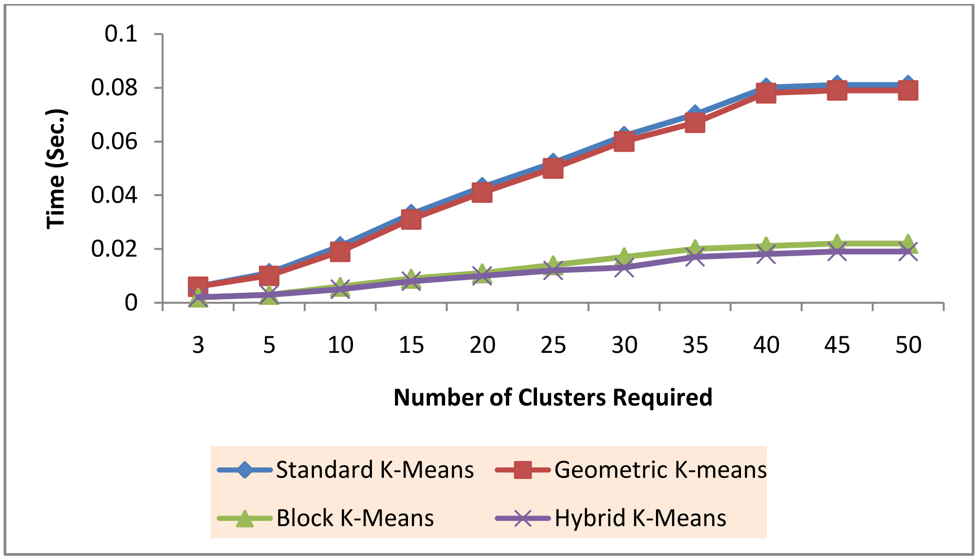


Figure (12) Execution Time (Lena Image).

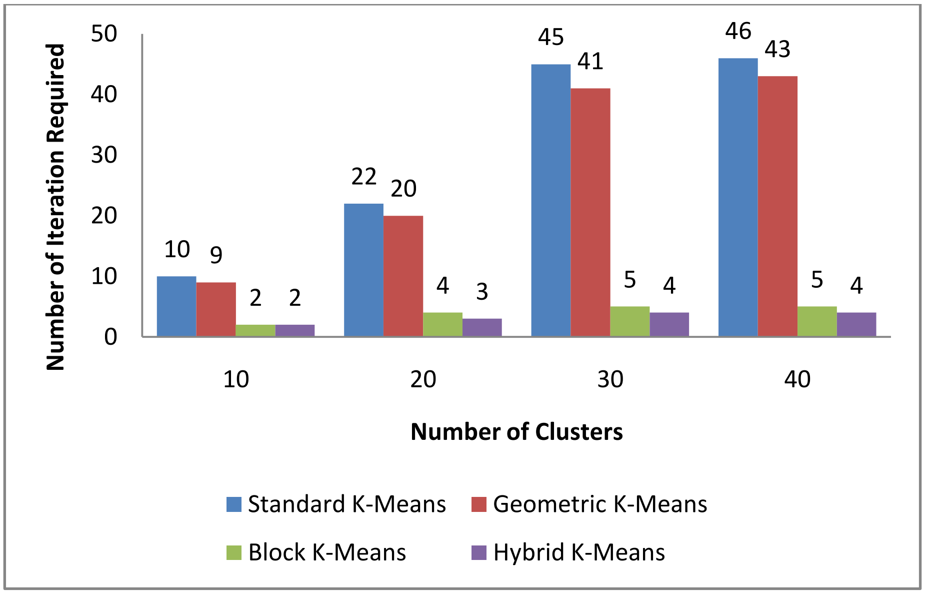


Figure (13) Number of Iteration (Lena Image).

References

[1] Yu-Jin Zhang, " Advances in Image and Video Segmentation", IRM Press, 2006.

[2] Ali Salem Bin Samma and Rosalina Abdul Salam, Adaptation of *K*-Means Algorithm for Image Segmentation, Int. J. Signal Processing 5:4 (2009) 270-274.

[3] Anil Z. Chitade, Dr. S.K. Katiyar, Colour Based Image Segmentation Using *K*-Means Clustering, Int. J. Eng. Sci. Tech. 2:10(2010) 5319- 5325.

[4] Paul S. Bradley and Usama M. Fayyad," Refining Initial Points for *K*-means Clustering", International Conference 15th on Machine Learning (ICML98), pp. 91-99. Morgan Kaufmann, San Francisco, 1998.

[5] Douglas Steinley, Michael J. Brusco, Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques, J. Classification 24 (2007) 99-121.

[6] Rajashree Dash, Debahuti Mishra, Amiya Kumar Rath, Milu Acharya, A Hybridized *K*-Means Clustering Approach for High Dimensional Dataset, Int. J. Eng. Sci. Tech. 2:2(2010) 59-66.

[7] Eshref Januzaj, Hans-Peter Kriegel, and Martin Pfeifle, " Towards Effective and Efficient Distributed Clustering", Workshop on Clustering Large Data Sets (ICDM), Melbourne, FL, 2003.

[8] Fisher, D., Knowledge Acquisition via Incremental Conceptual Clustering. Mach. Learn.1987.

[9] Jain, A.K., Murty, M.N., Flynn, P.J., Data clustering: A review, ACM Comput. Surveys 31:3(1999) 264–323.

[10] Anderberg, M.R., Cluster Analysis for Applications, Academic Press Inc.,1979.

[11] Xiaoping Qing, Shijue Zheng, A New Method for Initializing the *K*-means Clustering Algorithm**,** 2nd Int. Sym. Knowledge Acquisition and Modeling, (2009) 41-44.

[12] Joaquín Pérez O., Rodolfo Pazos R., Laura Cruz R., Gerardo Reyes S., Rosy Basave T., and Héctor Fraire H., " Improving the Efficiency and Efficacy of the *K*-means Clustering Algorithm Through a New Convergence Condition", O. Gervasi and M. Gavrilova (Eds.): ICCSA, LNCS 4707, Part III, pp. 674–682, 2007.

[13] Shehroz S. Khan, and Amir Ahmad, "Cluster center initialization algorithm for *K*-means clustering ", Pattern Recognition Letters 25, pp.1293–1302, 2004.

[14] Zuyi Huang, Fatih Senocak, Arul Jayaraman, and Juergen Hahn, " Solution of Inverse Problems for Obtaining Protein Concentrations from Fluorescent Microscopy Images ", American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, USA, pp. 1688-1693, 2009.

[15] Hong Liu and Xiaohong Yu, "Application Research of *K*-means Clustering Algorithm in Image Retrieval System", Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT'09) Huangshan, P. R. China, pp. 274-277, 26-28 Dec. 2009.