Hk-Means Clustering In Multimedia Applications For Pattern Recognition

$B.Padmini^1$, $K.Haripriya^2$
Department of Electronics & Communications Engineering[1,2] , Anurag Group of Institutions
RangaReddy Dist, Andhra Pradesh-501301

Abstract: **With the wide spread of embedded systems in various applications including in the medical field, complicated treatments are made easy by locating correct position of the affected area k-means algorithm is used to locate minute and deep affected areas for laser treatments. But the existing k-means architecture does not satisfy the cost of both computational time and hardware area. As the cluster number increases, struggle for maintaining the cost between computational time and hardware area also increases. To handle large cluster number, hardware architecture of hierarchical k-means (HK-means) is proposed to support maximum cluster number of 1024 for color clustering and image segmentation. It also offers maximum bandwidth to processing elements. The results show better implementation of video segmentation and color quantization based on proposed HK-means hardware.**

Key words: k-means, HK-means, Clustering.

1. INTRODUCTION

With the advances in imaging technology, diagnostic imaging has become an indispensible tool in medicine today. X-ray angiography, magnetic resonance angiography, magnetic resonance imaging, computed tomography and other imaging modalities are heavily used in clinical practice. Such images provide complementary information about the patient. While increased size and volume in medical images required the automation of the diagnosis process, the latest advances in computer technology and reduced costs have made it possible to develop such systems[1].

Blood vessel delineation on medical images forms an essential step in solving several practical applications. Segmentation algorithms form the essence of medical image such as visualization and computer-aided surgery[2]. Vessel segmentation algorithms are the key components of automated radiological diagnostic systems. Segmentation methods vary depending on the imaging modality, application domain, method being automatic or semi-automatic and other specific factors. Vessel segmentation techniques are classified as pattern recognition techniques, model based approaches, tracking based approaches, and neutral network based approaches and other tube like object detection methods. Clustering analysis is a part of pattern recognition technique. K-means is known as traditional clustering algorithm. K-means is mostly used for color clustering and image segmentation. For embedded systems high performance hardware for multimedia content analysis for customer electronics functioning is required. But due to tedious computations of K-means and design limits for hardware implementations need to accelerate the clustering process is to be proposed. But due to different applications the hardware specifications vary. *Jia-Li et al* proposed overall idea of K-means algorithm but as data scale increases rapidly it is difficult to use K-means and deal with massive data. *Duo et al* proposed algorithm transformation to map K-means clustering to FPGA hardware and the channel number is 10 for simulated multispectral thermal image data sets[3]. But in all the previous works K-means hardware architecture was designed for multimedia content analysis functionalities.

As the cluster number is increasing for newly developed algorithms various options for K-means algorithms with different hardware architectures for various applications are developed for large cluster number[4]. The K-means algorithms existing are not able to satisfy the costs of both hardware area and computational time. When cluster number is fixed or low both area costs and computational time are inverse proportional i.e. if the computational time is reduced, the number of parallel processing elements will increase. But when the cluster number increases the costs of embedded systems becomes very important as the struggle between the computational time and hardware area becomes severe. To overcome this drawback new hardware architecture based on hierarchical K-means algorithm is proposed. HK-means algorithm first initializes centroids uses memory structure to store the cluster centroids and total often processing elements for distance calculations and binary tree traversal are employed to compute nearest centroid process in pipeline. In the present work Architectural analysis of HK-means algorithm and design space exploration for high cluster number is proposed. The proposed architect is most suitable for multimedia content analysis in the next generation.

In Section 2 K-means algorithm is described, Section 3 describes proposed HK-means algorithm, Section 4 introduces to morphology, Section 5 describes Software and Hardware requirements, Section 6 shows experimental results and Section 7 gives the conclusion and list of references is followed.

2. K-MEANS ALGORITHM

Clustering can be considered the most important unsupervised learning problem. So, it deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. K-means is one algorithm which satisfies the above requirement and used more often. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters).[5],[6] The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bar centers of the clusters resulting from the previous step.

After we have these k new centroids, a new binding has to be done between the same data set points and the nearest centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function where is a chosen distance measure between a data point and the cluster centre is an indicator of the distance of the n data points from their respective cluster centers. The most widely used clustering error criterion is squared error criterion, it can be defined as

$$J_c = \sum_{j=1}^{c} \sum_{k=1}^{n_j} \left\| x_k^{(j)} - m_j \right\|^2 \quad (1)$$

Where J_c is the sum of square error for all objects in the database, x_k is the point in space representing a given object and m_j is the mean of cluster C_j . Adopting the squared error criterion, K-means works well when the clusters are compact clouds that are rather well separated from one another and are not suitable for discovering clusters with non convex shapes or clusters of very different size. For attempting to minimize the square error criterion, it will divide the objects in one cluster into two or more clusters. In addition to that when applying this square error criterion to evaluate the clustering results, the optimal cluster corresponds to the extreme. Since the objective function has many local minimal values, if the result of initialization is exactly near the local minimal point, the algorithm will terminate at a local optimum. So random selecting initial cluster center is easy to get in the local optimum not the entire optimal. For overcoming that square error criterion is hard to distinguish the big difference among the clusters.

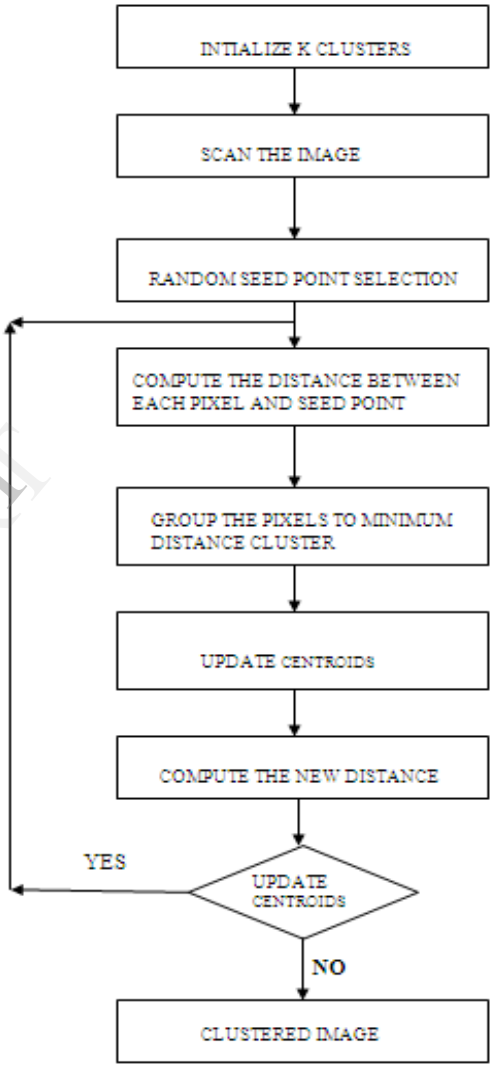

Fig 1. Flow of K-MEANS algorithm.

1. The Euclidian distance is used in this paper. The distance between one vector X = (x1, x2,….., xn) and the other vector Y= (y1,y2,…….yn) is described as follows

$$d(X,Y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_n-y_n)^2} \quad (2)$$

The distance between a data point X and a data

point d(X,V) = min (d(X,Y), Y EV). Suppose there are n data points in the population U and we want to partition U into k classes. Set m=l. Then the algorithm is described as follows. Compute distances between each data point and all of the other data points in U, find the two data points between which the distance is the shortest and form a data point set Am (l<m<k) which contains these two data points, delete these two data points from U.

2. Find the data point in U that is closest to the data point set Am add to Am and delete it from U.

3. Repeat step (2) till the number of data points in Am reaches

$$\alpha \times n/k \quad (0<\alpha<=1); \tag{3}$$

4. If m<k then m = m+1 find another pair of data points between which the distance is the shortest in U and form another data point set Am and delete from U then go to step (2).

5. For each Am(l<m<k) sum the vectors of data points and divide the sum by the number of data points in Am then each data point set outputs a vector and we select these vectors as the initial centroids.

6. Execute the process of the standard K-means algorithm from step 2.

The value of α is different with regard to different data. If the value of α is too small, all the centroids may be obtained in the same region that contains many similar data points, but if the value of α is too big, the centroids may stay away from the region that contains many similar data points.

3. HK-MEANS ALGORITHM

Original K-means algorithm choose k points as initial clustering centers, different points may obtain different solutions. In order to diminish the sensitivity of initial point choice, we employ a mediod which is the most centrally located object in a cluster to obtain better initial centers. The demand of stochastic sampling is naturally bias the sample to nearly represent the original data set, that is to say, samples drawn from data set cant cause distortion and can reflect original data's distribution. [7][8]Comparing two solutions generated by clustering sample drawn from the original dataset and itself using K-means, the location of clustering centroids of these two are almost similar. So the sample based method is applicable to refine initial conditions. In order to lessen the influence of sample on choosing initial starting points, following procedures are employed.

First drawing multiple sub samples (say J) from original dataset (the size of each sub sample is not more than the capability of the memory, and the sum for the size of J sub samples is as close as possible to the size of original dataset). Second use K-means for each sub sample and producing a group of mediods respectively. Finally comparing J solutions and choosing one group having minimal value of square error function as the refined initial points. To avoid dividing one big cluster into two or more ones for adopting square error criterion, we assume the number of clustering is K' (K'>K, K' depends on the balance of clustering quality and time). In general, bigger K' can expand searching area of solution space, and reduce the situation that there are not any initial values near some extremum. Subsequently, re-clustering he dataset through K-means with the chosen initial conditions would produce K' mediods, then merging K' clusters (which are nearest clusters) until the number of clusters reduced to k.

HK-Means refers to a top-down and divisive hierarchical clustering algorithm that adopts K-means clustering with cluster number in each stage. The concept of HK-means is usually applied to reduce the computational time in the software algorithm when the cluster number is large. The clusters are split into recursively in the Euclidian space, and K-means clustering with k=2 is performed locally in each level based on the clustering results of the previous level.

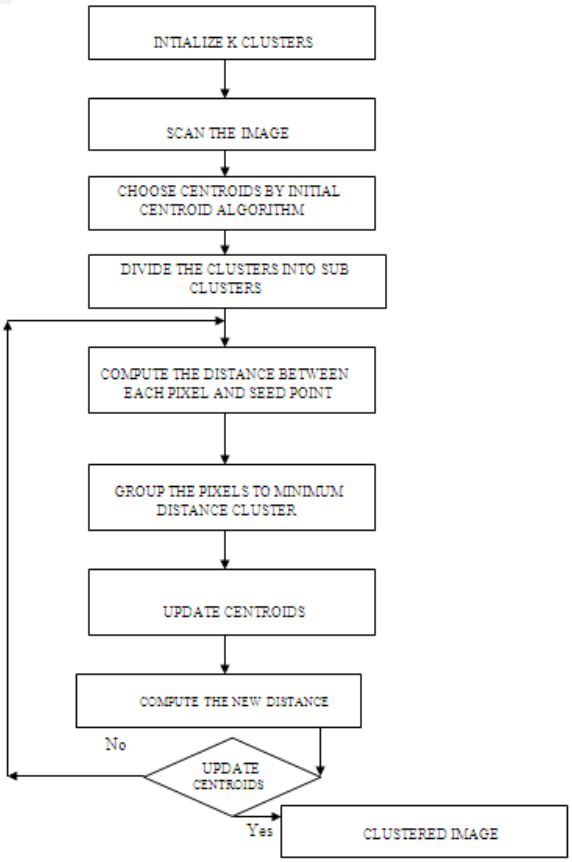

Fig 2. Flow of HK-MEANS algorithm.

4. MORPHOLOGY

The term morphology refers to the study of shapes and structures from a general scientific perspective. Also, it can be interpreted as shape study using mathematical set theory. In image processing, morphology is the name of a specific methodology for analyzing the geometric structure inherent withing an image. The morphological filter, which can be constructed on the basis of the underlying morphological operations, are more suitable for shape analysis than the standard linear filters since the latter sometimes distort the underlying geometric form of the image.

Some of the salient points regarding the morphological approach are as follows:

1. Morphological operations provide for the systematic alteration of the geometric content of an image while maintaining the stability of the important geometric characteristics.
2. There exists a well developed morphological algebra that can be employed for representation and optimization.
3. It is possible to express digital algorithms in terms of a very small class of primitive morphological operations.
4. There exist rigorous representations theorems by means of which one can obtain the expression of morphological filters in terms of the primitive morphological operations.

In general morphological operators transform the original image into another image through the interaction with the other image of a certain shape and size, which is known as structuring element. Geometric features of the images that are similar in shape and size to the structuring element are preserved, while other features are suppressed. Therefore morphological operations can simplify the image data preserving basic characteristics which can be used for edge detection, segmentation and enhancement of images.

Mathematical morphology is defined in Euclidian setting is called Euclidian morphology and that defined in a digital setting is called digital morphology. In the actual implementation we consider digital morphological setting. In binary and gray scale morphology the steps included are dilation, erosion, opening and closing of images.

5. HARDWARE AND SOFTWARE REQUIREMENTS

5.1 XILINX PLATFORM STUDIO (XPS):

XPS includes a graphical user interface (GUI), along with a set of tools that aid in project design. From the XPS GUI, we can design a complete embedded processor system for implementation within a Xilinx FPGA device.

XPS main window is divided into three areas:

The Projection Information panel: Projection information panel offers control over and information about project. The project information panel provides project, applications and IP catalog tabs.

The system Assembly Panel: The system assembly panel is where we view and configure system block elements. If the system assembly panel is not already maximized in the main window, click the system assembly tab at the bottom of the pane to open it.

The connectivity Panel: With the bus interface filter selected we see the connectivity panel, the connectivity panel is a graphical representation of hardware platform interconnection.

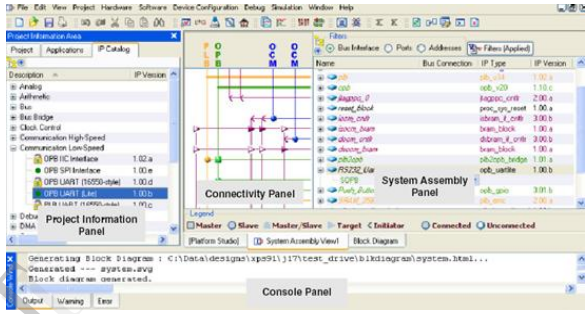

Fig 3. XPS GUI

5.2 INTEGRATED SOFTWARE ENVIRONMENT:

ISE is the foundation for Xilinx FPGA logic design. Because FPGA design can be an involved process, Xilinx has provided software development tools that allow the designer to circumvent some of this complexity. Various utilities such as constraints entry, timing analysis, logic placement and routing and device programming have all been integrated into ISE.

5.3 EMBEDDED DEVELOPMENT KIT:

Microprocessor Hardware Specification (MHS): XPS provides an interactive development environment that allows you to specify all aspects of hardware platform. XPS maintains hardware platform description in a high-level form, known as the microprocessor hardware specification file. The MHS an editable text file is the principal source file representing the hardware component of embedded system. XPS synthesis the MHS source files into Hardware Description Language net lists ready for FPGA place and route.

Microprocessor Software Specification (MSS): XPS maintains an analogous software system description in the MSS file. The MSS file together with software applications, are the principal source files representing the software elements of embedded systems. This collection of files allows XPS to compile applications. The compiled

Software routines are available as an Executable and Linkable Format (ELF) file. The ELF file is the binary ones and zeros that are run on the processor hardware.

Creating the project in XPS:

The following process is done to create the project in XPS;

- Building the user application.
- Compiling the code.
- Downloading the design.

5.4 SPARTAN -3E STARTER KIT:

The Spartan-3e Starter kit board highlights the unique features of the Spartan-3E FPGA family and provides a convenient development board for embedded processing applications.

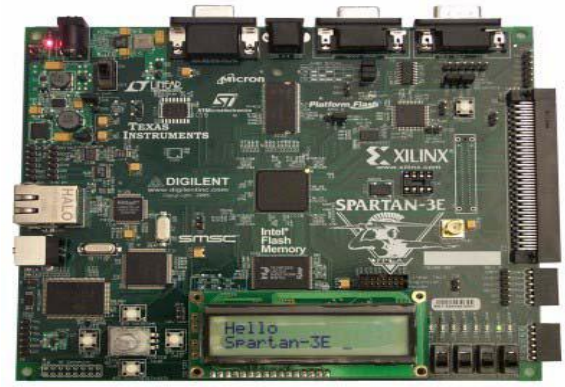

Fig 4. Spartan 3EDK

The board highlights following features:

Spartan-3E specific features:

- Parallel NOR Flash configuration.
- MultiBoot FPGA configuration from parallel NOR Flash PROM.
- SPI serial Flash configuration.

Embedded Development:

- Micro Blaze 32-bit embedded RISC processor.
- Pico Blaze 8-bit embedded controller.
- DDR memory interfaces.

5.5 VISUAL BASIC

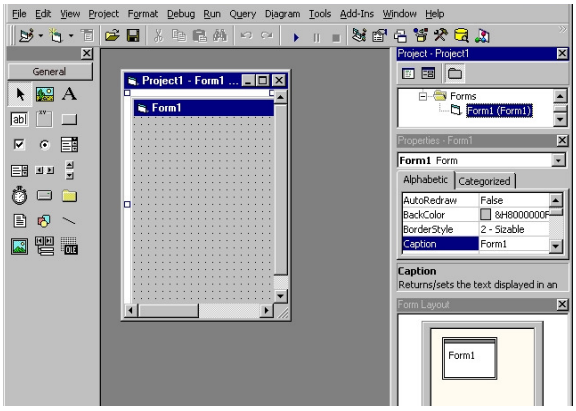

Fig 5. Visual Basic Environment

In Figure 5, the Visual Basic Environment consists:

- A blank form to design application's interface.
- The project window which displays the files that are created in applications.
- The properties window which displays the properties of various controls and objects that are created in application.
- It also has a Toolbox that consists of all the controls essential for developing a VB application.

First click on the project item on the menu then on the components item on the drop down list and lastly select the controls to use in the program.

6. EXPERIMENTAL RESULTS

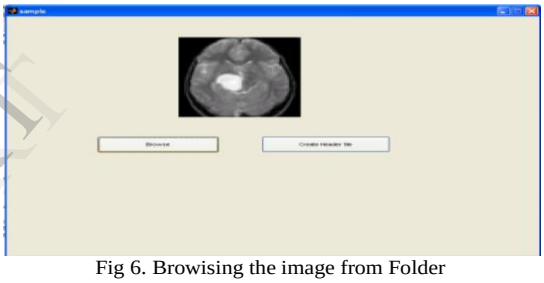

Fig 6. Browsing the image from Folder

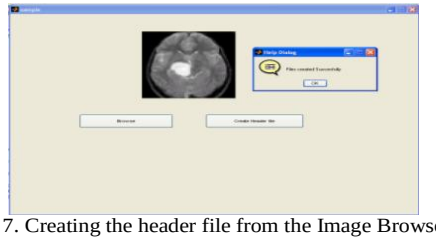

Fig 7. Creating the header file from the Image Browsed

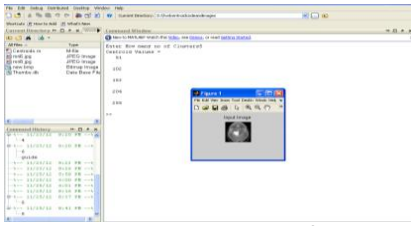

Fig 8. Calculating the Centroid values for input Image

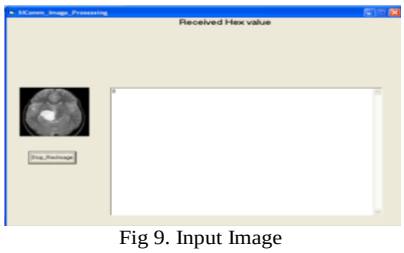

Fig 9. Input Image

Fig 10. Cluster1(first image)

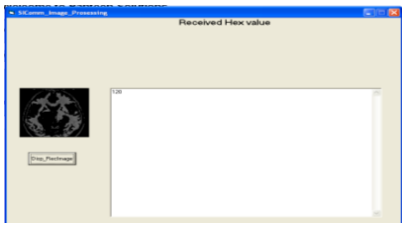

Fig 11. Cluster2 (second image)

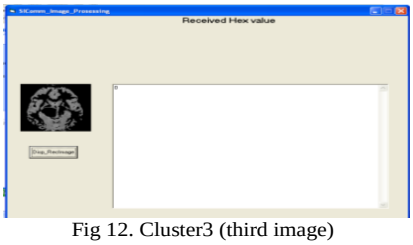

Fig 12. Cluster3 (third image)

Fig 13. Cluster4 (fourth image)

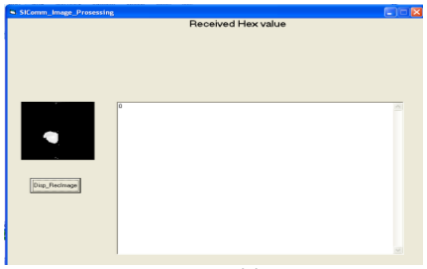

Fig 14. Cluster5 (fifth image)

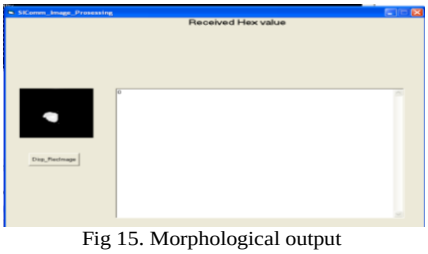

Fig 15. Morphological output

7. CONCLUSION

The above proposed architecture enables fast computation of K-means and morphological image filtering wit parallel processing to get accurate results in medical fields. By using both techniques in a combined manner we can achieve higher accuracy. More feasibility is achieved by HK-Means hardware architecture which is proposed. The proposed HK-Means contains 2046 centroids storage memory and 1280 bit/cycle of bandwidth. It also contains 10 sets of processing elements to perform binary tree traversal computations. The experiments show architecture enables adjustment of cluster number from 2 to 1024 with low gate count. The proposed architecture can be used for different medical diagnosis.

REFERENCES

[1] Yu-Fang Zhang, Jia-Li Mao, et al "An efficient clustering algorithm" International conference on machine learning and Cybernetics, 2003 Volume 1, 2-5 Nov.2003 Page: 261-265 Vol.1.
[2] Usama Fayyad, Cory Reina et la "Initialization of iterative Refinement Clustering Algorithms", Microsoft research Technical report MSR-TR-98-38, June 1998.
[3] D.Guo and P. Richardson, "Automatic vessel extraction from angiogram images", IEEE Computers in Cardiology, Vol.25, pp. 441-444, 1998.
[4] P.J. Yim, P.L.Choyke et al "Gray-scale skeletonization of small vessels in magnetic resonance angiography", IEEE Trans. On Med images, Vol.19, pp 568-576, June 2000.
[5] F.Zana and J.C. Klein, "Robust segmentations of vessels from retinal angiography", in IEEE International Conference on Digital Signal Processing, Vol.2, pp 1087-1090,1997.
[6] R.T Ritchings et al "Detection of abnormalities on carotid angiograms", pattern Rec. Let vol 4, pp. 367-374, October 1986.
[7] N.Otsu "A threshold selection method from gray-scale histograms", IEEE Trans on Sys, man and Cybernatics, vol.9, pp 62-66, 1979.
[8] Rui Xu et al "Survey of clustering algorithms" IEEE Tran on neural Networks Vol 16, issue 3, pp 645-678, May 2005.