International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Image Segmentation-A Novel Doping of Feature Extraction and Pixel Level Abstraction of MRI Medical Images

Dr K. M. Mohamed Sudheer

HOD, Govt Polytechnic College, Purapuzha, Thodupuzha

Abstract: ***Segmentation and feature extraction are important factors in most applications which uses images especially medical images as data. We all know that segmentation is a pre-requisite for quantification of morphological disease manifestations and radiation treatment planning. It also plays an inevitable role in the construction of anatomical models, definitions of flight paths in virtual endoscopy. Feature extraction assists in forensic studies and biometrics, automatic face recognition and recognising people by the texture of their irises etc. This paper deals with the synthesising of medical MRI brain image so as to segment out the various tissues of the brain image to perform morphological analysis by incorporating characteristics of feature extraction and k-means segmentation method.***

Keywords: Image Segmentation, Pixel, MRI

1. Introduction

In various computer vision applications, almost all of them widely use the process of retreiving desired images from a large collection on the basis of features that can be automatically extracted from the images themselves. The algorithms used for this purpose performs three basic tasks viz, extraction, selection and classification. The extraction task transforms rich content of images into various content features. Feature extraction is the process of generating features to be used in the selection and classification of tasks. Feature selection reduces the number of features provided to the classification task. Those features which are likely to assist in discrimination are selected and used in the classification task. Features which are not selected are discarded. Among the above mentioned task feature extraction is the most critical because the particular features which are made available for discrimination directly influence the efficacy of the classification task. The net result of the extraction task is a set of features, commonly called a feature vector, which constitutes a representation of the image. In order to identify and extract the feature of an image, first thing we have to do is to study the texture of the image. Texture is actually a nebulous concept, often attributed to human perception as either the feel or the appearance of fabric. i. e. representation of structure and detail of objects in art etc. Physically it can be defined as a database of images that researchers use to test their algorithms. Essentially there is no unique definition for texture and no unique mathematical model to synthesise texture. Images will contain samples of more than one texture. Accordingly we would like to be able to describe texture and then to classify it and then perhaps to segment an image according to its texture content. The purpose of texture description is to derive some measurements that can be used to classify a particular texture. There are invariance requirements on the measurements as there were for shape description. Actually the invariance requirements for feature extraction, namely invariance to position, scale, and rotation can be easily applied to texture extraction. Unlike feature extraction, which depends on edge extraction, texture description rarely depends on edge extraction. In order to segment an image according to it's texture, we can measure the texture in a chosen region and then classify it [1]. Most commonly used segmentation algorithm fortexture based segmentation is the k-nearest neighbour classifier. Naturally this is a computation demanding process. Another method is to simply classify regions as opposed to pixels. This is a tiled approach. Another approach is the uniform thresholding method which is equivalent to pixel segmentation based on brightness alone. But the result of thresholding depends on illumination level and on appropriate choice of the thresholdvalue.

The texture segmentation method is completely automatic and the measures are known to have invariance properties to illumination, as well as other factors. Also, in uniform thresholding there is no extension possible to separate more classes (except perhaps to threshold at differing brightness levels). A common approach to structural segmentation is the use of atlas registration-based techniques. In contrast to atlas-based approaches, there are low-level techniques in which the tissue content of each individual voxel in the image is identified. In this paper different imaging techniques used for brain segmentation has been explained. The different methods are threshold based segmentation, statistical method for brain segmentation and region growing methods. In the category of threshold-based segmentation was proposed the use of: iterative thresholding, histogram analysis and morphological operations. In the statistical method category, some statistical classifications combined with different image processing techniques were used in order to segment the MRI images [2]. The region growing techniques applied to MRI images represents the final category. The segmentation problems that are widely encountered by the researchers are the noise introduced with the acquisition of the image, the overlapping intensities (different brain structures have different tissue characteristics which results in various signal intensities and these intensities could overlap), the partial volume effect (when a pixel represents more than one kind of tissue type) and also some anatomical changes from one person to another. The blood vessels could also influence image that is taken and could introduce some noise. So, there are a lot of problems that researchers have to deal with in order to build an accurate segmentation system

2. Feature Extraction

The feature is defined as a function of one or more measurements, each of which specifies some quantifiable property of an object, and is computed such that it quantifies some significant characteristics of the object [3]. We can classify the various features currently employed as follows:

1) General features: Application independent features such as color, texture, and shape. According to the abstraction level, they can be further divided into:
   a) Pixel-level features: Features calculated at each pixel, e. g. color, location.
   b) Local features: Features calculated over the results of subdivision of the image band on image segmentation or edge detection.
   c) Global features: Features calculated over the entire image or just regular sub-area of an image.
   d) Domain-specific features: Application dependent features such as human faces, fingerprints, and conceptual features. These features are often a synthesis of low-level features for a specific domain.
2) Domain-specific features: Application dependent features such as human faces, fingerprints, and conceptual features. These features are often a synthesis of low-level features for a specific domain.

On the other hand, all features can be coarsely classified into low-level features and high-level features. Low-level features can be extracted directed from the original images, whereas high-level feature extraction must be based on low level features [4].

2.1 Colour

The colour feature is one of the most widely used visual features in image retrieval. Images characterized by color features have many advantages:

- **Robustness:** The colour histogram is invariant to rotation of the image on the view axis, and changes in small steps when rotated otherwise or scaled [5]. It is also insensitive to changes in image and histogram resolution and occlusion.
- **Effectiveness:** There is high percentage of relevance between the query image and the extracted matching images.
- Implementation simplicity. The construction of the color histogram is a straightforward process, including scanning the image, assigning color values to the resolution of the histogram, and building the histogram using color components as indices.
- **Computational simplicity:** The histogram computation has O (X, Y) complexity for images of size X × Y. The complexity for a single image match is linear; O (n), where n represents the number of different colors, or resolution of the histogram.
- **Low storage requirements:** The color histogram size is significantly smaller than the image itself, assuming color quantisation. Color is perceived by humans as a combination of three color stimuli: Red, Green, Blue, which forms a color space (Fig.1). This model has both a physiological foundation and a hardware related one. RGB colors are called primary colors and are additive. By varying their combinations, other colors can be obtained. The representation of the HSV space (Fig.2) is derived from the RGB space cube, with the main diagonal of the RGB model, as the vertical axis in HSV. As saturation varies form 0.0 to 1.0, the colors vary from unsaturated (gray) to saturated (no white component). Hue ranges from 0 to 360 degrees, with variation beginning with red, going through yellow, green, cyan, blue and magenta and back to red. These color spaces are intuitively corresponding to the RGB model from which they can be derived through linear or non-linear transformations.

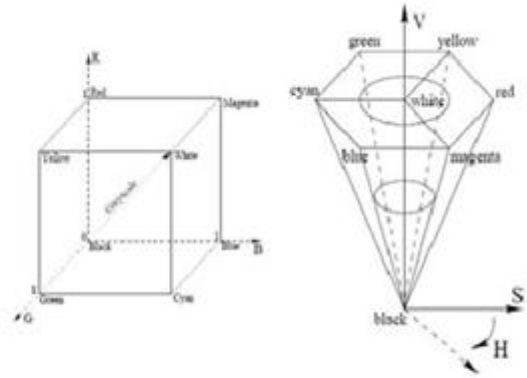


Figure 1: The RGB Color Space and HSV Color Space

Figure 2: Original Image

2.2 Texture

Texture is another important property of images. Texture is a powerful regional descriptor that helps in the retrieval process. Texture, on its own does not have the capability of finding similar images, but it can be used to classify textured images from non-textured ones and then be combined with another visual attribute like color to make the retrieval more effective.

Texture has been one of the most important characteristic which has been used to classify and recognize objects and have been used in finding similarities between images in multimedia databases. Basically, texture representation methods can be classified into two categories: structural; and statistical. Statistical methods, including Fourier power spectra, co-occurrence matrices, shift-invariant principal component analysis (SPCA), Tamura features, World decomposition, Markov random field, fractal model, and multi-resolution filtering techniques such as Gabor and wavelet transform, characterize texture by the statistical distribution of the image intensity.

1) Image (Pre) Processing for Feature Extraction

Pre-processing does not increase the image information content. It is useful on a variety of situations where it helps to suppress information that is not relevant to the specific image processing or analysis task (i. e. background subtraction). The aim of preprocessing is to improve image data so that it suppresses undesired distortions and/or it enhances image features that are relevant for further processing

The different types of pre-processing are (i) Enhancement (contrast enhancement for Contour detection) (ii) Restoration (aim to suppress degradation using knowledge about its nature; i. e. relative motion of camera and object, wrong lens focus etc.) (iii) Compression (searching for ways to eliminate redundant information from images)

2) Magnetic Resonance Imaging

Magnetic resonance (MR) images are acquired digitally, meaning that by virtue of the imaging process the area imaged, in this case the brain, is segmented into a uniform three-dimensional (3-D) array of volume imaging elements, or voxels. MR images can be acquired either in a 3-D mode or slice-by-slice, but in either event the images are displayed in a slice-by-slice fashion. Once acquired, the images are stored on a permanent archival medium, such as optic disk or magnetic tape.

3) Medical Image Segmentation

Segmentation of anatomical structures from medical images has very important applications in diagnosis, surgical planning, navigation, and medical image analysis. As already mentioned in the introduction a multitude of segmentation algorithms have been proposed to tackle these types of problems. viz edge-based, region-based, or a combination of these two. The edge-based methods (e. g. [6-11]) rely on the information of the edges, such as high magnitude of image gradient. The region-based methods (e. g. [12–15]) make use of homogeneity on the statistics of the regions being segmented. The algorithm developed in [16] integrates gradient and region information within a deformable boundary finding framework. The Geodesic Active Region models proposed in [17–18] integrate the edge and region-based segmentation methods into a variational approach.

Segmentation is the separation of structures of interest from the background and each other. A segmented image is the highest domain-independent or data-driven abstraction of an image. It is the input to a higher vision system that utilizes domain-specific information to further analyze and interpret the image. Most of the vision systems often progress from segmentation, to feature extraction from the regions, to object recognition. This approach may be called an open-loop approach. Also the segmentation and feature extraction methods may be complex and elaborate. To achieve good results the different steps of the object recognition process must interact. Knowledge of the object can aid in the segmentation process.

4) MRI Brain Stripping

A technique called Stripping is used in association with segmentation. Stripping is a type of segmentation that classifies head image elements into two rigid classes: brain and non-brain. Tissue segmentation divides brain matter into three or more labeled classes.

2.3 Segmentation Applications

1) To identify/evaluate anatomical areas of interest.
2) To preprocess for image registration.
3) To Preprocess for analysis with respect to functional metrics.
4) To Preprocess for surface extraction.

2.4 Identifying Areas of Interest

The areas of interest after segmentation are identified for the purpose of diagnosis of diseases like Multiple sclerosis lesions, tumors etc. The treatment for the abnormalities are prescribed based on the monitoring of lesion volume, radiation therapy planning, by differentiating edema, necrotic or scar tissue in response to therapy

2.5 Brain Segmentation Problems

The different problems that we come across while during segmentation are

a) Optimal selection of features for segmentation
b) Operator supervision of segmentation
c) Accurate segmentation over full field of view
d) Verification of results

2.6 Brain Stripping Steps

- Input image
- Preprocessing
- Feature extraction
- Segmentation
- Classification and description

2.7 Brain Segmentation Procedure

A dicom image (MRI) format of size 256 X 256 is used as the input. Then it is interpolated to give 3-dimensional cubic voxels i. e. orientation of the voxels is calculated. In the x and y axes the inclination of the different slices are found out and they are manipulated to get the proper orientation. The dimensions of the multidimensional array are rearranged by permute–order method. The order is found by trial and error method.

2.8 Thresholding and Segmentation

The choice of the selection of k different initial representatives is given to the user. Here the value of k is 3. These are the cluster centers or centroids. Each pixel will be placed in the cluster whose similarity measure is greater than or equal to a threshold. Repeat the above step until there is no change in clusters. Then apply some thresholding rules (simple global thresholding) to eliminate certain parts by using a threshold T. By selecting an appropriate threshold (40, 60 & 100) here a clean segmented result has achieved by eliminating the shadows leaving only the objects.

2.9 Erosion and Dilation Operations

Dilation and erosion are two fundamental morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

2.10 K Means Clustering Algorithm

This non hierarchical method initially takes the number of components of the population equal to the final required number of clusters. K-Means Clustering and Fuzzy Clustering are different than Hierarchical Clustering and Diversity Selection in that the number of clusters, K, needs to be determined at the onset [19]. In the first procedure, the objects are randomly assigned to one of the K clusters. Once this is done, the positions of the K centroids are determined, as is the value of the metric to minimize. A global optimization method is then used to reassign some of the objects to different clusters and their texture behaviour. There are several variants of the k-means clustering algorithm, but most variants involve an iterative scheme that operates over a fixed number of clusters, while attempting to satisfy the following properties:

1) Each class has a center which is the mean position of all the samples in that class.
2) Each sample is in the class whose center it is closest to.

The basic k-means algorithm consists of the following steps:

Initialization
Loop until termination condition is met:

1) For each pixel in the image, assign that pixel to a class such that the distance from this pixel to the center of that class is minimized.
2) For each class, recalculate the means of the class based on the pixels that belong to that class.

end loop;

Traditionally these steps are done iteratively

(i) Proposed Algorithms
The following is regular K-Means:
1) Select K data points as the initial representatives
2) For I =1 to N, assign item Xi to the most similar centroid (this gives K clusters).
3) For J = 1 to K, recalculate the cluster centroid Cj.
4) Repeat steps 2 and 3 until there is little or no change in clusters.

The expanding version of K-Means is as follows:
1) Select K data points as initial representatives.
2) For I = 1 to N, assign item Xi to the most similar centroid (this gives K clusters).
3) For J = 1 to K, calculate the mean and standard deviation of similarity measure between every data item in cluster j and cluster centroid Cj.
4) For I = 1 to N, if the similarity measure between item Xi and it's cluster's centroid has a z-score < threshold, place item Xi in cluster K+1.
5) If step 4 was applied to any data item Xi, then reassign K=K+1.
6) For J = 1 to K, recalculate the cluster centroid Cj.
7) Repeat steps 2 to 6 until there is no change in clusters.

2.11 Parameters and options for the k-means algorithm

2.11.1 Number of classes
Number of classes is usually given as an input variable. We have to chose a size that a human can visualize easily (more than 16 and the image starts to look cluttered). There are sometimes good reasons to use a specific number of classes. If we want to classify a scene into vegetation, soil and water, in which case 3 or 4 should suffice.

2.11.2 Initialization of K-means
K-means results are highly dependent on the initialization procedure used.

There are a number of different ways to initialize the algorithm [5]:

1) Arbitrarily assign classes to pixels. The straightforward way to do this is to assign the i^{th} pixel to the i modulo k^{th} class. This is a good approach when the number of channels is greater than the number of classes. We may need a large number of bits of precision just for the first few iterations-- to ensure that there is some distance between the different means. For this reason, it works better in software than hardware.
2) Distribute the mean table around the color space
This is a good approach if the number of channels is less than the number of classes. Otherwise it is difficult to distribute the means.
3) Initialize mean values with random pixels from the image.

To initialize k classes, choose k pixels at random from the image. (Note-they don't need to be that random.) Make sure that the pair wise distance between the k distance is large enough. How to ensure that 2 pixels are sufficiently far away from each other? One (compute intensive) way to do this is to choose p pixels at random from the image (where p >> k but smaller than all the pixels in the image) and then do k means clustering on those p pixels. (But how do you initialize the sub-problem? Arbitrarily assigning pixels to classes should work for this.)

2.11.3 Distance Measure

The meat of the k-means algorithm is calculating the distance between each pixel and each class center. There are different distance measures that can be used. The most common are:

L1 distance (Manhattan distance): The absolute value of the component wise difference between the pixel and the class. This is the simplest distance to calculate and may be more robust to outliers.

L2 distance (Euclidean distance): The square root of the component wise square of the difference between the pixel and the class. Since we are only comparing the results, you can omit the square root. Computing the L2 distance requires squaring the data, which introduces extra bits of precision into the data. The squaring operation is expensive in hardware. One advantage of this metric is that the distance is a sphere around the centroid

2.11.4 Termination of K-means algorithm

Theoretically, k-means should terminate when no more pixels are changing classes. There are proofs of termination for k-means. These rely on the fact that both steps of k-means (assign pixels to nearest centers, move centers to cluster centroids) reduce variance. So eventually, there is no move to make that will continue to reduce the variance. Running to completion (no pixels changing classes) may require a large number of iterations. In software, we typically terminate when one of the following criteria is met:

1) Terminate after fewer than n pixels change classes (we use n=1000 for 256x256 pixel images)
2) Terminate after J iterations. We use 50.

2.11.5 How to measure quality of a classification

One issue is how to measure the quality of the results provided by k-means classification. We use some internal measures:

A good classification has:
1) Low within class variance-- compactness
2) High distance between class centers-- isolation.

2.11.6 Ways to extract parallelism from k-means algorithm:

Assume you have multiple processing elements (PEs), each with local memory, and an arbitrary interconnection network for communication between PEs. There are several ways to parallelize the k-means algorithm:

1) Divide the image among processing elements. The pixel iteration can be done on any piece of an image without any communication with other PEs. The image can be divided in any manner. Spatial locality is not required. Communication is required after all pixels have been classified, in order to recalculate means.
2) Divide the mean table among PEs. If you have m+1 PEs, give the first m PEs 1/m of the mean table. Each PE communicates the class that x_i is closest to, of the class means it has, to the last PE, which determines the class this pixel gets assigned to. This requires more communication than scheme 1. Communication among PEs is required once per pixel, not once per pixel classification loop.
3) Divide the channels among PEs. Each PE determines the distance from the spectra that it has. This is combined with the distances calculated from other PEs to determine the closest class. Note that more than the minimum distance might be required to complete this calculation. This requires the most communication, and should be avoided. It may be needed for hyperspectral data.

It is noticeable that all these forms of parallelism can be combined. The first is the preferred mechanism since it minimizes communication and synchronization requirements among PEs.

2.11.7 Distance Measure

The meat of the k-means algorithm is calculating the distance between each pixel and each class center. There are different distance measures that can be used. The most common are:

1) L1 distance (Manhattan distance): The absolute value of the component wise difference between the pixel and the class. This is the simplest distance to calculate and may be more robust to outliers.
2) L2 distance (Euclidean distance): The square root of the component wise square of the difference between the pixel and the class. Since we are only comparing the results, you can omit the square root. Computing the L2 distance requires squaring the data, which introduces extra bits of precision into the data. The squaring operation is expensive in hardware. One advantage of this metric is that the distance is a sphere around the centroid.

2.11.8 Dealing with dead classes in the K-Means Algorithm

Frequently, some classes become dead during the course of running k-means. A class is "dead" if no pixels belong to it. There are a number of ways to deal with dead classes:

1) Mark a class as dead and never assign pixels to it in the future.
2) Keep the mean of the class the same as it was before the iteration where no pixels were assigned to this class, in the hope that pixels will be assigned to this class in the future.
3) Reassign the center of the class to a random value.

2.11.9 Variants on the basic K-means Algorithm

1) On-the-fly k-means: This involves incrementally updating the means. After each pixel classification, update the means of the class the pixel came from, and the class the pixel is moving to. This is very expensive in hardware, since it requires 2 mean calculations for every pixel classification.
2) 1- pass k-means (the leader algorithm): The basic k-means algorithm starts with the number of classes as a given, and iterates several times over image data. The 1-pass algorithm assigns each pixel to a class based on one decision. There is only one iteration through the data.

Instead of starting with the number of classes, the 1-pass k-means algorithm starts with the maximum radius of a cluster, epsilon.

Cluster center 1 is set to the values of pixel 1.

For each pixel in the image:

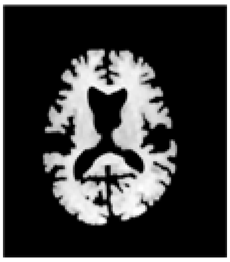
If this pixel is within epsilon of the centers of any existing class, then add that pixel to that class, else create a new class whose center is the values of this pixel.

Note that, worst case, each pixel is a class, and the number of comparisons for the last pixel is equal to the number of pixels in the image. This worst case is unlikely to happen in practice. The one pass algorithm is good for running on a host or for an implementation where k-means is a first step.

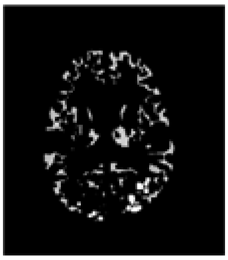
3. Test Results on the Image Img00025 on Various Methods

(i) K-Means Clustering Algorithm

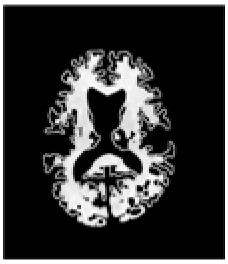
(a) Input image



(b) Brain Segmented



(c) Gray matter segmented



(d) White matter segmented

Figure 3: Segmentation of MRI Brain image using K-Means algorithm

(ii) Adaptive Region Growing Method

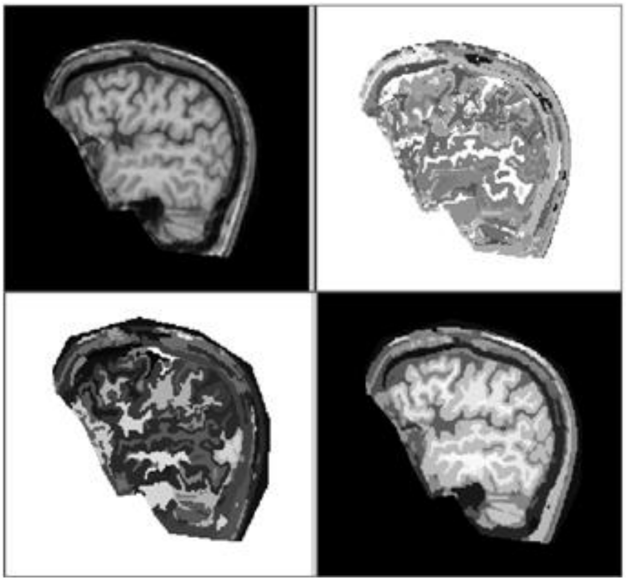
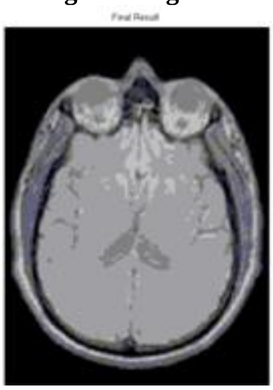


Figure 4: Segmentation of MRI Brain image using adaptive region growing

Segmentation using adaptive region growing method: (a) original image (b) result after automatic segmentation (c) labelling result after reclassification (d) regions are coded with their gray value.

(iii) Segmentation using Sobel

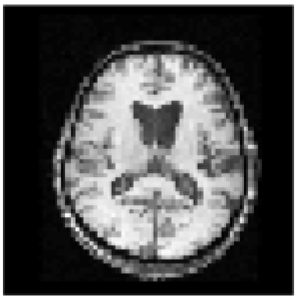
(iv) Segmentation using building detection samples



(v) Fuzzy-C-means method

Paper ID: SR23523122137 DOI: 10.21275/SR23523122137

(i) Background Segmented



(ii) Brain Segmented

4. Conclusion

Segmentation is a classical problem in computer vision. Different methods of segmentation have been discussed and their test results in same slice of dicom image are being compared. The test result is being compared with the result produced by other methods like adaptive region growing method which works by learning the homogenity criterion from the characteristics of the region to be segmented, sobel method, building detection method, fuzzy c-means method etc. The result shows that the method which uses the fusion of feature extraction and k-means clustering algorithm is giving better result than the other algorithm.

References

[1] Feature Extraction and Image Processing by Mark S. Nixon and Alberto S. Aguado.
[2] Multiscale Segmentation of Three-Dimensional MR Brain Images w. j. niessen, k. l. vincken, j. weickert, b. m. ter haar romeny and m. a. viergever.
[3] Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems by Ryszard S . Chora ́s-International Journal of Biology and Medical Engineering.
[4] E. Saber, A. M. Tekalp, ”Integration of color, edge and texture features for automatic region based image annotation and retrieval, ” Electronic Imaging, 7, pp.684–700, 1998.
[5] M. K. Hu, ”Visual pattern recognition by moment invariants, ” IRE Trans. on Information Theory, 8, pp.179–187, 1962.
[6] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models, ” International Journal of Computer Vision, vol.1, no.4, pp.321–331, 1988.
[7] V. Caselles, F. Catt ́, T. Coll, and F. Dibos, “A geometric model for active contours in image processing, ” Numerische Mathematik, vol.66, pp.1–31, 1993.
[8] R. Malladi, J. Sethian, and B. Vemuri, “Shape modeling with front propagation: A level set approach, ” IEEE Trans. Pattern Anal. Machine Intell., vol.17, pp.158–175, 1995.
[9] V. Caselles, R. Kimmel., and G. Sapiro, “On geodesic active contours, ” Intel. Journal of Computer Vision, vol.22, no.1, pp.61–79, 1997.
[10] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. J. Yezzi, “Gradient flows and geometric active contour models, ” Proc. ICCV’95, pp.810–815, 1995.
[11] A. Yezzi, S. Kichenassamy, A. Kumar, P. J. Olver, and A. Tannenbaum, “A geometric snake model for segmentation of medical imagery, ” IEEE Trans. Medical Imaging, vol.16, pp.199–209, 1997.
[12] S. C. Zhu and A. Yuille, “Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, ” IEEE PAMI, vol.18, pp.884–90, 1996.
[13] D. Mumford and J. Shah, “Optimal approximation by piecewise smooth functions and associated variational problems, ” Comm. Pure Appl. Math., vol.42, pp.557–685, 1989.
[14] D. Cremers, F. Tischhuser, J. Weickert, and C. Schnrr, “Diffusion-snakes: Introducing statistical shape knowledge into the Mumford-Shah functional, ” International Journal of Computer Vision, vol.50, no.3, pp.295–315, 2002.
[15] A. Tsai, A. Yezzi Jr.,, and A. S. Willsky, “Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification, ” IEEE Transactions on Image Processing, vol.10, no.8, pp.1169 –1186, 2001.
[16] A. Chakraborty, H. Staib, and J. Duncan, “Deformable boundary finding in medical images by integrating gradient and region information, ” IEEE Transactions on Medical Imaging, vol.15, no.6, pp.859–870, 1996.
[17] N. Paragios and R. Deriche, “Geodesic active regions for supervised texture segmentation, ” ICCV, Cofu, Greece, pp.926–932, 1999.
[18] N. Paragios, “Geodesic active regions and level set methods: contributions and applications in artificial vision, ” Ph. D. thesis, School of Computer Engineering, University of Nice/Sophia Antipolis, 2000.
[19] K-Means clustering by Brian T. Luke