CHAPTER 5

WAVELET BASED CLUSTERING ALGORITHMS

5.1 INTRODUCTION

Clustering is the organisation of a data set into homogeneous and/or well separated groups with respect to a distance or, equivalently, similarity measure. Its objective is to assign to the same cluster data that are more close to each other than they are in different clusters. Clustering methods fall into three types: Partitional, hierarchical and probabilistic approaches (Jain et al 1999). Partitional clustering algorithms use a single pass for clustering. Hierarchical approaches use many passes to obtain the clustering results. Probabilistic clustering is another way of dealing with clustering problems by a model-based approach (Bock 1996), which consists in using certain models for clusters and attempting to optimize the fit between the data and the model. Variants of K-clustering, such as K-means, ISODATA (Ball and Hall 1965; Takahashi and Abe 1999), and FCM are the partitional clustering methods that are most widely used (Gath and Geva 1989). Partitional clustering is computationally attractive, which makes it applicable for large data sets, but it is very sensitive to small clusters and outliers, i.e. noise or mixed pixels (Halkidi and Vazirgiannis 2002). Agglomerative Hierarchical Clustering (AHC) works well with small data sets and can handle outliers very well but its computation is very expensive and therefore it is not feasible for a large data set. Moreover, it also has a ‘chaining’ problem for a complex data set (Brereton 1992). In this work, a new clustering algorithm is designed to take advantage of the characteristics of these three clustering methods and
eliminate their potential limitations which can work well with a complex and large dataset, including small objects and outliers.

The first stage of image segmentation usually involves the development of a feature space. This comprises of calculating the values of several features for each pixel in the image. Each feature should in some way describe the appearance of the local area surrounding the pixel. These features make up a feature vector for each pixel so that each is represented by a point in a multidimensional feature space. Consequently, if the features used for the image are good descriptors, similar appearing regions in the image will contain pixels whose feature vectors occupy similar positions in the feature space. These are known as clusters and the purpose of clustering is to identify these clusters and classify the image’s pixel accordingly.

The quality and the accuracy of the segmentation ultimately depend on the type of features used. Therefore it is very important that the features should suitably characterize the aspects of the image on which the segmentation is based. For example, images consisting of a number of highly textured regions are well segmented using frequency based features, whereas images made up of smoother regions can more easily be segmented using local gray level mean and variance as features. Many real images are made up of both types of region and thus require different features to be used in different areas of the image. As a result, existing segmentation algorithms have failed to produce meaningful segmentations of many real images which are rich in color and texture. The proposed algorithm involves the use of wavelet analysis to offer effective segmentation results.

5.2 WAVELET ANALYSIS OF IMAGES

The 2D wavelet transform is a very popular tool in image processing. Its ability to repeatedly decompose an image in the low frequency channels makes it ideal for image analysis since the lower frequencies tend to
dominate real images. In this correspondence, the wavelet transform is used both to analyze the image prior to segmentation enabling feature selection as well as to provide spatial frequency based descriptors as features for segmenting textures.

It is expected that smooth images in which there are only gradual variations in gray level are to be dominated by low spatial frequencies, whereas textured images in which the gray level varies rapidly should be made up of a wide range of frequencies. Smooth and textured images can thus easily be distinguished from each other by examining their wavelet transforms. In order to extract useful information from this and hence discriminate between smooth and textured images, it is useful to group the wavelet coefficients into channels representing the various frequency/scale bands.

Discrete wavelet analysis is carried out using the concept of filter banks. Filters of different cutoff frequencies analyze the image at different scales. Resolution is changed by filtering, while the scale is changed by up sampling and down sampling. Once the filtered images have been obtained, the features are computed from local characteristics of the filtered image. In this work, four wavelet transformation techniques namely, Stationary Wavelet Transform (SWT), Intermediate features of Maximally Overlap Wavelet Transform (IMOWT), Dual Tree Complex Wavelet Transform (DTCWT) and M Band Wavelet Transform (MBWT) are explored. It is found that the energy values of the various channels of the 2D wavelet transform varied considerably between different textures, depending on their dominant orientation and spatial frequencies. The energy values are therefore used as features for segmentation in the textured areas of the image. Essentially the textures in the image are described by a series of features derived from their wavelet transforms at different levels of resolution. This leads to a classical
problem in pattern recognition that of choosing appropriate wavelet derived features to use when performing clustering analysis.

5.3 SUPERVISED CLASSIFICATION

The supervised classification is the essential tool used for extracting quantitative information from any image data. In supervised classification, the image analyst “supervises” the pixel categorization process by specifying to the computer algorithm. To do this, representative sample sites of known cover type, called training areas, are used to compile a numerical “interpretation key” that describes the spectral attributes for each feature type of interest. Each pixel in the data set is then compared numerically to each category in the interpretation key and labeled with the name of the category it “looks most like”. Supervised classification is much more accurate for mapping classes, but depends heavily on the skills of the image specialists (Kaya et al 2002). The steps involved in supervised classification algorithm (Jain et al 2000) can be grasped from Figure 5.1.

![Figure 5.1 Supervised Classification](image-url)
5.4 UNSUPERVISED CLASSIFICATION

Unsupervised classification does not require human to have the a priori knowledge of the classes and mainly using some clustering algorithm to classify an image data. The main input, a typical unsupervised clustering algorithm takes is the number of classes it should find. Unsupervised classifiers do not utilize training data as the basis for classification (Henry Selvaraj et al 2006). Like supervised classifiers, the unsupervised procedures are applied in two separate steps. The fundamental difference between these techniques is that supervised classification involves a training step followed by a classification step. In the unsupervised approach the image data are first classified by aggregating them into the natural spectral groupings or clusters, present in the scene (Lu and Weng 2007). The basic premise is that values within a given cover type should be close together in the measurement space, whereas data in different classes should be comparatively well separated. Then the image analyst determines the land cover identity of these spectral groups by comparing the classified image data to ground reference data.

The classes that result from unsupervised classification are spectral classes. Because they are based solely on the natural groupings in the image values, the identity of the spectral classes will not be initially known. The analyst must compare the classified data with some form of reference data (such as larger scale imagery or maps) to determine the identity and information value of the spectral classes. Thus, in supervised approach useful information is defined and their spectral separability is examined; but in the unsupervised approach spectrally separable classes are determined and their informational utility is defined. The steps involved in unsupervised classification algorithm are depicted in Figure 5.2.
5.4.1 Partitional Clustering

Partitional clustering directly divides data objects into some pre specified number of clusters. It is further classified into exclusive clustering and overlapping clustering.

**Exclusive Clustering:** In this clustering, data are grouped in an exclusive way (Celenk 1990), so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. An example for exclusive clustering algorithm is K-means clustering.
K-means is one of the simplest unsupervised clustering algorithms, in which the number of clusters is fixed a priori. The main idea is to define \( k \) centroids, one for each cluster. These centroids should be placed in a suitable way because 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 grouping is done. At this point \( k \) new centroids of the clusters resulting from the previous step are re calculated. Then, a new binding has to be done between the same data set points and the nearest new centroid. Thereby a loop has been generated. As a result of this loop it is noticed 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 is defined as,

\[
J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_{i}^{(j)} - c_{j} \|^2
\]

(5.1)

where \( \| x_{i}^{(j)} - c_{j} \|^2 \) is a chosen distance measure between a data point \( x_{i}^{(j)} \) and the cluster center \( c_{j} \), is an indicator of the distance of the \( n \) data points from their respective cluster centers. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The K-means algorithm can be run multiple times to reduce this effect. The algorithm is summarized in Figure 5.3.
Figure 5.3 K-means Clustering

**Overlapping Clustering:** In this overlapping clustering, fuzzy sets are used to cluster data, so that each point may belong to two or more clusters with different degrees of membership (Tapas Kanungo et al 2002). In this case, data will be associated to an appropriate membership value.

FCM is a method of overlapping clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. It is based on minimization of the objective function as given by,

\[
J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - c_j \|^2, 1 \leq m < \infty
\]  

(5.2)
where $m$ is any real number greater than 1, $u_{ij}$ is the degree of membership of $x_i$ in the cluster $j$, $x_i$ is the $i^{th}$ cluster of $d$-dimensional measured data, $c_j$ is the $d$-dimension center of the cluster, and $\| \cdot \|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown in Equation (5.2), with the update of membership $u_{ij}$ and the cluster centers $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{m-1}}$$  \hspace{1cm} (5.3)

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$  \hspace{1cm} (5.4)

This iteration will stop when $\max_\{ij\} \left( \left| u_{ij}^{k+1} - u_{ij}^k \right| \right) < \delta$, where $\delta$ is a termination criterion between 0 and 1, and $k$ is the number of iteration steps. This procedure converges to a local minimum or a saddle point of $J_m$.

### 5.4.2 Hierarchical Clustering

Data clustering algorithms can also be hierarchical (Martin et al 2004). Hierarchical algorithms find successive clusters using previously established clusters, whereas partitional algorithms determine all clusters at once. Hierarchical algorithms can be agglomerative (bottom-up) or Divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them in successively larger clusters. Divisive algorithms
begin with the whole set and proceed to divide it into successively smaller clusters.

A key step in hierarchical clustering is to select a distance measure. A simple measure is Manhattan distance, equal to the sum of absolute distances for each variable. A more common measure is Euclidean distance, computed by finding the square of the distance between each variable, summing the squares, and finding the square root of that sum.

Given a distance measure, elements can be combined. Hierarchical clustering builds (agglomerative), or breaks up (divisive), a hierarchy of clusters. The traditional representation of this hierarchy is a tree data structure called a dendrogram, with individual elements at one end and a single cluster with every element at the other. Agglomerative algorithms begin at the top of the tree, whereas divisive algorithms begin at the bottom.

5.4.3 Probabilistic Clustering

There is another way to deal with clustering problems: a model-based approach, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model (Fan and Xia 2003). In practice, each cluster can be mathematically represented by a parametric distribution, like a Gaussian (continuous) or a Poisson (discrete). The entire data set is therefore modeled by a mixture of these distributions. An individual distribution used to model a specific cluster is often referred to as a component distribution.
5.4.3.1 Gaussian Mixture Model (GMM)

The GMM is a type of density model which comprises of a number of component functions (Dempster et al 1977). These component functions are combined to provide a multimodel density. They can be employed to model the colors of an object in order to perform tasks such as real time color based tracking and segmentation (Mckenna et al 1999). The mixture model is typically an incomplete data structure model (Constantinopoulos et al 2006).

Given a set of all observed data points \( X = \{x_1, x_2, \ldots, x_N\} \), where \( x_i \) is a \( d \)-dimensional vector measurement. Assume that the points are generated in an independent and identically distributed (IID) from an underlying density. In the multivariate Gaussian mixture model, data \( X \) is assumed to be a sample from a probability distribution with density:

\[
p(x_j | \Theta) = \sum_{k=1}^{K} \alpha_k G(x_j | \theta_k)
\]

where \( \alpha_k \) is the mixture weight (corresponds to the prior probability) of component \( k \) and must satisfy

\[
\alpha_k \geq 0, \quad k = 1, 2, \ldots, K, \quad \sum_{k=1}^{K} \alpha_k = 1,
\]

and where

\[
\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}
\]

where each component \( k \) is a multivariate (\( d \)-dimensional) Gaussian density.
\[ G(x_i \mid \theta_k) = \frac{1}{(2\pi)^{d/2} |\varepsilon_k|^{d/2}} \exp \left( -\frac{1}{2} (x_i - \mu_k)^T \varepsilon_k^{-1} (x_i - \mu_k) \right) \] (5.8)

with its own parameters \( \theta_k = (\mu_k, \varepsilon_k) \), \( \mu_k \in \mathbb{R}^d \) is the mean, \( \varepsilon_k \in \mathbb{R}^{d\times d} \) is the covariance matrix.

The likelihood of a set of \( N \) samples \( X = \{x_i, \ldots, x_N\} \) is defined as follows:

\[
L(X \mid \Theta) = L(x(1), \ldots, x_i, \ldots, x(N) \mid \Theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \alpha_k G(x_i \mid \theta_k) \quad (5.9)
\]

the likelihood is thought of as a function of the parameters \( \Theta \) where the data \( X \) is fixed. In the maximum likelihood problem, our goal is to find \( \Theta \) that maximizes \( L \). That is, to find \( \hat{\Theta} \) where

\[
\hat{\Theta} = \arg \max_{\Theta} L(X \mid \Theta) \quad (5.10)
\]

Often \( \log[L(X \mid \Theta)] \) is maximized because it is analytically easier. Since the logarithm is monotonically increasing, maximizing the log-likelihood is equivalent to maximizing the likelihood. In order to estimate \( \Theta \), it is typical to introduce the log-likelihood, defined as,

\[
\log[L(X \mid \Theta)] = \sum_{i=1}^{N} \log \left[ \sum_{k=1}^{K} \alpha_k G(x_i \mid \theta_k) \right] \quad (5.11)
\]
It is well-known that the Maximum Likelihood (ML) estimate

$$\hat{\Theta}_{\text{ML}} = \arg \max_{\Theta} \log[L(X \mid \Theta)]$$ \hspace{1cm} (5.12)

cannot be found analytically, since

$$\frac{\partial}{\partial \Theta}\log[L(X \mid \Theta)]\bigg|_{\Theta = \hat{\Theta}_{\text{ML}}} = 0$$ \hspace{1cm} (5.13)

is difficult to solve, because the log-likelihood function is highly nonlinear and not easily maximized. The same is true for the Bayesian Maximum A Posteriori (MAP) criterion,

$$\hat{\Theta}_{\text{MAP}} = \arg \max_{\Theta} \{\log[L(X \mid \Theta)] + \log[P(\Theta)]\},$$ \hspace{1cm} (5.14)

given some prior $P(\Theta)$ on the parameters.

So the usual choice for obtaining ML or MAP estimates of the mixture parameters is the Expectation Maximization (EM) algorithm. The EM is an iterative procedure which finds local maxima of $\log[L(X \mid \Theta)]$ or $\log[L(X \mid \Theta)] + \log[P(\Theta)]$. This algorithm introduces an auxiliary function $Q$ (along with some auxiliary random variables) that has the same behavior as the log-likelihood function (in that when the log-likelihood function increases, so does the auxiliary function) but is much easier to maximize.

5.4.3.2 The Expectation Maximization (EM) Algorithm

The EM algorithm is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a
given observable data set $X$ when the data is incomplete or has missing values. We assume that a complete data set exists $Z = \{X, Y\}$ and also assume (or specify) a joint density function:

$$f(z | \Theta) = p(x, y | \Theta) = f(y | x, \Theta) p(x | \Theta). \quad (5.15)$$

In the cases (e.g., missing data values in samples of a distribution), it is assumed to have a joint relationship between the missing and observed values. With this new density function, a new likelihood function can be defined as,

$$L(Z | \Theta) = L(X, Y | \Theta) = p(X, Y | \Theta) \quad (5.16)$$

It is noted that this function is in fact a random variable since the missing information is unknown, random, and presumably governed by an underlying distribution.

For the finite Gaussian mixtures, the missing part (unobservable data) is a set of $N$ labels,

$$Y = \{y_1, ..., y_i, ..., y_N\} \quad (5.17)$$

associated with the $N$ samples, indicating which component produced each sample. Each label is a binary vector.

$$\bar{y}_i = \{y_i^1, ..., y_i^k, ..., y_i^K\} \quad (5.18)$$

where $y_i^k = 1$, and $y_i^p = 0$, for $k \neq p$, means that sample $x_i$ is produced by the $k^{th}$ component (i.e., $y_i^k = 1$ if and only if $x_i$ arises from $k^{th}$ component). The
complete log-likelihood (i.e. the one from which \( \Theta \) is estimated if the complete \( Z = \{X,Y\} \) is observed) becomes

\[
\log L(Z \mid \Theta) = \log L(X,Y \mid \Theta) = \log p(X,Y \mid \Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log[\alpha_k G(x_i \mid \theta_k)].
\]

(5.19)

The EM algorithm produces a sequence of estimates \( \{\Theta^{(t)} : t = 0,1,2,...\} \) by alternatively applying two steps until some convergence criterion is met.

**The Expectation Step:** The EM algorithm first finds the expected value of the complete log-likelihood \( \log L(X,Y \mid \Theta) \) with respect to the missing data \( Y \) given the observed data \( X \) and the current parameter estimates. The Q-function is defined as,

\[
Q(\Theta, \Theta^{(t)}) = E[\log p(X,Y \mid \Theta) \mid X, \Theta^{(t)}] = \log p(X,W \mid \Theta)
\]

(5.20)

where \( W = E[Y \mid X, \Theta^{(t)}] \) and \( \Theta^{(t)} \) are the current parameters estimates used to evaluate the expectation and \( \Theta \) are the new parameters optimized to increase \( \Theta \). Note that since \( \log p(X,Y \mid \Theta) \) is linear with respect to the missing data \( Y \), it is simple to compute the conditional expectation \( W = E[Y \mid X, \Theta^{(t)}] \) and \( \log p(X,Y \mid \Theta) \).

Since the elements of \( Y \) are binary, the conditional expectation of data point \( x_i \) in cluster \( k \) (i.e, calculation of the current conditional expectation of \( y_{ik} \) given the observed data \( x_i \)) is computed as,

\[
P_{ki}^{(t)} = E[y_{ik} \mid X, \Theta^{(t)}] = P[y_{ik} = 1 \mid x_i, \Theta^{(t)}] = \frac{\alpha_{k}^{(t)} G(x_i \mid \theta_{k}^{(t)})}{\sum_{k=1}^{K} \alpha_{k}^{(t)} G(x_i \mid \theta_{k}^{(t)})}
\]

(5.21)
where the last equality is simply Bayes rule, i.e., $\alpha_i^{(t)}$ is the a priori probability that $y_i^k = 1$, while $p_i^{(t)}$ is the a posteriori probability that $y_i^k = 1$, after observing $x_i$. For each data point $x_i$ the membership weights are defined such that $\sum_{k=1}^{K} p_{ki}^{(t)} = 1$. $p_{ki}^{(t)}$ for all data points $\{x_i\}_{i=1}^{N}$ and all mixture components $k \in [1,K]$ with the current parameter values are computed. This yields a $K \times N$ matrix.

The evaluation of this expectation is called the E-step of the algorithm. Notice the meaning of the two arguments in the function $Q(\Theta, \Theta^{(t)})$. The first argument $\Theta$ corresponds to the parameters that ultimately will be optimized in an attempt to maximize the likelihood. The second argument $\Theta^{(t)}$ corresponds to the parameters used to evaluate the expectation. The second step (the M-step) of the EM algorithm is to maximize the expectation.

The Maximization Step: The M-step of the EM algorithm is to maximize the expectation that is computed in the E-step. i.e.,

$$
\Theta^{(t+1)} = \arg \max_{\Theta} Q(\Theta, \Theta^{(t)}). \tag{5.22}
$$

In other words, $\Theta^{(t+1)}$ is the value that maximizes the conditional expectation of the complete data log-likelihood given the observed variables under the previous parameter value. This expectation is usually denoted as $Q(\Theta)$. In the continuous case, it would be given by

$$
Q(\Theta) = \int_{-\infty}^{\infty} p(Y \mid X, \Theta^{(t)}) \log p(X, Y \mid \Theta) dY. \tag{5.23}
$$
Now the membership weights and the data are used to calculate new parameter values.

- The new mixture weights:
  \[
  \alpha_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} p_{ki}^{(t)}, \quad 1 \leq k \leq K.
  \tag{5.24}
  \]

- The updated means are calculated:
  \[
  \mu_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} p_{ki}^{(t)}} \sum_{i=1}^{N} p_{ki}^{(t)} x_i, \quad 1 \leq k \leq K.
  \tag{5.25}
  \]

- The updated covariance are calculated:
  \[
  \Sigma_k^{(t+1)} = \left( \frac{1}{\sum_{i=1}^{N} p_{ki}^{(t)}} \sum_{i=1}^{N} p_{ki}^{(t)} (x_i - \mu_k^{(t)}) (x_i - \mu_k^{(t)})^T \right), \quad 1 \leq k \leq K
  \tag{5.26}
  \]

The equations in the M-step need to be computed in this order, i.e., first compute the K new \( \alpha \)'s, then the K new \( \mu_k \)'s, and finally the K new \( \Sigma_k \)'s.

After the computation of all the new parameters, the M-step is complete and the membership weights in the E-step are recomputed, and hence the parameters again in the E-step, and updating the parameters is continued in this manner. Each pair of E and M steps is considered to be as iteration. Hence, the EM algorithm is an iterative scheme that guarantees a local maximum of the likelihood of the data.
**Initialization of EM Algorithm:** The EM algorithm can be started by initializing the algorithm with a set of initial parameters and then conducting an E-step. The initial parameters or weights can be chosen randomly.

- Select K, so the weights \( \alpha_1 = \cdots = \alpha_k = \cdots = \alpha_K = \frac{1}{K} \),
- Select K random data points as initial means,
- Select the covariance matrix of the whole data set for each of the initial K covariance matrices.

**Cost Function:** EM algorithm does not consider the pixel neighborhood for clustering (Roula et al. 2002). The segmentation results can be improved, by considering the neighborhood effects. For considering neighborhood effects, cost function with Mahalanobis distance is used. Mahalanobis distance is based on correlations between the variables by which different patterns can be identified and analyzed. Cost function measures the cost of giving the label to a particular pixel. The segmentation results are improved by considering the neighborhood effects. The cost function \( C \) is defined as follows:

\[
C(x_i, l) = \lambda_p Dist(x_i, \theta_l) + \lambda_n V(x_i)
\]  \hspace{1cm} (5.27)

\[
V(x_i) = \sum_{s=1}^{N_k} V(x_i, s)
\]  \hspace{1cm} (5.28)

\[
V(x_i, s) = \sum_{x_j \in \mathcal{N}} \frac{\Delta(x_{i,s}, x_{j,k})}{\mathcal{N}}
\]  \hspace{1cm} (5.29)
\[ \Delta(x_{i,s}, x_{j,k}) = \begin{cases} 
+ \frac{1}{|k-s|+1} & \text{if } l_i = l_j \\
- \frac{1}{|k-s|+1} & \text{if } l_i \neq l_j
\end{cases} \quad (5.30) \]

The first term in the right side of Equation (5.27) is the distance between a sample and a given class \( l \). In the case of normal distribution this distance is called Mahalanobis distance and is defined as,

\[ \text{Dist}(x_i, \theta_l) = (x_i - \mu_l)^T \varepsilon_i^{-1} (x_i - \mu_l) \quad (5.31) \]

Size of neighborhood,

\[ \mathcal{N} = 9 \times N_B \quad (5.32) \]

for 3×3 neighborhood.

where \( \lambda_p, \lambda_n \) - Weight values

\( \theta_l \) - Parameter set for particular cluster \( l \)

\( N_B \) - Number of Bands

\( s, k \) - Individual bands

\( \mathcal{N} \) - Spectral neighborhood

\( \mu_l \) - Mean vector for the cluster \( l \)

\( \varepsilon_i \) - Covariance matrix for the cluster \( l \)

Figure 5.4 depicts clearly the processes involved in EM clustering with cost function.
Figure 5.4 EM Clustering with Cost Function
In this work, K-means clustering, agglomerative hierarchical clustering and probabilistic clustering are combined by introducing spatial refinement concept, to take advantage of the characteristics of these three clustering methods and eliminate their potential limitations. Spatial refinement can work with a complex and large dataset, including small objects and outliers.

5.5 EM WITH COST SPATIAL REFINEMENT ALGORITHM

This algorithm consists of three steps. They are K-means algorithm (for initial clustering), agglomerative hierarchical clustering (to find optimum number of clusters) and refinement step (EM algorithm with Cost Function) for relabeling. Initially, the input image is taken in vector form for each pixel location. These vectors can have number of elements equal to the number of bands of the given image. These vectors are processed as given in Figure 5.5.

![Figure 5.5 EM with Cost Spatial Refinement Algorithm](image-url)
In the proposed segmentation algorithm, k-means clustering is used as the initial step, to obtain the segmentation results for high number of clusters. Here, high number of clusters is set to be ten. Hence, the output from the k-means clustering is the labeled image consisting of ten clusters.

**Agglomerative Hierarchical Clustering (AHC):** AHC yields a hierarchical structure of clusters, representing how cluster pairs are joined. In principle, the algorithm starts with assigning each pixel to individual clusters. At each iterative step, the proximity matrix is calculated for all cluster pairs and the two ‘closest’ pair clusters are merged. The process will continue until there is only one cluster.

Depending on the definition of a distance between clusters, AHC are variants of single linkage, complete linkage, average linkage and Ward’s (Prasad Reddy et al 2004) algorithms. In single linkage, the distance of two clusters is the distance between two nearest points. Similarly, the distance is the maximal distance between points in different clusters in complete linkage, and the average distance of points in average linkage clustering. The distance in Ward’s method is defined as the squared Euclidean distance of the cluster mean vectors. Hence, Ward’s method is related to K-means through the minimum-variance criterion. In this work, agglomerative hierarchical clustering with Ward’s distance measure is used.

A dendrogram is produced, representing nested clusters and the similarity levels at which clusters are joined. The dendrogram can be cut at several levels in order to obtain an arbitrary number of clusters. It circumvents the problem of the pre-defined number of K clusters in K-means clustering algorithms. By starting with assigning each pixel to individual clusters the algorithm is not sensitive to outliers. Outliers will be kept in separate clusters, not influencing the other clusters.
Overall, agglomerative hierarchical clustering scheme considers only clusters that are obtained in the previous step. This means that once a point has been merged to a cluster, it cannot be considered for joining another cluster in later iterations. This rule is not optimal for complex data sets where cluster homogeneity levels are low or not uniform.

**Number of clusters:** Determining the number of clusters is a difficult problem in all clustering algorithms. Milligan and Cooper (1985) developed some measures of spread within and between clusters. The within cluster inertia, $W$, is defined as variation of individual points to their center and the between cluster inertia, $B$, is defined as the variation of cluster centers around the overall mean.

\[
W = \frac{1}{N} \sum_k \sum_{i \in c_k} d(x_i, c_k) \tag{5.33}
\]

\[
B = \frac{1}{N} \sum_k n_k d(c_k, c) \tag{5.34}
\]

In clustering algorithms, minimising the sum of squares criterion would thus minimise $W$. By keeping track of the within cluster inertia for a varying number of clusters, one can often observe a sharp increase at a certain level. Just before this increase, the spread of the clusters is minimal.

The number of optimum clusters is found out by, the ratio of within to between cluster inertia and it is given by,

\[
I = \frac{W}{B} \tag{5.35}
\]
For each merging stage, this index value is found out. A graph is drawn between these index values and the number of clusters, from which the optimum number of clusters for the input image is obtained by the position of sudden change in the graph.

**Spatial Refinement:** Spatial refinement is accomplished using EM algorithm with cost function on the initially clustered output with optimum number of clusters derived from agglomerative clustering. The distance of each pixel with the centroids is calculated and the pixels are reassigned to the ‘closest’ adjacent clusters. This refinement process iterates until there is no change in the pixel classification and reallocates misassigned points using the information of points in the spatial domain.

Spatial refinement is designed to use a combination of K-means clustering, agglomerative hierarchical clustering and probabilistic clustering to take advantage of the characteristics of three clustering methods and eliminate their potential limitations by introducing a refinement process using spatial information. In order to prevent the (expensive) application of AHC to a large data set, spatial refinement is first pre processed by K-means with a high number of classes, $M$. When $M$ is high enough, clusters can be considered as highly homogenous classes. These form the input to the agglomerative hierarchical process. The number of classes $M$ is much smaller than the total number of points $N$, typically in the order of 100. Determining the number of clusters in a data set by using the index function $I$ is very time consuming with K-means clustering, where the algorithm has to be run for each number of clusters $K$. On the other hand, it is much easier for agglomerative hierarchical clustering, where the index function at each merge level is calculated and is used in spatial refinement. For each level in the dendrogram, the clustering index $I$ is calculated and the ‘best’ choice of $K$ number of clusters thus is identified by observing the sharp change at the level $K$. 
The concept of spatial refinement is illustrated in Figure 5.6. It alleviates the inflexibility of agglomerative hierarchical clustering. By limiting the refinement only to boundary points, the clustering is expected to have a high continuity. At any iteration, let \( x_i \) be point in cluster \( S_c \) but not a border point. Even if there exists a cluster \( d \) such that \( d(x_i, c_d) < d(x_i, c_c) \) then \( x_i \) is not considered to be reassigned to cluster \( d \). It will only be joined to cluster \( d \) when it is at the boundary of cluster \( c \). Therefore, spatial refinement is fast, since only a limited number of reallocations have to be considered.

![Spatial Refinement Process Diagram](image)

**Figure 5.6 Spatial Refinement Process**

5.5.1 Experimental Results

A large variety of color textured images is employed in our experiments. The images originally are stored in RGB format. Each of the primitive colors (red, green and blue) takes 8 bits and has the intensity range from 0 to 255. Some experimental results are shown in Figure 5.7.
Figure 5.7 Segmentation Results of Clustering Algorithms with Spatial Refinement

a) Input images b) using Fuzzy C means Spatial Refinement c) using K means Spatial Refinement d) using EM with Cost function Spatial Refinement
It demonstrates the segmentation results obtained using spatial refinement in Fuzzy C means, K-means and EM with Cost function algorithms. It is very clear from the results that the proposed EM with cost spatial refinement outperforms both Fuzzy C means and K-means for all types of images.

The resulting images with much smaller number of segments can preserve the main features of the objects. But Fuzzy C means and K means spatial refinement algorithms produce over segmented outputs. In “Seagull” image, even the waves present in water are considered as texture and water is segmented uniformly whereas the other approaches segment each wave as a separate region and results in over segmented outputs. For the “Hill” image, our proposed approach correctly identifies the four regions present in the image and also the four regions are very well discriminated. But Fuzzy C means and K means spatial refinement does not discriminate effectively. Even in the “Shadow” image, the fire region is correctly segmented and the fine textured soil is identified as another region which results in smoothened output. But the other algorithms produce so many grains in the output that results in poor segmentation. The algorithm works quite well for multispectral images also. In “Volga” image, the streamlines are very well discriminated. Overall, our proposed method produces very good segmentation for smooth and fine textured images.

5.6 WAVELET BASED EM WITH COST SPATIAL REFINEMENT ALGORITHM: EXPERIMENTAL RESULTS

Even though the highly textured images are better segmented compared to other methods it does not produce very good segmentation results. To perform robust segmentation of such color textured regions, it is proposed to apply EM with cost spatial refinement algorithm in wavelet
domain. The algorithm is applied on the i) feature space derived from the wavelet coefficients and ii) feature space derived from the wavelet statistical features. The performance of the wavelet based segmentation algorithm is compared for various wavelet transforms by using SPOT satellite images and various kinds of color textured images created from VisTex database. Figures 5.8 - 5.15 depicts the superiority of the proposed approach applied on the feature spaces created.

It is observed that Fuzzy C means spatial refinement clustering algorithm does not produce good segmentation results for all kind of images. Almost over segmented images are obtained compared to other two algorithms. K-means spatial refinement algorithm produces better segmentation results for smooth textured images. But it is not superior to EM with cost spatial refinement algorithm. Specifically, in “HP2” image, eventhough, the three regions are correctly identified, Fuzzy C means does not separate the regions effectively whereas K-means produces inhomogeneous regions. Even in “Delta” image, two colors are enough to express the shape, material and its background. Our proposed method captures the two regions correctly and produces uniform segmentation output. In “Garden” image, over segmentation results in both Fuzzy C means and K means algorithms. But EM with cost produces homogeneous segmented output. In addition, test mosaics formed by various types of textured images are better segmented compared to the other two methods. In brief, the experimental results of wavelet based EM with cost spatial refinement algorithm are quite consistent with the visualized color distribution in the objects of the original images.
Figure 5.8 Segmentation Results Using SWT Coefficients

a) Input Images  b) Using Fuzzy C means Spatial Refinement

c) Using K means Spatial Refinement  d) Using EM with Cost Spatial Refinement
Figure 5.9 Segmentation Results Using SWT Statistical Features
a) Input Images b) Using Fuzzy C means Spatial Refinement c) Using K means Spatial Refinement d) Using EM with Cost Spatial Refinement
Figure 5.10 Segmentation Results Using IMOWT Coefficients
a) Input Images b) Using Fuzzy C means Spatial Refinement c) Using K means Spatial Refinement d) Using EM with Cost Spatial Refinement
Figure 5.11 Segmentation Results Using IMOWT Statistical Features
(a) Input Images  b) Using Fuzzy C means Spatial Refinement  c) using K means Spatial Refinement  d) Using EM with Cost Spatial Refinement
Figure 5.12 Segmentation Results Using DTCWT Coefficients
a) Input Images b) Using Fuzzy C means Spatial Refinement
c) Using K means Spatial Refinement d) Using EM with Cost Spatial Refinement
Figure 5.13 Segmentation Results Using DTCWT Statistical Features
a) Input Images b) Using Fuzzy C means Spatial Refinement c) Using K means Spatial Refinement d) Using EM with Cost Spatial Refinement
Figure 5.14 Segmentation Results Using MBWT Coefficients
a) Input Images  b) Using Fuzzy C means Spatial Refinement c) Using K means Spatial Refinement d) Using EM with Cost Spatial Refinement
Figure 5.15 Segmentation Results Using MBWT Statistical Features

a) Input Images  b) Using Fuzzy C means Spatial Refinement  c) Using K means Spatial Refinement  d) Using EM with Cost Spatial Refinement
Figure 5.16 shows the segmentation results in different color spaces. Among the five color spaces, RGB color space produces meaningful segmentation results for all types of images. The color textural features are well captured and the results successfully distinguish regions with different colors.

Figure 5.16 Segmentation Results Using Wavelet Based EM with Cost Spatial Refinement Algorithm in Various Color Spaces

a) Input Images b) in RGB c) in HSV d) in XYZ e) in YC_r C_r f) in LAB
The segmented images obtained using HSV, XYZ and YC_bC_r spaces are more noisy and also, all the regions are not correctly identified. The clustering information extracted is even worse in LAB space and therefore poor segmentation quality.

From this analysis it is observed that clustering algorithms based on spatial refinement work better for color textured images. Even though this method provides meaningful segmentation, it needs more time to perform segmentation. This can be overcome by applying clustering in 2D histogram of the image.

5.7 2D HISTOGRAM

2D histogram is the projection of 3D histogram onto band-pair planes. Consider the R plane of the color image $F = [f(x, y)]_{M \times N}$ of size $(M \times N)$ with $[0, ..., L-1]$, where $f(x, y)$ is the gray level of pixel location $(x, y)$. Consider also a G plane of the color image $G = [g(x, y)]_{M \times N}$, where $g(x, y)$ is the gray level of pixel location $(x, y)$. The 2-D histogram of images $R$ and $G$ is a $L \times L$ dimensional matrix $C = [C_{ij}]_{L \times L}$ which contains some statistical information (Eduardo et al 2003) regarding the number of gray level occurrences of two pixels at the same location $(x, y)$. $C = [C_{ij}]_{L \times L}$ from any pair of pixels $f(x, y)$ and $g(x, y)$ having respectively the gray levels $(i, j)$ can be formalized as follows:

$$C_{ij} = \text{card} \left\{ ( ( f(x,y), g(x,y) ) / f(x,y) = i, g(x,y) = j ) \right\} \quad (5.36)$$

where $\text{card}$ denotes cardinality.
5.7.1 2D Histogram Clustering

This method is based on the segmentation of band pair planes independently. For color images, the band subsets are denoted as RG, RB and GB pairs. First, 2D histogram is computed for each band pair. To eliminate the noise effects and spurious peaks in the histogram, they are smoothed and then down sampled by a factor of 2. It is usually considered that the different objects of the image are present in the histogram around the dominant peaks. The main peaks of the 2D histogram are the cluster centroids.

The proposed 2D histogram clustering method directly extracts the color cluster centroids without any assumption on the number of classes. Using these centroids as markers and applying watershed transform, clustered histogram is generated. The clustered histogram is then upsamped to its original resolution by replication of labels. From the resultant clustered histogram, a segmentation map is obtained. A label is then assigned to each region of the segmentation map and majority filtering is applied to suppress isolated pixels. The above clustering method is applied to the three 2D histograms of the input color image and three different segmentation maps are obtained. The final regions extracted correspond to homogeneous regions in the original color image. This method consists of three main stages namely, i) Clustering of color planes; ii) Label matching transformation and iii) Majority filtering.

5.7.1.1 Clustering of Color Planes

2D histogram of each color plane is constructed by summing up votes for all intensities occurring at each point of the color plane (Kurugollu et al 2001). To cluster this 2D histogram, it is assumed that the different objects of the image are present in the histogram around the dominant peaks.
(homogeneous regions in this gray level image). It is expected that these peaks will correspond to the cluster centroids in the spectral (RGB) measurement space. The main difficulty is to find the peaks without priori knowledge. The proposed method is based on the morphological clustering of the images. It is simple, fast and has proved their abilities in the processing of gray level images. The first stage of 2D Histogram clustering is illustrated in Figure 5.17.

![Diagram](image)

**Figure 5.17 First Stage of 2D Histogram Clustering Algorithm**

**Filtering the Histogram:** The goal of filtering is to remove or reduce the extent of small false peaks that occur in the histogram, while at the same time maintaining the original shape of the histogram. Instead of using a spatial mask to filter the histograms, the filtered histogram is obtained as a sum of normal distributions located at each cell. This procedure reduces the number of computations required to obtain the filtered histogram. The filtered histogram $H_{new}$, for a two dimensional case, is obtained at every cell $(k,l)$, $k = 0, 1, \ldots 63; \ l = 0, 1, \ldots 63$; as,
\[
H_{\text{new}}(k, l) = \sum_{i=0}^{i_{\max}} \sum_{j=0}^{j_{\max}} \frac{N(i, j, \sigma)H(k, l)}{\sum_{k=0}^{k_{\max}} \sum_{l=0}^{l_{\max}} H(k, l)}
\]  

(5.37)

where, \( N(i, j, \sigma) \) = normal distribution with mean, and the standard deviation \( \sigma \) which is chosen as scale of the filtering.

The filtered histogram is evaluated as a weighted sum of normal distributions centered at each point in the histogram. The peaks are then identified in \( H_{\text{new}} \).

**Down Sampling the Histogram:** To accelerate the cluster centroids determination (Lezoray and Cardot 2002), the smoothed histogram of size 256×256 is reduced to 128×128 by a down sampling factor of 2.

**Morphological Clustering:** In dilation techniques, the general approach is to take a binary image representation of the boundary and set to “1” any pixel that is currently at “0” but contacts a pixel with value “1”. This will add a layer of pixels to the boundary resulting in a small dimension change and will also fill in small gaps. Once gaps are filled, erosion technique is applied to restore the original size of the region.

In erosion methods, the general approach is to identify pixels with value “1” that contact a pixel with value “0” and set those to “0”. This could be modified to require contact with more than one pixel having value “0”.

The main peaks in the reduced histogram correspond to maxima in the 2D histogram; therefore, erosion applied on the latter reduces each bin to its main color. This property is used to extract the dominant colors. From this
reduced histogram, the ultimate erode set is extracted giving the main dominant colors of the 2D histogram. This method directly extracts the color cluster centroids without any assumption of the number of classes: the centroids considered to be the pixels which lastly disappear in an iterative process of morphological erosions (namely the ultimate erode set). These centroids are labeled and watershed transform of the 2D histogram is performed. The clustered histogram is finally upsampled to its original resolution (256×256) by replication of the labels. From the clustered 2D histogram a segmentation map is obtained since each region in the histogram image corresponds to a set of colors in the original image: to each (RGB) vector is associated the corresponding label in the clustered 2D histogram.

For each segmentation map of RG, RB and GB, the set of regions is denoted by $E(S)$ where $S$ is the segmentation map. For each segmentation map $S_i, \forall i \in \{\text{RG, RB, GB}\}$, the associated number of region is represented as

$$\theta(S_i) = \text{Card}(E(S_i))$$  \hspace{1cm} (5.38)

The fusion of the three segmentation maps is obtained by superimposition of the different region sets $E(S_{RG})$, $E(S_{RB})$ and $E(S_{GB})$.

### 5.7.1.2 Label Matching Transformation

In order to unify the segmentation maps, label matching transformation (Kurugollu et al 2001) must be constituted. A transformation that matches the labels of segmentation map $i$ to co-located segments in another map $j$ on the basis of maximum mutual overlap is defined as follows:

$$T_{ij} = y$$  \hspace{1cm} (5.39)
where \( x \) denotes the source label in the segmentation map \( i \), \( y \) is the target label in the segmentation map \( j \), which overlaps maximally with \( x \), and \( T_{ij} \) the label transformation. This equation implies that the region label \( x \) in map \( i \) must be same as the label \( y \) in the map \( j \) on the basis of being co-located and maximally overlapping. Thus, a total of \( 2 \times C(N, k) \), that is six transformations are formed: \( T_{RG,RB}, T_{RB,RG}, T_{RG,GB}, T_{GB,RB}, T_{RB,GB} \) and \( T_{GB,RG} \). Using these definitions of transformations, the bilateral matching cases are checked to find out regions to be identically labeled. A match is defined as:

\[
T_j(T_i(x)) = x \quad (5.40)
\]

This equation means that in map \( i \) and \( j \), there are two segments that are each others maximally overlapping counterparts, so that the \( x \)-labeled segment in \( i \) is mapped into \( y \) in \( j \), while the \( y \)-segment in \( j \) is mapped to \( x \) in \( i \). Notice that in general if \( T_i(x) = y \), then \( T_j(y) \neq x \)

1. RG-RB and RB-RB and RB-GB all match;
2. RG-RB and RG-GB match, RB-GB does not match) or (RG- RB and RB- RG-RB does not match);
3. (RG-RB matches only as change in only R-band forms a region) or (R-GB matches only G-band forms a region) or (RB-GB matches only as change in B-band forms a region).

The first scenario is conflict free since the same segment exists in all the three maps. However, in the second scenario one of the three matches is failing. This situation rarely occurs and it must be due to partial deformations and noise. This match is however overruled since it is not physics based and the labels are forced to be the same as matching ones in the
other band-pairs. The third scenario arises when a strong variation in only one color band is responsible for the creation of a segment. In this case, the labels of only the matching maps are made to agree while the corresponding pixels of the conflicting band-pair are left without a label. This means that this "conflicting" band will simply not play a role in the fusion stage.

5.7.1.3 Majority Filtering

Once the label concordance is achieved, the final segmentation is obtained by fusing the three band pair maps. Towards this aim, majority filtering operation is applied on both spatial and chromatic dimensions. The filtering operation is illustrated in Figure 5.18.

![Figure 5.18 Majority Filtering](image)

Figure 5.18 Majority Filtering
A 5×5 neighborhood of a pixel in each band pair is considered. Therefore the label of a pixel is declared as the most frequently encountered label in a 3×5×5 spatial–chromatic neighborhood. This fusion scheme incorporates spatial information into the segmentation process since the label of any pixel is determined by taking into account the labels of totally 25 neighbor pixels.

5.8 WAVELET BASED 2D HISTOGRAM CLUSTERING ALGORITHM: EXPERIMENTAL RESULTS

The proposed algorithm involves the use of wavelet transform to construct suitable feature space on which 2D Histogram Clustering is employed. To study the performance of this algorithm, a variety of color textured images created from VisTex database are used. The segmentation results are shown in Figures 5.19 and 5.20.

It is observed that wavelet based 2D Histogram clustering outperforms wavelet based clustering with spatial refinement. For Highly textured images, Coefficient method performs better than statistical feature based method. Among the various wavelet transforms, DTCWT derived coefficients offer very good segmentation results for highly textured images. The SWT has the second best results. IMOWT and MBWT coefficients are not so good for effective segmentation preferably for multi textured images.

For smooth textured images the statistical feature based method outperforms the coefficient based clustering method. It considers the neighborhood pixel values and produces the segmentation results accordingly. Among the various wavelet transforms, IMOWT derived statistical feature space produces very good segmentation results. MBWT performs better and SWT produces over segmented images.
Figure 5.19  Segmentation Results Using Wavelet Coefficients

a) Input Images  b) Using SWT  c) Using IMOWT  d) Using DTCWT  
  e) Using MBWT
Figure 5.20  Segmentation Results Using Wavelet Statistical Features
a) Input Images  b) Using SWT  c) Using IMOWT  d) Using DTCWT  
e) Using MBWT
The performance of the proposed wavelet based 2D Histogram clustering algorithm is applied in various color spaces. The results are shown in Figure 5.21.

Figure 5.21 Segmentation Results Using Wavelet Based 2D Histogram Clustering in Various Color Spaces

(a) Input Images (b) in RGB (c) in HSV (d) in XYZ (e) in YCbCr (f) in LAB
It is observed that the RGB color space is quite suitable for all types of images. LAB color space is the next best color space. XYZ suits for smooth textured images and HSV produces over segmented results.

5.9 CONCLUSION

In this chapter, we have presented algorithms to show how Wavelet theory and Clustering together can be applied in the domain of color imagery for segmentation. Features extracted by our method characterize the textures present in the color images in a better way by combining approximation and the selected detail sub images. This facilitates an improved segmentation quality of different textured regions. The results validate that the proposed methodology is indeed superior to some of the existing clustering algorithms especially for multi textured images. This work can further be extended for recently developed wavelet transforms to investigate the suitability to combine with clustering for effective segmentation of high dimensional images. The quantitative analysis is carried out in chapter 6.