# 6. FEATURE MATCHING

## 6.1 Overview

This Chapter describes the speech feature matching techniques that are used in speaker recognition systems. The vector quantization techniques used in this thesis is described in detail.

## 6.2 Speech Feature Matching

The problem of speaker recognition belongs to a much broader topic in scientific and engineering so called *pattern recognition* [17]. The goal of pattern recognition is to classify objects of interest into one of a number of categories or classes. The objects of interest are generically called *patterns* and in our case are sequences of acoustic vectors that are extracted from an input speech using the techniques described in the previous section. The classes here refer to individual speakers. Since the classification procedure in our case is applied on extracted features, it can be also referred to as *feature matching*.

Furthermore, if there exists some set of patterns that the individual classes of which are already known, then one has a problem in *supervised pattern recognition*. These patterns comprise the *training set* and are used to derive a classification algorithm. The remaining patterns are then used to test the classification algorithm; these patterns are collectively referred to as the *test set*. If the correct classes of the individual patterns in the test set are also known, then one can evaluate the performance of the algorithm.

The state-of-the-art in feature matching techniques used in speaker recognition include Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). In this thesis, the VQ approach [16] is used, due to ease of implementation and high accuracy. VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a *cluster* and can be represented by its center called a *codeword*. The collection of all codewords is called a *codebook* [18].

Figure 6.1 shows a conceptual diagram to illustrate this feature extraction process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2. In the training phase, using the clustering algorithm a *speaker-specific* VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The result codewords (centroids) are shown in Figure 6.1 by black circles and black triangles for speaker 1 and 2, respectively. The distance from a vector to the closest codeword of a codebook is called a VQ-distortion. In the recognition phase, an input utterance of an unknown voice is "vector-quantized" using each trained codebook and the *total VQ distortion* is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified as the speaker of the input utterance.

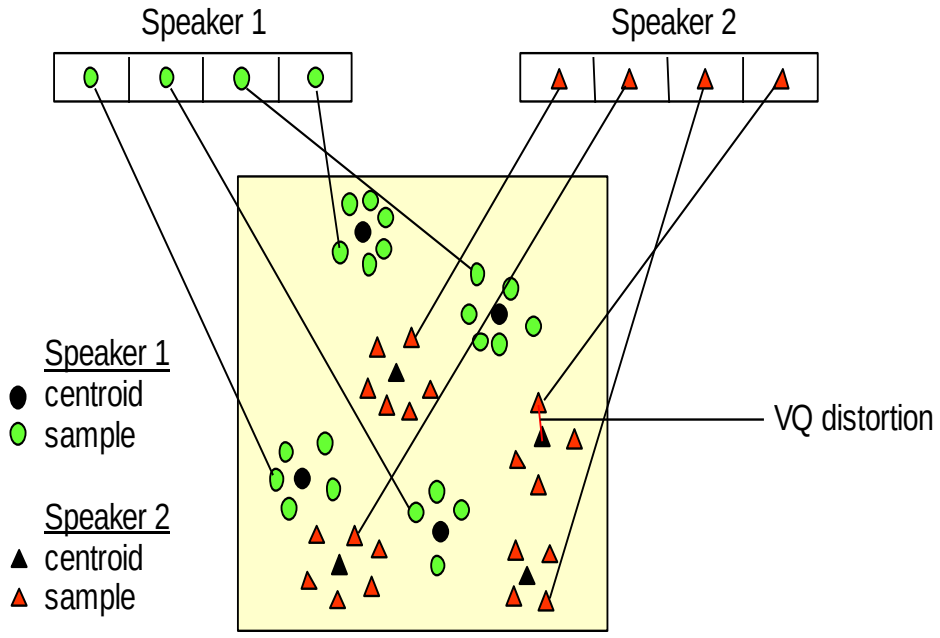



**Figure 6.1** Conceptual diagram illustrating vector quantization codebook formation. One speaker can be discriminated from another based of the location of centroids. [16]

### 6.3 Quantization

Quantization [18] may be applied to audio files directly or to its parameters, in which it encodes the data using the minimal amount of information. Due to this minimal representation

of the signal, high accuracy of the process to represent the original signal is required. Waveform coders are capable of representing these signals effectively. Vector quantization (VQ) is another quantization method that encodes groups simultaneously rather than individual data values.

Uniform or Linear Quantization is the most basic type of quantization where the range of values for the signal is divided into evenly spaced quantization levels. The number of quantization levels determines the number of codewords available (no. of quantization = no. of codewords) for quantization, where the codewords directly represent a quantized level of the signal. In short, a size of n bits give $2^n$ codewords and $2^n$ quantization levels.

For certain cases, non-linear spacing is employed between the quantization levels. Space settings are based on distribution of sample values, where smaller distances are set for larger sample values. These non-linear settings ensure smaller overall quantization error. Direct speech waveform coding uses logarithmically spaced quantization levels to best match the expected distribution of the speech signal. Figure 6.2 shows the distribution of quantization levels for a non-linear 3-bit quantizer.

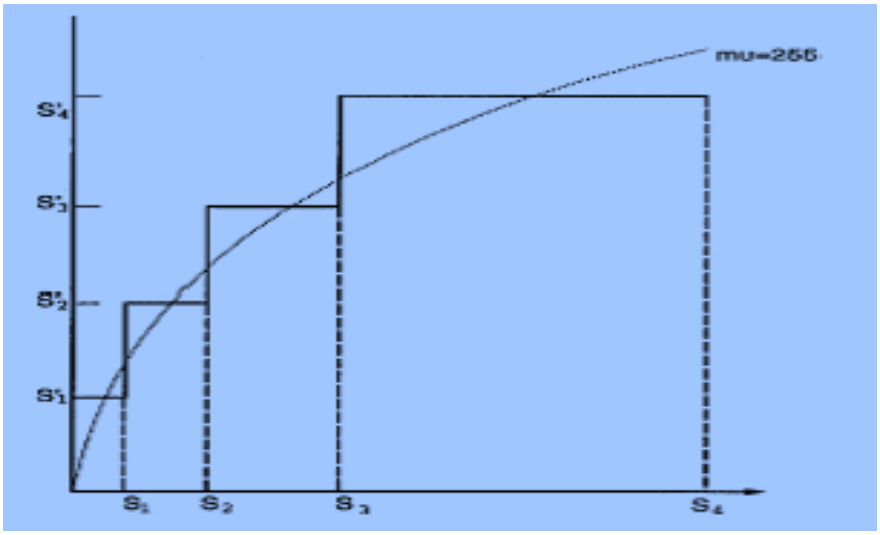



**Figure 6.2** Distribution of quantization levels for nonlinear 3-bit quantizer. [8]

### 6.4 Vector Quantization

Vector quantization (VQ) is a concept, which makes use of the relation among elements in the group to encode groups of data as a whole more efficiently rather than individual elements of data. This system is usually applied to a parameter representation of audio. Figure 6.3 and Figure 6.4 display a simple vector quantization system.

Data is encoded by feeding an audio segment $S_n$ into a parameter extraction algorithm (such as Mel Cepstral Processing, see Chapter 5). The parameters vectors $V_n$ are then compared with each vector using a distance metric to assess which vector best fits the input vector.

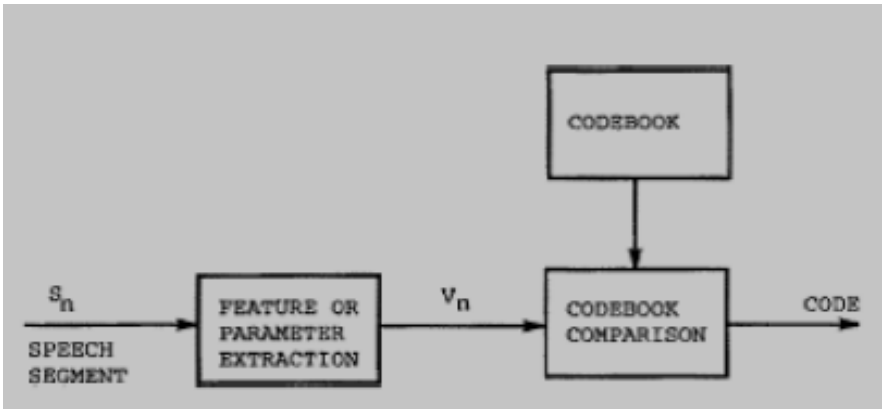



**Figure 6.3** Vector Quantization encoder. [16]

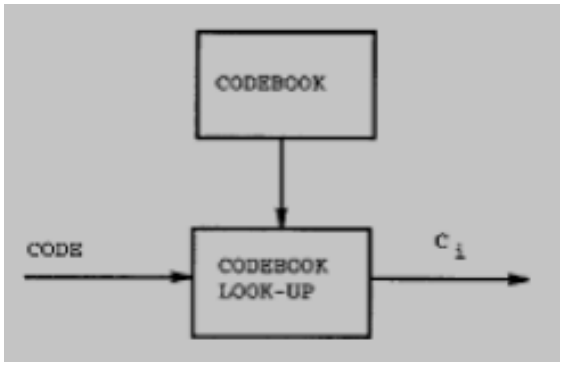



**Figure 6.4** Vector Quantization decoder. [16]

Basically, the codebook is calculated before hand and is stored in the decoder as it is in the encoder. In the decoding stage, the code is sent through a codebook lookup process. The

transmitted or stored codeword is an index in the codebook. This index is the same as determined during the coding process. Based on the index, the vector $C_i$ is retrieved. This vector is determined by the codebook generation process, to best represent vectors similar to the original vector $V_n$. The vector $C_i$ is further processed to produce synthesized speech, depending on the information it contains.

Numerous algorithms exist for codebook generation. For L codebook entries, the M-dimensional vector space is sectioned into L overlapping cells. This sectioning is usually performed based on a set of audio vectors referred to as training vectors. In many implementations, $C_i$ is the centroid of the training vectors within the cell i. The centroid is the multidimensional mean of those training vectors for a particular cell.

The centroids of the cells represent the output code vectors associated with the corresponding cells. In other words, during the encoding process, when an input vector falls within a particular cell, the index of the cell will be transmitted as the codeword. For the decoding process, the centroid of the cell will be the output vector. Figure 6.5 displays a two-dimensional vector space, partitioned by cell boundaries, with the centroids marked. The cells are numbered with a k-bit codeword where k=log2 L. The dimensions of vector space are $V_{n1}$ and $V_{n2}$, the first and second elements of the vector $V_n$ .

### 6.4.1 Distortion measure

A distortion measure indicates the level of similarity between two signals. These are usually compared in terms of vectors

A common distortion measure is the sum of the squared differences. It is computed as:

$$Squared\ Error(V_n, C_i) = \sum_{j=0}^{M-1} (V_{nj} - C_{ij})^2$$

Where $V_{nj}$ is the $j_{th}$ element of vector $C_i$. In this case, all difference between certain vector elements are weighted equally.

The distortion can be adjusted to weigh the difference between certain vector elements more than others. The weighted error is:

$$Weighted\ Error(V_n, C_i) = \sum_{j=0}^{M-1} [W_j (V_{nj} - C_{ij})]^2$$

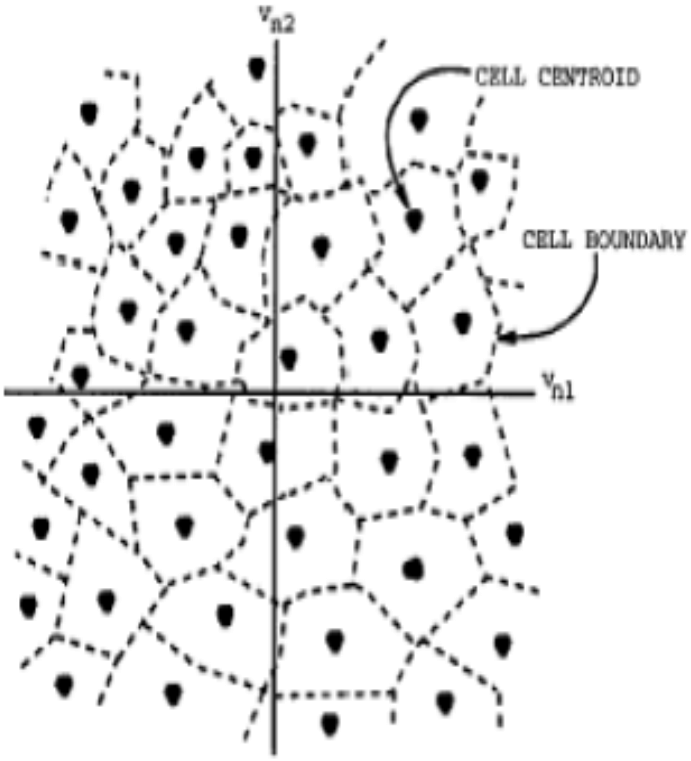



**Figure 6.5** Vector quantization partitioning of a two- dimensional vector space; centroids marked as dots. [16]

If the variance of the vector element $V_{nj}$ is different from that of $V_{kj}$, and differences relative to the respective variances are important, the weighting can be used to normalize by the standard deviation, $\sigma j$. The $\sigma j$ is estimated from the training data set as the square root of the variance of element j. The weighting then $w_j = {}^{1}/_{\sigma j}$. In this case, differences are treated inversely proportional to the variance of the element in the training set.

### 6.4.2 Clustering the training vectors

After the enrolment session, the acoustic vectors extracted from input speech of each speaker provide a set of training vectors for that speaker. The next important step is to build a speaker-specific VQ codebook for each speaker using those training vectors. There is a well-know algorithm, namely LBG algorithm [Linde, Buzo and Gray, 1980], for clustering a set of *L* training vectors into a set of *M* codebook vectors. The algorithm is formally implemented by the following recursive procedure:

1. Design a l-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook $\mathbf{y}_n$ according to the rule

$$\mathbf{y}_n^+ = \mathbf{y}_n(1+\varepsilon)$$

$$\mathbf{y}_n^- = \mathbf{y}_n(1-\varepsilon)$$

   where $n$ varies from 1 to the current size of the codebook, and $\varepsilon$ is a splitting parameter (we choose $\varepsilon$ =0.01).
3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).

4. Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.

5. Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold

6. Iteration 2: repeat steps 2, 3 and 4 until a codebook size of *M* is designed.

Intuitively, the LBG algorithm designs an *M*-vector codebook in stages. It starts first by designing a 1-vector codebook, then uses a splitting technique on the codewords to initialize the search for a 2-vector codebook, and continues the splitting process until the desired *M*-vector codebook is obtained.

Figure 5.6 shows, in a flow diagram, the detailed steps of the LBG algorithm. "*Cluster vectors*" is the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. "*Find centroids*" is the centroid update procedure. "*Compute D (distortion)*" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

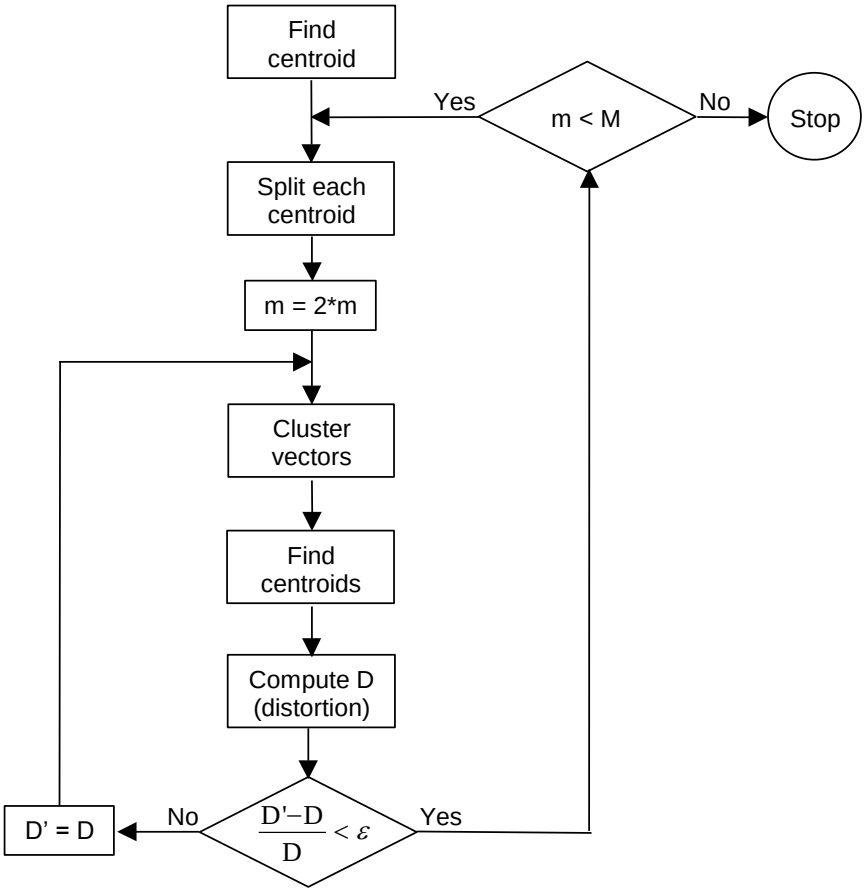



**Figure 6.6** Flow diagram of the LBG algorithm [16].

### 6.5 K-Means Clustering

In section 6.3, we talked about Vector Quantization which is used to reduce the size of the data for comparison where codebooks are generated using the k-means clustering algorithm. K-means clustering is one of the most conventional and successful methods employed as a summarization technique for cepstral parameters. Perceptual studies show that it is a good reduction method for modeling the coefficients and is a relatively efficient process. This often

terminates at a local optimum and involves 3 parameters; namely t, k and n where **n** is the number of objects, **k** is the number of clusters, and **t** is number of iterations. whereby, k, t << n.

For the clustering technique, the size of the initial segments controls the resolution of the results since all segments will be at least this long .This section aims to provide an understanding of K-means Clustering and its algorithms. We begin with an overview of Clustering. This will be followed by a discussion of hierarchical and non-hierarchical methods and finally we look at K means clustering.

### 6.5.1 Clustering overview

Cluster analysis provides a short description or reduction in the information of the waveform. It classifies a set of observations into two or more exclusive groups based on combinations of many variables. The objective of clustering is to regroup data in the manner where data in the same group is the same, while data in different groups are different. This is illustrated in Figure 5.7 using coloured balls.

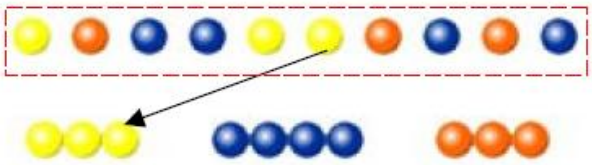


**Figure 6.7** Clustering Balls of the same Colour together. [25]

Thus from Figure 6.7, we see that clustering means grouping of data or dividing a large data set into smaller data sets of some similarity. Clustering works differently from discriminant analysis or classification tree algorithms. No prior information about classes is required, i.e., neither the number of clusters nor the rules of assignment into clusters are known. They are derived exclusively from the given data set without any reference to a training set. Cluster analysis allows many choices about the nature of the algorithm for combining groups.

In general, cluster analysis could be divided into hierarchical clustering techniques and nonhierarchical clustering techniques. Examples of hierarchical techniques are single linkage,

complete linkage, average linkage, median, Ward. Non-hierarchical techniques include K-means, adaptive K-means, K-medoids, fuzzy clustering. Since K-means clustering falls under the category of non-hierarchical techniques, we will only restrict our discussion on this type

### 6.5.2 Non-hierarchical clustering

Nonhierarchical clustering possesses as a monotonically increasing ranking of strengths as clusters themselves progressively become members of larger clusters. New clusters are formed in successive clustering either by merging or splitting of clusters.

Partitioning is one such method. This technique allows objects to be regrouped through a cluster formation process. Suppose we have k number of clusters as the objective and the partition of the object to obtain the required k clusters.

Partitioning starts with an initial solution, after which reallocation occurs according to some optimality criterion. Partitioning method constructs k clusters from the data as follows:

- Each clusters consists of at least one object n and each object k must be belong to one clusters. This condition implies that $k \leq n$.
- The different clusters cannot have the same object, and the construct groups up to the full data set.

The number of clusters k can be user defined or automatically generated to choose the best k. The following sections indicate the different clustering methods used. Of these, only K-means algorithm is used.

### 6.5.3 K-means method

K-means clustering is a partitioning method. That is, the function k-means partitions the observations in your data into K mutually exclusive clusters, and returns a vector of indices indicating to which of the k clusters it has assigned each observation.

K-means regard each observation in your data as an object located in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from

objects in other clusters as possible. Each cluster in the partition is defined by its member objects and by its centroid, or center. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized. Cluster centroids are differently computed for each distance measure, to minimize the sum with respect to the measure specified.

### 6.5.4 K-means implementation

To implement the K-means method, an iterative algorithm is used, which minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well separated as possible. Minimization of data can be specified by altering input parameters like the number of cluster centroids and the number of iterations.

The main idea is to define k centroids, one for each cluster. Centroids are to be placed appropriately because different location reaps different results. Ideally they should be place as far apart from one another as possible. Each point from the data set is then taken and associated with the nearest centroid. The k new centroids will be recomputed from the clusters resulting from the previous step and the process of grouping the data set points and the nearest new centroid will be repeated, thus we see a loop being performed, and this will continue until the centroids become stationary so no more changes are done.

Since K-means clustering finds a grouping of the measurements that minimizes the within-cluster sum-of-squares, we use the squared error algorithm described in section 6.3.1 for this purpose, which is also known as the Squared Euclidean Distance.
The steps for computing the K-means is shown below:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated

The loop procedure is expected to always end once the centroids remains unchanged. However it should be noted that the algorithm does not necessarily give the optimal results. Figure 6.8 helps visualize the steps mentioned above. These steps are similar to the above mentioned. Each of these steps below corresponds to the sequence of the diagram.

- Specify k number of clusters. e.g. k=5
- Randomly generate Cluster Centre locations
- Each Centre finds the centroid of the point it owns…
- And Jumps There
- Whole process repeated until terminated

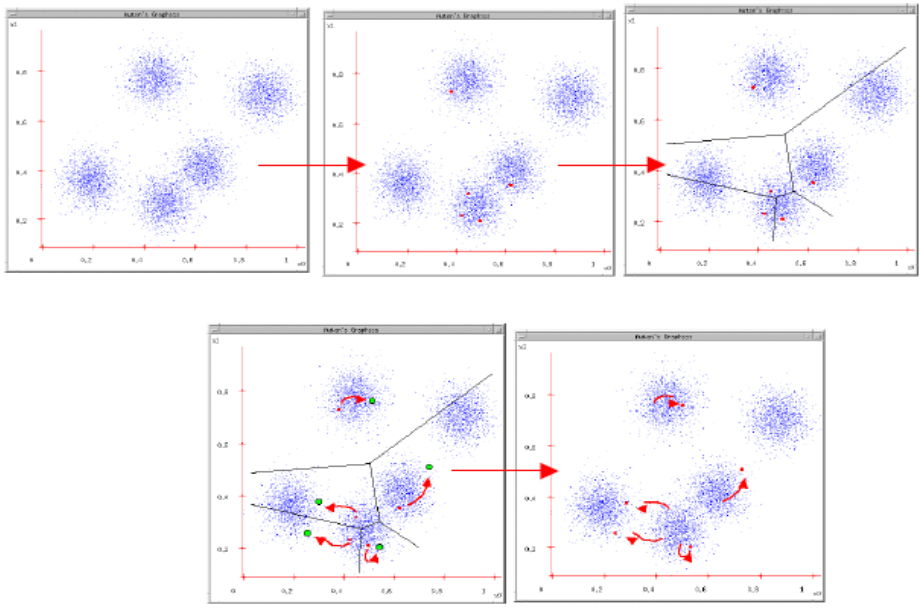
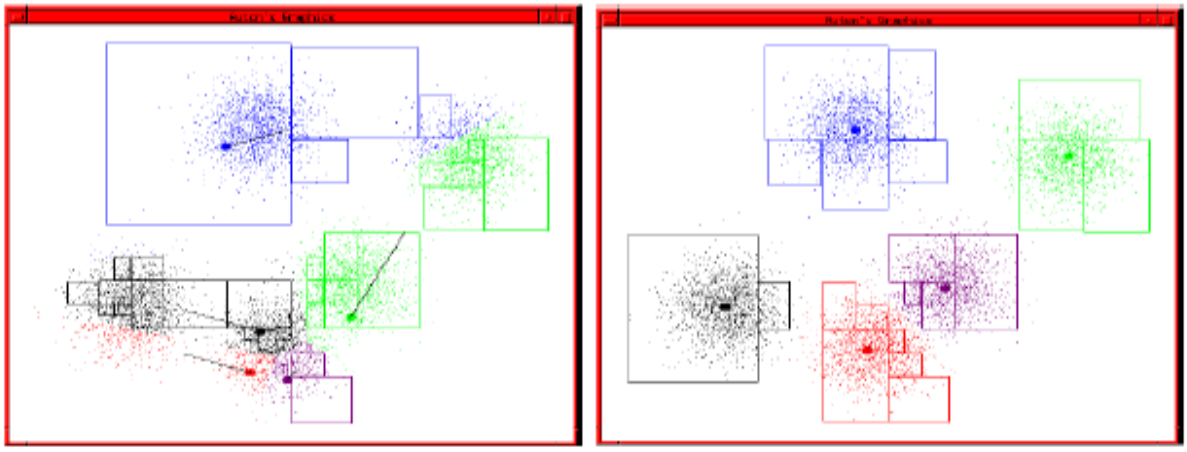


**Figure 6.8** The k-clustering method of Figure 6.9. Notice how similar data is grouped together. [8]



**Start** **Finish**

**Figure 6.9** K-means clustering. [8]

## 6.6 Summary

The Vector Quantization technique and Distortion measure that used for feature matching have been described in this chapter, and how to clustering the training vectors using the LBG (Linda ,Buzo and Gray) [18] method, the quantization and the Vector Quantization is also described separately, the K- means method and its steps. K-means falls under the category of non-hierarchical techniques and hierarchical techniques.