OBJECT RECOGNITION USING FORCE DATA CLUSTERING AND HMM BASED SHAPE RECOGNITION

by

Masoumeh Kalantari Khandani

B.Sc., Bahonar University of Kerman, Iran, 2005

THESIS SUBMITTED IN PARTIAL FULFILLMENT
FOR THE DEGREE
OF

MASTER OF APPLIED SCIENCE

In the School
of
Engineering Science

SIMON FRASER UNIVERSITY
Fall, 2010

Approval

Name: **Masoumeh Kalantari Khandani**

Degree: **Master of Applied Science**

Title of Thesis: **Object Recognition Using Force Data Clustering and HMM Based Shape Recognition**

Examining Committee:

Chair:

_____

Dr. Carlo Menon, P. Eng.
Assistant Professor, School of Engineering Science

_____

Dr. Parvaneh Saeedi, P. Eng.
Senior Supervisor
Assistant Professor, School of Engineering Science

_____

Dr. Shahram Payandeh, P. Eng.
Supervisor
Professor, School of Engineering Science

_____

Dr. Ivan Bajic, P. Eng.
Internal Examiner
Assistant Professor, School of Engineering Science

Date Defended/Approved: December 13, 2010

Declaration of
Partial Copyright Licence

Abstract

In this thesis the problem of detecting a known model object in a scene or database of images is addressed. We present two major components of a complete solution for this problem: a data clustering technique for image segmentation and feature extraction, and a shape recognition method. The presented novel data clustering method (Force) relies on the laws of electrostatic fields to find clusters of datapoints in a multiple-dimension space. Application of Force to image segmentation in gray level and color images is described in the thesis. We also show that Force can be successfully used for feature extraction from object images. We present a statistical shape matching method based on Hidden Markov Models (HMM) and then combine its recognition results with the recognition outcome of the Force based algorithm. We show improvement made when Force based features are added to the HMM based approach.

Keywords:

Object Recognition, Shape Matching, Data Clustering, Hidden Markov Models

Acknowledgments

I would like to thank my supervisor, Dr. Parvaneh Saeedi, for her support and encouragement and providing me with the opportunity to learn so much during my graduate studies. Her continuous support made my studies at SFU a good experience and allowed me to grow so much. Also, I would like to thank my family for all of their encouragement and support during my studies.

Table of Contents

List of Figures

List of Tables

Chapter 1 INTRODUCTION

The objective of this thesis is to provide image processing solutions for a computer vision system to identify a known object in a scene or database images. Such solutions can be used for robots equipped with a vision system to search for objects in an unknown environment, or for search tools that can identify object images in an online database (like Google image search, but using images instead of word tags).

Object recognition methods are in general comprised of several steps. For example, an image that is taken from an environment has to be segmented into candidate regions for further processing. Each region is then processed to find similarities with the known object that are searched for. As a result, an object recognition solution usually requires several image processing and analysis algorithms. In this thesis, we present two important contributions towards solving the object recognition problem: 1) a data clustering algorithm (called Force) that can be used for image segmentation, finding candidate regions of interest, and feature extraction, 2) a general method for shape recognition/matching based on the Force algorithm and Hidden Markov Models (HMM).

In the next sections, we explain in more detail the object recognition problem that we are aiming to solve, and the motivation behind the work presented in this thesis.

1.1 Motivation

For many years the problem of object recognition has been the subject of interest to many researchers and scientists. New methods and approaches have been introduced that take advantage of different aspects of natural science, human psychology, and technologies. While many of these specific methods deliver breakthrough results for certain domains of applications and sensor modalities, they may fail for others. For instance, object recognition methods that are designed for face detection, will fail in medical image applications. Some methods require human interventions which are called semi-automated methods [63] [64]. While, these methods produce better results they are

considered disadvantageous due to the human intervention. In general, the problem of object detection and recognition has not been solved and there are still so many potential applications that can benefit from a robust recognition solution. In this thesis the main focus is on developing a more general method for data clustering and image segmentation with applications in object's shape matching.

1.2 Problem Description

The problem addressed in this thesis can be described as follows: given a known object and a set of its views (taken from different angles and background removed), find instances of the object in a given scene image. The scene image in general can be the image of an actual environment that includes one or more instances of the object or a large online image database resource (for example Google internet image source). The main objective of this thesis is to address the shape recognition problem, and to provide tools, such as robust data clustering algorithms, that can be used for both identifying candidate regions of interest (ROI) in an image and for shape recognition. Figure 1 demonstrates an overall description of the problem addressed in this thesis. In this figure the red box object is given, the vision system's job is to identify the known object in the scene image. In later chapters of this thesis, we work with a standard set of objects' images from COIL-20 data base (Figure 2) in which 72 views of each object are included (5-degree apart) (Figure 3) [15][19].

As shown in Figure 1, for finding the desired object in a cluttered image, visual features and characteristics of the known object must be utilized to find the candidate regions in the scene. These visual features vary from colors and histograms to salient features, edges and contours.

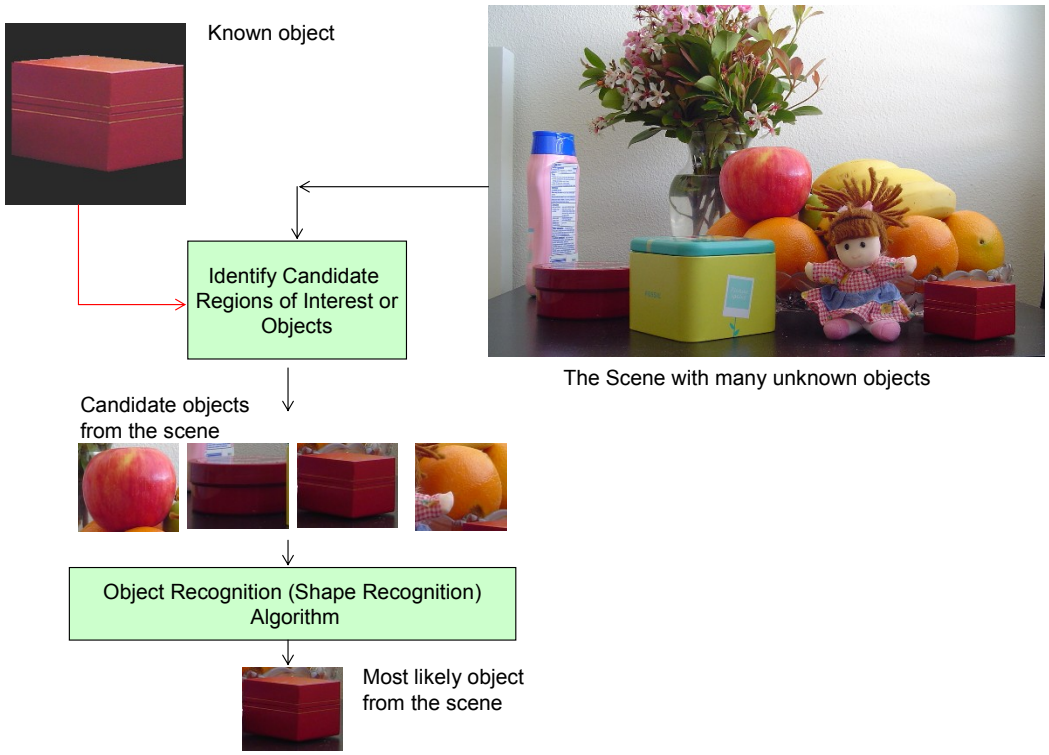


Figure 1. Problem description

Figure 2. Images of the COIL-20 dataset objects.

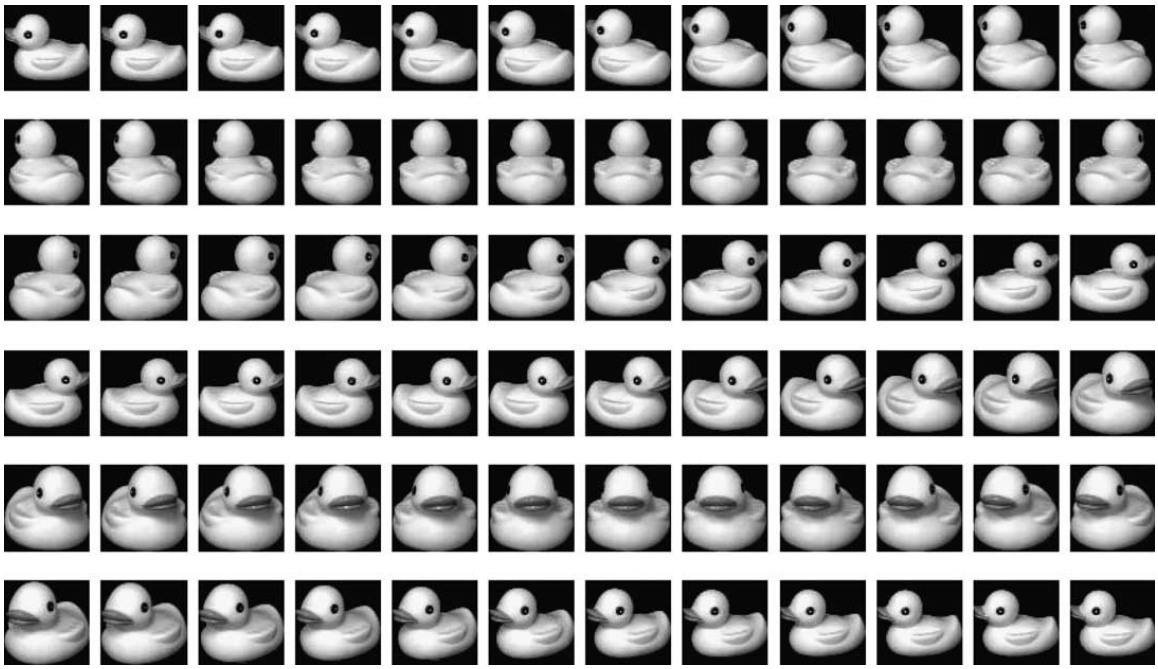


Figure 3. 72 views of the first object in the COIL-20 dataset [15]

One of the tools used in image analysis and in particular segmentation of an image into different sets of pixels is *data clustering*. In Chapter 3 of this thesis, we present a novel data clustering algorithm based on electrostatics field rules. The concepts of this data clustering algorithm are used in several applications and are presented in later chapters.

The second step of Figure 1 is a shape recognition procedure. This is also another focus of this thesis. Chapter 4 and Chapter 5 describe our proposed methods for the shape recognition problem. In the shape recognition application, we assume that candidate regions of interest are already found and available using our proposed method in Chapter 3.

1.3 Contributions

The main contributions of this thesis include a **data clustering** method for image segmentation and a novel algorithm for **shape recognition**.

Data Clustering: Image segmentation relies on similarity in intensity, color, texture, or statistical information to group image pixels. Features like color and spatial placement

of pixels of approximately the same color can be used to associate pixels together or "cluster" them. For this reason, data clustering algorithms have become a necessary tool in image processing. In this thesis, a new data clustering algorithm called Force is introduced. The applications of this method for grey level MRI images and color images are presented.

Shape Recognition: Shape recognition provides a method for identification of an object of interest using its shape. In this thesis, we present two different shape recognition methods each with different strengths and weaknesses. The first method uses Force algorithm for feature extraction and then utilizes two methods of comparing these features (Chapter 4). While this method is scale and rotation invariant, it could perform poorly under occlusion and clutters. The second method is a statistical method based on Hidden Markov Models which is more robust to clutters and occlusion but is not scale or rotation invariant (Chapter 5).We also present a method to combine these two methods together to provide better overall results.

1.4 Thesis Organization

The rest of the thesis is organized as follows: in Chapter 2 related previous works are reviewed. Chapter 3 presents a novel data clustering algorithm, Force, and several of its applications in image segmentation. Methods used to apply Force and Hidden Markov Models for the problem of object recognition are discussed in Chapter 4 and Chapter 5 respectively. A method to combine the results from methods of Chapter 4 and Chapter 5 is discussed in Chapter 6. Conclusion and future works are presented in Chapter 7.

Chapter 2 OVERVIEW OF PREVIOUS WORK ON OBJECT RECOGNITION

The objective of an object recognition process is to determine whether instances of a known object exist in an image. Thus object recognition can be described as a problem of *matching* models of a known object against representations of those models extracted from the unknown image.

Based on the structure and construction of these models, object recognition algorithms can be categorized into the following classes: model-based, shape-based, and appearance-based classes.

In model-based approaches [55] [56] and [57], an object could be represented as a collection of geometric primitives such as boxes, spheres, cones, cylinders, etc. In shape-based methods, an object could be described by its shape or contour features. Shape based methods have been studied since the early works of Fischler and Elschlager [60] in the 70's, but are still actively investigated (for example in the works by Malik, *et al.* [58] [59]. The appearance-based approaches use the appearance of the object as a whole [61] [62]. In these methods, the appearance is usually captured by various views of the object. Several statistical methods have been developed to work with the appearance of an object, as well as methods based on the color and texture of the object.

The proposed method in this thesis can be best classified as an appearance-based method. In the next section, we present an overview of several related previous works in the area of appearance-based object recognition. Following this review, in Section 2.1.1, we present an overview of several well known statistical methods that are commonly utilized in appearance based approaches.

Since an important contribution of this thesis is a novel data clustering method, which can be used in many computer vision applications including appearance-based object recognition methods, we will also review some of the related works on that topic in Section 2.2.

2.1 Appearance-based Object Recognition

Appearance-based object recognition methods rely on extracting features from several information spaces. For example, these methods may use primitive features such as color, intensity, or texture, or more complex features such as edges, corners, and transform domains such as wavelet or Fourier. Often these features are combined through multiple views of the same object from different point of views or perspectives.

Usually, methods that utilize color convert RGB images to other spaces [4], [5]. One of the early works on this topic was the work of Gevers *et al*. [3] in 1999 that examined different color spaces (Hue, Saturation, etc.). From their experiments they concluded that the RGB space has low performance due to its sensitivity to the illumination variation that often exists when changing viewing direction. Based on their results, the discriminative power of some colour spaces such as HSV (hue, saturation and illumination value), and *rgb* (normalized *RGB*) are higher than RGB. Also these spaces are more invariant to changes in the viewing direction, surface orientation, and illumination direction. Later, Giessen *et al*. [4] proposed to convert images from RGB to HSV (hue, saturation and illumination value) color space. After conversion to HSV, using these three channels they divided an image into regions. Utilizing a voting scheme on the regions extracted from the query image and images in a database, they tried to find the most likely matching image from the database. Their method was tested for two data bases of ZuBuD and COIL-100. While the result was rather satisfactory, occlusion, cluttered background or other image imperfections and different conditions were not considered in that work.

In 2008, Pandey *et al*. [5] presented a method based on converting images from RGB to YCbCr colour space using a set of linear mapping equations. They reported that outstanding objects were easily detected by the algorithm regardless of the environment that they were in. Considering the image as a composition of three individual components of Y, Cb and Cr, they reasoned that pixels with high and low intensity values concentrated over a considerable range indicate an outstanding object in an individual channel of YCbCr. As a result, the task of object detection was reduced to selecting pixel regions of intensity extremes in each channel.

Despite the benefits of analyzing color features in an image, it is usually not sufficient for detecting an object if the observed features such as color and intensity values are not complete or unique to an object. Aspects such as occlusions and clutter in the scene and similarities with other objects could affect the performance of any object recognition algorithm. Some of these problems can be alleviated if multiple views of on object are available. Therefore many researchers have used multi-view approaches for the object recognition problem. Selinger *et al*. [6] combined the results of a single view object recognition system applied to images obtained from multiple fixed cameras. Their experiments showed that an approach using multiple view images outperforms single view object recognition schemes; however, they concluded that if single view performance is already weak, use of multiple views does not noticeably improve the recognition results. Later an algorithm was presented in [7] in which a multi-view image sequence was utilized to establish feature point match correspondences across all images. They performed the recognition task by tracking feature points over the sequence. Although limited good results for small translation values (1 meter) were presented, the method was far from being capable of coping with larger view changes (including scale and rotation), lower image resolution and quality, and uncontrolled or variant lighting conditions.

A more recent example of the application of multiple views in object recognition was presented in 2010 by Westell [54]. In this work, depth information from multiple-view images was incorporated along with a novel way for representing local colour features to identify a known object in an unknown environment. This method is called the Saturation-Weighted Distributive Hue Histogram, which encapsulated the intrinsic colour information of the object as well as the spatial arrangement of colours within the object's boundaries.

In addition to the general methods described above, some appearance based methods rely on statistical approaches for the problems of feature comparison and match correspondences in images. We provide a review of some important methods in this domain in the next section.

2.1.1 Statistical Object Recognition Approaches

Statistical methods calculate the probability of occurrence of events to create some predictions that lead to better decision making. For example, Bayesian theory helps making better probabilistic predictions for unknowns, using information called the observations [12] [13]. With more observations, the probability associated with the unknowns is updated. Some important existing works based on statistical methods of object recognition are briefly explained here.

In [8]. Hetzel *et al.* proposed a recognition technique for objects in range images by analyzing a set of local features. Range images contain detailed information about the object shape in the form of distance; therefore, the preference is given to features that capture different aspects of the shape such as pixels depth, surface normal and curvatures. They combined these features to form a multi-dimensional histogram that was used as a discriminating classifier. Recognition was performed using histogram matching and a probabilistic recognition algorithm based on Bayesian decision theory. Histogram matching has low computational cost but it also has a coarse resolution. Since their method relies on histogram matching, it suffers from deficiency in partial occlusions and variation in lighting which could shift the object histogram.

A recent work that followed [8] was presented by Bang *et al.* in [9]. They presented a recognition method using the colour co-occurrence histogram and the spatial relation between image patches. In this method the candidate regions on the scene image similar to reference image patches were identified. Afterwards, a Bayesian method consisting of two steps of prediction and update was used to search the scene for highly probable candidate regions. This method was computationally expensive but with fewer chosen colours it would become less expensive. This method is also rotation-variant which is not desirable.

In addition to the above methods, the work in [10] suggested a Bayesian model for learning natural scenes. In this work, the scene image was represented by a collection of local regions. Each region was considered as part of a “theme” and was represented by a code-word from a large vocabulary of code-words. The goal of learning was to achieve a model that best represented the distribution of these code-words in each category of

scenes. After identifying the code-words in an unknown image, the model, which best fitted the distribution of the code-words, was found. Despite good performance for outdoor scenes, this method's performance for indoor scenes was not satisfactory.

While the above statistical methods rely on different schemes such as Bayesian decision theory, there are also a set of methods that rely on a statistical model known as Hidden Markov Model (HMM). Markov analysis is about modelling using a series of states/events that are dependent on previous states/events and independent from future events/states. In HMM modelling, a sequence of observed events is examined in order to find the hidden underlying model that created these events. HMM is naturally used for 1D data such as voice signals that have a serial nature. Application of HMM to 2D space or images is still a subject of interest to many research groups. An example of the application of Hidden Markov Models for 2D shape recognition was presented in 2001 by Bicego *et al.* [14]. They applied Hidden Markov Models for shape analysis using 'chain code' representation of the object's contours. HMMs were trained using the *Baum-Welch* [11] method with no assumption on the model's topology.

Chain Code [18] is a well known method for representing contours that specifies the edge direction of each contour in an edge list. Directions are coded into one of the eight directions of N, S, E, W, NE, NW, SE, and SW. Using an initial point and a symbol string (pointing to the direction of the following point in the contour (Figure 4)), the chain code of each 2D object is calculated. Perhaps, the main advantage of the chain code is in its invariance with respect to both rotation and translation. This method was shown to perform well for a small number of objects (three objects).

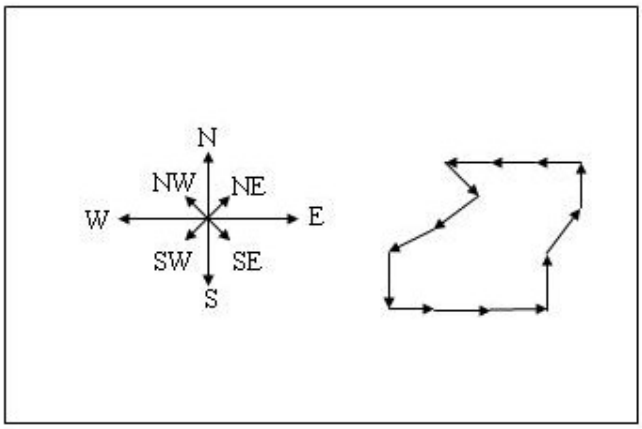


Figure 4. Chain code representation; each arrow shows the next direction, here starting form from left bottom the direction is E, E, E, N, NE, etc.

Bicego *et al.* also introduced another way of using HMM in the object recognition problem in 2005 [15] using wavelet transforms for the feature extraction. To obtain a sequence of partially overlapped sub-images, the objects were scanned in a raster-scan fashion. For each sub-image, wavelet coefficients were computed. Sequences of vectors (one for each view) were subsequently modelled using HMM. In this method, one model was trained for each object's view. The classification step was performed by assigning an unknown object's view to the class of the model with maximum likelihood. Their approach was tested on the images of COIL-20 database [19] using different window sizes and overlap ratios with various number of the wavelet coefficients. Although, their method presented some good results, in the case of partial occlusion or distorted objects, the method was very time consuming.

Recently, a new way of using HMM in image processing was introduced by Ma [16]. In their work, a form of two-dimensional hidden Markov model was used for image segmentation and classification for finding buildings in aerial images. The proposed model was based on identifying dependency among diagonal states. They made two assumptions; first, the transition probability of each state in the model depends on its adjacent neighbouring states in vertical, horizontal and diagonal directions; and second, the feature vector for each block must have a Gaussian mixture distribution (given its corresponding state) and it is independent of other feature vectors and their corresponding states. The algorithm was tested for a few images. The classification error rate for this

method (compared to manually labelled ground truth) was 11.67%. This work was mostly on segmentation and classification rather than object recognition.

Li *et al*. [17] reported that traditional block-based image classification algorithms ignored the statistical dependency among image blocks, which caused over-localization. They used a 2D model based on the concept of HMM, similar to [16], for image classification. In their work, HMM was used to create the inter-block dependency. This work also focused on image segmentation problem.

In Chapter 5 of this thesis, we present a statistical approach based on HMM for processing wavelet based image features. The recognition results from this HMM based method are used in combination with a non-statistical method that we propose in Chapter 4. Our proposed non-statistical approach is based on feature extraction using data clustering. A novel data clustering method along with its application for different image processing problems is contributed through this thesis research. In the next section we review some relevant previous work on the topic of data clustering.

2.2 Data Clustering

Data clustering is an essential tool in image segmentation as well as in data analysis and processing with numerous applications in pattern recognition, and image analysis. A data clustering method groups a set of datapoints into a few clusters of points based on some common properties (e.g., proximity of datapoints in some space). Perhaps the most popular data clustering algorithm is the K-means algorithm [24].

K-means clustering is a well-known partition-based technique for unsupervised learning. Some of properties of K-means algorithm include: applicability on numerical data, efficiency for processing large data sets, convergence to a local optimum, and generation of convex shapes. Despite moderate complexity, K-means algorithm is sensitive to initial seed selection [23]. To address some of K-means issues including sensitivity to the initial centers, several solutions have been proposed. For instance, ISODATA [26] found the optimal initial partitions by merging or splitting arbitrarily chosen initial partitions. [25] tackled the issue of sensitivity to the number of initial partitions by applying dynamic programming over the principal component direction. A

heuristic clustering dissimilarity function was incorporated into the sub-optimal K-means algorithm. The approach proposed in [27] used a combination of genetic algorithm and K-means to generate optimal clusters; the genetic algorithm was used to find good initial cluster centers and K-means algorithm was applied to find final partitions.

There are many other clustering methods that are suitable for large data sets. Among them there are two important algorithms CLARANS (Clustering Large Applications based on Random Search) [28], and BIRCH (Balanced Iterative Reducing and Clustering) [42]. CLARANS is a cluster analysis technique applied to spatial attributes of datapoints. It analyzes several random samples to find cluster centroids from the original dataset. However, CLARANS may not be equally efficient in all situations and may fail to find the real local minima [42]. BIRCH algorithm on the other hand keeps brief information about candidate clusters by applying a dynamic tree structure with leaf node representation of the clusters. BIRCH is claimed to be well capable of handling noise. Nonetheless, this method is sensitive to initial setting of its parameters.

In general, the most used clustering method is the well known K-means algorithm. As explained above, this algorithm has some shortcomings such as sensitivity to the selection of initial centroids. The enhancements proposed for K-means address some of the issues related to the sensitivity, but they are generally more complex to run and often do not resolve the problem completely. Another important shortcoming of existing methods such as K-means is the need for the number of clusters beforehand. This is in particular important if clustering is being used on an image where the number of segments (clusters) is generally unknown.

In this thesis we present a new clustering algorithm, Force [39], which attempts to resolve these issues by relying on laws of electrostatic to find global balance between points in the data set. Force simulates electrostatic fields in order to position cluster centroids in appropriate equilibrium locations. This algorithm is applicable for large data sets and can handle noise. Details of this algorithm are explained in Chapter 3.

Chapter 3 "FORCE" A NEW DATA CLUSTERING METHOD FOR IMAGE SEGMENTATION

Data Clustering is one of the most important topics in data processing and analysis, and has numerous applications in pattern recognition, image processing, and bioinformatics. Data clustering is essentially different from data classification. In data classification, the classes are predefined and datapoints are assigned to each class; whereas in clustering, datapoints are grouped into classes that have to be identified. Data clustering identifies the overall distribution patterns of the dataset, through finding clusters of data and their centroids. Our focus in this chapter is in presenting a new data clustering algorithm from the family of numerical and deterministic data clustering algorithms.

In general, data clustering is done based on measuring similarities between data items (for example, the proximity of datapoints in some space). There are many algorithms that cluster data; a review of them is presented in Section 2.2. The most common algorithm for data clustering is K-means algorithm [24]. In K-means a user-defined number of centroids are estimated. The datapoints are assigned to these centroids according to the minimum distance constraint. The algorithm iterates after modifying the centroids according to a set of predefined rules until the centroids do not move in two successive iterations. While K-means has been proven an effective algorithm, it could suffer from issues such as sensitivity to noise and initial centroids locations.

In this chapter, we present a new numerical-based data clustering algorithm that is inspired by rules of electrostatic fields. This novel approach allows efficient and robust clustering of multi dimensional data sets. It is suitable for larger data sets and produces predictable results, which are not sensitive to the initial centroid locations. The presented algorithm is of deterministic nature. Here, the main assumption is that datapoints are negative electrical charges scattered in a multi dimensional space. To cluster these charges, a number of positive charges (configurable parameter) will be released into the

space; these charges will move according to the electrostatic force so that all electrical charges reach to an electrostatic equilibrium or balance. When the balance is reached, positive charges will be at the equilibrium points, which we expect to be at the centers of data clusters. In certain exceptional cases unstable equilibriums may exist, which are not located at the data mass centers. However, these cases are rare due to the randomness of data and initial selections. Section 3.1 explains details of the proposed algorithm. An example of how Force is used for image segmentation in greyscale medical images is presented in Section 3.2 and the application of Force in color image processing is presented in Section 3.3.

3.1 Force Data Clustering Algorithm

The proposed data clustering algorithm employs the law of electrostatics that describes the nature of forces among electric charges. Direction of the electrical force is derived from electric fields around these charges. The main considerations of this algorithm are:

1- Cluster centroids have positive, variable charges and are spatially variable points.

2- Datapoints have single, negative, fixed charges and are spatially fixed points.

When positive charges are dropped amongst negative datapoints, an electric field is formed which forces the centers to move to places where forces are balanced. Under the balance condition, the centroids do not move any further. We call this configuration of points the electrostatic equilibrium. In Section 3.1.1 we first describe the general Force algorithm, and in Section 3.1.3 we present a modified version of this algorithm (called adaptive Force) that has a faster performance.

3.1.1 Data Clustering using the Concept of Electric Fields

Based on the stated assumptions in Section 3.1 the force between cluster centroids is repulsive, while the force between centroids and datapoints is attractive. These forces are calculated from:

$$F = c.\frac{q_1 q_2}{r^2} \tag{1}$$

Here c is a constant, r is the distance between two charges and q_1, q_2 are two charges that F is computed for. If the distance r is close to zero, it will be replaced with a constant small distance of R_0 to prevent F from becoming very large (infinity). The direction of the force between every two charges can be identified by the unit vector $(\vec{r}_1 - \vec{r}_2)/(\| \vec{r}_1 - \vec{r}_2 \|)$, where r_1 and r_2 are the coordinate vectors associated with the charges.

The centers naturally tend to move toward areas where datapoints are located. Meanwhile they repulse each other and therefore they will not land in the same cluster of datapoints. This implies that regardless of the initial random positions of the centers, they will move toward the center of clusters (if exist). The datapoints are associated with the centroid closest to them. Thus, the clusters are formed based on the minimum Euclidean distance constraint.

The charge of each datapoint is a constant value, but the charge of each centroid is updated dynamically and is set in proportion to the number of points associated with it (the presumed cluster that is associated with that center). For example, if N_j points are associated with a center j, the charge Q_j for the this center (assuming the charge of each point is one), is set as following:

$$Q_i=\alpha.N_j \qquad 0<\alpha<1 \tag{2}$$

The charge for each center (positive charge) is therefore less than the sum of all charges of the datapoints. This means that the centroids will be attracted to the datapoints and their mutual repulsive force cannot overcome the attractive force with the datapoints. If $\alpha >= 1$, the repulsive force will move the centroids far away from datapoints so that the steady state will be achieved at locations that are not cluster centers. If $\alpha << 1$, multiple centroids may be attracted to the same cluster and will be placed too close to each other. If $0<<\alpha<1$, centroids will end up near to the real centers of datapoint masses, the total force on each center is calculated by:

$$F_j=F_j^D+F_j^C \tag{3}$$

where F^C is the force applied to each centroid by other centroids (set C) and F^D is the force applied to each centroid by datapoints (set D) . The total force is calculated as:

$$\vec{F}_j = \sum_{i \neq j, i \in D \cup C} \frac{Q_j Q_i}{R_{ij}^2} \frac{(c_j - p_i)}{\|c_j - p_i\|}, \quad where \quad R_{ij} = \begin{cases} \|c_j - p_i\| & \|c_j - p_i\| > R_0 \\ R_0 & \|c_j - p_i\| \leq R_0 \end{cases} \tag{4}$$

Here c_j and p_i are vectors, describing the positions of centers and datapoints. Note that the value of Q_i for each datapoint i is *-1*. R_0 is the minimum distance between charges of Q_i and Q_j .

Based on the force on each centroid, the algorithm estimates a direction along which the centroid should move. The speed of the movement, or the size of each step taken at each iteration, is subject to several factors including weight of the centroids and charge masses. At this time, a fixed step size of η is utilized. Thus the direction of the force is the only required parameter for estimating the centroid's new position.

$$c_j^{(\tau+1)} = c_j^{(\tau)} + \eta \frac{F_j}{\|F_j\|} \tag{5}$$

Here $c_j^{(\tau+1)}$ is the new position of the cluster's centroid, $c_j^{(\tau)}$ is its previous position, and $F_j / \|F_j\|$ is the force's unit vector or the center's heading direction. There are several ways to control the speed of the centroid's movement. In this thesis, first a fixed step size is utilized. Adaptive adjustment of the step size is also explored in Section 3.1.3

After each iteration, new centroids' positions are updated and new clusters are formed based on the minimum distance constraint. For each new cluster, the charge of the centroid is updated according to equation (2), and the forces are recalculated. The algorithm will stop when the spatial variation of each cluster's centroid in two consecutive iterations is less than a predefined threshold. This condition happens only around electrostatic equilibrium points; the location of these points is independent of initial selections. This implies that the algorithm produces the same results in different runs unlike K-means for which the final centroids are sensitive to the initial selection of centroids. An advantage of Force algorithm over K-means is that Force moves towards the global equilibrium in each step. In other word Force calculates the global effect of all datapoints and moves the centroid in the direction of the electrostatic force. K-means on

the other hand moves the centroid in each step in the gradient direction to minimize a distance function within the current cluster (distance to the mean of the current cluster).

3.1.2 Algorithm Performance

To assess the performance of the proposed algorithm, several experiments are conducted. In the first experiment, two sets of normally distributed data sets are considered; some additive noise points (around 1% of all datapoints) are added to the datapoints. Figure 5-left shows the distribution of the datapoints. In this figure, the triangles represent the actual cluster centers around which the datapoints were normally distributed. Figure 5-right displays resulting centroids after running the Force algorithm. The final results are marked by diamonds. In this example the distance error between the true centroids and those found by the algorithm are smaller than 0.01 with η set to 0.02. As displayed in Figure 5-right, the centroids repel each other and move towards the centers of data masses.

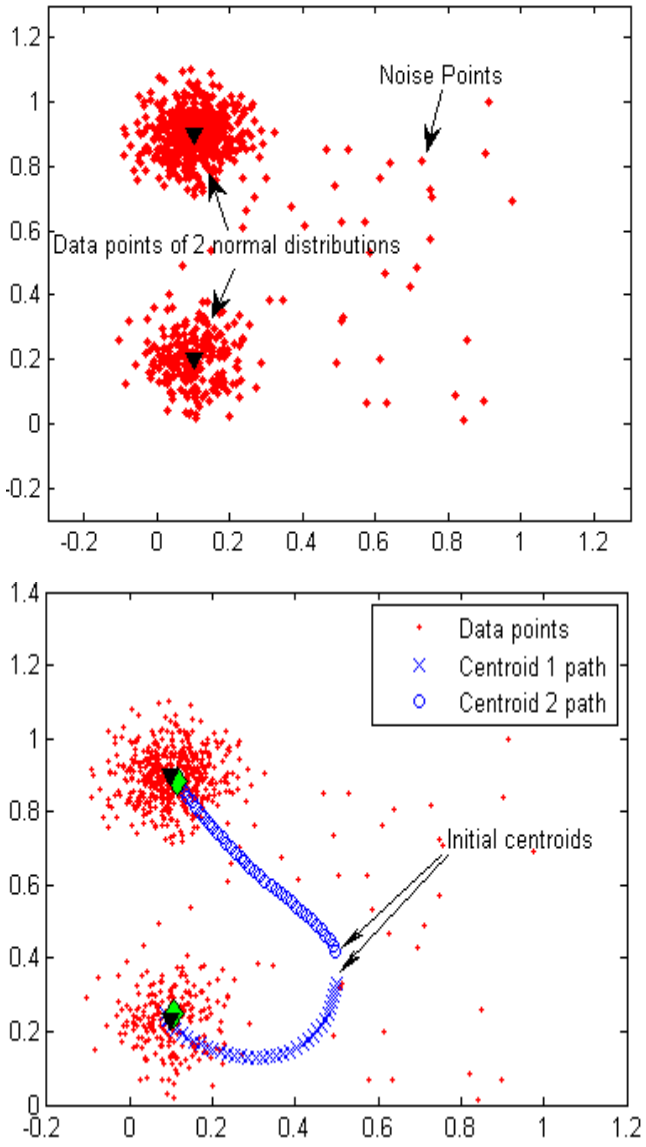


Figure 5. Datapoints and additive noise points presentation; triangles are the actual distribution centroids (top); movement path of centroids for Force algorithm, diamonds are the found centroids by the Force algorithm (bottom).

In the example depicted in Figure 6, in the earlier iterations both positive charges are attracted to the larger mass of datapoints; however, when they become closer to the mass and therefore each other, their mutual repulsive force will repel them and only one of them remains attracted to that mass while the other moves towards the second mass.

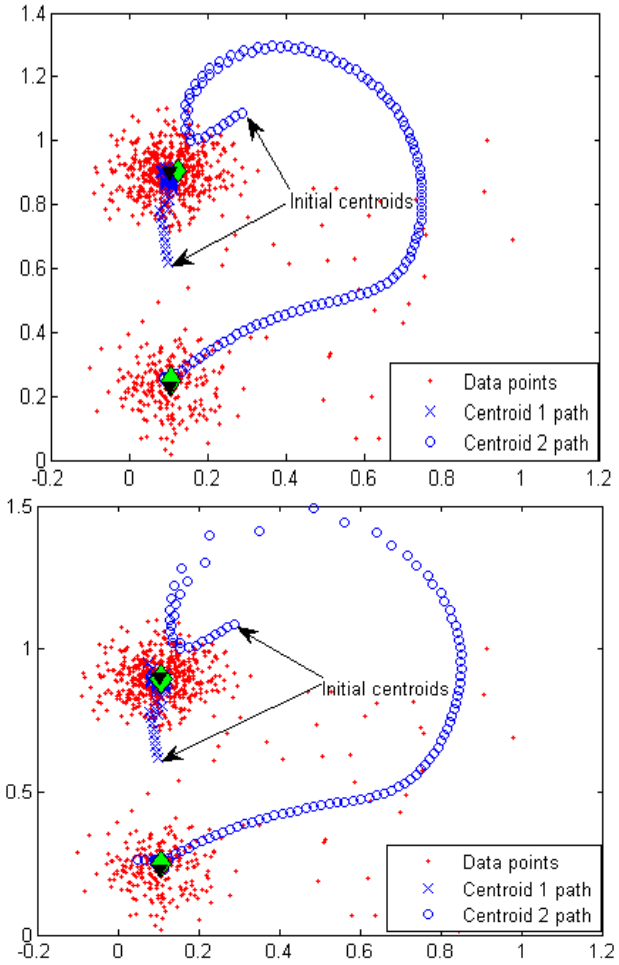


Figure 6. Centroid movement path:, Original algorithm using fixed step size (top); enhanced algorithm using adaptive step size (bottom)

3.1.3 Algorithm Improvement: Adaptive Force

In this section we describe some features that are incorporated in the system to improve the efficiency and accuracy of the proposed algorithm.

3.1.3.1 Adaptive Step Size

From Figure 6-left, it can be seen that the algorithm has taken many steps for the second centroid before reaching to its final destination. This initiated an adaptive step adjustment that forces the algorithm to choose longer steps if the centroid is far from the actual cluster centroid. Note that for centers located at a far distance of the clusters' datapoints, the computed force is small. Therefore, the step size is modified to be

inversely proportional to the amount of computed force. The modification of equation (5) is presented by:

$$c_j^{(\tau+1)} = c_j^{(\tau)} + \eta_1 \frac{F_j}{\left\|F_j\right\|} + \eta_2 \frac{F_j}{\left\|F_j\right\|^2} \tag{6}$$

Here η_1 acts as η in equation (5) and η_2 is the second step size. η_2 makes the movement magnitude dependent on the strength of the electrostatic force at the position of the centroid ($F/||F||^2$), whereas η_1 is independent of the strength of force as it is multiplied by a unit length vector ($F/||F||$) in (5).

The effects of this enhancement can be seen in Figure 6-right for which a fraction of the number of steps is reduced. In general the improvement is subjective to the case. For the case in Figure 6 the original algorithm takes 117 (left plot) steps while adaptive Force takes 102 steps (right plot). The reduction in the number of steps in this case is around 12.82 %.

Therefore, for the remaining work presented in this section, the adaptive step adjustments (η_1=0.04, η_2=1000) are incorporated. Also in all presented figures, the final centroids found by the Force algorithm are marked with diamonds, while K-means results are marked with squares.

3.1.3.2 Initial Centroids' Location Guesses

In order to improve the speed of the algorithm, in this section a simple yet effective adjustment is suggested. This adjustment is done by placing the initial location of the centers in the middle of the space where datapoints are scattered.

For this purpose, the center of gravity for the entire datapoint collection is found first. The initial centroids are then placed at that location with some small but random distance offset from each other. To see the effect of this adjustment, we plotted the traversing path of the centroids for the cases where the initial centroids are selected at the edge of the data range (Figure 7-left and Figure 8-left), and when they are selected near to center of gravity (Figure 7-right and Figure 8-right). The difference between Figure 7 and Figure 8

is the level of additive noise, respectively 5% and 20%. In these experiments the initial centroid locations and stop criteria for both Force and K-means were the same.

The results demonstrate that the suggested modification reduces the number of iterations in the algorithm. It must also be noted that the accuracy of the final solution is not jeopardized by this enhancement (maximum difference is smaller than the step size).

3.1.4 Adaptive-Force Performance Evaluation

The performance of the proposed method is examined for various initial centroid locations with different additive noise levels and distributions. The performance is also compared against K-means method. The code utilized for K-means is from Matlab's Statistics Tool Box.

To evaluate the performance of the algorithm for various initial centroid locations, we measured the Euclidean distance between the final estimated centroids and the actual cluster centers. Two initial centroids were placed at positions (β, β) and (β,*1-*β), where β changed from 0 to 1 in 100 steps. This is to sweep the range of *x-y* values in 0 to 1 range (data and noise points are distributed in this range as well, as shown in Figure 7).

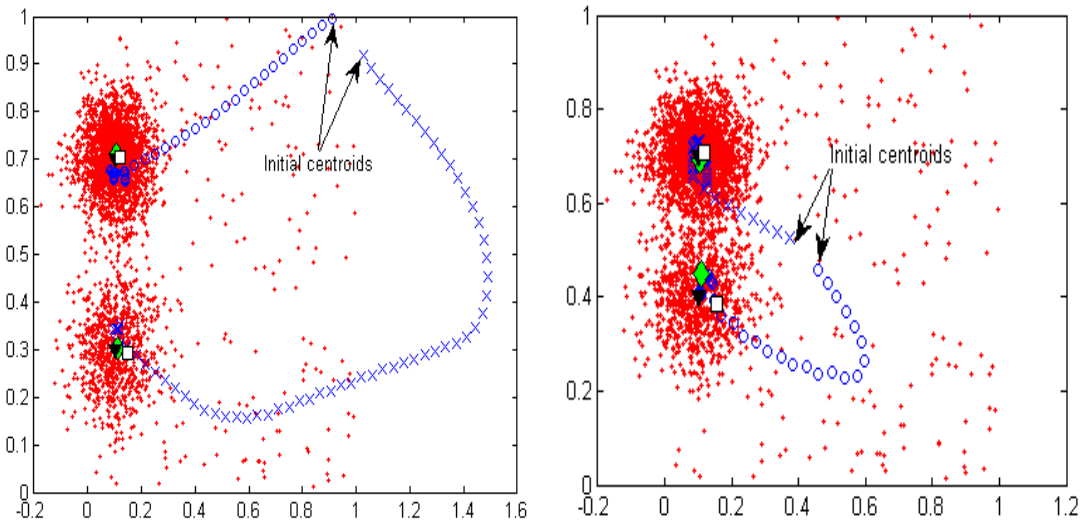


Figure 7. Different beginnings for Initial centroids: squares are final K-means centroids and diamonds are Force centroids. Initial centroids far from the data masses (left); initial centroids in the middle space (right) with low level of noise (5%) both cases.

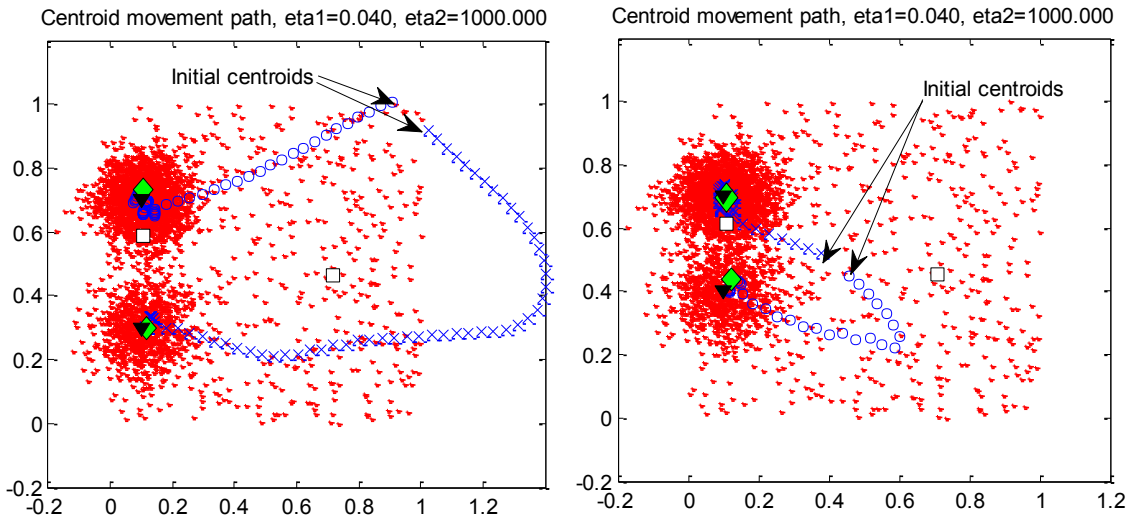


Figure 8. Variation in initial centroids starting point with 20% additive noise; Initial centroids are far from the data masses (left) ; initial centroids are in the middle space of the data masses (right).

The results depicted in Figure 9 show that the Force algorithm converges to the equilibrium point with an error less than 0.05 for 5% (Figure 9-left) and 0.1 for 20% (Figure 9-right) added noise

At the equilibrium state, the force F_j is large and the second term in equation (6) tends to zero. At this state, the movement of the centroid is around the equilibrium (otherwise the direction of electrostatic force would not reverse in consecutive steps to cause repetition). The repeated positions are at a distance of η_1 from each other (since the size of the first term in equation (6) is always η_1). This means that the centroids will stop at a point that is a maximum of η_1 from the equilibrium. Depending on the initial position, the ending point in each run may be different, but always at a maximum distance of η_1 from the equilibrium (in a circle with radius of η_1 around the true equilibrium).

Figure 9 also shows the Euclidean error distance for K-means algorithm. It is observed that under a low noise condition, K-means converges to the same solution, for this specific experiment; however, increasing the noise causes K-means algorithm to misidentify the cluster centroid for one case. This situation is also depicted in Figure 8, where additional noise causes K-means algorithm to converge to a wrong position. The Force algorithm output however remains unchanged and robust with respect to noise.

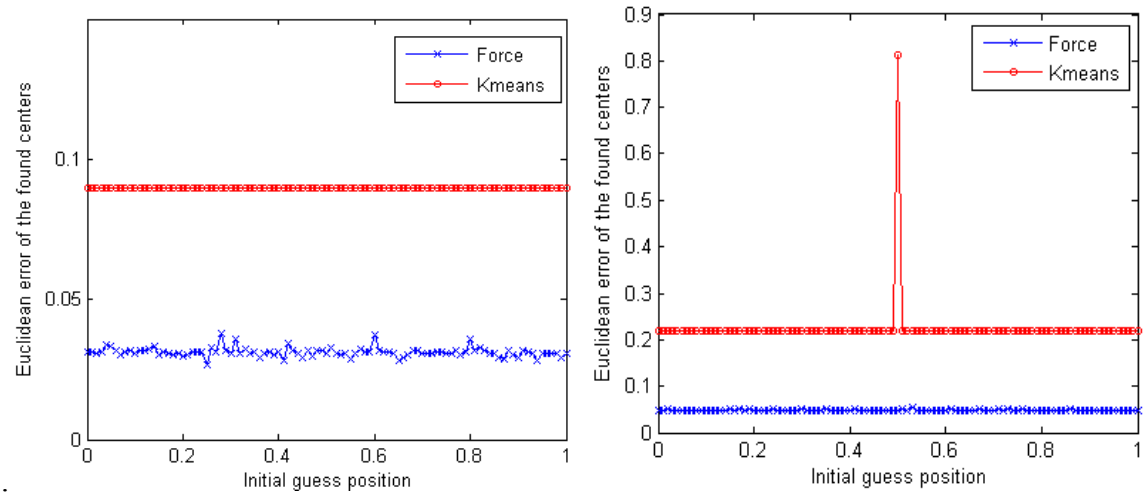


Figure 9. Euclidean error distance for the estimated centroids of K-means and Force algorithm for different starting points with 5% additive noise in left and 20% noise in right.

To further study the effect of the noise, we measured the Euclidean error distance at different levels and with various noise distributions in two additional test sets.

In the first set, the Euclidean distance between the final centroids (by K-means and Force) and the actual clusters' centroids at different noise levels are measured. Here, the noise was uniformly distributed in the space, [0 1], along both axes. Figure 10 compares the Euclidean error distance for both algorithms at different noise levels. The error was computed by averaging the results for 50 repetitions of the experiment using different initial points. The level of noise is specified by ratio of the number of noisy points to that of datapoints. It is expected that the error will increase as the noise level increases. It can be seen that the Euclidean error distance for K-means method increases rather quickly while the error for Force algorithm rises slowly. As mentioned earlier, this test was performed for a uniformly distributed noise.

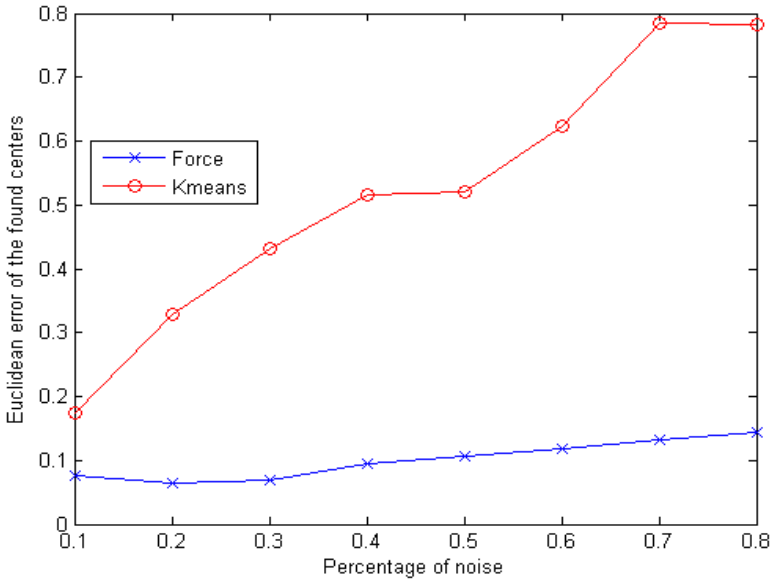


Figure 10. Euclidean error distance between the ground truth centroids and the final centroids using K-means and Force under linearly increasing noise level.

The second test set examines the behaviour of both methods under the condition where the mean value of the noise distribution varies. In this test, two levels of noise (30% in Figure 11-left and 40% in Figure 11-right) have been simulated. At each level, the mean location of the noise distribution is moved from -0.7 to 0.7 in the data space, while the noise samples still maintained a uniform distribution around the mean location within the [0,1] interval. Figure 11 depicts that around zero distance, the error is very small for both methods. However, as the noise distribution center moves to either sides, the error increases significantly and rather quickly for K-means. The Force seems to handle the additive noise up to 30% gracefully. At 40% additive noise (Figure 11-right) only when the noise distribution center moves to +/-0.5 away from the real data mass centroid, the Force's error becomes also large.

3.1.5 . Discussion

The basic assumption in the Force algorithm is that the equilibrium state (at which centroids will not move any further in the data space) represents the end of data clustering process. There are special cases that this might not be true. For instance, in the case of one negative data and two positive centroids with an equal distance from the

negative data the equilibrium has achieved but the data is not clustered properly. Future consideration can be made by adding randomly chosen small but unequal offsets to the centroids locations at the equilibrium. The Force algorithm is run one more time until the equilibrium is reached again.

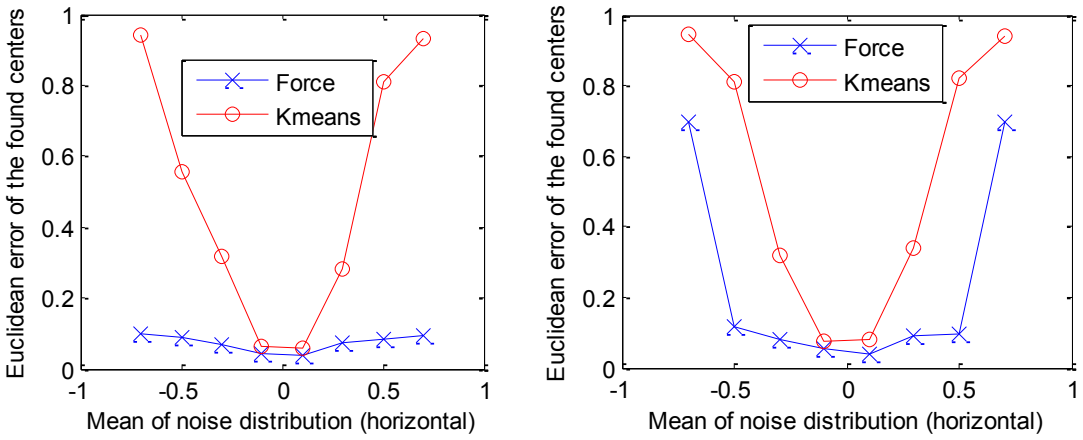


Figure 11. Mean location of noise distribution varies between -0.7 to 0.7: 30% noise (left), and 40% noise (right)

3.2 Application of Force in Gray Scale Image Segmentation

In this section we present an application that utilizes Force algorithm for image segmentation purposes [53]. Here we modified the original Force algorithm to be applicable for variable charge datapoints (the original Force assumed fixed charge). The example application in this section is an automated segmentation of the abnormalities such as brain tumors in MRI (Magnetic Resonance Imaging) image slices.

While detection and identification of abnormalities such as tumors is possible by experts, manual segmentation is usually tedious and time consuming [33] [37], and subject to error [34]. There are many methods that find a tumor in MRI images semi-automatically [47] [48]. In such methods, human intervention is required, which makes the process time consuming, expensive, and subjective to human error. The objective of this section is to present an automated unsupervised method for finding tumor regions

and their boundaries (high-grade gliomas) in slices of T2 FLAIR[1] MRI of Brain (no enhancements by contrast agent is done).

Usually one important characteristic in differentiating tumor tissues from the healthy tissues, which are gray matter (GM) white matter (WM) and cerebrospinal fluid (CSF), is the difference in intensity level. Relying only on the intensity value for this identification is usually not enough. The spatial information available in clusters of pixels that form a tumor could add to the robustness of the detection process.

The proposed method for segmenting brain tumors is comprised of three tasks: pre-processing, coarse detection of the tumor area, and fine detection of the tumor boundaries. Force is applied to a data set that is created from a pre-processed brain MRI image slices to find the center of clusters of pixels with higher intensity values. Once the clusters are found, further analysis is performed to identify those brain clusters that contain the tumor. In the last step, the identified tumor cluster is refined and the tumor boundaries are determined.

There has been significant effort to develop automated computer algorithms for identification and localization of tumors in brain MRI images. In general there are two classes of image segmentation methods: 1) supervised methods that require a training process using expert labelled data or ground truth, and 2) unsupervised methods that require no prior training. The method presented in this section falls into the group of unsupervised category.

Among supervised methods, the work in [34] combined information from registered atlas template, and user's input to train a supervised classifier. The method in [35][36] detected tumors based on outlier detection and affine transformation for registration. [36] proposed a method that utilized healthy brain images instead of tumorous images. To recognize deviations from normalcy, a multi-layer Markov random field was

[1] FLAIR: Fluid attenuated inversion. There are 3 basic scans of MRI: T1, T2, T2* weighted and spin density weighted. FLAIR is from specialized MRI scan types.

incorporated. In [32], the authors employed an atlas based pathological segmentation using affine transformation. Supervised methods are generally time consuming and require expert input for large training datasets. They also exhibited problems with reproducibility [30] which translates into different final results for different runs of the algorithm.

Unsupervised methods could potentially be faster than the supervised method. Unfortunately, current state of the arts for this group of works is not very efficient or accurate. For example, the work reported in [37] divides the image into few blocks, and calculates the number of edges in each block. It assumes that the tissues abnormalities occupy less that 10% of all pixels, and the blocks containing tumor pixels exhibit fewer edge pixels. Clearly large tumors could fall into different blocks which causes the algorithm to miss parts of the tumor. In another method presented in [38], color-based clustering is utilized. The MRI image is first transformed into the L*a*b* space and then K-means clustering [24] is used on the histogram of a* and b* planes to dynamically estimate thresholds and mark the tumor regions. The main issue with such methods is that they rely on intensity level classification, which is susceptible to misclassification. Also often methods based on unsupervised learning (such as [37] [38]) the spatial information is ignored.

Our proposed approach for tumor image segmentation is to use an unsupervised data clustering algorithm for initial estimation of the tumor region. Here, instead of classification based on the image histogram, both intensity level and spatial information are used. Also, the use of clustering/classification is not to directly determine intensity thresholds for segmentation but to use Force for finding the region of interest (ROI) (a rough estimation of the tumor's location).

Since tumors are more condensed than normal tissues, they reflect to brighter points in the image. Therefore, tumor detection mainly relies on finding clusters of pixels with different color/intensity values than their surroundings. The Force algorithm can be modified for this purpose. In the modified version, the assumption is made that each pixel of the image has a negative charge with charge magnitude set to the intensity of that pixel

raised to the power of k (here k is 4). Thus, if intensity of pixel i is υ_i, equation (4) can be rewritten as:

$$\vec{F}_j = \sum_{i \neq j, i \in D \cup C} \frac{Q_j \upsilon_i^{\,k}}{R_{ij}^{\,2}} \frac{(c_j - p_i)}{\|c_j - p_i\|} \quad Where \quad R_{ij} = \begin{cases} \|c_j - p_i\| & \|c_j - p_i\| > R_0 \\ R_0 & \|c_j - p_i\| \leq R_0 \end{cases} \tag{7}$$

Generally the intensity difference between tumor regions and their surroundings is not significant. Here, exponentiation is used to exaggerate the intensity difference between tumor and normal soft tissues, helping a faster and more accurate convergence for the Force algorithm. In the modified Force the original centroid position are updated according to equation (5).

3.2.1 Segmentation Steps

In this work, we first attempt to roughly identify regions containing the tumor, and then refine the tumor borders more precisely. Thus, our approach consists of two main levels: coarse detection and precise refinement. Coarse detection is done using Force. Precise refinement employs histogram analysis, thresholding, and region growing. The flow diagram of the proposed method is presented in Figure 12. Three main processing components of the proposed method include:

1. Pre-processing for clustering:
    - Skull removal,
    - Coarse removal of un-necessary information.
2. Finding the region of Interest (ROI) (Coarse Detection):
    - Force data clustering to find N clusters,
    - Determining ROI cluster.
3. Finding the boundary of the tumor in the ROI (Precise Refinement):
    - Histogram analysis and thresholding,
    - Region growing to identify tumor regions.

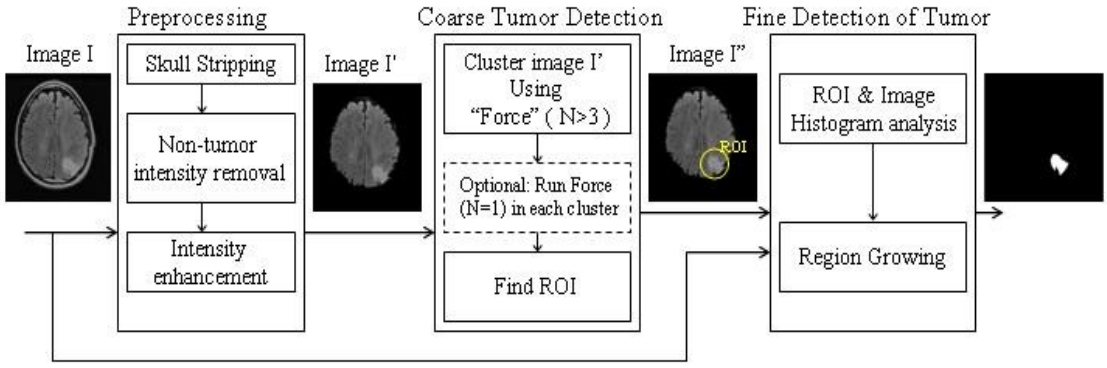


Figure 12. Proposed tumor detection procedure

3.2.1.1 Pre-processing

Before the coarse detection step, pre-processing is performed to remove the skull and some non-tumor pixels from the input image. Since, skull consists of very high intensity values in T2 images, usually it causes suppression of other intensity values. Therefore in the pre-processing stage, first the edges of the skull are found using Canny edge detection method [41]. Then, the area belonging to the skull that is contained within two strong edges are removed as shown in Figure 13.

It was noticed that sometimes it is better to remove some obvious non-tumor points which can be found using histogram analysis approach. Assuming that the tumor is smaller than the healthy tissue, we can remove pixels with intensity value below the histogram peak, since tumor is always brighter than brain soft tissues in T2 images. A similar gross separation of tumor and non-tumor pixels is also reported by [33]. For example the threshold can be the value of peak multiplied by 1.0 or 1.1 (we used 1.1). The resultant image is passed to the Coarse Tumor Detection process.

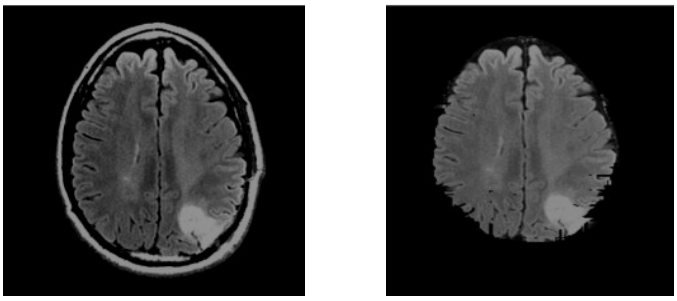


Figure 13. Original MRI of the brain (left); after skull stripping (right)

3.2.2 Force for Coarse Detection of Tumor

The resultant image from the pre-processing step is used to identify clusters of high intensity pixel values using Force. For this purpose, we release N (usually a value greater than 3), here N=4, centroids randomly in the image. Some of these centroids will move to clusters of distinctive pixels. If enough centroids are used, after running Force each true cluster attracts one centroid. If there are large scattered masses of data (which do not look like a cluster), some centroids may be associated with them. The remaining centroids that are not attracted by a specific cluster will end up outside of datapoint masses (brain) where the force effect from all other centroids is in equilibrium. The location where the repelled centroids end up is where the sum of electrostatic force from other centroids (repelling) and datapoints (attraction) is near zero. Note that a repelled centroid has at least a small positive charge (the initial small charge of 0.001), even if it has no datapoint in its cluster. Therefore, it moves until it reaches a point where attracting and repelling forces are almost equal. An example of such case is shown by centroid number 4 in (Figure 14-left) which has ended up outside the brain. After reaching the equilibrium, the charge of each centroid is equal to the total charge of its cluster points. This information can be used to remove centroids that are outside the data area and are not associated with high number of datapoints. An example of running Force on a pre-processed image is shown in Figure 14.

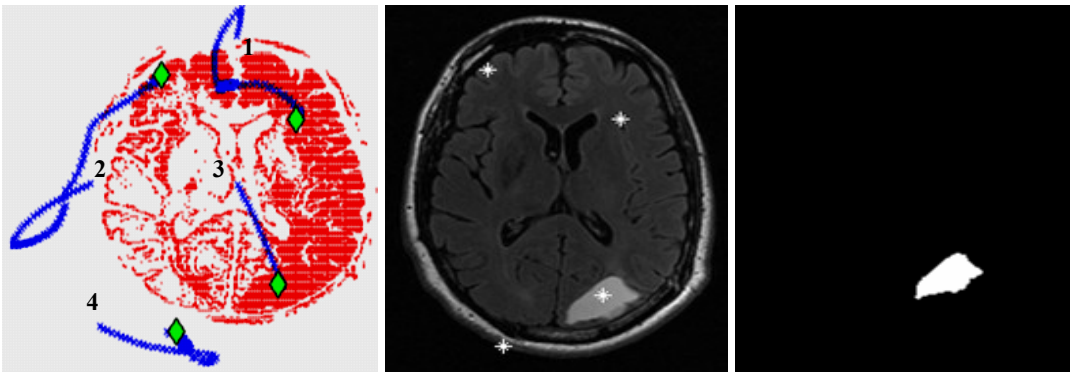


Figure 14. Clustering result on preprocessed image, the forth centroid is repelled (left), centroids on original image (middle); tumor boundary after region growing (right)

Here four centroids were released in random positions. They were gradually attracted towards data masses of either high number of datapoints or high intensity values (charges). Generally there are two conditions that may result in high charge condition for a cluster centroid: a vast cluster with each datapoint having a low charge or a small cluster but each datapoint having a high charge. Note that high charges for each datapoint and for centroids originate from high intensity values that are increased even more by the power of K operator. In Figure 14-left, two initial centroids were attracted toward the tumor which was high in the number of points and charges. Later the centroid number 3 that moved faster toward the tumor position repelled the centroid number 4. Since there was no large data mass nearby the repelled centroid stopped outside the brain area. The two other centroids were attracted towards other data masses. As shown in Figure 14-left, one of the centroids stopped inside the tumor; this centroid provides some rough information about the area where the tumor is located.

To ensure the robustness to imperfections that often exist in the pre-processing phase, the remaining steps of the algorithm (after clustering) use the original image and not the pre-processed resultant image.

3.2.2.1 Finding Region of Interest

After clustering, the next step is to identify which of these ***N*** centroids belongs to the tumor region. For this, few methods have been tested. With the assumption that the tumor

regions consist of high intensity values, a simple approach would be to select the center from the location where the intensity is at its maximum. The problem for this approach is that there is always a possibility that some small but very high intensity pixels exist in the image (after the pre-processing). These little spikes could attract one or more centroids to themselves.

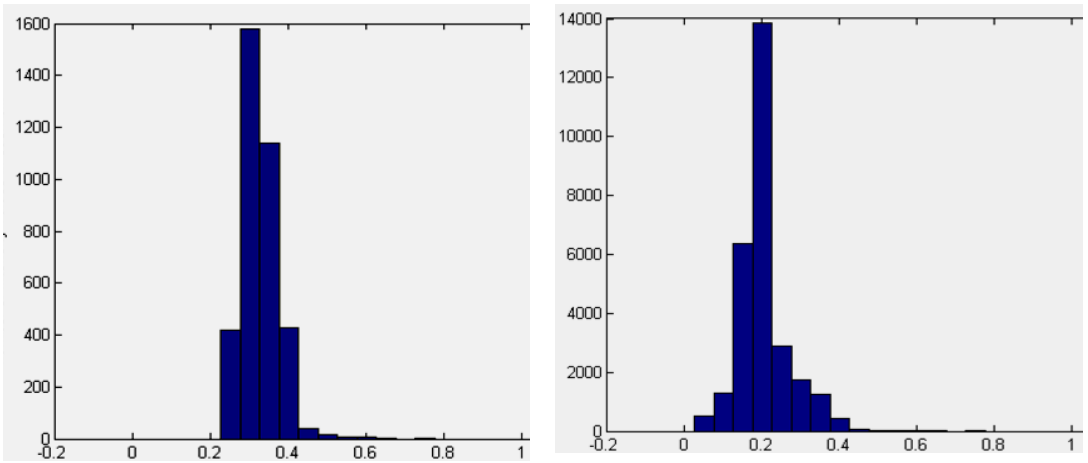
Another method would be to choose the centroid with the highest charge. The centroid's charge is proportional to the total intensity of its corresponding cluster. It is expected that the region containing tumor have the highest charge due to its brightness and size. However, sometimes scattered big clusters of low intensity pixels (soft tissues) form a cluster with a total charge higher than the tumor region.

Therefore it seems natural to combine the two above strategies and select the centroid with the highest total intensity with a limited spatial distribution. Here 5 to10% of the image size is chosen as the maximum area of a tumor region. This is in accordance with the general ratio between tumor and non-tumor regions in most T2 FLAIR MRI images of the patients with brain tumors.

3.2.2.2 Fine Tumor Detection

Tumor pixels are geometrically connected, but generally when moving from center toward outside of a tumor the intensity values decreases. To identify the boundaries of the tumor in the ROI, we use region growing. For this purpose, we estimate the threshold between the tumor and its surrounding using histogram analysis.

Since we already have a rough estimate of the tumor's location from previous step, the histogram analysis of this region is significantly less prone to misclassification of intensities than the histogram analysis of the entire image. First, a rough histogram of the tumor, considering a limited neighborhood (5% image dimension) of the centroid in the ROI, is estimated. Then the histogram of the brain soft tissues after removing the skull is generated. Figure 15 shows an example representation of these two histograms.



Intensity Histogram of tumor, bin size=0.05 Intensity Histogram of soft tissues, bin size=0.05

Figure 15. Rough histogram of the tumor (left), histogram of the brain tissues (right)

The difference between the peaks of the two histograms is found and their midpoint is used as the threshold. After estimating this threshold, the intensity value of each neighbouring pixel is checked, if it is above the threshold, it will be kept as a tumor candidate otherwise it will be removed. One may use the Otsu method [75] for finding the peaks and valley of these two histogram in order to estimate the proper threshold value. Later the 8-neighbor connectivity constraint is checked; if connected then that candidate is kept and otherwise discarded. The resulting area is declared as the tumor region, and the tumor boundary is consequently identified; this is shown in the right most image of Figure 14. In cases where cancerous tumor consists of multiple disconnected pieces, the above algorithm could only find the largest piece.

3.2.3 Segmentation Performance Evaluation

The performance of the proposed method has been evaluated on different MRI slices of 13 patients from the database images of University of California at San Francisco. Some of these results are depicted in Figure 16. Despite some difficult cases, our algorithm is capable of successfully identifying the tumor pixels. Comparing our results with the ground truth marked by the experts, we can see how well the proposed algorithm performs specially for parts of the tumor region where the intensity differences between the tumor and the surrounding tissues are low. These results show the robustness of our algorithm in handling different image intensity levels with dynamic ranges and different

tumor shapes and sizes (Figure 16). Later in this section we present the quantitative comparison between the ground truth and results found by our method.

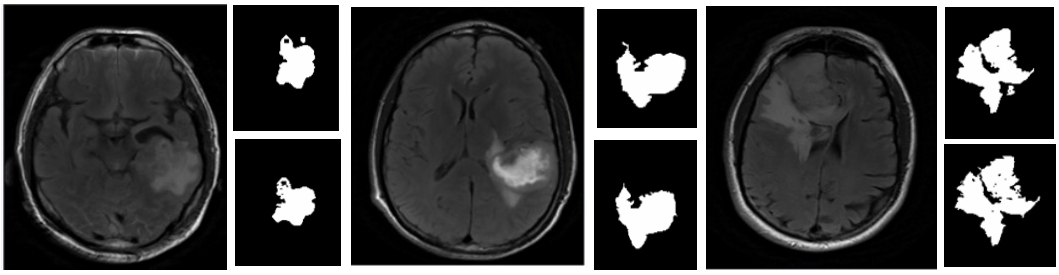


Figure 16. For each set, the original image is on the left; Top mask is the ground truth, bottom mask is the result of the proposed method

3.2.3.1 Robustness With Respect To Noise

To further evaluate the performance of the proposed method, noisy images are processed. Gaussian white noise, zero mean with variance of 0.02 has been added to the test images. An example is shown in Figure 17. The tumor is in a cluttered area and it is hard to see its true boundaries. Force is able to detect the tumor area, as one of the centroids moves inside the tumor. Using fine tumor detection and region growing the tumor area is identified more precisely.

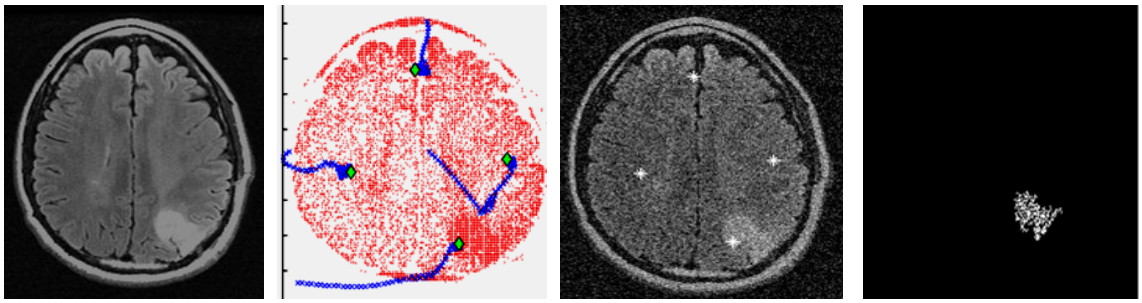


Figure 17. Original image; Force result; centroids on noisy image; tumor pixels

3.2.3.2 Comparison of Force with the Ground Truth

To investigate the accuracy of the proposed method, results are compared with the ground truth (manually marked tumors by experts). A number of slices are selected with different tumor shapes and sizes from different patients. Results in Table 1 are selected

from 100 slices that contain various conditions. In this table, three quantities are shown: True Positive (TP), False Positive (FP) and False Negative (FN). TP represents tumor points that are also identified as tumor points by this work. FP identifies those points that are identified as tumor point but in reality they are not. FN highlights those missing tumor points that the algorithm has missed to identify. The fact that a high percentage of TP is reported verifies the success of the algorithm in correctly identifying the tumor region. The presence of some FN and FP pixels (e.g., Slice24p9 or Slice24p13) imply that the fine detection step can be enhanced further.

Table 1 Results of our proposed method compared to ground truth (ratios, out of 1)

Image	TP	FP	FN.	Image	TP	FP	FN.
Slice28p1	0.920	0.056	0.079	Slice33p7	0.873	0.117	0.127
Slice35p1	0.900	0.065	0.099	Slice24p8	0.947	0.135	0.052
Slice25p2	0.876	0.007	0.124	Slice24p9	0.778	0.055	0.221
Slice35p3	0.901	0.144	0.098	Slice31p10	0.938	0.051	0.061
Slice24p4	0.939	0.110	0.098	Slice27p11	0.920	0.006	0.079
Slice18p5	0.959	0.041	0.040	Slice31p12	0.815	0.028	0.184
Slice26p6	0.967	0.072	0.032	Slice35p13	0.827	0.027	0.172
Slice38p6	0.933	0.024	0.066	Slice24p13	0.964	1.539	0.035

The proposed clustering algorithm can be employed in 3D space with no change. Therefore, the entire tumor detection approach is extendable to the 3D space. Moreover, the proposed clustering algorithm can be used for detection of tumors in other organs such as breast, liver and kidney. An example application of Force for detecting cancerous masses on a Breast MRI image is shown in Figure 18. This image is an exceptional case in which there are too many cancerous masses. To find these points, a large number of centroids were released to catch all the candidate cancerous masses.

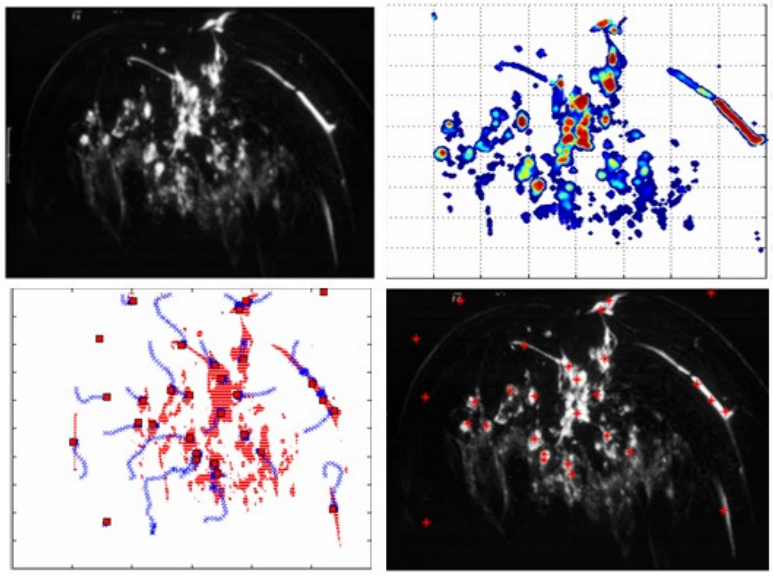


Figure 18. Effect of Force on finding cancerous point of breast MRI

From the above results and discussions it can be concluded that the proposed automatic method is capable of correctly identifying tumor locations in all test cases. The identified tumor boundaries are also relatively accurate and perhaps more sophisticated processes can improve the accuracy even more.

3.2.3.3 Remarks

The work in [37] states that under different clinical circumstances their work is capable of identifying tumor regions in T2-weighted images. Also the work presented in [38], based on L*a*b* color space and K-means clustering can successfully identify tumor regions. While both of these works ([37] and [38]) show limited examples (visual) of output results they fail to represent any quantitative or comparative results.

3.3 Application of Force in Color Image Segmentation

The purpose of this section is to show how Force data clustering approach can be utilized in segmentation of cluttered scenes, and further be used for detection of objects in cluttered scenes (details are presented in Chapter 4 and Chapter 5).

As mentioned earlier, detection of an object (called model object) in a cluttered scene involves finding candidate regions of the scene that comply with the model object

through specific rules and measures. These candidate regions can be further assessed using shape recognition algorithms to determine whether they have specific shape properties. In the previous section, we explained how Force could be used for segmenting medical gray scale images. Here, we describe the application of Force on color images.

The application in this section is only one example in identifying specific regions in color images and could be replaced by other methods such as color co-occurrence based method [43] or other color based segmentation such as the one presented in [44]. In our case, since we already have an image of the model object in the database, we can use the following proposed approach to find the ROI.

Given a sample image (view) of the model object, our objective is to process a cluttered scene image to find the location of one or more instances of the model object. Figure 19 shows a scene image with a model object (red box).

Figure 19. A typical scene image and the model object image.

3.3.1 Filtering the Scene Based on Color

For finding candidate ROI, the color information of the model object is used. First the RGB image is transformed into the L*a*b* [2] color space. Here the information of

[2] CIE L*a*b* (L*a*b*) was created to be an independent device model to be used as a reference. L* in this space represents the lightness (L* = 0 indicates black and L* = 100 indicates white), a* is the position between green (negative value of a*) to red (positive value of a*) b* is the position between yellow (negative value of b*) and blue (positive value of b*).

channels a* and b* of the object image are placed in a 2D space (Figure 21 for the objects shown in Figure 20). Then clusters of colors are found in the a-b space using the Force clustering algorithm. Here, two main colors are considered, since many objects have only 1 or 2 predominant colors but this number can be increased to include more colors.

Figure 20. Two model object samples

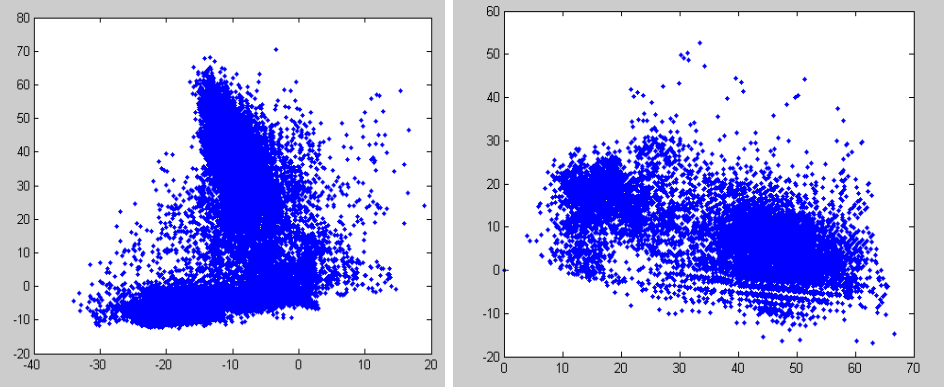


Figure 21. The distribution of a* and b* values for objects of Figure 20 (left: yellow box, right: red box)

By running the Force algorithm on the model object with 2 centroids, the center of clusters on the 2D space (on a* and b* spaces) are found. These cluster centroids represent the predominant colors of the objects. After finding the center of clusters, a threshold (for example, ±7) is placed around each centroid to create a color filter. For instance, if a pixel has a color that is within a threshold distance from the centroids, it is considered as a datapoint and otherwise it will be discarded. The same filter is also applied to the scene of interest. Figure 22 shows the result of this filtering procedure (using color of the model object in Figure 20) for the scene image in Figure 19.

Figure 22. Scene image after filtering out irrelevant pixels using model object color filter (shown by white pixels).

3.3.2 Candidate Regions Identification

After filtering the scene image using colors of the model object, a number of image pixels are identified as potential candidate regions that may in fact belong to instances of the model object in the scene. Since these pixels create clusters in the image space, a 2D clustering using Force is applied on them (there is no need for color any more). Force has the advantage that there is no requirement for knowing the exact number of seed points in advance compared to the K-means method. With K-means, if the number of centroids is more than the actual clusters, a cluster may attract more than one centroid and therefore it will be divided into more than one piece. With Force, a large number of centroids can be used and since Force repels the extra centers outside of the cluster regions (as shown in Figure 23) all extra centroids with no significant clusters will be eliminated**.**

After finding the center of each candidate region (cluster), a morphological dilation is utilized to clean up and fill potential gaps in each cluster. Since the scene image is acquired using an ordinary digital camera, it includes some noise that projects as scattered points. After dilation, the clusters have a cleaner shape and a rectangular wireframe is fitted around each cluster Figure 24.

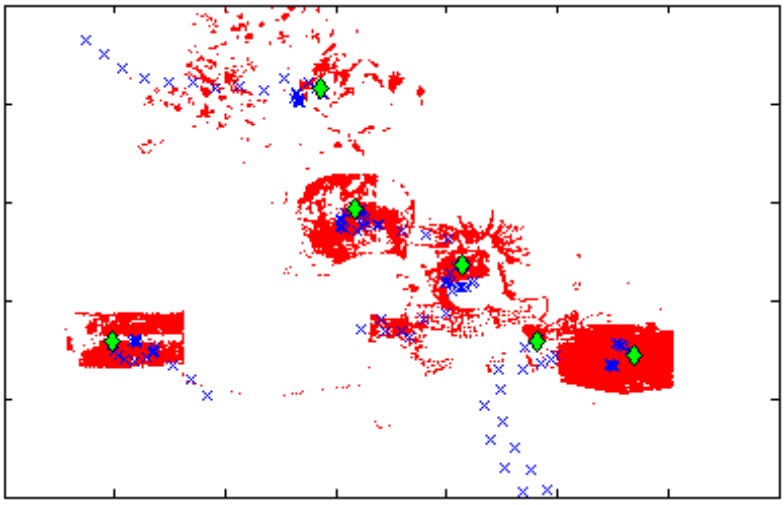


Figure 23. Force data clustering algorithm finding clusters of pixels with specific colors.

Figure 24. Wireframe rectangles are fitted to each found cluster.

3.3.3 Candidate Region Size Adjustment

After the filtering step, candidate regions can be analyzed using a shape recognition procedure (such as procedures presented in Chapter 4 and Chapter 5 of this thesis). An equal scale factor between the candidate scene regions and the model object is an essential factor in the quality of shape matching process. As shown in Figure 20, there are two specific model items that the search is carried for.

In order to match the candidate regions with model object's image, a rectangular wireframe is fitted to the image of model object (Figure 25). Dimensions of this rectangle are then used for scale alignment between the ROIs in the scene and the model object image.

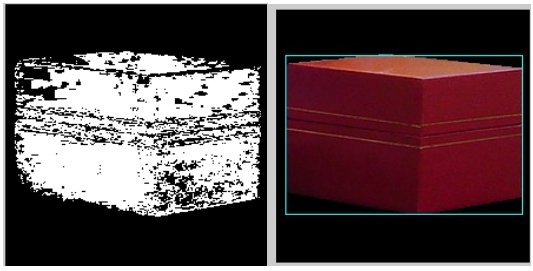


Figure 25. Model object and its color filtered version.

The filtered image of the model object (Figure 25-left) is used to distinguish and separate the object's region from its background. Although the recognition algorithm is not scale invariant, using this information and the dimensions of the model object could add to the robustness of the system with respect to the scale variations that naturally exist between the object images in the scene and the model object.

In order to scale align candidate object regions in the scene with the database images, the maximum length of the object in the scene and in the model object in x and y directions (equations (8) to (11)) are calculated.

$$length_{x\_obj\_scene} = max\ (scene\_obj\_x) - min\ (scene\_obj\_x) \quad (8)$$

$$length_{y\_obj\_scene} = max\ (scene\_obj\_y) - min\ (scene\_obj\_y) \quad (9)$$

$$length_{x\_mod\_obj} = max\ (model\_obj\_x) - min(model\_obj_x) \quad (10)$$

$$length_{y\_mod\_obj} = max(model\_obj_y) - min(model\_obj_y) \quad (11)$$

The scale factor and the starting point coordinats are computed by:

$$scale = max\ (length_{x\_obj\_scene}/length_{x\_model\_obj},\ length_{y\_obj\_scene}/length_{y\_model\_obj}) \quad (12)$$

$$S_{x\_roi} = -scale.S_{x\_mod\_obj} + S_{x\_obj\_scene} \quad (13)$$

$$S_{y\_roi} = -scale.S_{y\_mod\_obj} + S_{y\_obj\_scene} \quad (14)$$

In equation (13), $S_{x\_obj\_scene}$ is the starting point in the candidate region in the scene in x direction and $S_{x\_mod\_obj}$ is the starting point of the object in the model object image in x direction; equation (14) repeats the same definition in y direction. Using these two equations, the starting points of ROI for each candidate region in the scene image is calculated. Through equations (15) and (16) the size of ROI in direction x and y is calculated.

$$length_{x\_scene\_roi} = -scale \times length_{x\_image\_mod\_obj} \quad (15)$$

$$length_{y\_scene\_roi} = -scale \times length_{y\_image\_mod\_obj} \quad (16)$$

Here $length_{x\_image\_mod\_obj}$ is the length of the model object image in the x direction. The final result is shown in the Figure 26. After this process, the resultant candidate regions are processed by a shape recognition algorithm for further assessment (details are described in Chapter 4 and Chapter 5).

Figure 26. Drawing a green box around borders of each region

Chapter 4 SHAPE RECOGNITION USING FORCE

In this chapter, a shape recognition algorithm based on Force clustering method is presented. This algorithm extracts features from a model object image, and compares them with those of unknown objects by means of a similarity score. The database object with the highest similarity score is identified as the database representative of the model object. In the next sections, first a feature detection method is described, and then a strategy is presented on how the similarity score is computed. In this work, the COIL-20 database images (20 objects each with 72 views, with a total of 1440 images) is used to evaluate the performance of the proposed method.

4.1 Feature Extraction using Force

The modified version of the Force clustering algorithm (as described in Section 3.2) could detect clusters of high intensity in MRI images. Here we propose to use the centroids of image clusters of specific characteristics as image features. Our proposed method in this section is based on gray scale images. It, however, can be extended to color images. The algorithm has two main steps:

1- Transform the image into a space where each pixel has a single value (e.g., greyscale). For color images each one of the bands could be considered here or a mean value of the three RGB bands could be used.

2- Run the Force algorithm with K centroids (usually $K > 4$). Here the pixel values serve as datapoint charges.

After running Force, the K identified centroids represent features of the image which can be interpreted as local centers of gravity of the image clusters. An example is displayed in Figure 27. Here K was set to 5 for the left image and 15 for the right image. It can be seen that in the case of K equal to15, four centroids (centroid number 2, 4, 12 and 15) are repelled from the datapoint areas. These centroids have very small charges

and can be removed. This makes the number of valid centroids variable. Therefore, in order to have a constant number of centroids for each image, we suggest setting the value of K to a small number (3-5, depending on the complexity of the image). In the experiments for COIL-20 database [19] *K* was set to 5.

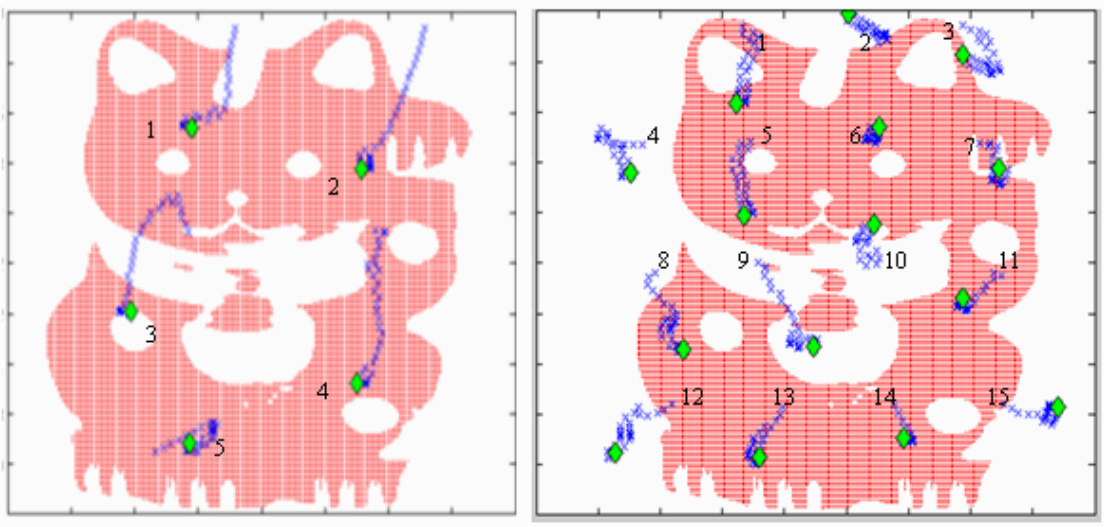


Figure 27. Centroids found by Force for *K=5* and *K=15*

To measure the similarity of image features, one way could be to use their spatial information. Therefore locations of these features are compared against the locations of their corresponding features in the other image. Another way could be to utilize the relative locations or angles of specific number of centroids in each image and then measure the similarity of these features as a group with their correspondences in the other image. In the next section we propose three different approaches to compare features of the two images and to establish matching between a model image and images of unknown objects from COIL-20 object data base.

4.2 Feature Comparison

Here we discuss two different methods and evaluate each using the COIL-20 database images. To prevent any bias towards known images, image used as the model object image is removed from the test set. This removes the possibility of identifying the exact same view in the database and therefore allows testing the algorithm with views that are at least 5-degree apart from the known model object view. The test for each object is

carried out against all other images in the database (a total of 1439 images for each case). The three proposed methods are location-based, edge-based, and angle-based methods. The first method is not invariant with respect to the scale or rotation, but the two other methods are.

4.2.1 Location-Based Feature Comparison

In this method, the distance of the centroids' locations for the model object image and database images are compared. The view with the smallest sum of all distances is considered as the most similar view to the model object. Here, the closest neighbour constraint (between the centroids of the two images) is utilized to associate corresponding centroids between the two images. If the centers are matched exactly, their distance and therefore their sum will be zero or very close to it. The more similar the two images are, the smaller the difference measure will be. The sum of differences for features of image *I* and image *J* is computed by:

$$diff(I,J) = \sum_{i=1}^{K} \left\| c_i^I - c_i^J \right\| \tag{17}$$

The location based feature comparison was evaluated for objects in COIL-20 database using the above difference measure. The results are shown Table 2. From this table, the mean recognition rate is 78%. The lowest recognition results belong to objects 3 and 6 with 51% success rate. A closer look at the COIL-20 database (Figure 28) reveals that the low score is indeed due to high similarity of objects 3, 6 and 19.

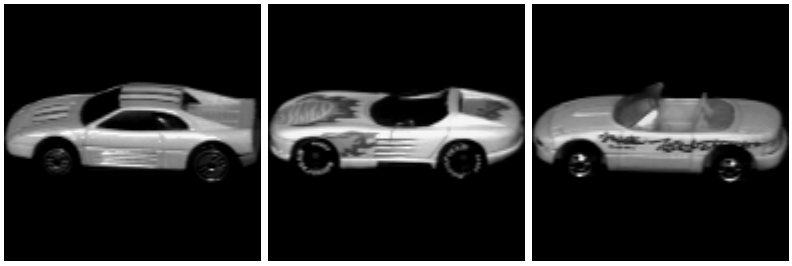


Figure 28. Objects 3, 6 and 19 from COIL-20 database

Table 2. Location-Based Feature Comparison Results

Object No.	Success rate	Object No.	Success rate	Object No.	Success rate
1	0.8333	8	0.9861	15	0.7500
2	0.8611	9	0.7361	16	0.6806
3	0.5139	10	0.8472	17	0.8056
4	0.8750	11	0.9722	18	0.8056
5	0.6111	12	0.5833	19	0.6944
6	0.5139	13	0.8194	20	0.8889
7	0.9167	14	0.9028		

The main problem with this approach is that wrong matches (even if only one) could heavily affect the sum of distances and therefore cause the algorithm to dismiss the true match. For example, if from 6 centroids, 5 match well, but one matches with a wrong feature located at a far distance, the summation of distances will become large. This difference is dominated by one centroid and does not count for many good matches that do exist between the two feature sets. To address this issue, instead of adding the distances, we calculate the similarity measure using a function F that outputs a value of 1 for centroids located at the same location, and a value of zero for far centroids. Then the sum of these scores is used as the similarity score. Three potential definitions for F are presented as shown in equation (19).

$$score = \sum^{K} F(dist), \tag{18}$$

$$F = c.\exp^{-d} \quad or \quad F = c.10^{-d} \quad or \quad F = \frac{1}{1 + (c.d^{N})} \tag{19}$$

Here d is the distance, and c and N are constant values that are used for adjustment purposes and in this work they are set to c=700, N=3. The results of matching using the new score function in equation (18) for the 20 objects in COIL-20 database are shown in Table 3.

Table 3. Modified Location-Based Feature Comparison Score.

Object No.	Success rate	Object No.	Success rate	Object No.	Success rate
1	0.8889	8	0.9861	15	0.9028
2	0.9444	9	0.7917	16	0.7222
3	0.5556	10	0.8750	17	0.8056
4	0.9028	11	0.9722	18	0.7917
5	0.6389	12	0.7778	19	0.7778
6	0.4722	13	0.9167	20	0.9167
7	0.9306	14	0.9722		

The overall mean success rate is improved to 82% success. The location-based method relies on absolute positions; moreover, it is sensitive to "scale" and "rotation" and if the image is resized or rotated, it could easily fail. In order to resolve these issues, we should incorporate strategies that are not based on absolute location distance. For instance, we can use the relative distance or angle between a group of features for comparison purpose. The next section describes the proposed algorithm based on using connecting line segments (we call them edges) and the angle of those line segments with reference to x or y axis.

4.2.2 Angle Based Similarity Assessment

In an image with K centroids, there are ($K! /2!(K-2)!$) line segments (edges) that could connect each pair of centroids together. For an image and its rotated version, the lengths of the edges are not dependent on the absolute location of these centroids (Figure 29). This is under the assumption that the rotation has not caused a significant shape variation. Therefore, if the matching is performed using these edges, the matching method could inherently become rotationally invariant. Also, if all the length of these edges are normalized, the method could become scale invariant.

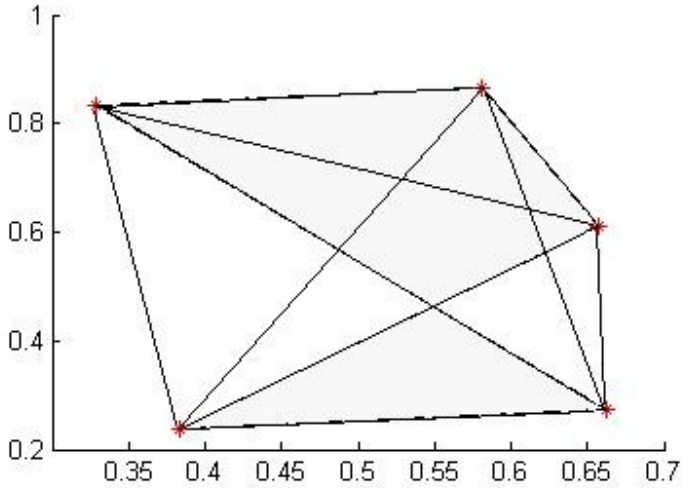


Figure 29. Centroids as red stars, edges are the lines between these centroids

Unfortunately, there are two problems with using edge lengths in computing the similarity measures between features. First, since typically there are several centroids and therefore many edges in an image, the correct association of edges based on only length turns out to be very difficult if not impossible. Second, the normalization of edge lengths may also become problematic. For instance a wrong centroid at a far distance can affects the normalized lengths and therefore increases the matching error. These problems make the comparison based on edges sensitive to errors in the centroid locations. To overcome these issues, we propose utilizing angles between edges in the comparison process.

The angle based similarity assessment relies on comparing angles formed between centroid edges. An immediate advantage of this method to the edge-based method is that it does not require edge length normalization. Therefore wrong or far centroids will not cause the algorithm to fail. Moreover, the angles naturally describe the shape of an image, regardless of the size or orientation (see example case in Figure 30).

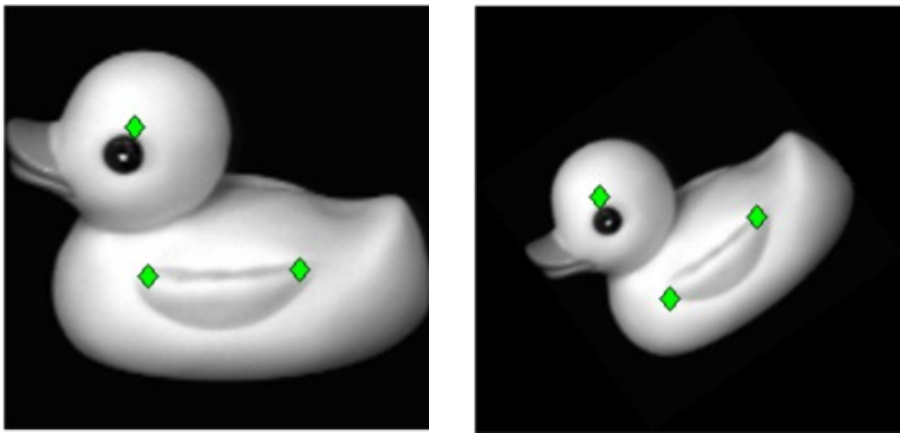


Figure 30. Result of Force on original (Left) and on 35 degree rotated and scaled down image (right)

This method could be scale and rotation invariant for objects with distinct clusters. If however an object does not have distinctive clusters, the method might not perform well under rotation. An object with less distinctive features is shown in Figure 31.

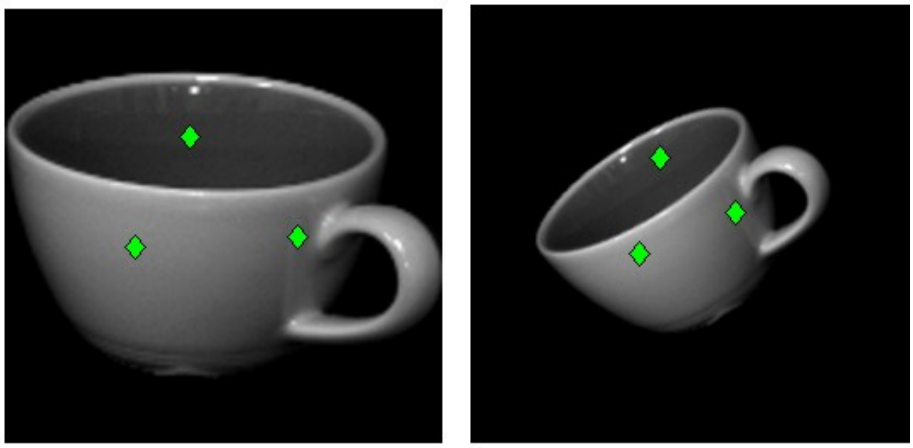


Figure 31. An example of a case where rotation invariance fails, since there are no distinct clusters in the image.

There are two ways of forming angles. The angles can be between each two edges, or between each edge and *x* or *y* axes.

The disadvantage of the first method is that there are too many angles and associating them in two images becomes very difficult. For example, the number of all angles formed

by edges of 6 centroids (15 edges) is 75. It is easy to match wrong angles together and match a model object with a wrong database object.

Therefore using the angle between an edge and x or y axis could simplify the task of comparison since each angle depends on two feature points instead of three. In the case where an image is rotated, the amount of rotation is added to all angles (as an offset). To associate angles in two images, we must devise a method that is capable of coping with such offsets.

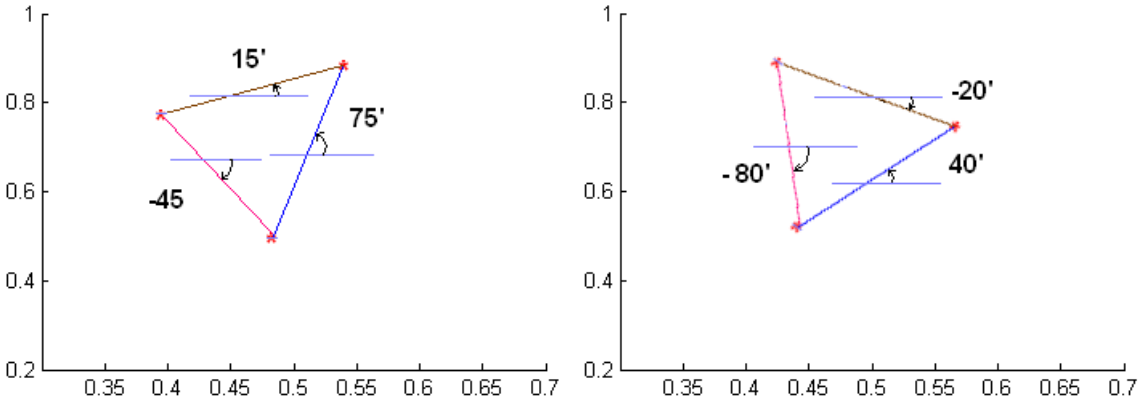


Figure 32. An example of angles for an image and its rotated version (35' rotation)

To calculate the set of angles (there are N angles for N edges), first the vector $V=(v_1,v_2)$ between each 2 centroids in one image is computed. Here $v_1 = x_{c1}-x_{c2}$ and $v_2 = y_{c1}-y_{c2}$. The angle between V and the x-axis is then calculated (Figure 32). Since here relative angle between the edge and the x-axis is computed (instead of the absolute angle between edges), it is required to estimate the rotation angle θ between the two images before any comparison between features of the two images can be done. The procedure to find the rotation angle θ between the two image using N edges is described below:

1. The angles between all the edges and x-axis are computed for the two images J and I and placed in sets α_j and α_i.
2. The difference between each angle of the 1st image and all other angles in the 2nd image is calculated and placed in matrix D $[N^2]$.
3. The histogram of all elements in D using B bins is calculated. In this work B is set to 30 with a uniform spacing in the range of (-180,180) degrees. A search

for the most populated bin is carried out. The center of the most populated bin is considered as the initial estimate of the potential rotation angle between the two images (θ_1). This angle will be subtracted from all angles of matrix *D*.

$$D' = D - \theta_1.$$

4. Step 3 is repeated using a finer resolution bin size for the corrected angles in D'. This is just to increase the accuracy of the rotation estimation process.

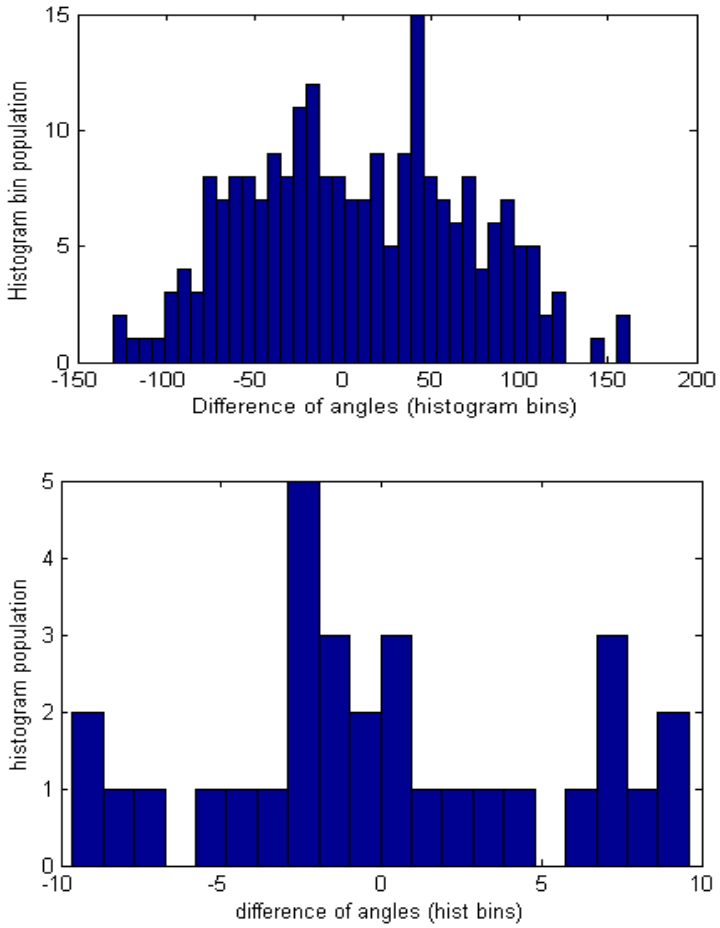


Figure 33. Histogram of difference of angles in 2 images, showing a max at around 40 degrees in the first plot (top), and a maximum at 5 degrees after first offset removal (bottom), second angle adjustment, showing a max at -3. The total adjustment is 37 degrees (compare to the 35 applied to the image)

The reason for incorporating step 3 is that if two images belong to the same object, a rotation of θ would cause N edges to be exactly θ degrees different. So there should be a

bin (where θ belongs to) with a population of at least N samples. Also because the difference between non-corresponding pairs of angles is randomly distributed, their bins will have lower populations than N. Therefore, a bin with the highest population most likely represents the rotation of θ. Figure 33 depicts example histograms for a 35 degree rotated and scaled version of two similar (but not identical) views of an object.

Since the precision of θ will be limited by the number of bins, this procedure could be repeated one more time after the initial deduction of θ from all elements of matrix D. The newly estimated angle (θ_2) is now expected to be closer to 0. This angle is deducted from all elements of matrix D'.

Now the corresponding edges can be found by looking at the angle values in the matrix (since compensation for rotation has been done already). To do this, in each row of the matrix, the minimum absolute value is found. The column index of this minimum identifies the matching edge in the second image corresponding to the edge of the first image associated with that row.

We can sum all these differences or apply function F on them to find the overall similarity measure:

$$S = \sum_{i \in all\ rows} (F(a_i^{min})) \tag{20}$$

Here a_i^{min} is the minimum value in row i.

The results for the angle based feature comparison are shown in Table 4.

Table 4 Results for angle-based similarity assessment

Object No.	Success rate	Object No.	Success rate	Object No.	Success rate
1	0.7917	8	0.8056	15	0.9444
2	0.6528	9	0.7222	16	0.9444
3	0.2917	10	0.6389	17	0.9861
4	0.6944	11	0.6944	18	0.5694
5	0.5278	12	0.9444	19	0.4444
6	0.2778	13	0.5556	20	0.8472
7	0.6806	14	0.8472		

The overall mean recognition is 69%. Note that the success rate is lower than the location based method, but the angle based method is in fact invariant with respect to rotation and scale. The location based approach for the above cases would completely fail because we compare the absolute location of centroids and if one image is rotated absolute locations will definitely change.

The performance of this method is evaluated under addition of Gaussian noise, zero mean and variances of 0.005, 0.01 and 0.02 (shown in Table 5). As the level of noise increases the recognition degrades. Figure 34 displays a sample image along with its degraded version.

Table 5 Recognition results under additive Gaussian noise

Noise variance value	Zero	0.05	0.01	0.02
Recognition Ratio	69%	50%	45%	40%

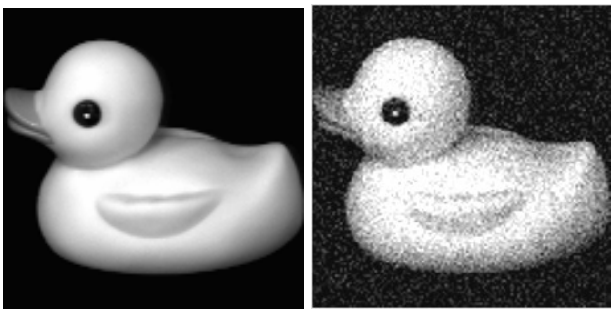


Figure 34. Original image (left), Gaussian noise; mean zero variance of 0.01 is added (right)

Chapter 5 SHAPE RECOGNITION BASED ON HIDDEN MARKOV MODELS

In this chapter we present a new method for shape recognition using Hidden Markov Models (HMM). A Markov model is a probabilistic process over a finite set of states, where the probability of being in any state, conditioned on the current state, depends only on the current state of the model and the transition probabilities (and not the previous states). If the states are hidden from a system observer, the model becomes a Hidden Markov Model (HMM). HMM is usually used when there are "sequences" of observed data generated by an underlying hidden phenomenon. Our initial intention was to incorporate the Force algorithm for detecting features and incorporating such features in Markov based shape recognition. Since Force is region based feature detection, unless the image has many different colors, the number of features at each view is between 2 to 6 features. Through implementation of this part of the project, we realized that unfortunately the HMM requires a large number of object features to be able to successfully identify that object. For that reason, we decided to switch the features from Force features to wavelet based features. Therefore in reading this chapter one might feel a bit of disconnection from the previous chapters. We thought that presenting these results and adding this method as a complementary method to our shape recognition method based on Force features (presented in the previous chapter) could still be useful in improving the final results of the recognition process.

Previous work using HMM for object recognition problem was presented earlier in this thesis Section 2.1.1. The proposed method in this chapter is inspired by the work in [15]. Here our main focus was to reduce the computational complexity of that work and address some of its remaining issues. The work in [15] proposed to use overlapped sub-images (50% or 75% overlap) of size 8x8 or 16x16 pixels and then used wavelet transform to produce sequences of coefficients as input data for a continuous-output HMM. The result is a sequence with length in the order of the number of pixels in the image.

In this chapter, we first present a brief overview of HMM concept and how it is related to our shape recognition problem in Section 5.1. Following the description of HMM, we describe the steps and configuration of the HMM based object recognition approaches in Section 5.2. These steps are then elaborated more in Sections 5.3, 5.4 and 5.5, which respectively describe feature extraction from images for creation of sequential data, design of an observation space from the extracted feature for training HMMs, and the topology of the HMM.

5.1 HMM Definition and Steps

An HMM has a finite set of states and in each time instance the system is assumed to be in one of these states (see Figure 35). At each time instance, the system transits from one state to another, based on time invariant transition probabilities. In each state, the system generates an output according to observation emission probability. The observed output is either from a finite discrete space, or a continuous space. The model may start from any initial state. The transition between states is at a discrete time step. HMM is specially used for modelling a sequence of observations such as voice samples. Application of 1-D HMM in image processing requires acquiring image sequences.

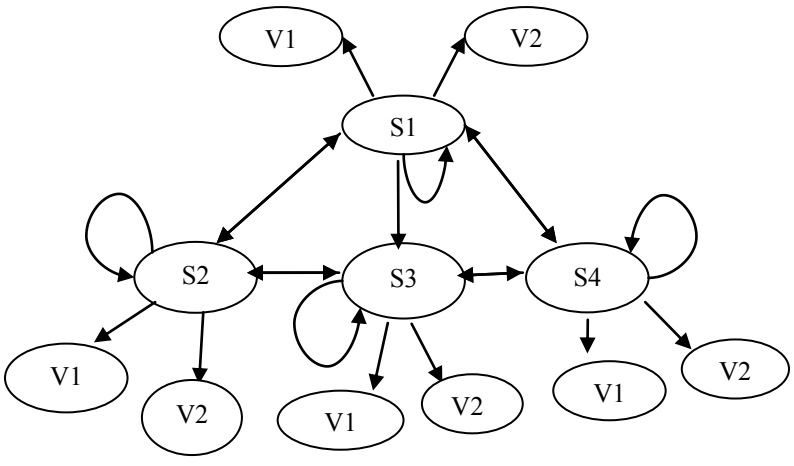


Figure 35. Hidden states: *S*, visible states: *V*; In the HMM, there is only access to *V*s

There are three main tasks associated with Hidden Markov Models: *evaluation, decoding*, and *learning*. The objective of the *evaluation* task is to find the probability of a

specific sequence being observed. For evaluation, often one of the forward or backward algorithm is used [11] [20]. The forward algorithm presents the probability of a particular observation sequence given any sequence of hidden states. The backward algorithm provides the same probability but the calculation is in the reverse order (starts from the end of the sequence).

In the *decoding* task, the most likely sequence of hidden states that could have produced a particular sequence of observations is found. This is similar to forward algorithm but here the objective is to find a sequence of states that maximizes the probability of a particular sequence of observations. For decoding usually the *Viterbi* algorithm [12] is used.

The *learning* task describes how the transition probability matrix (A) and the observation generation probability matrix (B) should be found using a set of training data. The initial state vector (π) is also found for the probabilities of being in any particular state initially. For the learning task, often a forward-backward algorithm called *Baum-Welch* is used [11].

Hidden Markov models could be used for object recognition purposes. For this, the required condition is that the extracted data must have some *sequential* relationships. This means that each element of the data is dependent on its previous elements (property of Markov models), and each sequence pattern must be different from each other but dependent on the underlying object that the sequence is generated from.

5.2 Shape Recognition Algorithm Setup

In the proposed shape recognition algorithm, we assume that a single HMM can be designed for each unique view of the object (image), or for multiple similar views (with small viewing differences) of the object. The general steps of the algorithm are as following:

Training:

1- Extracting features from model object image (known object),

2- Creating sequential data (Z) from features,

3- Training and producing an HMM model (A, B, π) using Z.

Testing:

1- Extracting features from the unknown objects,

2- Creating sequential data (O) from features,

3- Evaluation (finding log-likelihood) of the sequential data (O) with the HMM model (A,B, π) of the known object

The above steps describe that each image must be converted into a sequential data set that is used for training a specific HMM model. After training the HMM model for an image (or a set of views), this model is used to examine an unknown sequence (evaluation task) to estimate the likelihood of that unknown sequence being generated by the same image that the HMM was trained for. If the test (evaluation) is done for many sequences, the sequence with largest likelihood is declared as the most similar image to the training image. Figure 36 shows an example that depicts how the evaluation task works. For verification of the proposed algorithm, in all cases that are explained in this chapter, one HMM is trained for odd views of the database objects, and tested for all even views. This ensures the potential bias in the training database for odd views at the recognition time. In this work the COIL-20 database (Figure 2) is utilized. We only use the grey scale images in this chapter.

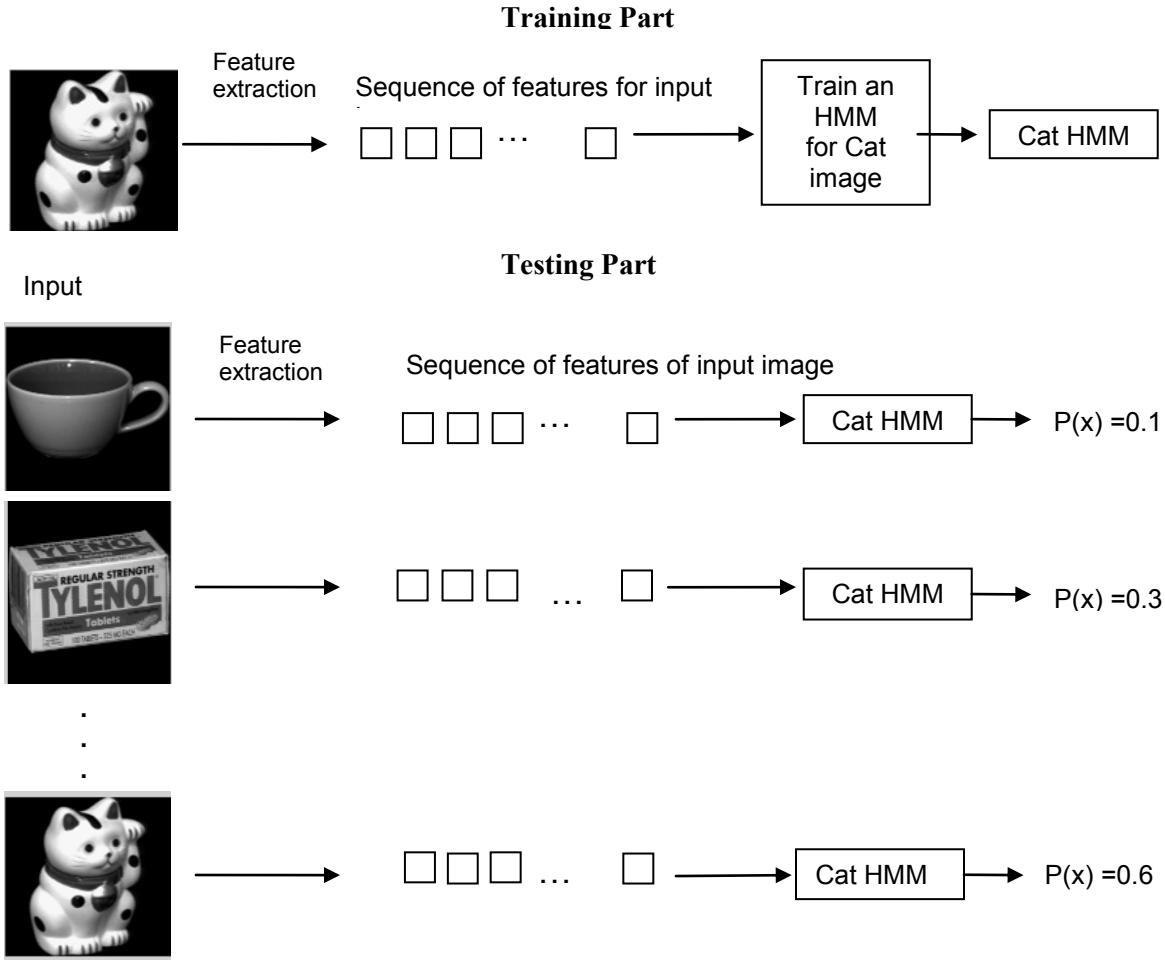


Figure 36. Using HMM in shape recognition.

The HMM utilized in this chapter is a discrete first order HMM consisting of N states and S hidden states defined by $S=\{S_1, S_2,..., S_N\}$. The observation sequence is of a length T and denoted by $O = O_1,O_2, ...,O_T$; the underlying hidden sequence is identified by $Q = Q_1,Q_2, ..., Q_T$. The transition matrix $A =\{a_{ij}\ 1<i, j <N\}$ represents the probability of transiting from a state S_i at time t to a state S_j at time $t+1$. The initial state distribution π, represents the probability of starting form state S_j. E Each row of a_{ij} represents the probability mass function for transition out of state i, and sums to one over all j's. Here:

$$0 < a_{ij} < 1, \sum_{j=1}^{N} a_{ij} = 1 \quad , for\ all\ i$$

If the observations are from a finite discrete space of size E ($V=\{ V_1, V_2, ..., V_E \}$), the observation matrix B holds all the probabilities for being at state S_j at time t and emitting symbol V_k. Here k is the index and is defined by $1 < k < E$.

For example, if an HMM has 16 states ($N=16$), $S=\{1, 2, 3, ...,16\}$, an observation sequence example could be like $O=\{a\ b\ d\ g\ g\ e\ f\ a\ c\ e\ d\ b\ ...\ d\ e\ a\ f\ a\}$ for instance of length T=100, which is generated from set $V=\{a,b,c,d,e,f,g\}$ with E=7. A sequence of states corresponding to the observation vector O could be like $Q= \{2\ 5\ 2\ 4\ 9\ 1\ 2\ 3\ 14 ...\ 4\ 3\ 10\ 2\}$, which will have the same length as O. The length of this vector is equal to the length of O (each observation is from one of the possible states in S).

An example of matrices A and B, for N=16 and E=7 is given below. Here the transition matrix is a 16x16 matrix since it represents the probability of moving from a hidden state to another one (a state can also repeat itself). The observation matrix B includes 16x7 elements, each showing the probability b_{jk} of emitting (observing) a specific output v_k from set V, while at state j (from set S):

$$A = \begin{bmatrix} 0.2 & 0.3 & 0.07 & \dots & 0.1 & 0.04 \\ . & . & & & & . \\ . & & . & & & . \\ . & & & & . & . \\ 0.1 & 0.09 & 0.05 & \dots & 0.4 & 0.3 \end{bmatrix} \quad B = \begin{bmatrix} 0.12 & 0.34 & \dots & 0.03 & 0.1 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ 0.19 & 0.02 & . \ . \ . & 0.4 & 0.02 \end{bmatrix}$$

The probability of starting the HMM from any state is vector π, also known as the prior. In this example π is 16x1. At each iteration in the training phase, A, B, and π are updated and usually after a few iterations they converge into stable values.

In order to have a model for an object, A, B, and π must be determined. If A, B, and π are known, the probability of an observation sequence can be estimated using the forward algorithm given the model. For example, if the HMM model is trained for the object "cat", we can use the forward algorithm to evaluate $P(O|\ Model="cat")$ for each sequence O of an unknown object images. Figure 36 represents an example of how the test phase would work.

Methods based on HMM require determining/setting a few system parameters before the actual training. Some of the general parameters for HMM are as following: the number of elements (set to T=256), the number of iterations (150 iterations in this work). For this work, Matlab's HMM library and Murphy's HMM code [22] are used. Setting of some other parameters are discussed in following sections of this chapter. In general, a proper observation space and HMM topology (connectivity and number of states N) are difficult to determine. Finding proper values for these parameters requires knowledge of the patterns in the data sequences. For example, if there are repeating pattern with length N we can expect the number of states to be N too. For this reason, we first look at how a sequence of data can be created using images in the next section.

5.3 Feature Extraction and Serialization

The first process before training an HMM is the feature extraction and creation of a data sequence O (observation sequence). This process consists of 2 tasks, feature extraction and serialization of the extracted features.

For the feature extraction, wavelet transformation is used; for serialization a zigzag fashion reading and serialization of the wavelet coefficients are chosen. Other methods for both feature extraction and serialization may be used; however since the wavelet transform (WT) and wavelet coefficients convey shape information at different levels, a zigzag scan of these coefficients can preserve the similarity of neighbouring coefficients better.

5.3.1 Using Wavelet Transformation for Feature Extraction

Wavelet transform can be applied on the entire image or small image blocks or patches. In general, wavelet transforms capture both frequency and location information. To keep the samples more correlated, some have used overlaps between image blocks [15]. In this work, the wavelet transform is applied on the entire image at multiple levels. Also, low-pass band coefficients are utilized since they encapsulate most of the image energy in a very small number of coefficients. This results in a much faster feature extraction process. The work in [1] also reports the use of wavelet coefficients in low-

pass bands. It, however, incorporated the histogram of coefficients in the recognition process, whereas in this work we use the coefficients in the HMM.

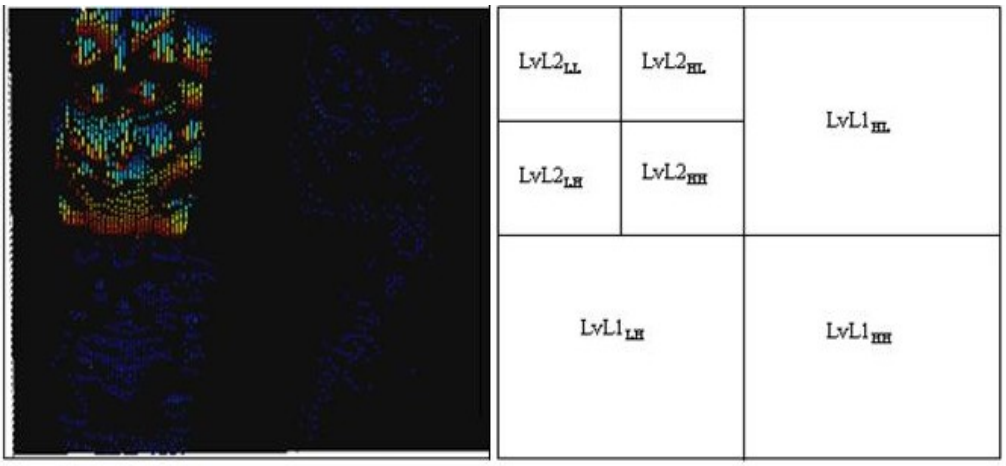


Figure 37. The "cat" image is decomposed into one level transformation (left image), $LvL1_{LL}$ (level one low pass band) can again be decomposed to create four more sub-bands (right image)

To extract features, we used a discrete wavelet transform (wavelet decomposition 2D Le Gall 5/3 Wavelet). Using a one-level wavelet decomposition, each image is transformed into low-pass and high-pass bands through filtering process At each level, the low-pass band is once again decomposed into low-pass (*LL*) and high-pass (*LH*) bands. While the basic energy of the image is preserved in the low pass bands, the image details such as edge information are preserved in high-pass bands. An example of the wavelet transformation for the object "cat" is shown in Figure 37. In the low pass band (top left corner of the left image) the cat can be seen, and in the *HL* and *LH* its vertical and horizontal edges are obvious [1].

In this work, 4 levels of decomposition (on resized images of 256x256 pixels) are used. This results in 16x16 pixel images (using *LL* band) for the serialization as shown in Figure 38 while still keeping the essential information. The main reason for using *LL* versus other components such as *LH*, *HL*, and *HH* is that the LL component preserves the shape and the connectivity of similar regions. These two characteristics are required by the proposed algorithm.

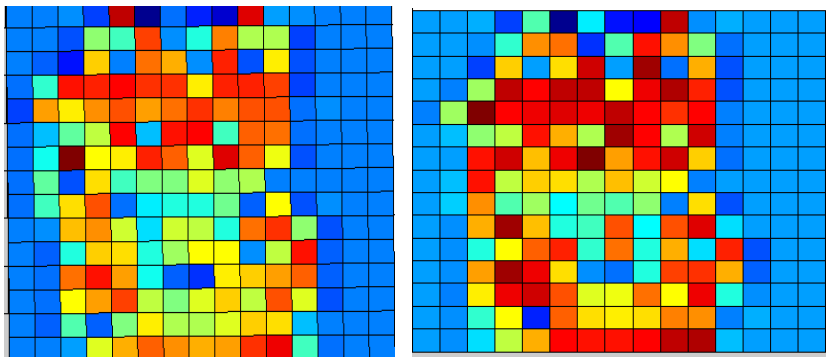


Figure 38. Two different views of the object cat wavelet coefficients after 4 level transformations.

5.3.2 Serializing Wavelet Coefficients

After the decomposition using the *2-D Le Gall 5/3* Wavelet transform method, the low pass part is read in a zigzag fashion (Figure 39). The zigzag way of reading maintains the relationship between data at the edges locations. The resulting sequence is shown in Figure 40 left. If the raster fashion was used (reading every row from the beginning), the connection between edge coefficients would be broken.

Since the number of coefficients is set to 256, the length of the observation sequence will also be T=256. Moreover, the pattern of each row will be repeated after every 16 (in one passes) or 32 coefficients (in two pass). Therefore one can guess that the number of states in HMM could be at least 16, possibly 32, or any multiple of 16.

After creating the serialized sequence of the data, we must process the sequence to make it usable for the HMM. Each data element in the sequence belongs to an observation space for the HMM. Therefore, we must describe these samples in the same space that is used by the HMM. This is explained in the next section.

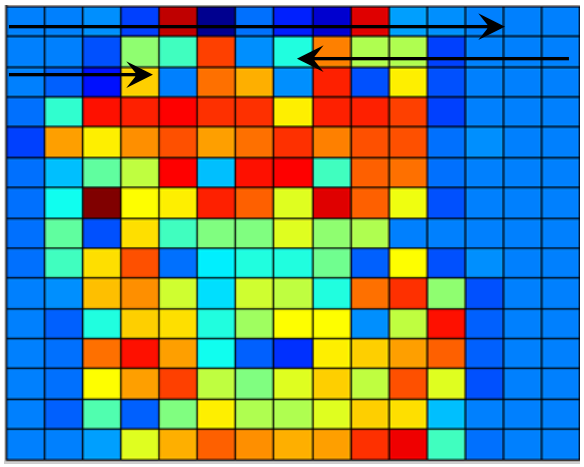


Figure 39. Object cat after 4-level transformation; it is now a 16x16 matrix.

5.4 Observation Space

Since the wavelet coefficients are real values, we must quantize them to discrete values so the HHM can utilize them. An example set of results after quantization is shown in Figure 40 right. For quantization two pieces of information are required: the range of the wavelet coefficients, and a criterion for dividing the range. Such criterion could be, for instance, placing the quantization levels at equal intervals or at locations where the population of data is higher. For example, if the serialized sequence is quantized into 8 levels, the observation sequence becomes like O={1 5 8 3 7 4 1 6 3 1 1 8 5 6 5 5 4 3 2 1 2 8 . . . . 7 7 4} where each one of the numbers in vector O is taken from the set of discrete output values V=(1, 2,…8). Figure 40-left shows an example of serialized data for the "car" object, where each datapoint is in range [-5 21]; the result of quantizing this data into 8 levels is shown in figure 40-right.

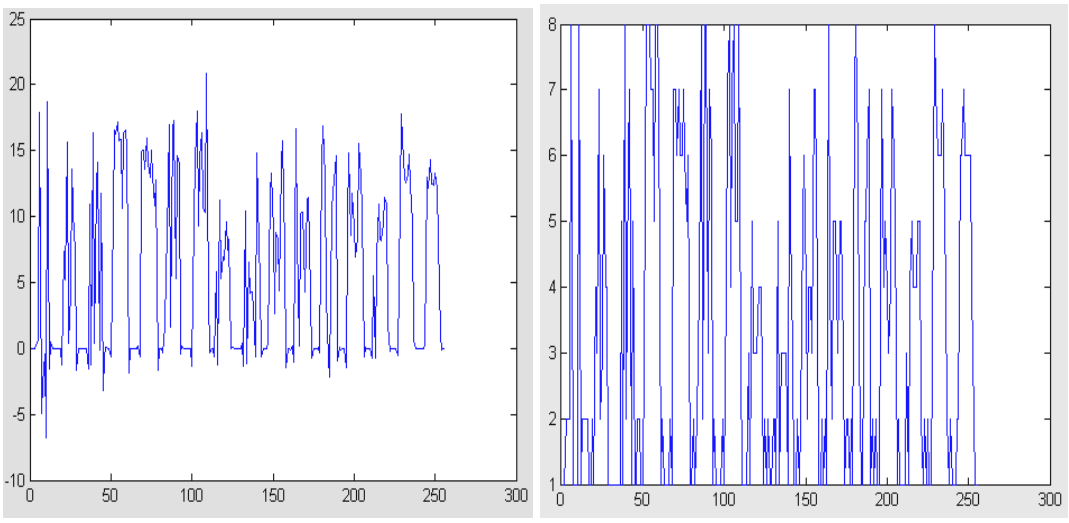


Figure 40. Serialized sequence representing samples after reading in zigzag fashion (left) and quantized sequence of "cat" object after 8 levels of quantization(right)

5.4.1 Discrete Quantization of Features: Uniform Quantization

Uniform quantization utilizes equal size quantization levels for the range of coefficients. In our study, it provided a good separation between values that resulted in good results. However, in general the best separation intervals for each view were found manually, which therefore means this is not a suitable approach for our work.

In order to check the performance of the recognition method based on the uniform quantization of the wavelet coefficients, we trained the HMM with 3 consecutive views of each object (10 degree difference between each) and then tested it with all other views. We used 3 views that included more fluctuation of data in the 3 successive views to increase the robustness of the recognition process.

The wavelet coefficients in this test ranged from -5 to 21. We utilized intervals of size 3 between -5 to 21. Therefore the quantizer had 9 levels from {-5,-2, …, 19}. This meant that the observed output set had 9 possible levels.

Since here we are only interested in the performance of the quantization methods, we set the HMM number of states to 32. The initial observation and transition matrices were set randomly. One can consider the number of states equal to the number of coefficients to keep all potential relationships; such strategy, however, is very time-consuming and

for practical purposes we require to find the fine balance between the numbers of states versus the recognition results quality. The recognition results for the uniform quantization are shown in Table 6.

Table 6 Uniform quantization recognition result, 12 of 20 objects produced correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	x	11	√	18	√
5	x	12	√	19	√
6	√	13	x	20	x
7	√	14	x		

5.4.2 Discrete Quantization of Features: Non-Uniform Quantization

A further look at the Figure 40-left reveals that the wavelet coefficient is not uniformly distributed. Therefore, it makes more sense to use a non- uniform quantization instead of a uniform one. We reviewed many coefficients and signals created from COIL-20 database and decided to create 13 following levels [-1 1 4 5 6 7 8 9 10 12 14 16 17] according to most populated places. Thus for the non- uniform quantization size of V becomes E=13. The transition and observation matrices are set randomly.

The recognition results of the HMM based method using non- uniform quantization are provided in Table 7. Here there were 15 (out of 20) correct outcomes. These results were better than the linear distribution case; however the main problem here was that we had to find the quantization levels manually by looking at the most populated areas for each object and through trial and error. Thus, we have to find an automated method to identify the proper levels.

Table 7 Non-uniform quantization recognition result, 15 of 20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	√	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	x	11	√	18	√
5	√	12	√	19	√
6	√	13	x	20	√
7	√	14	x		

5.4.2.1 Non-uniform Quantization with Automatically Found Levels

One simple way for finding the levels automatically is to utilize histogram of the coefficients. Finding local maxima in histograms is fairly simple and robust. For this, a bin size of 1 is initially used to create the histogram as shown in the example of Figure 41. Once the local maxima are found, they can be considered as the centers for clusters of coefficients. Each coefficient is then assigned to a center according to its distance from all centers. For view 3 of object 1, the local maxima are [0 3 5 7 9 12 14 16]. We used these bins to assign each coefficient to its closest bin. Given these maxima, we have 8 levels (E). For view 3 of object 2, the levels are [0 3 7 11 16], and therefore E is 5. As it can be seen, the number of quantized levels changes from view to view since the number of local maxima in each histogram is changing in each view. The results of identification using this method are provided in Table 8. The quantization method described here can also be replaced with the Lloyd-Max quantization scheme [73][74].

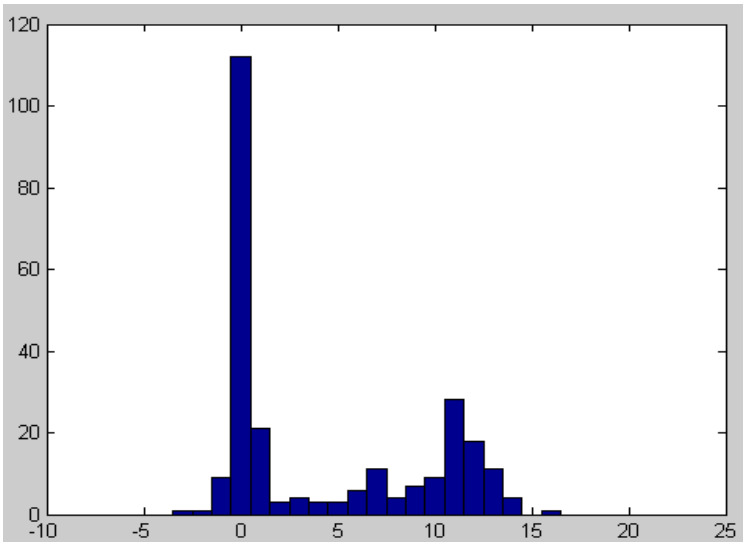


Figure 41. Histogram of coefficients, object 2, view 3 local maxima: [0 3 7 11 16]

Table 8 Automated method: histograms' local maxima results, 12 of 20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	x	11	√	18	√
5	x	12	√	19	√
6	x	13	x	20	√
7	√	14	x		

This method seems to perform poorly (60%). The main problem with the local maxima method is that two close views of a single object may generate coefficients that are close in value but are quantized into two different discrete levels. This means that in discrete method they may be considered completely unrelated, while in fact they are slightly different.

To address this issue, experiments were repeated using the most populated bins in the histogram, instead of the local maxima. Therefore, at each image, we find E most populated bins as the center of the quantization levels. Given the complexity and the type

of objects in the COIL-database, a value of 5 to 8 for E should be chosen to lead to a successful recognition. For E= 5 and E=8 the results are shown below in Table 9 and Table 10. In these tests A and B and π were randomly initialized and N was set to 32.

Table 9 Results for non- uniform automated quantization with E=5, 14 of 20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	√	8	√	15	√
2	√	9	√	16	√
3	√	10	x	17	√
4	x	11	x	18	x
5	√	12	√	19	√
6	√	13	x	20	√
7	x	14	√		

Table 10 Results for non- uniform automated quantization with E=8, 14 of 20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	√	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	x	11	√	18	x
5	x	12	√	19	√
6	√	13	x	20	√
7	x	14	x		

Using the number of quantized levels of 5 and 8 produced the similar number of correct answers. The advantage of this method over the initial non- uniform quantization method is its automatic selection of the bin centroids.

While the recognition results seem relatively good (around 70% success rate), in order to improve them, we must have a closer look on how the topology of HMM and its initial values can be adjusted. This is presented in the next section.

5.5 HMM Structure and Initialization Effects

In this section, we investigate the structure of HMM (connectivity of the graph) and the initial setting of the transition and observation matrices on the performance of the recognition system.

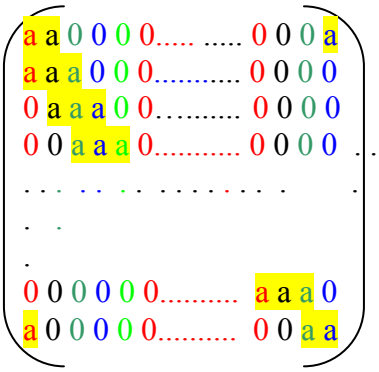
For the previous results, we initialized the HMM randomly. This means that we assumed a fully connected graph for HMM and all states can have transitions to other states. Also the initial values for transition probabilities were set randomly. After training the HMM, some of the transition probabilities became smaller, and some became larger than before. This means that in the training process, some connections (transitions) from the HMM graph were removed, due to the content of the image that was used for the training. The emission matrix was also initialized in the same way; each state of the HMM had transition probabilities for all possible outputs. Once again after training the HMM some of the emission probabilities will change. The initial state probability π (the state in which the HMM starts) was set randomly. After training, π changes to match the A and B matrices.

When A, B, and π are randomly set, there is a high possibility that the results from one training run differs from another. Although on average the recognition rate is almost the same, individual runs are not guaranteed to produce the exact same results. With fixed initial values, even though the final recognition result may sometimes be lower, it is more trustworthy since it is repeatable. Also on a random setting, the model cannot be finalized. In each run of algorithm we may get different recognition results. Therefore, it was decided to start the training from fixed initial values for A, B, and π. In the following subsections, we explain the effect of different initial settings for A, and B. For the prior (π), it was noticed that with fixed value probabilities, the HMM may start from any state, and the result will not be much different. The trained A, and B will be different but the recognition results are the same. Therefore, the initial state vector (π) which is the probability of being in any particular state initially, from here on is set to $\pi = [1\ 0\ 0\ \ldots\ 0]$ to enforce the system to start from the first state always.

5.5.1 Transition Matrix

The parameter to set in designing an HMM is the number of states (*N*). In the feature extraction part, we produced a 16x16 matrix of wavelet coefficients (low pass level). We then read them in a zigzag fashion to create a sequence in which every 16 values belong to a line of reading. All the points are related to their neighbours in each line, but datapoints at locations that are multiple of 16 are also related to the next line's values (Figure 42). At multiples of 32, the points are related to their previous line neighbours as well. Thus, it is intuitive that the number of hidden states can be 16 or multiples of 16. If we want to use the connection between cells better, we may consider number of states *N*=32. This requires more processing time but the results are considerably better.

The structure of the transition matrix is also very important in the performance of the HMM based application. The structure is shaped by setting some of the elements of the transition matrix into zero while allowing the others to be trained by the HMM. Given that the data is read in a zigzag fashion, one expects that at each point in a state to be related to its neighbours (one or more points before and after). Therefore, we can intuitively consider a transition matrix like the following, where the width of the non-zero diagonal elements has to be found:



A single-element transition matrix and its corresponding data sequence are shown in Figure 42; the elements of transition matrix are the states and not the actual values of the elements.

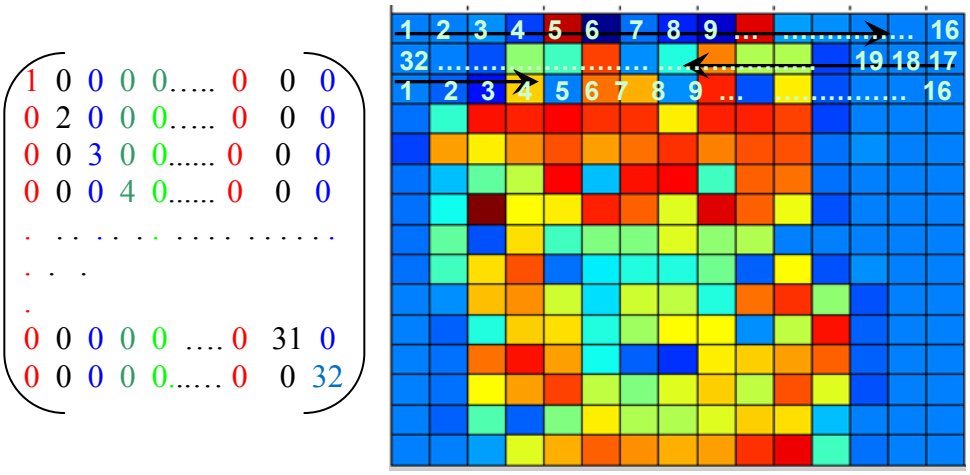


Figure 42. A matrix (this is not the transition matrix) indicating the time instance of being in each hidden state if a 1-element diagonal matrix was used as transition matrix and *N*=32

Our experiments showed that for some objects one-element diagonal transition matrices provides good results but for others it did not. We tested several examples and here report the results of two different modes of single diagonal element and multiple diagonal elements. The results for both of these transition matrix styles are shown in Table 11 and Table 12 (random observation matrix).

Table 11 Results of training using single element diagonal transition matrix, average 8 of 20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	√	8	√	15	√
2	√	9	x	16	√
3	x	10	x	17	√
4	x	11	x	18	x
5	x	12	x	19	x
6	x	13	x	20	√
7	√	14	x		

The Table 12 shows the results of using transition matrix with addition of 2 neighbours from each side (left and right). Therefore the width of the diagonal elements

is 5. The results are much better than the single diagonal case; however, without adjusting the observation matrix the overall result is still not satisfactory. In the next section we try some methods to set the observation matrix properly.

Table 12 Results of training using 5-element diagonal transition matrix, average 13/20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	√	8	√	15	√
2	√	9	√	16	√
3	√	10	x	17	√
4	x	11	x	18	x
5	√	12	√	19	√
6	√	13	x	20	x
7	√	14	x		

5.5.2 Observation matrix

Here we assume that the non- uniform discretization method according to Section 5.3 is used. Each element in the observation matrix represents the probability of observing a specific output at a specific state of the HMM. Therefore, one intuitive method to initialize the observation matrix could be to measure how many times each output value is seen in each state (similar to calculating histogram of output values in each state). Clearly, this is only possible if we knew what datapoints in the sequence belong to which hidden state. This is easy for the single-element diagonal transition matrix as in Figure 42, but for multiple-element diagonal matrixes we cannot be very sure that a specific datapoint is generated by a specific state. Here we assume that the center element in each row is the main state that ultimately generates an output in the data sequence (like a single-element matrix of Figure 42). We also consider two different sizes of observation space (5 and 8), and two structures for transition matrix (single-element and 5-element diagonal).

In the case of single element model for the transition matrix, a special case exists where the training data is such that a given state never produces a specific output. So, if we take the output histogram, the corresponding bin becomes zero indicating that the

probability of this state generating that specific output is zero. When we run our test with this HMM, if that given state is tested for the output that it has never seen, the total likelihood becomes zero, regardless of all other states. This is because the likelihood is calculated in the HMM as a multiplication of several factors. In order to prevent this problem, after first step of calculating the likelihoods based on histogram, a small margin value (0.01) is added to all values of observation matrix and then the matrix is normalized. Through this modification, the training and testing algorithms have the chance to train and assess all cases.

To calculate the likelihood based on histogram, we assume that the states and data sequence values are as shown in Figure 42. For each state, the number of times that each output is observed for that state are computed and the probability of producing each output (the ratio of seeing that output to the total outputs) is assigned. Then the value of 0.01 is added to each probability and it is normalize. For example, if state 1 observes the discrete output values of 1, 3 and 4 (out of E=5 discrete values), 2, 2, and 4 times respectively (in a 32 state HMM with T=256 sequence size, each state produces 8 output), the observation matrix elements for the first row are b_{11}=0.25, b_{12} =0, b_{13}=0.25, b_{14}=0.50, and b_{15}=0. These values are adjusted by adding a value of 0.01 to each element and then and normalize to sum to one. For example these values at the end become (b_{11}=0.2476, b_{12} =0.0095, b_{13}=0.2476, b_{14}=0.4857, and b_{15}=0.0095). After applying the above method, the result for an HMM with single-element diagonal transition matrix HMM, and E=5 and E=8 is found (Table 13 and Table 14).

We have also checked the recognition outcome for the 5-element transition matrix with observation matrix as described in this section for E=5 and E=8 (Table 15 and Table 16). In this case however, there is no need to add the small offset to the initial setting of the observation matrix since the transition matrix has many connections and the likelihood does not become zero, especially if the training data does not have a specific output associated with a specific state.

Table 13 Results of using 1-element diagonal transition matrix, observation space *E*=5, histogram based initialization of *B*; average 13/20 correct answers

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	√	17	√
4	√	11	√	18	x
5	X	12	√	19	√
6	√	13	x	20	√
7	X	14	x		

Table 14 Results of using 1-element diagonal transition matrix, observation space *E*=8, the histogram based initialization of B; average 14/20 correct recognition.

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	√	17	√
4	x	11	x	18	√
5	x	12	√	19	√
6	√	13	x	20	√
7	√	14	√		

Table 15 Results of using 5-element diagonal transition matrix; observation space *E=5*; the histogram based initialization of B; average 11/20 correct recognition.

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	x	11	x	18	x
5	√	12	√	19	√
6	x	13	x	20	√
7	√	14	x		

Table 16 Results of using 5-element diagonal transition matrix, observation space *E=8*, the histogram based initialization of B; average 13/20 correct recognition.

Object No.	Successful recognition	Object No.	Successful recognition	Object No.	Successful recognition
1	x	8	√	15	√
2	√	9	x	16	√
3	√	10	x	17	√
4	√	11	√	18	√
5	x	12	√	19	x
6	√	13	x	20	√
7	√	14	x		

Under the above settings (since all initial values are fixed), the results are robust and reproducible. From these results, the single element method will produce better results than the 5-element method, but the difference is small.

5.6 Model Fitting

The previous set of results shows that the proposed adjustments work for some objects but not for all. This motivated us for finding a way to incorporate a best set of parameters (*A, B*, and *E*) for each object. For example, if single-element method with *E*=5 produces the best result for object 4, then we should use that specific setting for

object 4. In order to see how much the recognition result increases by applying this flexibility, we have considered 2 settings for transition matrix (single- and 5-element diagonal matrices) and 4 settings for the number of output states (quantization levels) E=5, 6, 7, and 8. Therefore in total 8 settings will be possible (Table 17).

To find which mode performs the best for any given object, we must process the object images using all potential modes and pick up the mode with the best generated likelihood. These modes, however, are different from each other in terms of their matrix sizes and likelihood values and therefore values from two different modes are not comparable. Therefore, it is required to use a normalized recognition score, instead of pure likelihood, to present the model quality in presenting an object. This way, each mode will receive a score and the mode with the highest score is chosen as the best mode for that object.

Table 17 Modes with observation space from 5 to 8, two cases for transition matrix: one elemental and 5 elemental.

Mode	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
TM	1-element	1-element	1-element	1-element	5-element	5-element	5-element	5-element
OS size	5	6	7	8	5	6	7	8

5.6.1 Normalizing Recognition Score

For each mode, the test is run for all views of the object and the mode with highest probability (highest log likelihood) is picked. Therefore, for each mode, after running the HMM algorithm, first G (for example G=50) highest log likelihoods are chosen. Since there are multiple views of the object in the dataset, the view with the highest likelihood is picked. If the highest likelihood found for the desired object i is $LL(i)$, the score is calculated as:

$$score(i) = \begin{cases} 1 & object\ \ i\ \ has\ the\ highest\ \ LL \\ 0 & object\ \ i\ \ not\ \ found\ amongst\ \ G\ highest\ \ LL \\ 1 - LL\,(i) \,/\, LL\,(G) & otherwise \end{cases} \qquad (21)$$

If the object associated with the highest likelihood is in fact the desired object, the score will become 1, otherwise it will be a number smaller than 1. The higher the difference between the 50^{th} object and the first few objects, the closer the score to 1 (higher the chances that the object is indeed found in the first few results). The reason for this is that G^{th} log likelihood is the largest in absolute value and if the object is amongst the first few tested objects then $LL(i)<<LL(G)$. For example, these log likelihood can be sorted from high to low this way: -127(first), -128(second), -129,…,-190(50th). The 50th value is the smallest considering the sign but its absolute value is the highest. Thus $LL(i)<<LL(G)$, $0<i<50$.

At the end of this stage, there is a score for each mode. This procedure is performed for all views of an object and the average score value for all views is calculated. The mode which produces the highest score is chosen as the best mode for presenting that object. The above scheme has been tested with the database objects. The recognition results are checked for the above 8 modes. Table 18 and Table 19 represent results for the best found mode and the highest score for each object.

Table 18 Finding best mode for each object; best modes and scores utilizing 8 modes

Object No.	Best Mode	Score	Object	Best Mode	Score	Object No.	Best Mode	Score
1	8	0.91717	8	1	1.00000	15	1	1.00000
2	1	1.00000	9	5	0.87181	16	1	1.00000
3	1	1.00000	10	1	1.00000	17	1	1.00000
4	1	0.99158	11	1	1.00000	18	2	1.00000
5	5	0.96403	12	1	1.00000	19	1	1.00000
6	1	1.00000	13	1	0.92336	20	2	1.00000
7	1	1.00000	14	1	0.94864			

Table 19 Results of tests using 8 modes, average success rate over all objects: 0.64

Object No.	Success Ratio	Object No.	Success Ratio	Object No.	Success Ratio
Object 1	0.2500	Object 8	1.000	Object 15	1.000
Object 2	0.8889	Object 9	0.2500	Object 16	1.000
Object 3	0.6111	Object 10	0.6111	Object 17	.8056
Object 4	0.4167	Object 11	0.2500	Object 18	1.000
Object 5	0.3888	Object 12	1.000	Object 19	.6667
Object 6	0.5278	Object 13	0.3889	Object 20	.7778
Object 7	0.6111	Object 14	0.3889		

To review how results shown in Table 18 indeed improve the overall recognition results (Table 19), comparison is made between cases where only mode 1 is used for all objects (since it produced better results for higher number of cases) versus the best selective mode. The results of this comparison are depicted in Table 20. From these results it can be seen that for the objects that their mode was something other than mode one, the recognition success rate is very poor (objects 1, 5, 9, 18 and 20). Clearly, these bad results lower the overall recognition results to 53% (from 64% in the above). In comparison to Force based method, the proposed HMM based method has a 5% lower success.

Table 20 Results of tests, using only mode 1, average success rate over all objects: 0.53

Object No.	Success Ratio	Object No.	Success Ratio	Object No.	Success Ratio
1	0.1944	8	1.000	15	1.000
2	0.8889	9	0.1667	16	1.000
3	0.6111	10	0.6111	17	0.8056
4	0. 4167	11	0.2500	18	0.000
5	0.0278	12	1.000	19	0.6667
6	0.5278	13	0.3889	20	0.1944
7	0.6111	14	0.3889		

5.7 Recognition Performance for Occluded Objects

Using the fitted model, a set of experiments was carried out to assess the performance of the HMM based method under partial occlusion. The occlusion level was changed 10-40% in each image dimension and it was applied to the middle part of the image. The recognition results for one view of each object are shown in Table 21. The trend of the success rate is shown in Figure 44.

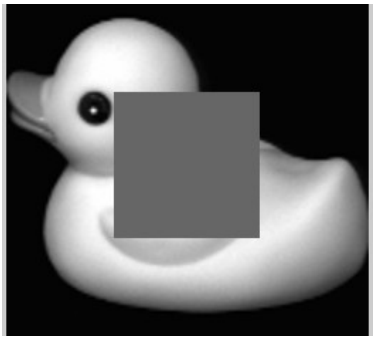


Figure 43. Occluded object, 40% in each dimension, covering the middle of the image

From these results, the HMM method handles occlusion rather gracefully. The occlusion was introduced in the middle of the image, causes more wavelet coefficients to be distorted.

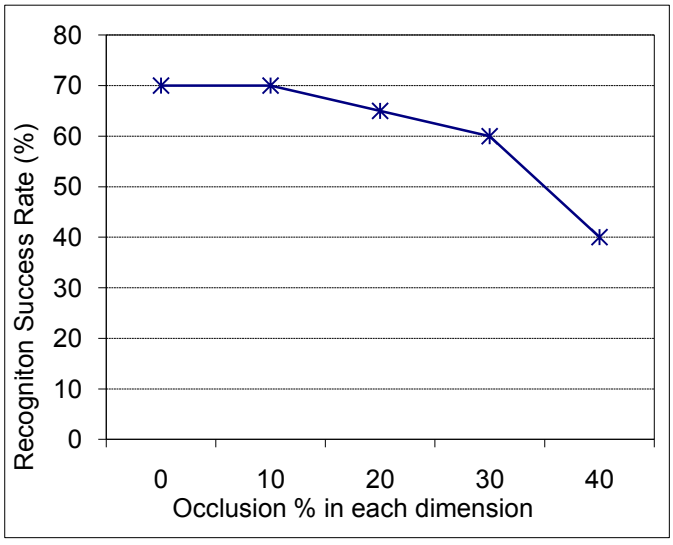


Figure 44. Success Rate vs. Occlusion percentages (for one view of each object only)

Table 21 Occlusion results for one view of each object (occlusion centered in the middle of image)

Object	0%occlusion	10% occlusion	20% occlusion	30% occlusion	40% occlusion
Overall result	70	70	65	60	40
1	x	x	x	x	x
2	√	√	√	√	√
3	√	√	√	√	x
4	√	√	√	√	x
5	√	√	x	√	√
6	√	√	√	x	x
7	x	x	x	x	x
8	√	√	√	√	x
9	x	x	x	x	√
10	√	√	√	√	√
11	x	x	x	x	x
12	√	√	√	√	√
13	x	x	x	x	x
14	x	x	x	x	x
15	√	√	√	√	√
16	√	√	√	√	√
17	√	√	√	√	x
18	√	√	√	√	x
19	√	√	√	x	x
20	√	√	√	√	√

5.8 Remarks

The method presented in this chapter provides a similar performance to [15] in the situation where 1 to 3 views of a known object are available. However, the proposed method is much faster.

It is then shown that when two views of an object are used for matching, this method produces 67% correct results for the COIL-20 database. This is similar to what was obtained with our proposed method based on using1 to 3 views with a discrete HMM. They showed that the accuracy of detection could be improved to 98% if all 36 views

were used in the matching process. In our method we utilized only with 1 to 3 views of the known model.

We were not able to identify how coefficients within a sub-image are serialized to create a sequence in [15](it was not stated in their work). Moreover, the performance of that method in terms of its computation time is not presented. This method was tested for the continuous-output HMM and it was noticed that with large datasets with large number of states the method is very slow. This finding is also was also acknowledged by [70] [71] [72]. For [15] the significant time associated with the training and testing processes is due to two factors: the use of continuous HMM and multiple overlapping sub images. A quick examination of this method reveals that the number of datapoints per image (for an NxN image) is at least 0.5xNxN and at most 3.5xNxN. In comparison, our proposed method utilizes NxN/256 datapoints. For N=256, our method generates a sequence of length 256, while that method uses a number of datapoints between 32768 and 229376. The difference is in an order of 128 to 896 times more for [15] .

In this chapter we presented a method using discrete HMMs instead of continuous HMM. The proposed method used 4 levels of wavelet transform with no overlapping sub-images. In our tests using the MATLAB code from[22], the speed of discrete method was (on average) at least 20 times faster. The advantage of the method presented in this chapter is it's significantly faster processing time due to using discrete HMMs (also mentioned in [70] [71] [72]) and non-overlapping sub images.

Chapter 6 COMBINING RESULTS OF HMM AND FORCE BASED METHODS

In previous chapters, both HMM or Force based methods were used to identify objects based on comparison with the trained data. Both methods utilized measurement scores that required being at their maximum. It was noticed that in many cases where the correct object was not found, the correct object was indeed the second or third best object. Therefore a post processing strategy may be added to the system to incorporate the likelihood from HHM and score information from Force to improve the overall recognition results.

In this chapter, we describe a strategy for combining the results from HMM and Force based algorithms to improve the recognition results. These algorithms were individually able to detect the correct shape with success rates of 64% and 69% respectively (on the COIL-20 database). Since the HMM method, the algorithm relies on largest likelihoods while Force depends on the smallest angle difference, it is required to find a way to convert these two measures into one single quantity. The next subsection describes our proposed approach for this.

6.1 Combination Methods

In order to be able to use both Force algorithm and HMM to improve our recognition results, we have created a system for score assignment.

1. Score for the Force-based Algorithm

In the Force algorithm the score for object *n* is computed by:

$$SF(n)=\frac{\Delta(N)-\Delta(n)}{\Delta(N)-\Delta(1)} \qquad for\, n=1,..,N \tag{22}$$

Here *SF* stands for Score of the Force algorithm and $\Delta(i)$ is the summed difference of angles for image *i* (difference between angles of image *i* and model object). The *N*'th object has the highest value of angle difference and therefore is the least likely match.

This formula results in a score value that is normalized between 0 and 1. The closer the score is to 1, the higher the likelihood of the image being from the desired object.

2. Score for the HMM based Algorithm

For the HMM method, the score *SH* is calculated from:

$$SH(i)=\frac{LL(i)-LL(N)}{LL(1)-LL(N)} \qquad for\, i=1,..,N \tag{23}$$

Here *LL(i)* is the log likelihood of image *I*, tested with the HMM of the model object. *LL(1)* is the largest log likelihood and belongs to the most likely match (*i*=1). *SH* is normalized in the interval [0 1]. The most likely object has score 1 and the least likely has the score 0.

3. Combining *SH* and *SF*

The sum of the scores of the same object in the *SF* and *SH* is computed, if there is an object from Force which is not in the first 10 candidate of HMM, only the score in *SF* is used as the final score. The object that obtains the highest score is the answer to the recognition problem.

For example for view 13 of object 1, the candidate objects and their score by Force algorithm are:

Object	**1**	**7**	**12**	**2**	**10**	**11**	**4**	**15**	**14**	**13**
Score	1.00	0.73	0.53	0.37	0.27	0.26	0.19	0.13	0.0536	0

The candidate objects according to the HMM are:

Object	**2**	**1**	**19**	**15**	**11**	**7**	**12**	**14**	**10**	**13**
Score	1.00	0.68	0.42	0.33	0.29	0.25	0.238	0.143	0.12	0

The combined results are:

Object	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Score	**1.68**	1.372	0	0.198	0	0	0.988	0	0	0.391
Object	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Score	0.556	0.771	0	0.197	0.476	0	0	0	0.427	0

Combining the results of HMM and Force produces results with higher certainty and accuracy.

6.2 Result of Combining HMM and Force Based Methods

We tested the above combine score on the COIL-20 database objects. The recognition rates are shown in Table 22.

Table 22 Results of recognition based on combined Force and HMM based methods

Object No.	**success rate**	**Object No.**	**success rate**	**Object No.**	**success rate**
1	0.8611	8	0.9444	15	1.000
2	0.9167	9	0.5556	16	1.000
3	0.4444	10	0.7500	17	1.000
4	0.7222	11	0.8611	18	0.8611
5	0.7222	12	1.0000	19	0.4722
6	0.5278	13	0.6667	20	0.8056
7	0.7500	14	0.7778		

The recognition result now become 74% percent, almost 5% better than Force and 11% better than HMM alone.

Chapter 7 CONCLUSIONS AND FUTURE WORK

In this thesis the problem of detecting a known object in a scene or database of images was investigated. The object recognition problem has many aspects and a complete solution requires the use of many image processing techniques and tools. In this thesis, we presented two major components for a complete solution to this problem: a data clustering technique for image segmentation and feature extraction, and shape recognition methods for identification. The presented data clustering method (Force) relies on the laws of electrostatic fields to identify clusters of datapoints in 2D space. Applications of Force to image segmentation in gray level and colour images were described to show the versatility of this algorithm. In particular an MRI brain tumor detection technique based on the use of Force was shown to produce accurate and robust results.

Following the image segmentation methods, we presented a method to use Force for feature extraction from database images. These features were later utilized in a shape matching algorithm. We presented a scale and rotation invariant scheme based on angle similarity assessment. This scheme was shown to be robust to scale and orientation difference between the model and test images. The shape recognition technique was shown to have a success rate of 70% for the COIL-20 database. To further enhance the recognition results, and also to handle non-ideal situations like object occlusion (which Force based method does not handle), we considered an alternative shape recognition technique. For this purpose, we presented a statistical object detection method based on feature extraction using wavelet transforms and modelling an image using hidden Markov models. This approach produced around 64% correct recognitions for the COIL-20 database and was shown to be robust with respect to occlusion. We then considered combining the results of HMM based object recognition with the results obtained from the Force based method. The combination was done through assigning similarity scores

to each tested image and then combining the results to create an overall similarity measure. This method improved the overall recognition results to 74%.

7.1 Summary of Contributions

This thesis presents four contributions towards the goal of object recognition.

1. The development of a novel data clustering algorithm, Force, capable of handling noise and outlier points and different initial settings.

2. The development of image segmentation schemes for gray and color images based on the Force data clustering algorithm.

3. The development of the shape recognition scheme based on features extracted using the Force algorithm and compared through angle similarity assessment approach which is rotation and scale invariant.

4. The development of a statistical object recognition scheme based on hidden Markov models applied to features extracted using wavelet transform.

7.2 Future Research

The data clustering method presented in this thesis has been shown to perform well in applications such as image segmentation and feature extraction. However, a mathematical analysis of its properties is still an open research subject. We expect that through a formal analysis of the presented algorithm, one can optimize the algorithm for faster convergence. The Force algorithm relies on the laws of electrostatic to find electric field equilibrium points as cluster centers. However, we can also consider other field concepts, for example a field in which the force between points is proportional to the inverse of distance (or cubed inverse of distance) instead of the squared inverse of distance (as in Electric fields). This is still an open research problem.

In addition, we have shown that Force can be easily modified and used in different applications ranging from gray level image segmentation of MRI images to feature extraction from optical image. These applications were all inspired by the fact that clusters of datapoints existed in some space that represented the image (for example,

intensity space for MRI). An immediate extension of the current work is to extend the use of Force to higher dimensional spaces by adding other features to each datapoint.

Another possible future work could be on the development of a 2D HMM model for the specific wavelet transform of the images instead of 1D HMM model that was used in this thesis; ideally a 2D model should be designed for images. The reason is that the concept of Markov modeling cannot be simply extended to a 2D domain (Markov modeling specifies the transition between states in two consecutive 'time instances', which cannot be directly mapped to a 2D space instance). Therefore, 2D models only use concepts from HMM, and are developed for specific applications.

References:

[1] W. Zou and Y. Li, "Image Classification Using Wavelet Coefficients in Low-pass Bands," in proceedings of the International Joint Conference on Neural Networks, IJCNN 2007, pp. 114-118, Orlando, Florida, USA, August 12-17, 2007.

[2] P. M. Roth and M. Winter "Survey of Appearance-Based Methods for Object Recognition," Technical Report ICG-TR-01/08, Inst. for Computer Graphics and Vision, Graz University of Technology, Austria; January 15, 2008.

[3] T. Gevers and A. W. M. Smeulders, "Color based object recognition," Pattern Recognition., vol. 32, pp. 453-464, 1999

[4] M. v. d. Giessen and J. Schmidhuber, "Fast color-based object recognition independent of position and orientation," Proc. Intl. Conf. on Artificial Neural Networks ICANN'05, LNCS 3696, pp. 469-474, Springer-Verlag Berlin Heidelberg, 2005.

[5] S. Pandey and S. Kumar, "A novel approach to automatic object detection using intensity pattern recognition in YCbCr color space," Proceedings of the International Conference on Audio, Language and Image Processing, ICALIP pp. 1274-1277, 2008.

[6] A. Selinger and R. C. Nelson, "Appearance-Based Object Recognition Using Multiple Views," 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01) vol. 1, pp. 905-911, Kauai, Hawaii, USA, December 2001.

[7] Y. Wang, M. Brookes and P. Dragotti, "Object Recognition using Multi-view Imaging," Proc Intl Conf on Signal Processing, pp. 810-813, 2008.

[8] G. Hetzel, B. Leibe, P. Levi, B. Schiele, "3D object recognition from range images using local feature histograms," in proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 394-399, 2001.

[9] H. Bang, S. Lee, D. Yu, and H. Suh, "Robust object recognition using a color co-occurrence histogram and the spatial relations of image patches," Artificial Life and Robotics, vol.13, no.2, pp.488-492, 2009.

[10] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05, vol. 2, pp. 524-531, 2005.

[11] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," Ann. Math. Statist, vol. 41, no. 1, pp. 164-171, 1970.

[12] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006, ISBN 0-387-31073-8.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, Wiley-Interscience, 2001, ISBN 0-471-05669-3.

[14] M. Bicego, V. Murino: "2D shape recognition by Hidden Markov Models," Proc. of 11th IEEE Int. Conf. on Image Analysis and Processing (ICIAP01), pp 20-24, 2001.

[15] M. Bicego, U. Castellani, V. Murino: "A Hidden Markov Model approach for appearance-based 3D object recognition". Pattern Recognition Letters, vol. 26, no.16 , pp. 2588-2599, 2005.

[16] X. Ma, A. Khokhar, and D. Schonfeld, "A general two-dimensional hidden Markov model and its application to image classification," in Proc. IEEE Int. Conf. Image Processing, pp. 41-44, San Antonio, TX, 2007.

[17] J. Li , A. Najmi , R.M. Gray RM, "Image classification by a two dimensional hidden Markov model," IEEE Transactions on Signal Processing, vol. 48 , no. 2, pp. 517-533, 2000.

[18] H. Freeman. On the encoding of arbitrary geometric configurations, IRE Transactions on Electronic Computers EC- 10 pp. 260-2686, 1961.

[19] S.A. Nene, S.K. Nayar, and, H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep. CUCS-005-96, Columbia University, 1996.

[20] Hidden Markov Models: Decoding and Evaluation, Chuan Sheng Foo, Lecture 5, 01/23/2007.

[21] http://ai.stanford.edu/~serafim/CS262_2007/notes/lecture5.pdf

[22] http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[23] A.K. Jain, M.N. Murthy, P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, Sept. 1999.

[24] J. McQueen, "Some methods for classification and analysis of multivariate observations," Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.

[25] M. Xu and P. Franti, "A heuristic K-means clustering algorithm by kernel PCA," International Conference on Image Processing, ICIP, vol. 5, pp. 3503-3506, 2004.

[26] G. H. Ball, and D.J. Hall, "ISODATA, a novel method of data analysis and classification," Tech. Rep. Stanford University, Stanford, CA, 1965.

[27] G.P. Babu, and M.N. Murty, "A near optimal initial seed value selection in K-means algorithm using a genetic algorithm," Pattern Recog. Lett. 14, pp. 763-69, 1993.

[28] R. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," Proc. 20th Conf. Very Large Databases, pp. 144-155, 1994.

[29] J.C. Bezdek, L.O. Hall, and L.P. Clarke, "Review of MR image segmentation techniques using pattern recognition," Medical Physics, vol. 20, no 4, 1033-1048, 1993.

[30] K. Held, K.E. Rota, B.J. Krause, W.M Wells, R. Kikinis, and H.W. Muller-Gartner, "Markov random field segmentation of brain MR images," IEEE Trans. Med. Imaging, pp. 878-886, 1997.

[31] L.P. Clarke, R. P. Velthuizen, M.A. Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, and M.L. Silbiger, "MRI segmentation: methods and applications." Magn Reson Imaging, vol. 13, no. 3, pp. 343-68, 1995.

[32] M.B. Cuadra, J. Gomez, P. Hagmann, C. Pollo, J.G. Villemure, B.M. Dawant, and J. Thiran, "Atlas-based segmentation of pathological brains using a model of tumor growth," Medical Image Comp. & Computer-Assisted Intervention MICCAI , Springer, pp. 380-387, 2002.

[33] M.C. Clark, L.O. Lawrence, D.B. Golgof, R. Velthuizen, F.R. Murtagh, and M.S. Silbiger, "Automatic tumor-segmentation using knowledge-based techniques," IEEE Trans. on Medical Imaging, vol. 17, pp. 187-201, 1998.

[34] M.R. Kaus, S.K. Warfield, A. Nabavi, E. Chatzidakis, P.M. Black, F. Jolesz and R. Kikini, "Segmentation of meningiomas and low grade gliomas in MRI," Lecture Notes in Computer Science, MICCAI, vol. 1679. Springer Verlag, pp. 1-10, 1999.

[35] M. Prastawa, E. Bullitt, S. Ho, G. Gerig, "A Brain Tumor Segmentation Framework Based on Outlier Detection," Medical Image Analysis, vol. 8, pp. 275-283, 2004.

[36] D. Gering, W. Eric, L. Grimson, and R. Kikinis, "Recognizing Deviations from Normalcy for Brain Tumor Segmentation," Medical Image Computing and Computer-Assisted Intervention, pp. 388-395, 2002.

[37] P.Y. Lau, Frank C.T. Voon, and S Ozawa, "The detection and visualization of brain tumors on T2-weighted MRI images using multiparameter feature blocks," 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5104-5107, September 1-4, Shanghai, China, 2005.

[38] M. Wu, C. Lin, and C. Chang, "Brain Tumor Detection Using Color-Based K-means Clustering Seg.," IEEE Computer Society (IIH-MSP) pp. 245-250, 2007.

[39] M. Kalantari Khandani, P. Saeedi, Y. P. Fallah, and M.K. Khandani, "A Novel Data Clustering Algorithm Based on Electrostatic Field Concepts," IEEE Symposium on Computational Intelligence and Data Mining (IEEE CIDM), pp. 232-237, 2009.

[40] J. Mc Queen, "Some methods for classification and analysis of multivariate observations," in Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability, pp. 281-297, 1967.

[41] J.F. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Analysis and Machine Intelligence, pp. 679-698, 1986.

[42] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," SIGMOD Rec. vol. 25, no. 2, pp. 103-114, 1996.

[43] P. Chang and J. Krumm, "Object Recognition with Color Cooccurrence Histograms," IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) vol. 2, pp. 498-504, Fort Collins, CO, June 23-25, 1999.

[44] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using Region," Proc. of IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp.1030-1037, Miami, USA, 2009.

[45] D. Comaniu and P. Meer, "Robust analysis of feature spaces: color image segmentation," Proc. of IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp. 750-755, 1997.

[46] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," IEEE Trans. on Image Processing, vol. 6, no. 11, pp. 1530-1544, 1997.

[47] G. Hamarneh, J. Yang, C. McIntosh and M. Langille, "3D live-wire-based semi-automatic segmentation of medical images," Proceedings of the SPIE medical imaging: image processing, vol. 5747, pp. 1597-1603, 2005.

[48] K. Karsch, Q. He, and Y. Duan, "A fast, semi-automatic brain structure segmentation algorithm for magnetic resonance imaging," in: Proc. of IEEE BIBM 2009, pp. 297-302, 2009.

[49] S. Ito, M. Yoshioka, S. Omatu, K. Kita and K. Kugo, "An image segmentation method using histograms and the human characteristics of HSI color space for a scene image," Pattern Recognition Letters archive vol. 27, no. 13, pp. 1515-1521, Elsevier Science Inc. , New York, NY, USA, 2006.

[50] M. Mancas, B. Gosselin, and B. MACQ, "Segmentation Using a Region Growing Thresholding," Proceedings of the SPIE 5672, pp. 388-398, San Jose, CA, USA, 2005.

[51] H. Digabel and C. Lanteujoul, "Iterative algorithms," in Proc. 2nd Europ. Symp. on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, pp. 85-99, Caen, France, 1977.

[52] N. Senthilkumaran and R. Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches," International Journal of Recent Trends in Engineering, vol. 1, no. 2, pp. 250-254, May 2009.

[53] M. Kalantari Khandani, R. Bajcsy, and Y. P. Fallah, "Automated Segmentation of Brain Tumors in MRI Using Force data clustering Algorithm," Lecture Notes in Computer Science (LNCS) 5th International Sym. Visual Computing (ISVC), vol. 5875/2009, pp. 317-326, December 2009.

[54] J. A. Westell, "Object recognition via multiple-view Inspection Using Saturation Weighted Distributive Hue Histogram and Depth Information," Master's Thesis, Simon Fraser University, 2010.

[55] M. J. Tarr, P. Williams, W. G. Hayward, and I. Gauthier, "Three-dimensional object recognition is viewpoint dependent," Nature Neuroscience, pp. 275-277, 1998.

[56] P. Besl, R. Jain "Three-dimensional object recognition," ACM Computing Surveys, vol. 17, no. 1, pp. 75-145, 1985.

[57] I. Biederman, "Recognition-by-components: A theory of human image understanding," Psychological Review, pp. 115-147, 1987.

[58] S. Belongie, J. Malik, and J. Puzicha, "Matching Shapes," IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 454-461, July, 2001.

[59] G. Mori, S. Belongie, J. Malik, "Efficient Shape Matching Using Shape Contexts," IEEE PAMI, vol. 27, no. 11, pp. 1832-1837, Nov. 2005.

[60] M. Fischler, R. Elschlager, "The Representation and Matching of Pictorial Structuresv, "IEEE Computer Society, vol. C-22, no. 1, pp. 67-92,1973.

[61] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[62] T. Tuytelaars and L. Van Gool, "Content-based image retrieval based on local affinely invariant regions," International Conference on Visual Information and Information Systems, pp. 493-500, 1999.

[63] G. Friedland, K. Jantz, and R. Rojas, "SIOX: Simple Interactive Object Extraction in Still Images," Proceedings of the IEEE International Symposium on Multimedia (ISM), pp. 253-259, Irvine (California), December, 2005.

[64] A. X. Falcão, "Paradigmas de Segmenta¸c˜ao de Imagens Guiada pelo Usu´ario: Live-Wire, Live-Lane e 3D-Live-Wire," Tese (Doutorado) — Universidade Estadualde Campinas, 1997.

[65] E. N Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors," Graph. Models Image Process., vol. 60, no. 5, pp. 349-384, 1998.

[66] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik pp. 269-271,1959.

[67] N. Salman, "Image Segmentation Based on Watershed and Edge Detection Techniques: The International Arab Journal of Information Technology, vol. 3, no. 2, pp. 104-110, 2006.

[68] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, W. G. Aref, "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing," IEEE Transaction on Image Processing, vol. 10, no. 10, pp.1454-1466, 2001.

[69] R. Ohlander, K. Price, and D. Raj Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, vol. 8, pp. 313-333, 1978.

[70] F.R. Chen, L.D. Wilcox, D.S. Bloomberg, "A Comparison of Discrete and Continuous Hidden Markov Models for Phrase Spotting in Text Images," Document Analysis and Recognition, Proceedings of the Third International Conference (ICDIR), vol. 1, pp. 398- 402, 1995.

[71] V. Digalakis, S. Tsakalidis, C. Harizakis and L. Neumeyer, "Efficient speech recognition using subvector quantization and discrete-mixture HMMS," IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '99, vol. 2, pp. 569- 572, 1999.

[72] C. Neukirchen, J. Rottland, D. Willett, G. Rigoll, "A Continuous Density Interpretation of Discrete HMM Systems and MMI-Neural Networks," Speech and Audio Processing, IEEE Transactions on, vol. 9, no. 4, pp. 367-377, 2001.

[73] S. P. Lloyd, "Least squares quantization in PCM,"Institute of Mathematical Statistics Meeting, Atlantic City, NJ, September 1957; IEEE Transactions on Information Theory, pp. 129-136, March 1982.

[74] J. Max, "Quantizing for minimum distortion," IRE Trans. Information Theory, it-6, pp. 7-12, 1960.

[75] N. Otsu "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9: pp. 62–66, 1979