Pattern Based Object Segmentation (POS) Algorithm

Md Ziaul Karim

A thesis submitted to the Department of Computer Science and Engineering in partial fulfilment of the requirements for the

degree of

Master of Science in Engineering (Computer Science and Engineering)

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

July 2009

The thesis "***Pattern Based Object Segmentation (POS) Algorithm***", submitted by Md Ziaul Karim, Roll No. 100105039P, Registration No. ...., Session October 2001, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Master of Science in Engineering (Computer Science and Engineering) and approved as to its style and contents. Examination held on July 25, 2009.

Board of Examiners

1. ______________________________

Dr. M. Mahfuzul Islam
Assistant Professor
Department of CSE
BUET, Dhaka–1000

Chairman
(Supervisor)

2. ______________________________

Dr. Md. Saidur Rahman
Professor and Head
Department of CSE
BUET, Dhaka–1000

Member
(Ex–officio)

3. ______________________________

Dr. M. Kaykobad
Professor
Department of CSE
BUET, Dhaka-1000

Member

4. ______________________________

Dr. Md. Mostofa Akbar
Professor
Department of CSE
BUET, Dhaka-1000

Member

5. ______________________________

Dr. Ashfaqur Rahman
Assistant Professor
Department of CS
AIUB, Banani, Dhaka

Member
(External)

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Md Ziaul Karim

Date: 25 July 2009

Dedicated to my parents and wife for all their love and inspiration

Acknowledgements

First I express my heartiest thanks and gratefulness to Almighty Allah for His divine blessings, which made me possible to complete this thesis successfully.

I feel grateful to and wish to acknowledge my profound indebtedness to my supervisor Dr. M. Mahfuzul Islam, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology. The deep knowledge and keen interest of Dr. Islam in the field of image processing influenced me to carry out this thesis. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this thesis.

I would like to express my heartiest thanks, profound gratitude and deep respect to my classmate Dr. M. Ameer Ali, Assistant Professor, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh, for his opinion, advice, continuous encouragement, exchange of knowledge and valuable suggestions.

I would like to also recognize Dr. Manoranjan Paul, and Dr. Manzur Murshed, Gippsland School of Information Technology, Monash University, Australia, for their contribution in the pattern based video processing that motive me to use those patterns for segmenting objects using split and merge scheme.

Abstract

In the wake of increased use of multimedia applications for day-to-day normal activities, including business and entertainment, the pressure of introducing a generalized technique for segmenting an object in effective and efficient way becomes crucial for creative animations and providing low bit rate communications. All the existing techniques of image segmentation either require prior knowledge about the number of objects in the image or misclassify pixels to be included within the object or the both. The above limitations, along with the complexities of the existing segmentation techniques, make the process ineffective as the requirements of segmentation cannot be met in many cases. This thesis proposes a novel image segmentation algorithm called the *robust object segmentation based on pattern matching*, which introduces the region stability, connectivity and pattern matching techniques for splitting and merging an image to identify objects within it. The stability test classified the regions as the background, object and mixed regions, of which the most interesting mixed regions will be splitted recursively until a small block containing 16x16 pixels is obtained, otherwise being classified to be either the background or the object region. Each of the smallest mixed regions is then matched with one of the suitable patterns for replacing the contour of an object. The algorithm for the first time solves key problems of segmentation, is able to segment all types of known and unknown images and provides the superior performances compared with recently developed suppressed fuzzy c-means and object based segmentation using fuzzy clustering schemes in terms of its complexity and ability to segment objects, and therefore, hastening the arrival of new era in the world of multimedia applications.

Abbreviations

AMB	Active micro-blocks
B	Background
CT	Computerized tomography
dB	deciBel
EM	Expectation maximization
FCGS	Image segmentation using fuzzy clustering and integrating generic shape information
FCM	Fuzzy c-means
FCS	Fuzzy circular shell
FISG	Fuzzy image segmentation of generic shaped clusters
FKE	Fuzzy elliptic-ring
FKR	Fuzzy k-ring
GK	Gustafson-Kessel
HC	Hard clustering
ID	Identifier
LI	Light intensity
M	Mixed
MAP	Maximum a posteriori
MB	Micro-blocks
MRI	Thermal and magnetic resonance images
O	Object
OSF	Object based segmentation using fuzzy clustering
OSSM	Object segmentation based on split and merge algorithm
PC	Pattern codebook
PCM	Probabilistic c-means

PE	Processor element
RAG	Region adjacency graph
RCFCM	Rival checked fuzzy c-means algorithm
RI	Range image
RMB	Active-region micro-blocks
ROSP	Robust object segmentation using pattern matching
SE	Standard error
SFCM	Suppressed fuzzy c-means
SM	Split-and-merge
SMB	Static micro-blocks
SMEM	Split-and-merge expectation maximization
SMG	Split-and-merge with grouping segmentation
SPECT	Single photon emission computed tomography

Key Publications

[1] Karim Z., Paiker, N. R., Ameer Ali, M. Sorwar, G., and Islam, M. M., "Pattern based object segmentation using split and merge", IEEE International Conference on Fuzzy Systems, August 20-24, 2009. (Accepted).

[2] Faruquzzaman, A. B. M., Paiker, N. R., Arafat, J., Karim, Z., and Ameer Ali, M., "Object segmentation based on split and merge algorithm", IEEE TENCON, IEEE Region 10 Conference, pp 1-4, 2008.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

Image segmentation is the process of separating mutually homogeneous regions of interests i.e., assigning pixels to a region having common properties. This is particularly important as it is often the pre-processing step in many image processing algorithms. The objects are partitioned into a number of non-intersecting regions in such a way that each region is homogeneous and the union of two adjacent regions is always non-homogeneous. Most natural objects are non-homogeneous however, and the definition of what exactly constitutes an object depends very much on the application and the user, which contradicts the generic definition of image segmentation. There is actually no formal unified standard definition of image segmentation, which can be viewed therefore, as an ad hoc process which depends on the emphasis given to the specific object features to be segmented in a particular application, and in the way they balance and compromise one desired property against another [1, 20, 26, 54, 64]. That is the reason for image segmentation being a very difficult and challenging research topic.

Though image segmentation is very difficult, it plays a fundamental role in the field of image processing, image analysis, and coding with a wide range of applications not limited to: automatic car assembling in robotic vision, airport identification from aerial photographs, security systems, object-based image identification and retrieval, object recognition, second generation image coding, criminal investigation, computer graphic, pattern recognition, and diverse applications in medical science such as cancerous cell detection, segmentation of brain images, skin treatment, intrathoracic airway trees, and abnormality detection of heart ventricles [29, 54, 58, 73].

An image may be defined as two dimensional function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the grey level intensity of the image at that point. When x, y, and amplitude value of f are all finite, discrete quantities, the image is called digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels and pixels. Pixel is the term most widely used to denote the elements of a digital image [36].

There are no clear-cut boundaries in the continuum from image processing at one-end to computer vision at the other. One useful paradigm is to consider three types of computerized processes in this continuum: low, mid and high level processes. Low-level processes involve primitive operation such as image processing to reduce noise, contrast enhancement and image sharpening. Mid-level processing on image involves description of those objects to reduce them to a form of suitable for computer processing and classification for individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images. Finally, high-level processing involves "making sense" of an ensemble of recognized objects, as in image analysis, and at the far end of the continuum, performing the cognitive function normally associated with vision [36].

The application of digital images is rapidly expanding due to the ever-increasing demand of computer, Internet and multimedia technologies in all aspect of human lives, which makes digital image processing a most important research area. Digital image processing encompasses a wide and varied field of applications from medical science to document processing and generally refers to the manipulation and analysis of pictorial information. Image processing is mainly divided into six distinct classes: i) representation and modelling, ii) enhancement, iii) restoration, iv) analysis, v) reconstruction, and vi) compression. Image analysis embraces feature extraction, segmentation and object classification [11, 26, 36, 45, 48], with segmentation for instance, being applied to separate desired objects in an image so that measurements can subsequently be made upon them.

Segmentation is the process of subdividing an image into its constituent regions or objects. The level to which the subdivision is carried depends on the feature used for measuring the homogeneity of objects. That is, segmentation should stop when the objects of interest in an application have been isolated [64].

Image segmentation algorithms generally are based on one of the two basic properties of intensity values: discontinuity and similarity. In the first category, the approach is to partition image based on abrupt changes in intensity, such as edges in an image. The principal approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined criteria. Thresholding, region growing and region splitting and merging are examples of methods in this category. The segmentation approaches used in compound document compression can be grouped into 3 classes: (i) object-based segmentation (ii) layer-based segmentation, and (iii) approximate block-based segmentation.

In object-based segmentation, an image is divided into regions, where each region follows exact object boundaries. An object may be a photograph, a graphical object, a letter, etc. In principle, this method may provide the best compression, since it provides the best match between a data type and the compression method most suitable for this data type. In reality, the best compression may not be achievable for the following reasons. Coding the object boundaries requires extra bits, and the typical algorithms, used for lossy image compression [2], are designed to operate on rectangular objects. They can operate on objects with nonrectangular boundaries, but the compression performance will suffer. Complexity is another drawback of this method, since precise image segmentation may require the use of very sophisticated segmentation algorithms [29, 58, 73].

Layer-based segmentation can be regarded as a simplified version of the full object-based segmentation. The original page is divided into rectangular layers, where each layer can have one or more objects, and "mask" planes. A mask plane tells which pixels of a particular layer should be included in the final composite page. Each layer is compressed with a specific compression method. The advantages of this approach are simplified segmentation (when the number of layers is less than the number of objects), and a better match between layer boundaries and the compression algorithms (since layers are rectangular regions). Standard, off-the-shelf compression methods can be easily incorporated into this structure. The drawbacks of this method are: mismatch between the compression method used for a particular layer and the data types (when several various objects are included in the same layer), mismatch between the object boundaries and the compressed region boundaries, and an intrinsic redundancy, due to the fact that the same parts of the original image appear in several layers [29, 58, 73].

Approximate block-based segmentation can also be regarded as a simplified version of the full object segmentation. Each region follows approximate object boundaries, and is made of rectangular blocks. The size of the blocks may vary within the same region to better approximate the actual object boundary. The advantages of this approach are: simplified segmentation, better match between region boundaries and the compression algorithms, and the lack of redundancy, which may be present in the layer-based approach. The potential drawbacks are the potential loss in the compression performance compared to the true object-based segmentation, and the need to slightly modify the off-the-shelf algorithms to work on nonrectangular regions. Note that the segmentation performed in this case is done with the purpose of optimizing the compression performance, and may not be appropriate for other uses, such as OCR, image enhancement, etc [29, 58, 73].

Different applications require different types of digital image. The most commonly used images are *light intensity* (LI), *range (depth) image* (RI), *computerized tomography* (CT), *thermal and magnetic resonance images* (MRI). The research published to date on image segmentation is highly dependent on the image type, its dimensions and application domain and so for this reason, there is no single generalized technique that is suitable for all images [49, 58].

In some situations, such as industrial inspection applications, at least some measure of control over the environment is possible at times. The experience image processing system designers invariably pays considerable attention to such opportunities. In other applications, such as autonomous target acquisition, the system designer has no control of the environment. Then the usual approach is to focus on selecting the types of sensors most likely to enhance to the objects of interest while diminishing the contribution of irrelevant image detail. A good example is the uses of infrared imaging by the military to detect objects with strong heat signatures, such as equipment and troops in motion.

Bandwidth is a very important limiting factor in application of image segmentation. Several segmentation schemes require morphological analysis of the different regions, and multiple passes over the image being segmented. However, each pass normally requires loading memory data from slow to fast memory (L1 cache, etc.), which is a slow process. Segmentation solutions based on multiple passes are much slower (or costly) than what can be expected by, for instance, counting the number of operations. Thus, an ideal solution would use a single pass to decide on the type of image region. Such solution would be very difficult with arbitrary shapes of segmentation regions, but it is feasible if we consider only a pre-defined shape. For example, we can use rectangular blocks, and decide the image type based only on the properties of the pixels inside the block. We call such technique as block classification. This technique is theoretically sub-optimal, since it must classify all pixels in the block in the same manner, even if the block contains the boundary between two regions. Another potential problem would be the fact that it does not consider the pixels in the block's neighbourhood [49].

Despite the numerous techniques developed in the literature [1, 20, 54], image segmentation is still a subject of on-going investigations and it cannot be conclusively stated that the segmentation problem has been completely solved because of the applications, and image and object diversities. As a consequence, the task of choosing the best method for a specific application is still a challenging task. Several survey papers [1, 20, 54, 58, 64] cover the major image segmentation techniques available in the literature. The tree diagram of image segmentation techniques is shown in Figure 1.1. Segmentation techniques can be broadly categorized into two approaches: (i) boundary-based methods and (ii) region-based methods. Basically, the first approach is based on discontinuity and tends to partition an image by detecting isolated points, lines and edges according to abrupt changes of local properties. The regions are then deduced from their boundary. The usual tools that are employed in boundary-based methods include local filtering approaches such as Canny edge detector [16] or energy minimization like the active contour model (i.e. snake model) [50] and balloon models [22]. First approach is not efficient in segmentation process due to the following reasons:

(i) They do not exploit spatial information.

(ii) They are domain-dependent method.

(iii) Thresholding is static in nature.

(iv) Two adjacent regions do not share the same boundary information.

(v) Some simply closed discrete curves do not satisfy Jordan theorem.

The algorithms from the second approach exploit the homogeneity of spatially dense information (e.g. intensity, colour, texture properties, etc.) to produce the segmentation result. It includes thresholding [68], clustering [72], region-growing [66], region splitting [44] and merging [40]. Both types of approaches have their advantages and disadvantages. To improve the segmentation results, a strategy consists in combining these two approaches in order to obtain a robust segmentation by exploiting the advantages of one method to reduce the drawbacks of the other. Different frameworks have been proposed [17, 67]. Zhu [74] develops a unifying framework that combines the attractive geometrical features of deformable models and the statistical techniques of region growing. Germond *et al.* [35] proposed to mix several types of information and knowledge provided and used by complementary individual systems like a multi-agent system, a deformable model and an edge detector within a cooperative framework. Moreover, Geiger *et al.* [33] developed another approach based on mathematical models. This is an attempt to unify different methods of image segmentation under a common framework based on the Bayesian theory.

A mixed strategy in combination of these two approaches are also available but they are less efficient as they suffer from huge computational complexities and inefficient with respect to time requirement. There are also other image segmentation techniques which are not classified in either category, including those based upon Markov random models, Bayesian principles and the Gibbs distribution, with further details being given in [7, 34]. These approaches are also not widely acceptable for their shortcomings. Thereby researchers now-a-days are highly interested to improve the performance and quality of the second approach to meet the image segmentation demand as they provide following advantages:

(i) It is very simple in nature.

(ii) It evaluates all the spatial properties of the image.

(iii) Data structure representation is easier than boundary based image processing.

(iv) It gives dynamic solution to image segmentation.

(v) It is based on a specific region model described by a homogeneity predicate.

(vi) It provides well organized iterative solution with less computational complexities and hazards.

Segmentation is certainly one of the most challenging tasks in image processing and computer vision for many reasons, some of which are [1, 20, 54, 58]:

(i) Image types such as MRI, CT or *Single Photon Emission Computed Tomography* (SPECT) contain inherent constraints that make the resulting image noisy and may include or introduce some visual artefacts.

(ii) Image data can be ambiguous and susceptible to noise and high frequency distortion as in SPECT imaging for instance, where object edges become fuzzy and ill-defined.

(iii) The shape of the same object can differ from image to image due to having different domain and capturing techniques as well as various orientations. An object's structure may not be well defined in many natural images and can also be very hard to accurately locate the contour of an object.

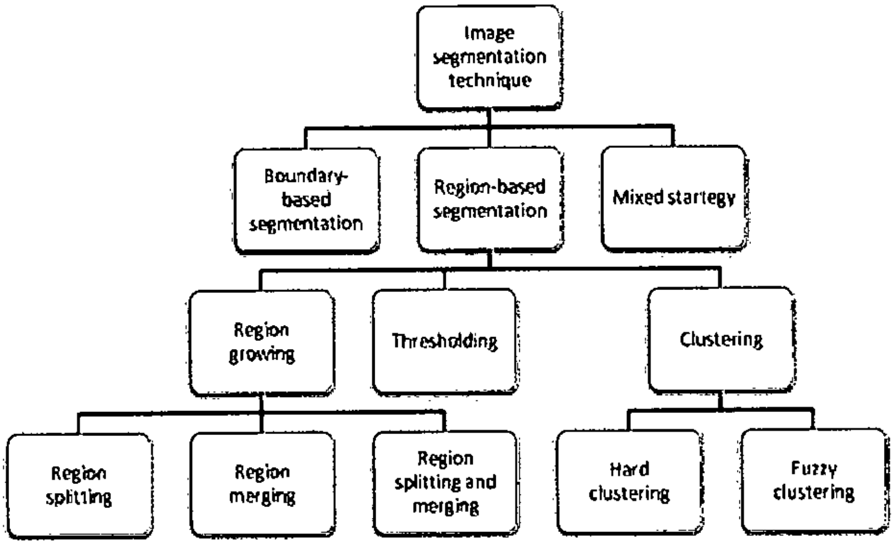


Figure 1.1: General classification of image segmentation technique

(iv) The distributions of gray-scale pixel values of the same object are not the same for all images and even in the same image; pixels belonging to the same class may have different intensities and distributions.

(v) Objects to be segmented are highly domain and application dependent; for example, in order to automatically estimate the myocardial wall thickness from a captured X-ray image of the human heart region, the inner and outer contours of the heart's left ventricle may be the two objects required to be segmented, while for another application, the entire heart may need to be segmented.

(vi) The properties of an object can differ in their representation depending upon the type of image and its domain, so there needs to be a trade-off between the desired properties that are to be employed for segmentation. For example, some gray scale images have a Poisson distribution, though this would not be true for either an RI or MRI image, so the segmentation strategy requires both semantic and *a priori* information concerning the image type and with other relevant object information such as the number of objects in the image.

Region-based image segmentation is a procedure to subdivide an image into its constituent parts or objects, called regions, using image attributes such as pixel intensity, spectral values, and/or textural properties [69]. The basic scheme consists of joining adjacent pixels to form regions; adjacent regions are then merged to obtain larger regions. Region growing by merging is a bottom-up procedure that groups pixels or regions into larger regions. Region growing by splitting, a top-down approach, is just opposite to the merging. If a region does not satisfy the homogeneity criteria then it is subdivided into sub-regions. Region splitting-and-merging technique solves the region growing problem in two stages: the Split and the Merge stages. The Split stage is a pre-processing stage that aims to reduce the number of merge steps required to solve the problem. This approach firstly splits the objects based on the threshold used for homogeneity and then merges splitted regions those are identical with respect to some predefined threshold used for homogeneity to form final segmented regions or objects.

Fuzzy image segmentation techniques [54 depends on fuzzy set theory of mathematical models, genetic algorithms and neural networks, and are used in diverse applications including image processing, pattern recognition, robotic vision, engineering tools, security and computer vision systems. Fuzzy image segmentation techniques, are broadly classified into six categories [25]: i) fuzzy geometric, ii) fuzzy thresholding, iii) fuzzy integral-based, iv) fuzzy rule-based, v) soft computing-based and vi) fuzzy clustering. Clustering is the process of separating or grouping a given set of unlabeled patterns into a number of clusters such that the patterns drawn from the same cluster are *similar* to each other in some sense, while those are assigned to different clusters are *dissimilar* [11, 26, 36]. Most of the time, objects are defined by a set of features and so those with similar features are classified into one cluster [26].

There are mainly two types of clustering, namely *hard* (*crisp*) (HC) and fuzzy-based [45]. In a HC algorithm [48, 54], the decision boundary is fully defined and one pattern is classified into one and only one cluster, i.e. clusters are mutually exclusive [11, 26]. However in the real world, the boundaries between clusters are not clearly defined. Some patterns may belong to more than one cluster and so in this case, fuzzy-based clustering techniques [1, 5, 20, 64, 73] provide a better and more efficient approach for classifying these patterns by assigning a membership value to each individual pattern. Fuzzy clustering algorithms are broadly classified into two groups: i) classical and ii) shape-based [36]. There exist many classical fuzzy clustering algorithms in the literature, among the most popular and widely used being: i) *fuzzy c-means* (FCM) [12], ii) *suppressed fuzzy c-means* (SFCM) [27], and iii) *probabilistic c-means* (PCM) [51], while from a shape-based fuzzy clustering viewpoint, well-established and popular algorithms include: i) *Gustafson-Kessel* (GK) [39], ii) circular shape-based [55], iii) elliptical shape-based [32] and iv) *object based segmentation using fuzzy clustering* (OSF) [5] techniques.

Segmentation is however, very challenging because of the multiplicity of objects in an image and the large variation between them. In many object based image segmentation applications from robotic car assembly through to medical imaging, the number of clusters is known *a priori*, for this reason, clustering algorithms are used for object based image segmentation, though even so, the segmentation performance of these clustering algorithms such as [12, 27, 51] is still highly dependent on the features used and types of object in an image, which ultimately limits their generalization capability. As humans exploit shape as a perceptual attribute in both detecting and recognising objects, this motivates the exploration of ways to integrate shape-based information into the clustering framework in order to segment objects in an image. Existing shape-based clustering techniques, such *fuzzy k-ring* (FKR) [55] and *fuzzy elliptic-ring* (FKE) [32] are all characterized as able to only accurately segment objects having ring, compact spherical, elliptical or a combination of ring and elliptically shaped objects. Most natural objects however, are neither ring nor elliptic in shape so the performance of these algorithms is compromised. An alternative shape-based clustering approach is the GK algorithm [39] which adopts the local topological structure of the shape of a cluster though this does not explicitly consider shape information and so is unable to segment arbitrary shaped objects satisfactorily. To achieve this goal, strategies that embed generic shape information within the fuzzy clustering framework need to be developed. Ameer *et al.* [6] introduced the *fuzzy image segmentation of generic shaped clusters* (FISG) algorithm that attempted to incorporate generic shape information, and while it segmented certain arbitrary shaped objects well, it possessed a number of limitations including: i) inaccurate updating of the object shape contour, ii) erroneous shape representation due to using a Bezier curve with a large number of control points; and iii) the overlapping of initial shape descriptors. To explicitly address these issues a new OSF algorithm [5] that reduces over-scaling and generates a shape contour representation either automatically by any clustering algorithm or using B-splines for a given set of significant shape points. To incorporate generic shape information, shape initialisation for each object has to be either manually or automatically provided, so OSF uses the GK algorithm for automatic shape initialisation because it locally adapts the distance metric to the shape of the cluster by estimating the cluster covariance matrix [39]. Though OSF outperforms above mentioned clustering algorithms it does not provide better result where number of clusters are not known *a priori*. At the same time it also suffers from scaling and other problems in object diversities.

Though, several algorithms [1, 20, 29, 54, 58, 64, 73] are developed to overcome various difficulties in image segmentation, the *split-and-merge* (SM) algorithm [58] is one of the efficient and well recognized algorithms which directly or indirectly widely used in many applications. The SM algorithm firstly splits the objects based on the threshold used for homogeneity and then merges the splitted regions those are identical with respect to some predefined threshold used for homogeneity to form final segmented regions or objects. But, the problems of the SM algorithm are that both splitting and merging steps are highly dependent on the thresholds used. Also, there are a huge number objects having different variations among them [4]. For these reasons, the SM algorithm is unable to segment all types of objects in an image within its general framework. This motivated to develop a new segmentation algorithm based on the SM algorithm considering the following motivated works as mentioned below. The motivated work diagram is shown in Figure 1.2.

(i) The pattern based video processing is well known algorithm in the literature [60]. But, the existing SM algorithm does not consider pattern based segmentation that will reduce the segmentation processing time. This motivates to consider region based image segmentation technique.

(ii) The basic scheme of the SM algorithm is that it is highly dependent on the threshold value. The existing SM algorithm along with its modified algorithms available in the literature do not provide any means of updating this threshold value based on the object feature diversities as they do not consider the inter-object variability [1, 20, 29, 54, 58, 64, 73]. For this reason unsupervised image segmentation cannot be obtained by the existing SM algorithms and this motivates to consider a fiducial limit of T-test as threshold value for both splitting and merging based on the inter-object variance which is updated after each merging operation.

(iii) Since the block of pattern is matched with the region of the image, the accuracy of the segmentation performance will highly dependent on the patterns that are used for matching purpose. In [56, 59, 61, 62, 63,] 32 patterns are used in video processing, these patterns will be considered to assess the segmentation performance of the proposed algorithm.

(iv) Most of the existing algorithms fail to segment real time images as the number of objects are not known *a priori*. The static thresholding of the existing SM algorithms results wrong segmentation of real time images. This motivates to incorporate fiducial limit as thresholding when applied T-test for checking object stability and for this reason the number of objects need not to be known *a priori* as the threshold is generated from the data available to the input images and dynamically updates itself throughout the merging process.

(v) To improve the performance of the block based or pattern based clustering algorithm, it needs to incorporate 8-connecitivity instead of 4-connectivity in the segmentation framework for merging process to produce final segmented regions or objects due to considering the weakly connected regions also.

(vi) In the pattern matching procedure, the image region is replaced by the existing pattern which causes some pixel loss. Since the pattern matching procedure will cause some pixel loss during segmentation process, this motivates to consider small sized pattern with different shapes and orientation to reduce the pixel loss.

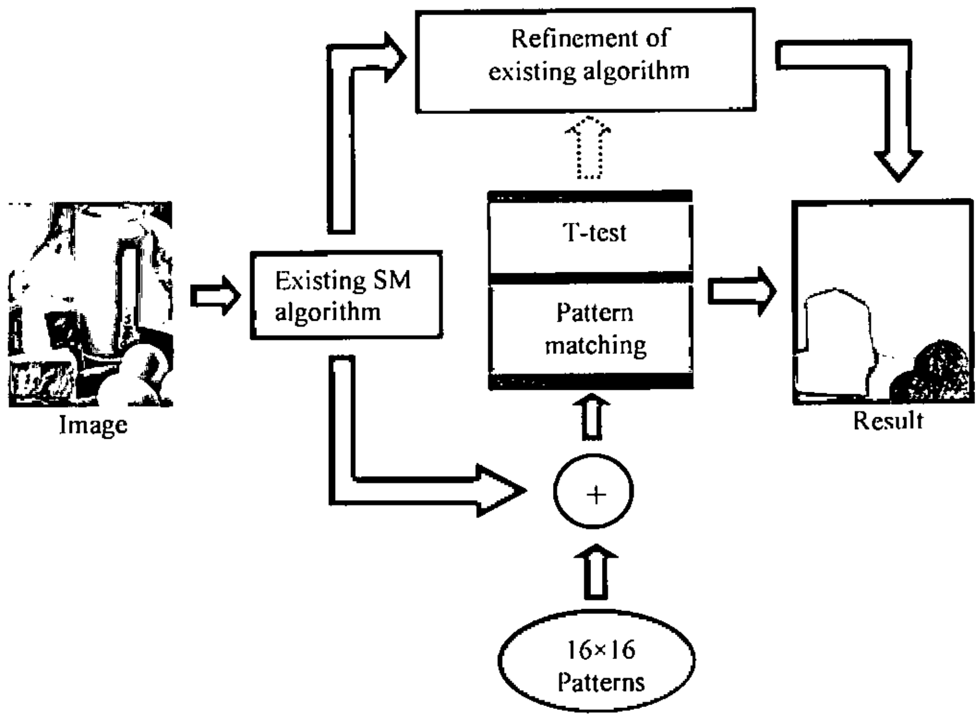


Figure 1.2: Motivated work diagram.

Addressing the above mentioned issues and with the aim of generalizing the SM algorithm and to introduce a new segmentation dimension, this thesis contributes to the following fields:

(i) This thesis has analysed the existing SM algorithm along with its modified algorithms available in the literature and has consolidated the useful consideration and proposal of various researchers to formulate a strong base of knowledge on segmentation techniques. In real time application it is better to accept little bit erroneous information rather than high access time. With the aim of reducing segmentation processing time, a new dimension of image segmentation technique is introduced in this thesis considering pattern based segmentation. This thesis introduces a newly developed algorithm namely *robust object segmentation using pattern matching* (ROSP).

(ii) First time in the ROSP algorithm, 8 patterns having 16×16 size are used in segmentation process. Since, the partially accepted image blocks (mixed regions) are replaced with the matched pattern, these causes some pixel loss and hence shape distortion. To reduce the pixel loss and shape distortion, 32 patterns are used instead of 8 that are already applied in video processing.

(iii) The existing region based segmentation algorithms could not dominate in image segmentation domain over the clustering algorithms due to its aforementioned inherent drawbacks. To enhance the performance of the proposed ROSP algorithm, the image feature stability applying T-test is used for splitting the objects in an image and then image feature stability, 8-connectivity, and pattern matching are considered for merging the splitted regions to form the final segmented regions (objects). The 8-connectivity is applied to consider the weakly connected regions also instead of 4-connectivity and hence improves the segmentation performance of the ROSP algorithm. Moreover, unlike the various clustering algorithms the proposed ROSP algorithm number of object need not to be known *a priori*.

(iv) To assess the robustness of the proposed algorithm, the experiments were conducted for different images having different objects. The segmented results of the proposed algorithm are compared with the *suppressed fuzzy c-means* (SFCM) [27], and a newly developed shape-based fuzzy clustering algorithm namely *object based image segmentation using fuzzy clustering* (OSF) algorithm [5]. The qualitative analysis shows that the ROSP algorithm outperforms the SFCM and the OSF algorithms. Thus, it can be stated that the ROSP algorithm will assist the region based segmentation algorithms to be the most acceptable algorithms in image segmentation where the real time image segmentation is needed and will give the region based segmentation algorithms a dominating boost over the clustering algorithms.

The rest of the thesis is organized as follows: Chapter 2 contains the basic concept of the original SM algorithm as proposed by Pavlidis [64] and includes the brief description of few modified algorithms which are based on the original SM algorithm to outperform the SM algorithm by reducing few of its drawbacks. It has shown the motivation for multi-stage merging since the single-stage merging does not provide better performance to all type of objects. Finally, literary background of the proposed ROSP algorithms is discussed with mathematical and philosophical logics.

Chapter 3 introduces the proposed ROSP algorithm. This chapter describes the operating procedure of the ROSP algorithm, chronologically mentioning the consideration used by each step and finally unfolds the algorithm itself.

The qualitative result and analysis of the proposed ROSP algorithm compared with the SFCM and the newly accepted OSF algorithm is provided in Chapter 4. This chapter describes the comparative evaluation of each experiment of the ROSP.

Finally, Chapter 5 provides the conclusion which includes the merits and demerits of the ROSP algorithm and explores some future direction to improve the ROSP algorithm with a view to overcome the problem.

Chapter 2

Literature Review

A number of algorithms have been developed to overcome various difficulties in image segmentation. The *split and merge* (SM) algorithm is one of the efficient and well recognized algorithm which directly or indirectly used in many applications. Numerous researchers have performed their research work on this algorithm to triumph over its drawbacks for its sustainable and competent implementation. This Chapter has consolidated the useful consideration and proposal of various researchers to formulate a strong base of knowledge on segmentation techniques. It has also tinted few unsettled drawbacks of the SM algorithm that has opened the casement of brainstorming. The single stage employment of the SM algorithm cannot be applied globally on all types of images thereby multistage merging become essential to withstand feature diversities. This concept motivates for developing the proposed *robust object segmentation based on pattern matching* (ROSP) algorithm. This Chapter, therefore, describes the theoretical background of the ROSP algorithm.

2.1 Basic Idea on Region Based Image Segmentation

As mentioned in the previous Chapter, image segmentation is a vital field in image analysis, coding and understanding, with a wide diversity of application, ranging from car assembly, airport security, object recognition and second generation image coding, through to criminal investigative analysis and medical imaging. Image segmentation is also a key step in many approaches to data compression and image analysis. Region based image segmentation is a procedure to subdivide an image into its constituent parts or objects, called regions, using image attributes such as pixel intensity, spectral values, and/or textural properties [66]. Let R represent the entire image having different objects. Segmentation may be viewed as a process that partitions R into n sub-regions, R_1, R_2, ..., R_n, such that:

(a) $\bigcup_{i=1}^{n} R_i = R$,

(b) $\forall i : R_i$ is a connected region,

(c) $\forall i, j : i \neq j$ and $R_i \cap R_j = \phi$,

(d) $\forall i : P(R_i) = TRUE$,

(e) $\forall i, j : i \neq j$ and $P(R_i \cup R_j) = FALSE$,

Where, $P(R_i)$ is a logical predicate over the set of pixels in the set of pixels in R_i and ϕ is the empty set [64].

Proposition (a) indicates that segmentation must be completed, that is, every pixel must be in a region while the second one indicates the pixels belonging to a region must be connected. The proposition (c) represents that the regions must be disjoined and the proposition (d) deals with the properties that must be satisfied by the pixels in every segmented region R_i in such a way $P(R_i) = TRUE$ if all the pixels in R_i are equivalent with respect to predicate P. Finally, the proposition (e) indicates the regions R_i and R_j are different in the sense of predicate P.

It is very difficult to identify features for this method because features are dynamic in nature. The features appropriate for one image segmentation may not be appropriate for another image. Usually following features are used for region based image segmentation of gray scale image:

(i) Intensity of gray level,

(ii) Position of gray level,

(iii) Connectivity of regions,

(iv) Texture information (mean covariance, standard deviation, maximum or/and minimum gray level intensity),

(v) Boundary information,

(vi) Size, and

(vii) Shape.

Among these features few are selected optimally to form the feature vector. Different algorithms use different set of features. Ultimate goal of feature selection is that the image can be segmented into various regions in such a way that the aforementioned proposition (a-e) is fulfilled [15].

Basically, region growing is a general technique for image segmentation. The basic scheme consists of joining adjacent pixels to form regions; adjacent regions are then merged to obtain larger regions. Based on the use of fusion (or merge) and split operators, region extraction techniques may be classified as:

(i) The bottom-up approach which leads to merging algorithms consisting in aggregating small regions into larger ones.

(ii) The top-down approach which leads to splitting algorithms consisting in recursively dividing an image into smaller and smaller regions.

(iii) The mixed approach which leads to splitting algorithms introduced by Horowitz and Pavlidis [64], and combining splitting and merging.

All these methods partition original image by recursively splitting and/or merging its regions. Bottom-up, top-down and split and merge methods are regrouped into the general term of region based methods.

The definition of region-based methods required the knowledge of different features about regions. These features can be grouped into six basic requirements:

(i) Traverse a region in order to extract parameters such as mean, variance, histogram etc.

(ii) Find the set of regions being adjacent to another one.

(iii) Find the boundary of a region.

(iv) Find the boundary element shared by two regions or the two regions sharing a same boundary element.

(v) Traverse a region boundary in order to extract parameters: length, mean, gradient along the boundary, etc.

The calculation of all of these features involves the topology and/or the geometry of the segmented image. There is also a sixth requirement involving a hierarchical organization.

(i) Find the split that has produced a given region and the other regions generated by this split [14].

This list summarizes the basic requirements of region-based algorithms. Other thing level features like the set of regions included in another one can be required by region-based algorithms. Only a subset of these requirements is used by top-down and bottom-up methods. Bottom-up methods mainly use requirements 1 and 2 while top-down methods use requirements 1 and 6. Most of split and merge methods use about all of six requirements. Data structure used to find efficiently information required by split algorithms are often made of squares organized in quad trees or pyramids. These structures are then broken and reorganized into a Region Adjacency Graph in order to perform merge operations [14]. Various types of region growing methods are discussed in the following Sections.

2.2 Region Growing by Merging/Splitting

Region growing by merging is a bottom-up procedure that groups pixels or regions into larger regions. The simplest approach is pixel aggregation, which starts with a set of '*seed*' pixel. From these seed pixels regions grows by appending neighbouring seed pixels that have similar properties in the sense of the predicate *P*, which defines the process. Two immediate problems of this approach are:

(i) The selection of initial seeds that properly represent regions of interest.

(ii) The selection of suitable properties for including pixels into regions during the growing process [59].

When no *a priori* information about how to select the seed pixels is available, the procedure consists of computing the same set of properties for every pixel that will ultimately be used to assign pixels to regions during the growing process. The selection of similarity criteria (predicate P) depends on the problem under consideration [53].

Region growing by splitting a top-down approach is just opposite to the merging. If a region does not satisfy the homogeneity criteria then it is subdivided into sub-regions. This procedure continues until the entire regions satisfy the homogeneity criteria. The main drawback of this method is that adjacent regions may have identical properties, thus it does not give a global solution rather give local solution.

2.3 Region Splitting and Merging

An alternative to the previous methods as mentioned in Section 2.2 consists of initially subdividing the image into a set of arbitrary, disjoint regions and then to merge and/or to split the regions in an attempt to satisfy the following (a) to (c) conditions [64]. The *split-and-merge* (SM) algorithm being developed by Pavlidis [64] in 1974 based on this concept is still one of the most popular classical image segmentation algorithms and is widely used directly or indirectly in image processing. The SM algorithm is a useful and important algorithm in image processing and pattern recognition due to its simplicity, effectiveness, un-supervision and relative cost minimization. The SM algorithm firstly splits the objects based on the threshold used for homogeneity and then merges the splitted regions those are identical with respect to some predefined threshold used for homogeneity to form final segmented regions. Let R and P be the entire image and the predicate, respectively. If R is a square image, an approach for segmenting R consists of successively subdividing it into smaller and smaller square regions so that, for any region $R_i \subset R$, $P(R_i) = TRUE$. That is, if $P(R_i) = FALSE$; R_i has to be newly subdivided and so on. To satisfy the constraints (a) to (c) below only merging adjacent regions whose combined pixels satisfy the predicate P is required where two adjacent regions R_j and R_k will be merged if $P(R_j \cup R_k) = TRUE$. The SM algorithm may be summarized by the following steps:

(a) Split any region R_i into four almost equal regions where $P(R_i) = FALSE$.

(b) Merge any adjacent regions R_i and R_j for which $P(R_i \cup R_j) = TRUE$.

(c) Stop when no further merging or splitting is possible. Otherwise repeat steps (a) and (b).

The Split-and-Merge algorithm as proposed by Horowitz and Pavlidis [64] solves the region growing problem in two stages: the Split and the Merge stages. The Split stage is a preprocessing stage that aims to reduce the number of merge steps required to solve the problem. Though there exists a huge number of segmentation algorithm in the literature [1, 19, 20, 29, 54, 57, 58, 64, 71, 73], the SM algorithm is only considered in this thesis. For this reason, the related literature based on the SM algorithm is presented here. Major difficulties in the SM algorithm of Pavlidis are:

(i) It is difficult to identify the point to split a region.

(ii) It does not provide a unique solution: when starting with a different initial partition, we get a different result. A small difference introduced at a low level of the merging stage propagates and induces severe modifications in the final result.

(iii) The SM algorithm only gives a local optimization and not a global one. For instance, this procedure does not guarantee that each subset is merged with the best of its four supersets, but only that each free subset encountered in the scanning process does.

(iv) The consequence of the limitation of local optimization appears as instability in the final areas. Due to the competition between adjacent subsets, a small variation in the segmentation process induces large differences in the result. This variation can be:

(a) A geometrical translation of the picture (registration error),

(b) A scale factor on the luminance (photometric error),

(c) A noise on the signal.

Thus we say that the SM algorithm is not robust with respect to geometric and photometric disturbances.

(v) It is sensitive to a variation of the threshold value in the homogeneity predicate.

(vi) Feature selection need to be dynamic to get a better result.

(vii) Huge computation is required in each iteration and these computations are interdependent, thereby it slow downs the overall processing. Parallel implementation is also difficult due to interdependencies.

(viii) Even if it is possible to continue with parallel processing using a *region adjacency graph* (RAG) data structure, the worse case condition may raise load unbalancing problem.

(ix) Coherent regions created during the split may remain in the partitioning, because their existence is only reconsidered after a sequence of split operation.

(x) Two separate operations such as split and merge require a huge amount of processing time.

(xi) Smoothness of the image boundary is not maintained since the region is grown by combining the squared regions partitioned through a hard constant of homogametic criteria.

(xii) From a computational point of view, this region growing algorithm is representative of the class of irregular and dynamic problems which are most difficult to solve and need extreme knowledgeable agent to perform the task.

(xiii) Non-uniform problems are characterized by a behaviour that is data dependent and cannot be determined at compilation time.

(xiv) The SM algorithm has a very volatile behaviour, starting with a high degree of parallelism that very rapidly diminishes to a much lower degree of parallelism.

Based on the SM algorithm as proposed by Pavlidis [64] different models have been developed so far. SM modelling depends on the merging criteria for its constituent regions. Basically a merging criterion consists of two parts: a region model, describing each image region with a set of features, and a dissimilarity measure, defining a metric on the features of the region model. The range of possible region models reaches from simple models like uniform intensity values up to texture, shape or motion parameters. This thesis has considered gray scale image only. However, all presented criteria can be readily generalized to work on colour images. The better a region model matches the real image-data, the longer the minimum edge-weights remain small and the steeper is the relative increase as soon as the segmentation has reached its final state. This makes the segmentation process more robust to the selection of the fixed threshold for the stopping condition. A region is describes by its homogeneity property. A region is said to be unique if it satisfies the condition $P(Ri \cup Rj) = FALSE$, $\forall_{ij} i \neq j$. This homogeneity property basically defines a region model. Various types of homogeneity property measurement criteria [15, 40] are discussed in the following Sections.

2.3.1 Mean Gray Level Difference

The simplest region model is to describe each region R_i by its mean luminance μ_i. A straightforward possibility to define a dissimilarity measure on this model is to use the squared difference as follows [15]:

$$\omega_{ij}^{M} = (\mu_i - \mu_j)^2 \tag{2.1}$$

2.3.2 Ward's Criterion

Another measure which operates on the mean-gray level model is the Ward Criterion. The idea is to consider the model error for a region R_i that is defined as:

$$\xi_i = \sum_{x \in R_i} (f(x) - \mu_i)^2 \tag{2.2}$$

The dissimilarity associated with a pair of regions is defined as the additional total error that is introduced by merging the two regions and can be calculated by $\omega_{ij}^{W} = \xi_{ij} - \xi_i - \xi_j$ (with ξ_{ij} being the error after a hypothetical merge of R_i and R_j) [15]. After elementary implications, this can be expressed as:

$$\omega_{ij}^{W} = \frac{|R_i||R_j|}{|R_i| + |R_j|}(\mu_i - \mu_j)^2 \tag{2.3}$$

2.3.3 Mean/Ward Mixture

Neither the Mean-criterion nor the Ward-criterion produces a subjectively appropriate segmentation. A better criterion may be a compromise between the characteristics of Mean and Ward. For this reason, the geometrical mean of both the criteria ($\omega_{ij}^{G} = \sqrt{\omega_{ij}^{W} . \omega_{ij}^{M}}$) is treated as a new Mean-Ward Criterion [15]. Since the absolute value of the criterion is not important, the square-root can be ignored and hence resulting as follows:

$$\omega_{ij}^{G} = \frac{|R_i||R_j|}{|R_i| + |R_j|}(\mu_i - \mu_j)^4 \tag{2.4}$$

2.3.4 Linear-intensity Model

Because of intensity effects, natural images seldom consist of completely homogeneous regions. Almost all regions that are perceived as homogeneous contain a small gray level intensity gradient. Therefore, it is sensible to use a region model that is capable of describing slowly varying intensity gradients. A possible region model defines the gray level distribution as $f'(x,y) = \alpha + \beta x + \gamma$ with the three parameters α, β, γ. For each individual region, these parameters are estimated from the image data using a least-squares approach [15]. Comparable to the Ward-criterion, we define this model error as:

$$\xi_i^L = \sum_{x \in R_u} (f(x,y) - f'(x,y))^2 \tag{2.5}$$

and the region dissimilarity as:

$$\omega_{ij}^L = \xi_{ij}^L - \xi_i^L - \xi_j^L \tag{2.6}$$

2.3.5 Border Criterion

Although the linear-gray level model handles well most regions occurring in natural images, the model has two main drawbacks such as (i) it is rather computationally intensive, and (ii) it still cannot handle all cases of small gray level variations. Especially curved surfaces have complicated gray level distributions. Both problems can be circumvented by using the following Border criterion. Let $B_{ij} = \{x_k, x_l\}$ be the set of pairs of pixels along the common boundary between region R_i and R_j (with $x_k \in R_i$ and $x_l \in R_j$). The Border-criterion can be defined as the sum of squared differences along the boundary [15]:

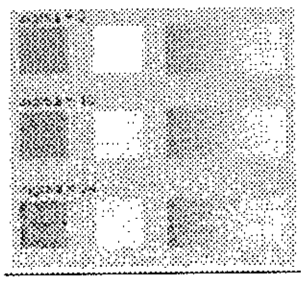
$$\omega_{ij}^B = \frac{1}{B_{ij}} \sum_{x_k, x_l \in B_{ij}} (f(x_k) - f(x_l))^2 \tag{2.7}$$

2.3.6 Comparison

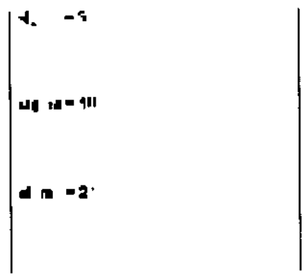
In natural image segmentation, several classes of commonly occurring difficulties can be identified. The robustness of each criterion on the problem classes can be evaluated [15]. In the following Sections, some problem classes are described in more detail:

2.3.6.1 Noise

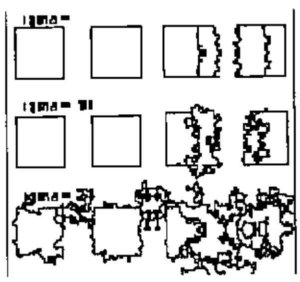
Camera noise has a well visible effect at the beginning of the segmentation process. Dissimilarity measures which are normalized to their region sizes, like Ward's criterion, give superior results, because single noisy pixels introduce no large overall error [15]. Noisy image segmentation based on various criterions is shown in Figure 2.1 for visual perception.



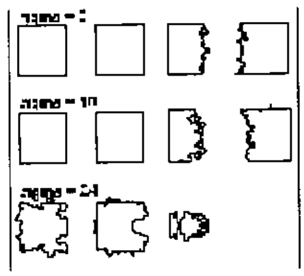
(a) Input Image



(b) Mean criterion



(c) Ward criterion

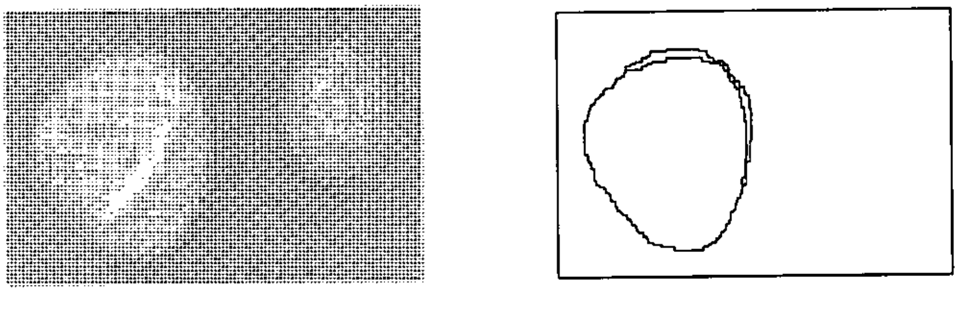


(d) Linear-intensity criterion

Figure 2.1: Noisy image segmentation based on various criterions [15].

2.3.6.2 Blurred and Double Edges

Objects which are out of camera focus appear with blurred edges in the image. This can lead to an over segmentation into many thin rings around the object boundary. The Linear- gray level criterion can approximate the blurred edge with a single region if the object boundaries are straight lines. Curved boundaries can be handled by the border criterion. However, the more general model of the Border criterion has the disadvantage to ignore the pixels inside a region. Thus, it is possible for an object to grow along its unsharp border until it is completely merged with the background as shown in Figure 2.2. After segmentation with a too low threshold, objects seem to have double edges. Because of its tendency to merge small regions, the Ward-criterion can remove these double edges well.

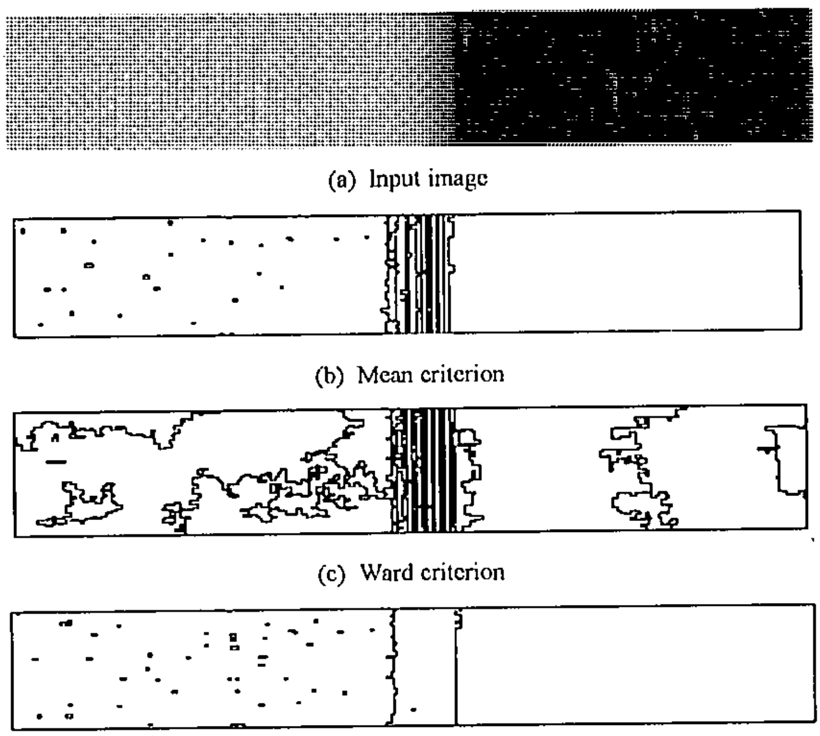


(a) (b)

Figure 2.2: (a) Input Image (b) The right object is removed by the Border criterion [40].

2.3.6.3 Subjective Evaluation

As can be seen in Figure 2.3, the Ward criterion has a tendency to split large regions into several segments, while mean criterion removes large regions equally likely as small regions.



(a) Input image

(b) Mean criterion

(c) Ward criterion

(d) Linear-Intensity criterion

Figure 2.3: Unsharped image segmentation based on various criterions [15].

Figure 2.1 shows the segmentation using the Mean-Ward criterion. The result is much more subjectively pleasing as the large regions are preserved and much of the text is kept.

2.3.7 Stopping Criterion

To ease the selection of robust stopping thresholds, it is desirable that the weights remain small for as long as possible and show large growth as soon as the model cannot adapt to the enlarged regions anymore. The Mean criterion increases before a subjectively recognizable reason is visible. On the other hand, both Ward and Linear-Luminance show a very steep increase in model error [15].

2.3.8 Motivation for Multistage Merging

Each criterion, as mentioned above, shows both advantages and disadvantages. To choose a single criterion for the complete segmentation process results in a dissatisfactory segmentation and hence motivates a multi-stage approach. A criterion is used as long as it can well handle the current configuration. Then, the criterion is exchanged by another one. By using several stages, the selection of an appropriate threshold in the stopping criterion is not critical. It should be chosen sufficiently low to ensure passing the control to the next criterion before the situation exceeds the capabilities of the criterion's region model. A sequence of criteria which produced good results is:

- Ward criteria: remove much of the image noise and eliminate double edges.
- Mean-Ward criteria: does the main work of merging regions.

Border criteria: finally merges regions in which pixel intensity effects play a central role in the segmentation process [15].

2.4 Modification of the SM Algorithm

Since the SM algorithm as mentioned in Section 2.3 can not give optimal result due to image character diversities, various researcher have been developed new algorithms which basically based on the concept of the SM algorithm. Their effort could bring the SM concept to a formidable stage but could not establish its total acceptability. Few of those modified SM algorithms are highlighted in the following Sections.

2.4.1 Modification of the SM Algorithm Based on Region Variability

Generally objects within an image minimize their intra-object variability while maximize the inter-object variability. Basing on this concept, another region model namely *Split-and-Merge with Grouping Segmentation* (SMG) is proposed by Veenman *et al.* [71] based on a common criterion to quantify region homogeneity that is the sum-of-squared-error and is as follows:

$$\sum_{i=1}^{M}\sum_{x\in R_i}\|x-\mu(R_i)\|^2 \quad \text{where,} \quad \mu(Y)=\frac{1}{|Y|}\sum_{x\in Y}x \tag{2.8}$$

To minimize the intra-region variability, the sum-of-squared-error criterion is used. Further, the joint intra-region variability constraint is used as a minimum variance for the union of two regions [71]. This leads to the following region model:

$$\min\sum_{R_i\in R}|R_i|\operatorname{var}(R_i) \quad \text{where,} \quad \operatorname{var(Y)}=\frac{1}{|Y|}\sum_{x\in Y}\|x-\mu(Y)\|^2 \tag{2.9}$$

Subject to,

$$\forall R_i R_j, i\neq j : \operatorname{var}(R_i \cup R_j) > \sigma^2_{\max} \tag{2.10}$$

In general, the optimization of this model leads to region having a variance below $\sigma^2_{\max}$. However, there are some rare situations in which the variance of individual region can exceed this limit. The solution of this problem is if there exists a region R_a for which the variance exceeds the maximum variance limit $\sigma^2_{\max}$, then all sub-regions of R_a have a distance smaller than $\sigma^2_{\max}$ to the centre of region R_a or they can not be separated from R_a without violating the joint variance constraint in equation above with some region R_b which is:

$$\operatorname{var}(R_a)\geq\sigma^2_{\max}\Rightarrow\forall_x\in R_a:\left(\|x-\mu(R_a)\|^2\langle\sigma^2_{\max}\vee\exists R_b:\operatorname{var}(R_b\cup\{x\})<\sigma^2_{\max}\right) \tag{2.11}$$

Prove of the above assumption is given in [71]. In this model, the connectivity is imposed as hard constraints because:

- Connected segments are usually bigger so that more reliable statistics can be maintained.
- The optimization problem simplifies into a one criterion optimization problem, since it does not need to optimize a connectivity criterion as well.
- Can avoid the introduction of an additional parameter that is needed in case of resulting multi-criterion optimization problem which is solved by weighting homogeneity and a connectivity term.

This region model is implemented through split-and-merge algorithm based on [42] and [64]. The homogeneity predicate used in these models is the Chi-squared test, where the rejection probability is set to 0.5. Since only merges on the same level of the quad tree are allowed, a final grouping step is incorporated. In the grouping stage, all regions those are homogeneous are grouped together. The regions are homogeneous if the distance between their means is below a certain maximum threshold δ_μ. Finally, these regions are grouped together with their neighbouring segments if their size is relatively small (default 0.2%) with respect to their largest neighbour. This split-and-merge algorithm does not need to know the number of segments a priori. Moreover, the segment connectivity is always satisfied [71]. The result is shown in the Figure 2.4 below:

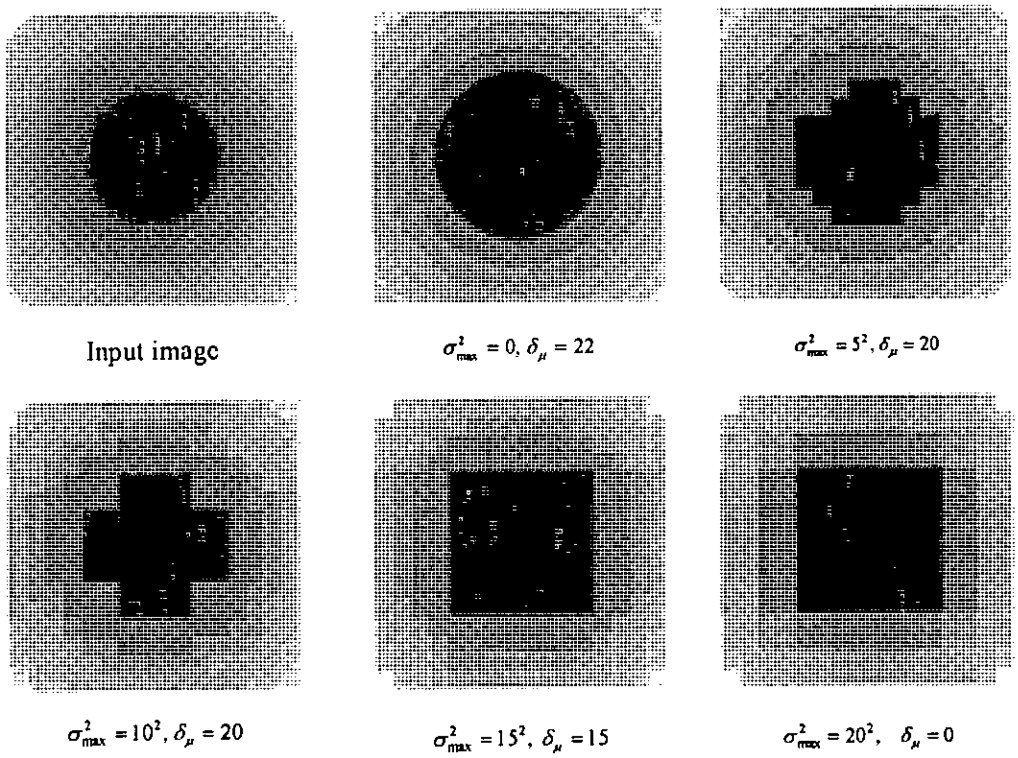


Figure 2.4: Application of the SMG algorithm to circular input image with various σ^2_{max} and δ_μ parameter settings [71].

The basic limitation of this algorithm is that, it behaves abruptly different with the change of threshold value as the threshold is user defined. More so the object smoothness cannot be maintained in this algorithm [28-29].

2.4.2 Modification of the SM Algorithm for Parallel Implementation

One of the major problems of the SM algorithm is that it requires more processing time than expectation [42, 54]. It is because of calculation for both splitting and merging differently and in both stages a number of iterations is required to attain the threshold level. This problem can be solved by applying various techniques, such as (i) using multiprocessor to segregate the processing and distributing the load to various processors, (ii) having multi-agent environment where we can blend genetic algorithm with the SM algorithm, and (iii) may be done by recursively executing the splitting and merging process simultaneously.

One algorithm is proposed by M.D.G. Montoya *et al.* [54] where the region growing process transforms a set of pixels into a set of labelled regions and goal is to design strategies for these transformations to be carried out concurrently, in such a way that each processor must work on disjointed subsets of pixels or regions. The Split phase works on the set of individual pixels of the image and is an inherently parallel task. So, initially, an image of size $N=N_1 \times N_2$ is partitioned into $P=P_1 \times P_2$ sub-images containing $\frac{N_1}{P_1} \times \frac{N_2}{P_2}$ adjacent pixels. Neighbouring sub-images will be located into neighbour processor elements (PEs). At the beginning of the merge phase the computational work load of a processor element depends on the set of local regions created during the split phase and on the set of edges created between local regions and neighbouring processor regions. The implementation of the Split-and-Merge parallel region growing algorithm can be described by the following steps [54]:

(a) A set of $P_1 \times P_2$ sub-images are created and mapped onto the set of PEs.

(b) Each PE executes the split phase on its own sub image using a bottom-up strategy and assigns an identifier (*ID*) to each of the created regions.

(c) Each PE builds its own local weighted graph by creating the set of edges between its own local regions which satisfy the vicinity and homogeneity criteria.

(d) PEs exchange information about the regions at the border of the initial sub-image and create the set of weighted edges between regions that belong to different neighbouring PEs.

(e) Each region determines the linked regions that best suit the homogeneity criterion as candidate regions to be merged with.

(f) PEs must exchange information about the best selection for those linked regions located at different PEs.

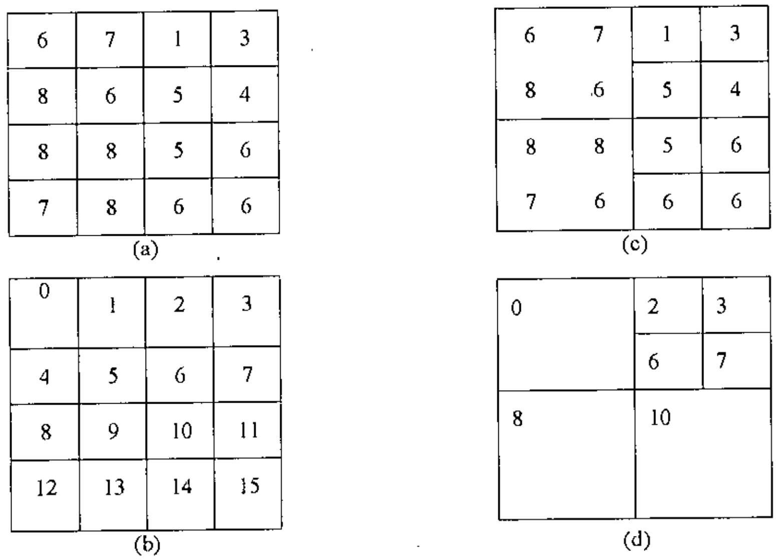


Figure 2.5: Split phase (a) pixel values and (b) pixel *ID* numbers at the start of the split phase, (c) pixel values and (d) Region *ID* after split first and final split iteration [54].

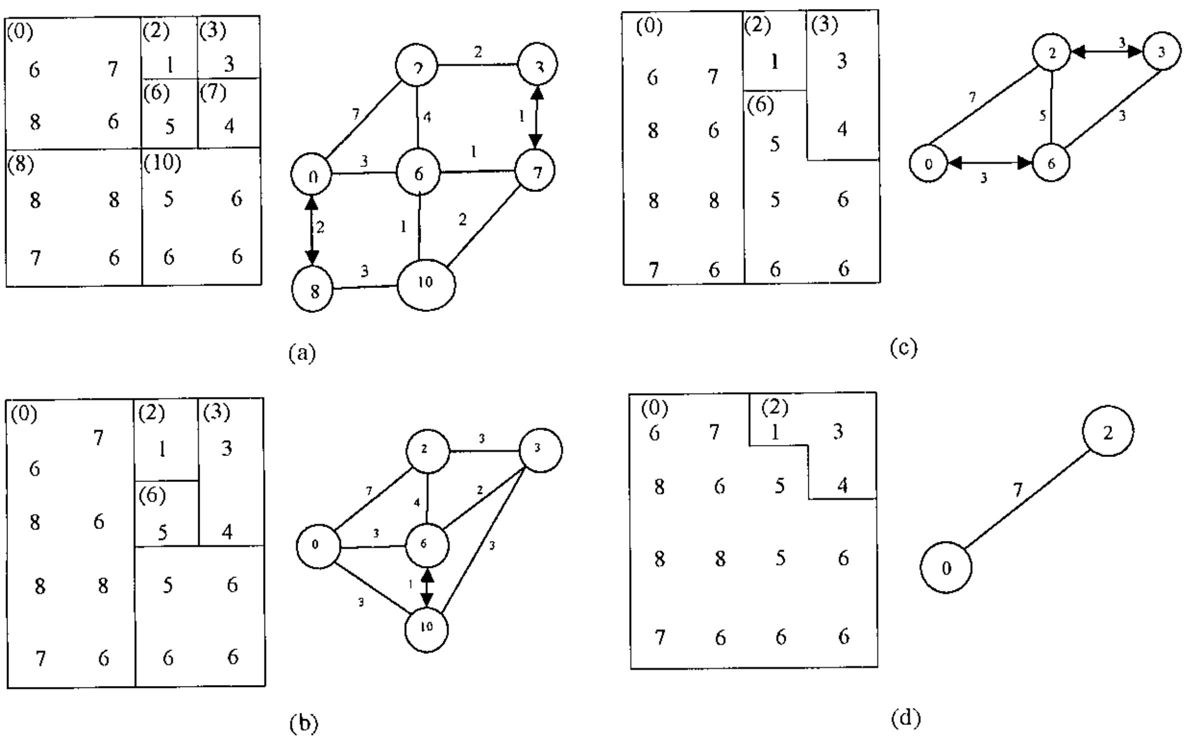


Figure 2.6: Merge phase (a) at start of merge phase, (b) after first iteration of the merge iteration, (c) after second iteration, and (d) after final iteration [54].

(g) Vertices and edges of the graph are updated for the current set of regions and only those edges that do not satisfy the homogeneity criterion are removed.

(h) Repeat steps (c)–(g) while linked regions still exist. Otherwise the algorithm ends.

At step (b) of the algorithm, when the split phase is finished, each PE assigns an identifier to each of their created regions. These *ID*s are calculated as a function of the processor element identifier and depend on the maximum number of regions generated locally by a PE. The split and merge phase of this algorithm are shown graphically in Figure 2.5 and Figure 2.6 respectively. Three methods for defining the region's *ID*s can be applied. Examples of region *ID* method are shown in Figure 2.7.

2.4.2.1 Consecutive

The set of regions obtained at the end of the split phase are sequentially examined from left to right and top to bottom and consecutive *ID*s are assigned to consecutive regions. In this method, the *ID* given to a region, located at the processor p is

$$ID = p \times \frac{N}{P} + i; \quad \text{where, } 0 \leq p < P, \; 0 \leq i < \frac{N}{P}, \tag{2.12}$$

where p is the processor identifier.

So, *ID*s of regions located at the same PE are a block of consecutive integers. This method suffers a strong load unbalance effect.

2.4.2.2 Cyclic

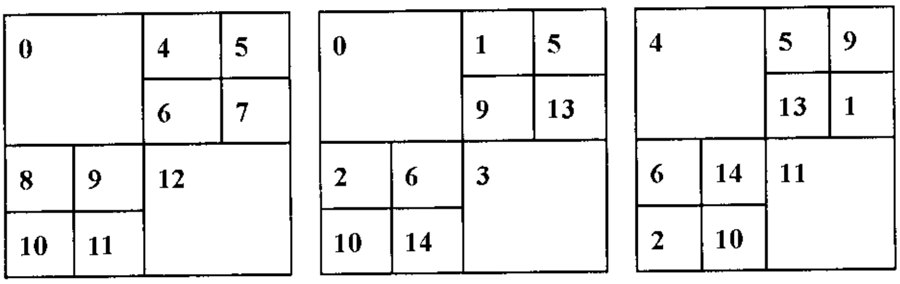
IDs of the regions are cyclically distributed among PEs. The *ID* assigned to a region located at the processor p is determined as

$$ID = p + P \times i; \quad \text{Where, } 0 \leq p < P, \; 0 \leq i < \frac{N}{P}. \tag{2.13}$$

In this method the load unbalance effect can be reduced. Moreover, this helps to break the chains of merge dependencies in the boundaries of the processors when ties are produced. It reduces the number of iterations with respect to the Consecutive method which retains the serialization problem related to the merging order within every processor element, due to the fact that the region identifiers follow an increasing order [54].

2.4.2.3 Random

Region *ID*s are cyclic and randomly generated. The execution time is also reduced. In this method the good load balance effect is maintained. It avoids the serialization problem. Random method eliminates the chains of merge dependencies in the selection of the regions to be merged, introducing some additional degree of parallelism in the algorithm.



(a) Consecutive (b) Cyclic (c) Random

Figure 2.7: Examples of region *ID* method

From a parallel point of view the region growing problem is an irregular and dynamic problem, i.e., the size and the amount of regions as well as the number of edges between regions are data dependent and they evolve during algorithm execution. To solve this kind of problem on a multiprocessor system, it seems to be very convenient to apply load-balancing strategies and to establish efficient communication patterns [54, 66]. At the beginning of the merge phase the number of edges at each processor is data dependent, so it will be necessary to apply a work load-balancing strategy. Montoya *et al.* [54] proposed a static load-balancing strategy as a previous step to the merge phase and a dynamic load-balancing scheme after each iteration of the merge phase.

The main limitation of this parallel SM algorithm is that, large images with small objects yielded the best results but small image with more objects does not provide expected results as well as requires more execution time [28]. This is due to the fact that small objects require less inter processor merging and allow a higher degree of parallelism since there are fewer dependencies between merging regions. The algorithm also produces a more restricted ordering for merging sequence and requires a greater amount of inter processor communication while the image contains a single large object or few large objects.

2.4.3 Modification of the SM Algorithm for Recursive Implementation

If the split and merge process can be performed simultaneously then the computational time is reduced significantly, Til Aach *et al.* [1] proposed a segmentation algorithm where recursive SM is applied. It starts with the whole image being one region which is described by its overall statistics. The basic idea is to extract those areas from this region where local statistics deviate significantly from the global overall statistics. New regions are formed from these statistically deviating areas by sets of connected pixels. The result is a partition consisting of several major regions. The global parameters for all of these regions can then be computed. This process is continued for each region: We compare local statistics inside each region against the newly acquired global region statistics, and extract the deviating areas. The process comes to an end when no further inhomogeneities can be detected in any region. This recursive segmentation scheme, however, is not complete due to the following reasons:

> **First,** there is a trade-off between 'reliability' and spatial resolution inherent with the used significance test, which depends on the size of the sliding window inside which the local statistic is computed. The reliability increases with the window size.
>
> **Secondly,** only the region internal grey values are described by a stochastic model, but not the region shapes. This leads to noisy boundaries between those regions which differ only slightly in their respective statistics.

Both of these shortcomings can be overcome by a relaxation procedure, which examines each pixel situated at a region boundary, and assigns it to that neighbouring region to which it fits best. Since this procedure predominantly affects region shapes, it is reasonable to incorporate a stochastic model for these shapes, which tends to smooth noisy boundaries at statistically 'uncertain' transitions. Each step of the recursive segmentation process hence consists of two parts.

(a) Detection of inhomogeneities (object detection).

(b) Contour relaxation.

2.4.3.1 Object Detection

Consider a given partition Q for the image data Y. This might be the initial one-region partition, or one of the partitions emerging during the process. The notion 'inhomogeneity' is specified as a significant deviation of local mean from the global estimate, $\hat{m}(R_j)$. The detection procedure works as follows: First, an error image $E = \{e_{m,n}\}$ is computed by normalizing the grey values inside each region according to

$$e_{m,n} = \frac{y_{\min} - \hat{m}(R_j)}{\hat{\sigma}(R_j)} \tag{2.14}$$

where, R_j is the region to which the pixel (*m, n*) belongs. Inside a sliding window of dimensions $d \times d$ on the error image E the local average $\mu_{m,n}$ is computed, and $\mu_{m,n}$ is assigned to the pixel (*m, n*). If the local statistics comply with the global ones, $\mu_{m,n}$ is distributed according to $N(0, d^2)$. Each pixel (*m, n*) with $|\mu_{mn}| > t$ is then marked as an inhomogeneity[1].

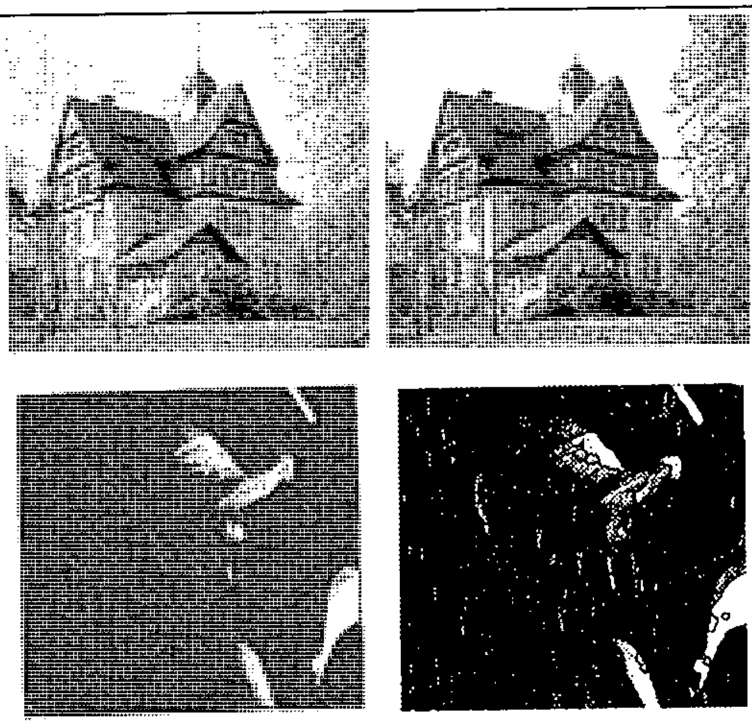
2.4.3.2 Contour Relaxation

The application of object detection provides a partition Q_0 which suffers from inaccurate region boundaries due to the limited spatial resolution of the detection scheme. Hence Q_0 has to be modified until it matches the given image best. The 'best match' is evaluated on the basis of the Maximum a Posteriori (MAP) criterion, that is, we modify Q_0 until a local maximum of the a Posteriori density *p(Q, Y)*. Ideally, the process of re-labelling stops when no more pixels whose labels have to be changed are found during a scan. The result of this method is shown in the Figure 2.8.

These problem solving steps as mentioned above gives raise of computational complexities and excessive time requirement. This algorithm depends on the seed pixel which is not always possible to provide, mainly in case of real time image processing. Again the object boundary may not be clearly defined for few objects which will be wrongly segmented by this algorithm [28].

2.4.4 Modification of SM Algorithm for Dynamic Thresholding

When merging splitted regions to its accurate neighbour, the homogeneity criterion of a region is measured to merge it with one of its connected neighbours and hence it gives the local optimization. This affects the performance of final segmentation. Considering this issue,



(a) Input image (b) Output image

Figure 2.8: Examples of recursive SM with object detection and contour relaxation [1].

Cheevasurit *et al.* [20] gives a modification of SM algorithm. When scanning the picture in a regular order, a subset of splitted regions will be merged with those of its neighbours which fulfil the most homogeneity criteria. For the subset S_{22}^{n}, this modification requires the computation of the four heterogeneity measures as follows:

$$d\left(S_{11}^{n+1} = S_{11}^{n} S_{12}^{n} S_{21}^{n} S_{22}^{n}\right)$$

$$d\left(S_{12}^{n+1} = S_{12}^{n} S_{13}^{n} S_{22}^{n} S_{23}^{n}\right)$$

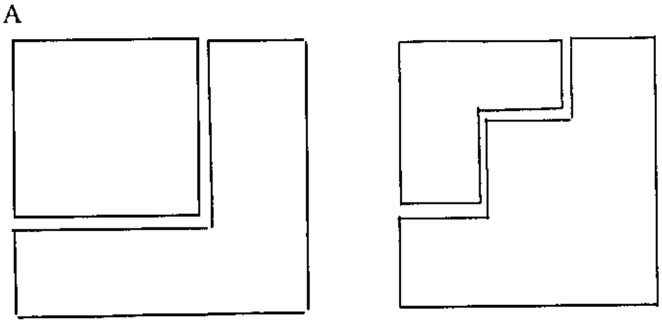
$$d\left(S_{21}^{n+1} = S_{21}^{n} S_{22}^{n} S_{31}^{n} S_{32}^{n}\right)$$

$$d\left(S_{22}^{n+1} = S_{22}^{n} S_{23}^{n} S_{32}^{n} S_{33}^{n}\right) \tag{2.15}$$

The superset with the smallest heterogeneity measure will be kept as usual if it is lower than the threshold ε. The quality of the merging is notably improved by this modification as shown in Figure 2.9.

5	5	0
5	1	0
0	0	0

(a)



(b) (c)

Figure 2.9: improvement after modification of local optimization [20]

The number of region, its size and shape varies with the threshold value. It is very difficult to optimize a threshold value for any particular image. It is also difficult to evaluate a global threshold value which can best fit in any SM algorithm for all the images.

Cheevasuvit [20] addressed this issue in his proposed algorithm where several threshold values are applied to the same image to find out the stable regions in the image. Let $E = \{\varepsilon(k)\}$ be an ordered set of threshold values; let $h_i(k)$ be the i-th subset of picture X after the whole SM process with $\varepsilon(k)$ as a threshold. Let $H_i(k)$ be an m-dimensional feature vector of R^m describing this subset. H contains information on the geometry of h (position and shape) and on its radiometry. A distance function Δ between two subsets $h_i(k)$ and $h_j(k')$ and denote it Δ (i, k; j, k'), where Δ is any measure on R^m, acting on H vectors. Two subsets at adjacent threshold levels are similar if their distance is less than a given value δ :

$$h_i(k) \text{ - } h_j(k+1) \text{ If } \Delta\,(i,\, k; j,\, k+1) < \delta \tag{2.16}$$

For all the subsets h and all the ε values of E, we build a graph representation $G(X, E, \Delta)$, where the subsets h are nodes and the similarity relation links. For this complete graph $G(X, E, \Delta)$, it is possible to select the areas which are stable for many successive thresholds and can be determined precisely for which range of the threshold they remain nearly unmodified. For each zone the best thresholding in terms of stability and robustness can be determined from this graph. The graph is a robust description of the picture, because, if each node is still sensitive to small perturbations, the global graph is not. It is required to compare two subsets obtained for different thresholds $\varepsilon(k)$ and $\varepsilon(k')$, in order to determine whether they are provided by the same area or not. The distance between two subset $h_i(k)$ and $h_j(k')$ is measured as

$$\Delta\left(i,k,j,k'\right)=\frac{x_{ij}^2}{x_i x_j} \tag{2.17}$$

A complete combinatorial comparison was then made between all the subsets at threshold $\varepsilon(k)$ and all the subsets at threshold $\varepsilon(k+1)$ and the graph $G(X, E, \Delta)$ was obtained. An area is defined as stable if it belongs to long path in G. The Figure 2.10 to 2.13 show the sequential procedure of this algorithm.

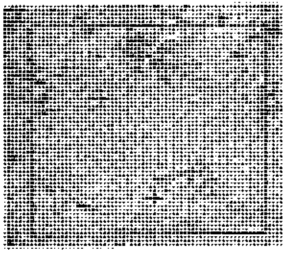


Figure 2.10: Original NOAA-7 channel 4 picture

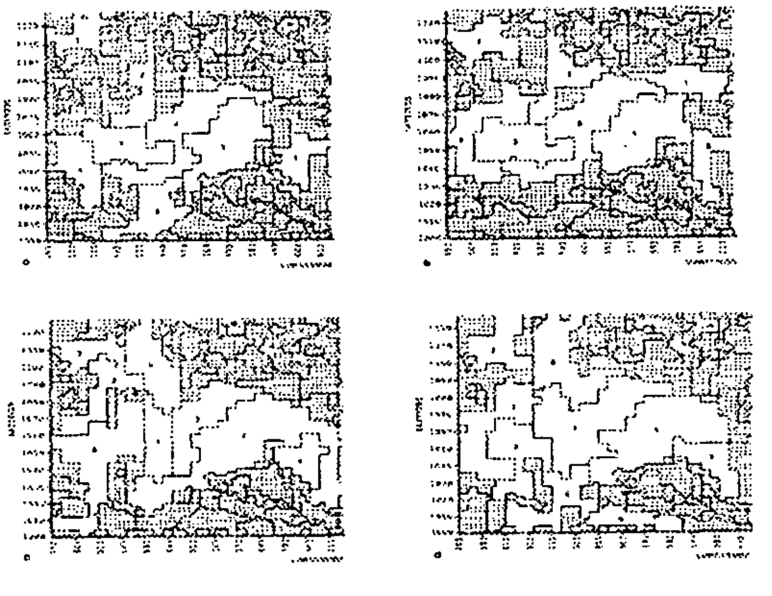


Figure 2.11: Four results of SM modified algorithm of Cheevasuvit *et al.* [20] with (a) feature value ε = 20, (b) ε = 20, (c) ε = 22, (d) ε =24.

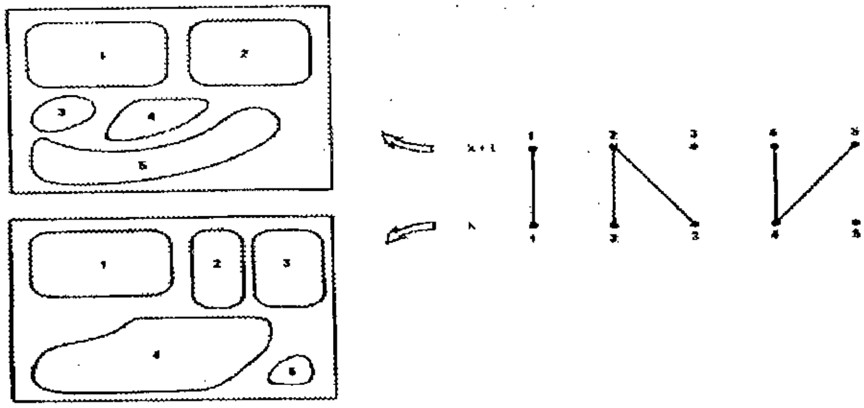
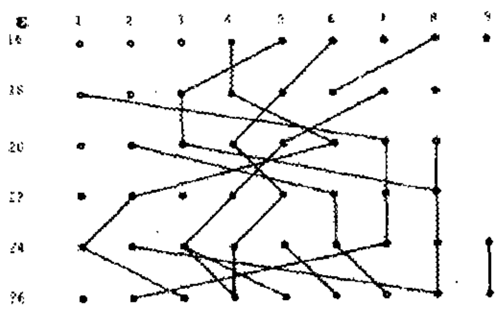
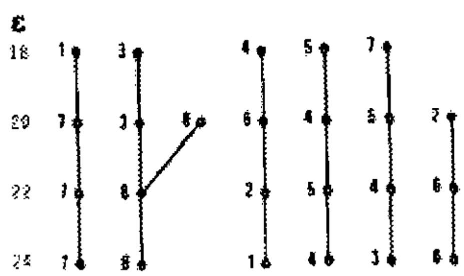


Figure 2.12: Typical graph representation with two threshold levels k and $k+1$. All possible situations are represented: simple similarity, merging, appearance, splitting, and disappearance [20].



(a)



(b)

Figure 2.13: (a) A complete graph representation of Figure 2.10 with six threshold levels. The total dynamic range is 128. The four central threshold values give the results of Figure 2.10.(b) Five stable regions extracted from the graph in (a) [20].

This algorithm suffers from huge computational complexities as it requires obtaining objects for different threshold values to represent them in the complete graph. Again, the range of thresholds varies from image to image thereby the static nature can not be eradicated in this algorithm. Though this algorithm point out the local optimization problems of the SM algorithm it cannot overcome that as it consider only neighbours of an object while merging.

Though this algorithm performs better than those mentioned above, it is too complex to implement and less efficient due to time complexities. More so the noisy and arbitrary shaped images are segmented further to its constituent parts by restricted merging criteria. Like other algorithm, this algorithm is highly dependent on threshold values also [28].

2.4.5 Modification of SM Algorithm for Splitting Points and Direction of Splitting

A significant drawback of SM method splitting phage is to find out points where it can be splitted into four region and the direction along which the region need to be splitted into sub-regions. Chaudhuri *et al.* [19] has proposed a splitting algorithm along the sparsely populated strip. According to his algorithm of splitting, strips of finite width at different directions around the centroid of the data are considered. The data is split across the sparsely populated strip. For a two-dimensional data set four directions at the centre of the data are considered as shown in Figure 2.14. The greater the number of directions for strips the better the accuracy of the result is.

The width of a strip along any direction is to be found out before actually constructing the strip. The width is an important impediment in deciding whether the data is indeed sparsely populated in that direction, if the width is very large then all the points in the data set may belong to the strip. If it is very small then the strip may not be amenable for making any decision. This algorithm proposed that the width of a strip should be $2h_n$ where $h_n = a\varepsilon_n$,

$$\varepsilon_n = \frac{1}{\eta^p},\ 0 < p < \frac{1}{q} \tag{2.18}$$

where, 'a' is a constant, n is the total number of points in the q-dimensional data set. The algorithm is as follows:

(a) Find b_1, b_2, b_3 and b_4 where b_i are the number of points in each of the strip.

(b) Find n_1=min{ b_1, b_2, b_3 , b_4 }

(c) Find the strip in which the number of points is $n_1 : \sum f \frac{n_1}{n} \times 100 < l_1$, then split the set S along the corresponding direction. Here l_l is a small quantity, called splitting restriction dependent on the width at the strip. Strip $|b_i| = n_i$, is called the sparely populated strip [19].

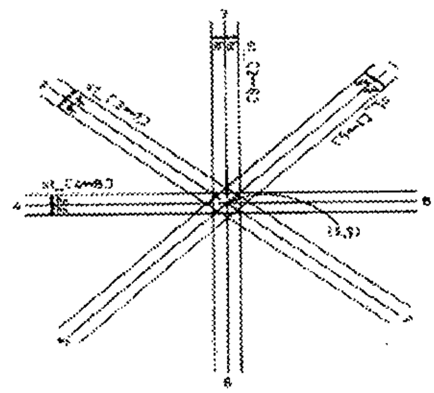


Figure 2.14: Various directions to find sparsely populated strip [19].

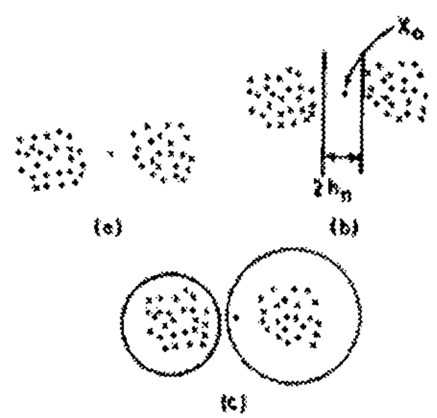


Figure 2.15: Width of a strip [19].

2.4.6 Modification of SM Algorithm for Global Convergence Capability

One limitation of the SM algorithm as proposed by Pavlidis is that it gives only local optimization rather global one. If, for instance, the decision of merging S_{22}^{n} in S_{22}^{n+1} (shown in Figure 2.16), it is not sure that this operation will be the best merging for S_{23}^{n} or (S_{32}^{n} or S_{33}^{n}). Clearly, when refusing a cascading verification, a suboptimal solution is chosen.

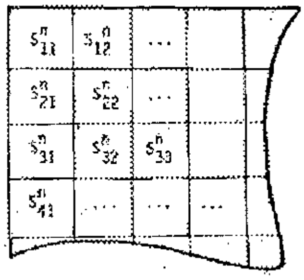


Figure 2.16: Index of the regions [73].

Numerous research works are going on at present on this topic but still widely acceptable dynamic solution could not be discovered, Zhang *et al.* [73] has blended the SM method with expectation maximization (EM) method and proposed an algorithm known as the *split-and-merge expectation maximization* (SMEM) algorithm. This algorithm uses the membership value concept of Ueda *et al.* [70] in both split and merge step of the SM algorithm where mean and covariance are used as parameters. But it solves the ill posed problem of the concept by applying a split method based on the singular value decomposition or Cholesky decomposition [73].

The key of the SMEM algorithm is to define a proper split-and-merge criterion, and to construct an efficient split-and-merge method. Ueda *et al.* [70] presented a method of initializing the parameter $\Theta = (\pi_1, \pi_2, \ldots, \pi_n, \theta_1, \theta_2, \ldots, \theta_n)$ where π is the membership value and θ is the parameter composed of mean and covariance as $\theta_m = (\mu_m, \Sigma_m)$. Suppose i' be the merged component of components i and j, its initial parameter values are set as linear combinations of the original ones. The merging equation can be defined as follows:

$$\pi_{i'} = \pi_i + \pi_j \tag{2.20}$$

$$\theta_{i'} = \frac{\pi_i \theta_i + \pi_j \theta_j}{\pi_i + \pi_j} \tag{2.21}$$

On the other hand, if the operation is splitting component k into two component i' and k', the initial parameter values of components j' and k' are set as

$$\pi_{j'} = \pi_{k'} = \frac{\pi_k}{2} \tag{2.22}$$

$$\theta_{j'} = \theta_{k'} + \varepsilon \tag{2.23}$$

$$\theta_{k'} = \theta_k + \varepsilon' \tag{2.24}$$

where, ε or ε' is some small and random perturbation vector or matrix. Because the covariance matrices should be positive definite and cannot ensure this, Ueda *et al.* [70] defined them as

$$\sum_{j'} = \sum_{k'} = \det(\sum_{k})^{1/n} \ln \tag{2.25}$$

From the merge equations, we can see that the mean vector and the covariance matrix of component i is a linear combination of the corresponding parameters of components i and j, and the merge of the mean vector is independent of the covariance matrix. This means that the merge of the mean vector does not consider the influence from the covariance matrix, and vice versa. We know that mean vectors and covariance matrices are first and second moments, respectively. They are closely related, so it is not appropriate to treat them independently. From the split equations, the same problem still exists in the split operation. Moreover, although Ueda's equation can guarantee covariance matrices being symmetric positive definite, this destroys the anisotropy of covariance matrices.

More so, the mean vectors and the covariance matrices obtained with the method by Ueda *et al.* [70] is not the solutions of these split equations. Solving the split equations is an ill-posed problem because the number of the equations is less than the number of the unknowns. This ill posed problem is solved by applying a split method based on the singular value decomposition or Cholesky decomposition [73]. The experimental results of this algorithm are shown in the Figure 2.17.

Though this algorithm performs better than those mentioned above, it is too complex to implement and less efficient due to time complexities. More so the noisy and arbitrary shaped images are segmented further to its constituent parts by restricted merging criteria. Like other algorithm, this algorithm is highly dependent on threshold values also [28].

Addressing the performance and shortcomings of the aforementioned modified SM algorithm Faruquzzaman *et al.* [30] proposed a new *object segmentation based on split and merge algorithm* (OSSM). The proposed OSSM algorithm uses the concept of multistage SM algorithm to be robust in segmenting objects of an image dynamically. The algorithm used T-test for both splitting and merging regions. Besides, it has used intra-region variance minimization and intra-region variance maximization throughout the merging process to maintain the objects identity.

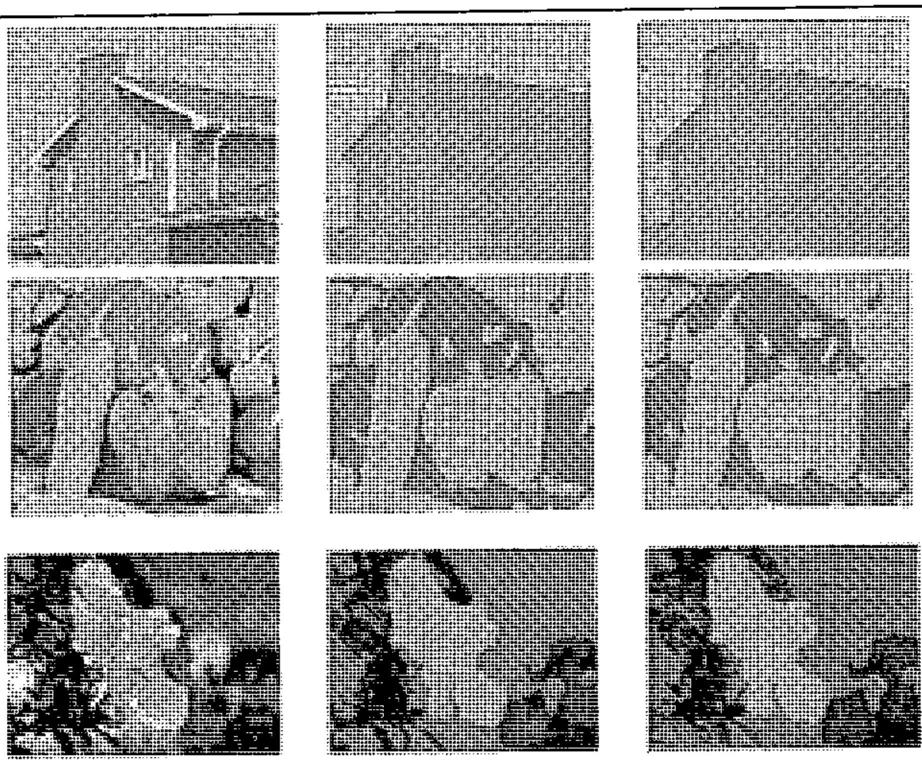


Figure 2.17: Example of SMEM algorithm implementation [57].

2.5 Fuzzy Clustering Algorithms

Beyond the SM algorithm and its variants, analysis of fuzzy clustering algorithms is another interesting domain of image segmentation now-a-days. To evaluate the performance of any proposed algorithms, it is necessary to compare it with the recent fuzzy clustering algorithms. Hence, despite the dissimilarity this section addressed some well-known fuzzy clustering algorithms for the completeness of this literature review chapter.

Clustering algorithms that use general feature sets such as PL, PI or CIL are generally treated as classical fuzzy clustering techniques. These are dependent on both the features used and the type of objects in an image. A review of some main classical fuzzy clustering techniques is now detailed.

2.5.1 Fuzzy c-Means Algorithm

The FCM algorithm [12] was developed by Bezdek in 1981 and is still the most popular classical fuzzy clustering technique, widely used directly or indirectly in image processing. It performs classification based on the iterative minimization of the following objective function and constraints [12]:

$$J_q(\mu,V,X)=\sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^q D_{ij}^2 \tag{2.26}$$

subject to

$$0\le \mu_{ij}\le 1;\quad i\in\{1,\ldots,c\} \text{ and } j\in\{1,\ldots,n\} \tag{2.27}$$

$$\sum_{i=1}^{c}\mu_{ij}=1;\quad j\in\{1,\ldots,n\} \tag{2.28}$$

$$0<\sum_{j=1}^{n}\mu_{ij}<n;\quad i\in\{1,\ldots,c\} \tag{2.29}$$

where n and c are the number of data and clusters respectively. μ is the fuzzy partition matrix containing membership values $[\mu_{ij}]$, q is the fuzzifier where $1<q\le\infty$, V is cluster centre vector $[v_i]$, X is a data vector $[x_j]$ and $D_{ij}=d(x_j,v_i)$ is the distance between datum x_j and v_i. Using a Lagrangian multiplier [3], the following can be derived by optimizing the objective function in (2.26) with respect to μ and V.

$$\mu_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\frac{D_{ij}}{D_{kj}}\right)^{2/(q-1)}} \tag{2.30}$$

$$v_i=\frac{\sum_{j=1}^{n}(\mu_{ij})^q x_j}{\sum_{j=1}^{n}(\mu_{ij})^q} \tag{2.31}$$

The membership values are initialized randomly and both these and the cluster centres are iteratively updated until the maximum change in μ_{ij} becomes less than or equal to a specified threshold ξ. q is normally set to 2 as this is the best value for the fuzzifier (Step 0) while the membership μ_{ij} is randomly initialized in Step 0. The cluster centre v_i and membership values

μ_{ij} are then iteratively updated using (2.30) and (2.31) respectively (Steps 3.1-3.2) until either the maximum number of iterations ($\max\_\text{Iteration}$) or threshold ξ is reached (Step 3.3). The complete FCM algorithm is given in Algorithm 2.1, which for n data points incurs $O(n)$ computational time complexity [7, 37].

Algorithm 2.1: *Fuzzy c-means* (FCM) algorithm.

Pre condition: Objects to be segmented, number of clusters c, threshold ξ and the maximum number of iterations $(\max\_\text{Iteration})$.

Post condition: Final segmented regions $\Re$.

Fix $q = 2$.

Initialize μ_{ij}.

FOR $l = 1, 2, 3, \ldots, \max\_\text{Iteration}$

3.1 Update cluster centres v_i using (2.31).

3.2 Update membership values $\mu_{ij}^{(l)}$ using (2.30).

3.3 IF $\left\|\mu_{ij}^{(l)} - \mu_{ij}^{(l-1)}\right\| \le \xi$ THEN STOP.

The number of clusters c, fuzzifier q and threshold ξ all need to be set manually. The selection of q is especially important because if $q = 1$ then FCM produces crisp *hard clustering* (HC) instead of fuzzy regions. Also (2.30) and (2.31) are not sufficient to achieve the local minimum of (2.26) [7, 37], since if any of the distance value $D_{ij} = 0$, (2.30) will be undefined. FCM strongly supports probability, but not the degree of *typicality* because it has the constraints in (2.27)-(2.29) which preclude the trivial solution $\mu_{ij} = 0$.

2.5.2 Suppressed Fuzzy c-Means Algorithm

By using a fuzzifier q and membership value μ_{ij}, the performance of FCM is better than any HC technique [12], though the convergence speed is much lower. Moreover, if the fuzzifier is large $(q > 2)$, it increases the gap between the membership values which may lead to a decrease the overall segmentation performance of FCM [36]. To address these issues, the *rival checked fuzzy c-means* (RCFCM) algorithm [27] was introduced on the basis of competitive learning, by magnifying the largest membership value and suppressing the second largest membership value. The main step in the RCFCM algorithm is to modify μ_{ij} in the FCM algorithm as follows.

Assume the largest membership value of datum x_j for the p^{th} cluster is μ_{pj} and its second largest membership value in the s^{th} cluster is μ_{sj}. After modification, the membership value of x_j belonging to each cluster is then:

$$\mu_{pj} = \mu_{pj} + (1-\alpha)\mu_{sj} \tag{2.32}$$

$$\mu_{sj} = \alpha\mu_{sj} \tag{2.33}$$

where $0 \le \alpha \le 1$. The main problem with RCFCM is that it only pays attention to the largest and second largest membership values, so if the choice of α is unsuitable, it can lead to the second largest membership value to be modified being actually less than some others, which causes a disturbance in the original order [27]. For this reason, the convergence of RCFCM is not assured and so to solve this, the suppressed fuzzy c-means (SFCM) algorithm was introduced to magnify only the largest membership value and to suppress the rest [27]. If μ_{pj} is the largest membership value for datum x_j, the modified values are:

$$\mu_{pj} = 1 - \alpha\sum_{i \neq p}^{c} \mu_{ij} = 1 - \alpha + \alpha\mu_{pj} \tag{2.34}$$

$$\mu_{ij} = \alpha\mu_{ij}; \quad i \neq p \tag{2.35}$$

where the various parameters are as defined above. Since SFCM prizes the largest and suppresses all other membership values, it does not disturb the original order and so eliminates the drawback of RCFCM. When $\alpha = 0$, SFCM produces the same results as HC, while for $\alpha = 1$ it becomes the FCM algorithm, so this establishes a more natural and realistic relationship between the HC and FCM algorithms, so that for a suitable α value, SFCM can compromise the advantages of faster convergence speed of HC techniques, with the better clustering performance of FCM without impacting on the time complexity which remains the same as FCM, i.e., $O(n)$.

2.5.3 Gustafson-Kessel Algorithm

The *Gustafson-Kessel* (GK) algorithm [39] is a powerful clustering technique that has been used in various image processing, classification and system identification applications [12, 23]. It is characterised by adapting automatically the local data distance metric to the shape of the cluster using a covariance matrix and adapting the distance inducing matrix correspondingly [8, 39, 43, 52]. The GK algorithm is based on the iterative optimization of the following FCM-type objective function [12]:

$$J_q(\mu, V, X) = \sum_{j=1}^{n}\sum_{i=1}^{c}(\mu_{ij})^q D_{ij}^{'2} \tag{2.36}$$

$$0 \le \mu_{ij} \le 1;\ i \in \{1,\ldots,c\} \text{ and } j \in \{1,\ldots,n\} \tag{2.37}$$

$$\sum_{i=1}^{c}\mu_{ij} = 1; \qquad j \in \{1,\ldots,n\} \tag{2.38}$$

where $D_{ij}^{'}$ is the data distance norm calculated for clusters of different shapes in one dataset that is given by:

$$D_{ij}^{'2} = (x_j - v_i)^T A_i (x_j - v_i) \tag{2.39}$$

where A_i is the *norm inducing* matrix, which allows the distance to adapt to the local topological structure of the data [8, 43, 52]. Using the Lagrangian multiplier in (2.36), the membership value μ_{ij} can be calculated as follows:

IF $(D_{ij}^{'} = 0)$

THEN $\mu_{ij} = 1$ maintaining $\sum_{i=1}^{c}\mu_{ij} = 1$ (2.40)

ELSE $$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{D_{ij}^{'}}{D_{kj}^{'}}\right)^{\frac{2}{q-1}}} \tag{2.41}$$

The cluster centre v_i is updated as:

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^q x_j}{\sum_{j=1}^{n} (\mu_{ij})^q} \tag{2.42}$$

To adapt to the structure of the cluster shape, the distance norm inducing matrix A_i is used which increases the distance of the furthest data points while decreasing those data points close to the cluster centre. A_i is defined as:

$$S_{fi} = \frac{\sum_{j=1}^{n} (\mu_{ij})^q (x_j - v_i)^T (x_j - v_i)}{\sum_{j=1}^{n} (\mu_{ij})^q} \tag{2.43}$$

$$A_i = \left| \rho_i \det(S_{fi})^{1/P'} (S_{fi})^{-1} \right| \tag{2.44}$$

where S_{fi} is the *fuzzy covariance* matrix, P' is the dimension of hyper-spherical cluster, and ρ_i is the cluster volume, which is usually set to 1. In the GK algorithm, the parameters values are set to $q = 2$ and $\rho_i = 1$ (Step 1) followed by the initialization of membership values μ_{ij} (Step 2). The cluster centre v_i is updated using (2.42) in Step 3.1, while the data distance norm is calculated (Steps 3.2 and 3.3) to iteratively update the membership value μ_{ij} using (2.40) and (2.41) (Step 3.4) until either fulfilling the specified threshold ξ or the maximum number of iterations is exceeded (Step 3.5). The detailed steps of the GK algorithm are given in Algorithm 2.2.

The performance of the GK algorithm is not very good for either small datasets or when data within a cluster are (approximately) linearly correlated, because in such cases the covariance matrix becomes singular. Babuska *et al.* [8] overcame these drawbacks by considering the ratio of the maximum and minimum eigen values in calculating the *fuzzy covariance matrix*.

In summarising, the GK algorithm adapts the local structure of the cluster shape using a *distance norm inducing matrix* A_i, with the *modified GK* algorithm [8] able to effectively handle both the large and small datasets. These characteristics are exploited by using the GK

Algorithm 2.2: *Gustafson-Kessel* (GK) algorithm

Precondition: Objects to be segmented, number of clusters c, threshold ξ and max_Iteration

Post condition: The final segmented regions $\Re$.

Fix $q = 2$ and set $\rho_i = 1$.

Initialize μ_{ij}.

FOR $l = 1, 2, 3, \ldots, \text{max\_Iteration}$

3.1 Update cluster centre v_i using (2.42).

3.2 Compute cluster covariance matrix using (2.43) and (2.44).

3.3 Calculate data distance norm by (2.39).

3.4 Update $\mu_{ij}^{(l)}$ using (2.40) and (2.41).

3.5 IF $\left(\left|\mu_{ij}^{(l)} - \mu_{ij}^{(l-1)}\right| \leq \xi\right)$ THEN STOP.

as key part of the shape-based algorithm [5] for integrating generic shape information into the clustering framework.

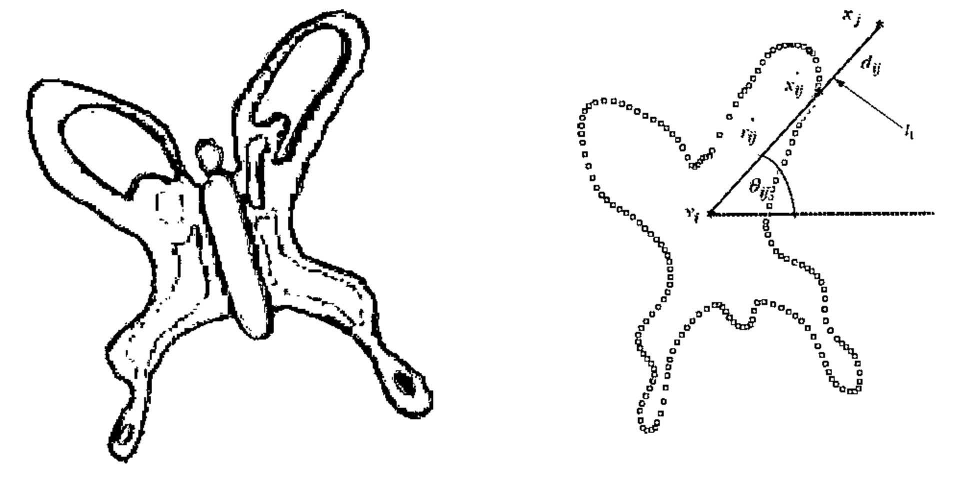
2.5.4 Object Based Segmentation using Fuzzy Clustering

As the circular and elliptical shape based algorithms are unable to segment arbitrary-shaped objects, to address this limitation, a new shape-based clustering algorithm called *image segmentation using fuzzy clustering and integrating generic shape information* (FCGS) has been introduced, which incorporates generic shape information into the fuzzy clustering framework by exploiting a B-splines [31, 41] representation of an initial object's shape [3]. In the FCGS algorithm, the generic shape information is integrated considering the same technique used in the Gustafson-Kessel algorithm [39]. The shape is initialized using B-splines while the intersection point is calculated using the polar coordinate system detailed in [3]. The shape of the cluster is iteratively updated using the concepts of circular shape clustering algorithm used in [32]. Even though the FCGS algorithm is able to segment arbitrary shaped objects in an image well, it suffers from the disadvantages including: (i) it does not follow the optimization criteria clustering algorithm, (ii) the FCGS algorithm is unable to find out the proper intersection point, and (iii) the scaling of clusters during iterations is not proper. Addressing these issues, Ameer *et al.* [5] has developed a new *object based segmentation using fuzzy clustering* (OSF) with the meaning of fuzzy k-generic shaped algorithm that incorporates generic shape information by introducing a shape constraint to make certain the optimization criteria is upheld to both ensure the convergence of the objective function and preservation of the original object shape during iterative scaling [5]. To seamlessly integrate generic shape information into the segmentation process, an object shape descriptor i.e., a contour, needs to be provided as part of the initialization process.

To initialize the OSF algorithm, the initial shape contour points can be either provided manually or generated automatically which is adopted in FCGS [3]. In order however, to automatically initialize the contour of the corresponding object in the OSF algorithm the following two steps are required: (i) the GK algorithm is used for initial segmentation and (ii) the boundary points of each region are scanned with all outliers discarded and these are then treated as the corresponding contour points. These contour points are used to measure the data distance from the boundary point in a similar strategy to that employed in the FCS, FKR, FCES, FKE, and FCGS algorithms.

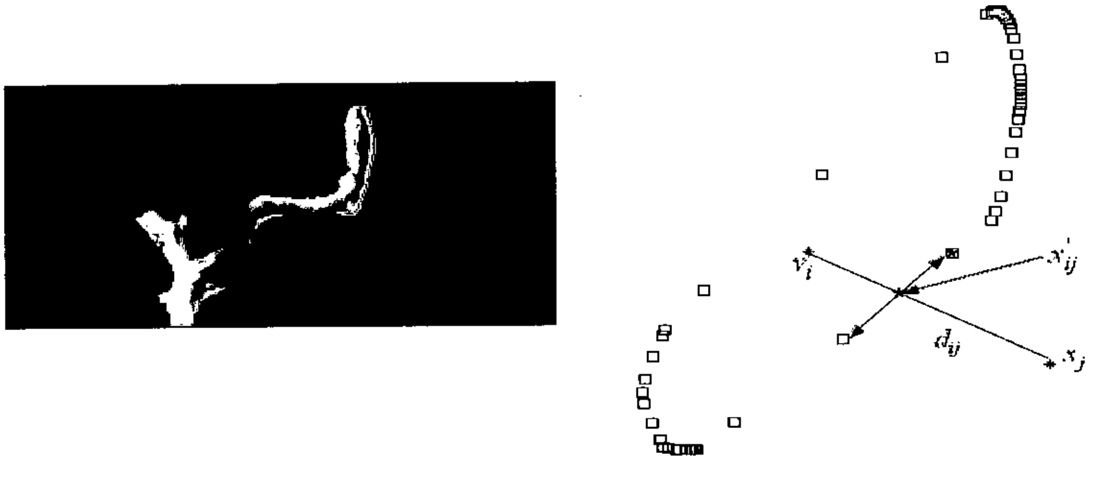
It is the paramount of importance in any clustering-based segmentation strategy of calculating the data distance d_{ij} of a datum for subsequent use in an objective function. Using an analogous strategy to that used for the FCGS algorithm, d_{ij} is calculated from the respective contour shape points in the OSF algorithm. For the *butterfly* image and its corresponding B-spline shape representation in Figure 2.18 (a) and (b), the intersection point x'_{ij} on the contour of datum x_j can be calculated using either a Polar or Cartesian coordinate system.

In the OSF algorithm, the actual intersection point is calculated as shown in Figure 2.19 (c) and comprises the following key processing steps: (i) find two points on the contour of the curve those are closest to and lie on opposite sides of the line l_1 and (ii) as the shape descriptor usually generates a straight line between these two consecutive contour points, the meeting point between line passing through these two points and l_1 is the *intersection point* x'_{ij} of the corresponding datum x_j.



(a) (b)

Figure 2.18: (a) Original *butterfly* object; (b) Example of the *intersection point* between datum x_j and the B-splines shape contour of (a) along line l_1 [5].



(a) Original Image (b) Intersection point using Cartesian system

Figure 2.19: (a) Original *branch* image, Intersection point calculation of datum x_j on the B-splines representation shape using the Polar (b) and Cartesian (c) coordinate system [5].

A unique feature of the OSF algorithm is that for the first time a dedicated shape constraint is incorporated into the fuzzy clustering framework to ensure convergence by preserving the initial shape of an object during the subsequent iterative shape scaling process. The shape constraint used for segmenting arbitrary-shaped objects is formally defined as:

$$\frac{r_{ij}^{'}}{\sum_{t=1}^{n} r_{ij}^{'}} = k_{ij} \tag{2.45}$$

where $r_{ij}^{'}$ is the distance between the intersection point $x_{ij}^{'}$ and the i^{th} cluster centre v_i and k_{ij} is a constant of the j^{th} datum in the i^{th} cluster. The objective function for the OSF algorithm can now be formally defined based upon FCM [12] as follows:

$$J_q(\mu, V) = \sum_{j=1}^{n} \sum_{i=1}^{c} (\mu_{ij})^q d_{ij}^2 \tag{2.46}$$

subject to
$$\sum_{i=1}^{c} \mu_{ij} = 1 \text{ and } r_{ij}^{'} \Big/ \sum_{t=1}^{n} r_{it}^{'} = k_{ij} \tag{2.47}$$

where $d_{ij} = D_{ij} - r_{ij}^{'}$

The objective function in (2.46) with its constraints in (2.47) is iteratively minimised using Lagrangian optimisation techniques [3]. For optimisation, if $d_{ij} = 0$ then a HC (crisp) decision is necessary and the j^{th} data will be classified into i^{th} cluster, otherwise the membership value will be updated based on the value of d_{ij}. The membership value is defined as:

$$\text{IF } d_{ij} = 0 \text{ THEN } \mu_{ij} = 1 \text{ maintaining } \sum_{i=1}^{c} \mu_{ij} = 1 \tag{2.48}$$

$$\text{ELSE } \mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{q-1}}} \tag{2.49}$$

The contour radius $r_{ij}^{'}$ is updated as follows:

$$r_{ij}^{'} = D_{ij} - \frac{k_{ij} \sum_{t=1}^{n} D_{ij} - D_{ij}}{k_{ij} \sum_{t=1}^{n} \frac{1-k_{it}}{\mu_{it}} - \frac{1-k_{ij}}{\mu_{ij}}} \left(\frac{1-k_{ij}}{\mu_{ij}} \right) \tag{2.50}$$

When the second term in (2.50) becomes small i.e., tends to zero, the initial shape may become over-scaled when it is updated and so to reduce the impact of this effect, the $r_{ij}^{'}$ update is controlled as follows:

$$r_{ij}^{'}(new) = \lambda r_{ij}^{'} + (1-\lambda) r_{ij}^{'(0)} \tag{2.51}$$

where $r_{ij}^{'}$ and $r_{ij}^{'(0)}$ are the current and initial values of $r_{ij}^{'}$ respectively, and λ is an empirically selected data constant which is a trade-off between the current and initial object shapes. Using the same optimisation technique as in (2.46) and (2.47), the i^{th} cluster centre v_i can be calculated from:

$$f_x = x_{j1} - D_{ij}\frac{x_{ij1}^{'} - v_{i1}}{r_{ij}^{'}} + x_{ij1}^{'} - r_{ij}^{'}\frac{x_{j1} - v_{i1}}{D_{ij}} \tag{2.52}$$

$$f_y = S_{j2} - D_{ij}\frac{x_{ij2}^{'} - v_{i2}}{r_{ij}^{'}} + x_{ij2}^{'} - r_{ij}^{'}\frac{x_{j2} - v_{i2}}{D_{ij}} \tag{2.53}$$

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^q \begin{pmatrix} f_x \\ f_y \end{pmatrix}}{2\sum_{j=1}^{n} (\mu_{ij})^q} \tag{2.54}$$

where for an image, the 2-D data and cluster centre are given by $x_j = \begin{bmatrix} x_{j1} \\ x_{j2} \end{bmatrix}$ and $v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix}$ respectively. The complete steps for the OSF algorithm are provided in Algorithm 2.3.

Algorithm 2.3: Object based segmentation using fuzzy clustering (OSF) algorithm.

Precondition: Objects to be segmented, the number of clusters c, max_Iteration and threshold ξ.

Post condition: Final segmented regions $\Re$.

Initialize the shape.

Find intersection point and calculate initial $r_{ij}^{'}$.

Calculate k_{ij} using (2.47).

FOR $l = 1,2,3,\cdots,$ max_Iteration

4.1 Update $\mu_{ij}^{(l)}$ using (2.48) and (2.49).

4.2 Update $r_{ij}^{'}$ using (2.50) and (2.51).

4.3 Update v_i using (2.54).

4.4 IF $\left\|\mu_{ij}^{(l)} - \mu_{ij}^{(l-1)}\right\| \le \xi$ THEN STOP.

2.6 Patterns used in Video Processing

The concept of segmenting the video frame into separate segments containing one or more moving objects is first used in MPEG-4 video standard [9]. These segments are 16×16 blocks which contain motion area and non-motion area. These blocks are referred to as micro-blocks (MBs). *Wong et. al.* [18] classified the micro-blocks into three categories: 1) *Static MB* (SMB): MBs that contain little or no motion; 2) *Active MB* (AMB): MBs which contain moving object(s) with little static background; and 3) *Active-Region MB* (RMB): MBs that contain both static background and part(s) of moving object(s). Patterns P_1-P_8 of Figure 2.20 was used by *Wong et. al.* to represent the moving objects [9]. In [9, 10, 13] a complete 32 *pattern codebook (PC)* is proposed which is shown in Figure 2.20. These patterns are regular, clustered where all the pixels are connected and boundary adjoined.

In PSM, these patterns are used for object extraction with the basic assumption that in the segmentation procedure the exact shape of the object is avoided and some approximate shape is used. In the proposed algorithm, each pattern of 16×16 block contains background and objects (Figure 2.21). In the proposed model, the unit blocks are referred to as micro-blocks (MBs) as they were called for video coding. These patterns are matched with the splitted regions of an image to find the best matched patterns. Then the regions are replaced with the pattern which leads to subdividing the region into two parts: accepted (objects) and not accepted (background). By merging all the regions which are connected and accepted together the final segmented objects or regions can be identified.

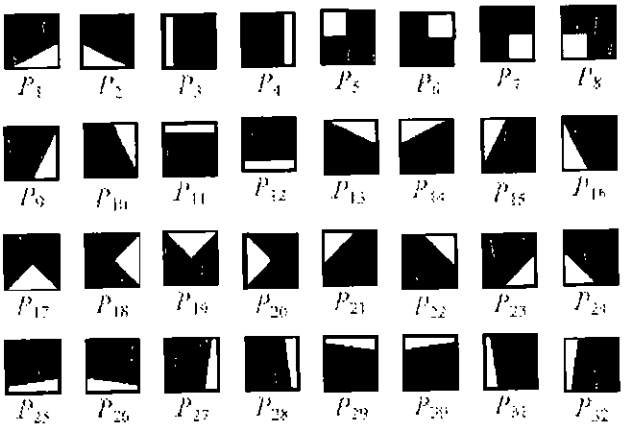


Figure 2.20: The pattern codebook of 32 regular shaped, 64- pixel patterns, defined in 16×16 blocks, where the white region represents 1 (motion) and black region represents 0 (no motion) [56].

(a) (b) (c)

Figure 2.21: Different types of patterns: (a) Object in the whole pattern, (b) the pattern containing both object and background, and (c) the pattern containing only background [56].

2.7 Region Stability Test

There are many theorem to identify region stability such as Z-test, T-test, Chi-square test, etc. among them T-test is considered in this thesis. T-test is universally proved and well recognized statistical theorem to identify the stability of a sample space from its sample values. Let S be a sample space and s be a set of samples such that $s \in S$. Applying T-test, it can identify whether the sample set s is stable within its sample space S or not [38]. For large sample set (over 30 samples), the 95% fiducially limits of population mean are given by

$$\overline{X} \pm t_{0.05} SE \tag{2.55}$$

And 99% fiducially limits of population mean are given by

$$\overline{X} \pm t_{0.01} SE \tag{2.56}$$

where, standard error of mean

$$SE_{\overline{X}} = \frac{\sigma}{\sqrt{n}} \tag{2.57}$$

$$t_{0.05}=1.96 \text{ and } t_{0.01}=2.58 \text{ [17]} \tag{2.58}$$

For small sample set (less than 30 samples), the decision depends on the value of t which is given by the following equation:

$$t = \frac{\overline{X} - \mu}{S}\sqrt{n} \tag{2.59}$$

Where,

$\overline{X}$ = mean of the sample,

μ = mean of the population,

n = size of the sample,

S = standard deviation of the sample,

95% fiducially limits of population mean are given by

$$\overline{X} \pm \frac{S}{\sqrt{n}} t_{0.05} \tag{2.60}$$

99% fiducially limits of population mean are given by

$$\overline{X} \pm \frac{S}{\sqrt{n}} t_{0.01} \tag{2.61}$$

Depending on degree of freedom value of $t_{0.05}$ varies from 12.71 to 2.04 and that of $t_{0.01}$ varies from 63.66 to 2.75. Degree of freedom for small sample is given by n-1 and for large sample it is infinity [38].

2.8 Intra-variance and Inter-variance Test

The OSSM algorithm [30] leads to produce regions having less intra-variance and more inter variances. The global solution can be achieved if the intra-and inter-variance test is applied. In the image segmentation domain, the number of clusters is not given *a priori*. Since, the number of clusters in segmentation algorithm is either not fixed or not manually provided, the minimization of the intra-region variability and the maximization of inter-region variability in the union of two regions are considered [71]. However, both straight minimization of intra-region variability and maximization of the inter-region variability lead to undesirable trivial solutions, being N regions or 1 region, respectively. The OSSM algorithm minimizes the intra-region variability while at the same time constraining the inter-region variability in the union of two regions. In this way, the inter-region variability constraint defines the scale at which two regions can be differentiated from each other. To minimize the intra- region variability, the sum-of-squared-error criterion is used such that it fits better to the model. Further, it implements the joint intra-region variability constraint as minimum variance for the union of two clusters [30]. At the same time, joining two clusters maximizes the inter-cluster variability. This leads to the following region growing model:

$$\min_{R} \sum_{R_i \in R} |n_i| \ \mathrm{var}(R_i) \tag{2.62}$$

where, n_i is the number of pixels in region i and $var(R_i)$ is the variance of the region i

$$var(Y) = \frac{1}{Y}\sum \| x - \mu(Y) \|^2 \tag{2.63}$$

Subject to,

$$\forall R_i, R_j, i \neq j;\ \max\ \mathrm{var}(R - \{R_i, R_j\} \cup (R_i \cup R_j)) < \max\ \mathrm{var}(R) \tag{2.64}$$

where, $maxvar(R) = var(var(R_k))\ \forall R_k \in R$, k is the number of splitted regions (2.65)

In general, the optimization of this model leads to region having intra-variance less than joint-intra variance of its neighbour and merging any two regions R_i and R_j decreases the inter-variance of the image. After application of these test stable regions that have minimum variance within themselves are obtained. At the same time, their inter-region variances are maximized. This test allows finding a global solution. Here threshold is dynamically updated thereby eliminates the problems of result anomalies due to threshold change. However, there are some rare situations in which individual clusters can violets these constraints [30, 71].

2.9 Concept of Human Perception

If the change of any object or feature is less than or equal to 0.5dB, then human perception is unable to detect the change [4]. Now, let the larger region be R_l and the small region be R_s. Presence of R_s collocated to R_l create distortion. We can merge R_s with R_l considering them belongs to the same object if they satisfy the following equation [4]:

$$0.5 = 20\log\frac{R_l}{R_s}$$

$$\Rightarrow \text{anti-log}((\frac{0.5}{20})) = \frac{R_l}{R_s}$$

$$\Rightarrow 1.059 = \frac{R_l}{R_s}$$

$$\Rightarrow R_l = 1.059 R_s. \quad (2.66)$$

So, it can be said that, for any two neighbouring regions, if the size of one region is less than or equal to 6% of another one then human perception cannot differentiate these two regions. This concept can be used to allow the SM to identify any non-patterned shaped object totally without any distortion.

2.10 Summary

In this Chapter, the basic concept of region based image segmentation is discussed first where special consideration is given to the widely accepted algorithm namely the split and merge (SM) algorithm. According to the SM algorithm, a larger region is splitted into four smaller regions if it exceeds the threshold. Again two adjacent connected regions are merged to form a larger region if its value is below the threshold limits. SM algorithm provides localized solution in image segmentation. Various modifications of the SM algorithm to outperform the original one are discussed. From those discussions it has been observed that single-stage merging as proposed in the SM algorithm can not come out of the local optimization due to feature diversities. Therefore, multi-stage merging has been thought off. Thresholds in the SM algorithm are fixed in nature thereby threshold need to be changed manually for different images to cope up with its object's feature diversities. A generalized thresholding method is needed to be applied on all types of images. Based on these two inherent problems of the SM algorithm a new algorithm is proposed namely the *robust object segmentation based on pattern matching* (ROSP) which is detailed in the next Chapter.

Proposed Pattern Based Object Segmentation

Split and merge (SM) algorithm is a well recognized widely used algorithm, but is unable to segment all type of objects in an image as huge variations in size, shape, colour and intensity are observed among objects within an image. The performance of the SM algorithm is also highly dependent on threshold values used for the split and the merge stages. Moreover, the existing SM algorithms require prior knowledge about the number of objects in an image. In many applications, it is not possible to know the number of objects in an image in prior to segmentation and hence fail to take advantages of the image segmentation concepts. Addressing these issues, a new algorithm namely *robust object segmentation based on pattern matching* (ROSP) algorithm is proposed in this chapter by integrating the region stability and the patterns for object extraction into the split and merge framework. The experimental results as discussed in Chapter 4 prove the superior segmentation performance of the ROSP algorithm in comparison with the *suppressed fuzzy c-means* (SFCM), and *object based image segmentation using fuzzy clustering* (OSF) algorithm.

3.1 Introduction

Object-based image segmentation, as described in Chapter 1, is the process of separating mutually exclusive homogeneous regions of interest in an image. Segmenting images for extracting objects is a very difficult and challenging task due to the huge number of objects with innumerable variations among them. Even though there are many techniques exist in the literature for image segmentation, the *split and merge* (SM) algorithm [64] is one of the most well known and widely used one, which works based on the region based image segmentation technique [1, 20, 29, 54, 58, 64, 73]. The concept of this algorithm firstly splits the image foreground into small regions and then merges the splitted regions to form final segmented regions. Both the splitting and merging are done based on the manually specified thresholds. A number of variants are also available for SM algorithms as discussed in Chapter 2 due to its popular capability of segmenting images in an efficient manner. Though all of these SM algorithms perform effective image segmentation, none of these are able to segment all types of objects in an image perfectly due to various limitations. For example, it is experimentally observed that large images with a small number of objects lead to good results for parallel SM algorithm while with the increase of the number of objects lead this algorithm to perform gradually poorer. The trade off analysis between the reliability and the spatial resolution inherent with the significance test, moreover, yields this algorithm incomplete.

All of the existing clustering image segmentation algorithms are not able to be applied in an unknown image where the number of objects is not known in prior. This put limitations to segmentation algorithms of not being able to be applied in many applications.

In this thesis, a new algorithm called *robust object segmentation based on pattern matching* (ROSP) is proposed considering the basic SM algorithm, the region stability, and the pattern matching for object extraction. The region stability implies the concepts of finding the bounded homogeneity of an object within the images. Different techniques for measuring the region stability include the Z-test, the T-test, the Chi-square test and so on. The concept of pattern matching is very much familiar in video coding [56, 59, 61-63] for improving very low bit rate coding techniques. In these techniques, 16×16 micro-blocks (MB) are normally used for motion estimation. The size of MBs can be varied from 16×16 to 4×4. The larger the size of MBs, the smaller the size of information, while small sized MBs lead to better quality images or videos [56, 59, 61-63]. The blocks are three types: full motion, partial motion and no motion. The partial motion blocks can be represented efficiently by using some predefined patterns. The ROSP algorithm proposed in this thesis uses the concepts of patterns in the SM algorithm to increase the performances and efficiency of it significantly, which reduces the complexity of the algorithm leading to the ability of segmenting any kind of images. Hence, the novelty of this algorithm lies on the ability of separating objects from an unknown image, besides its efficiency.

The proposed ROSP algorithm is divided into three stages: the split, the pattern matching and the merge stages. The split stage divides the image recursively based on some measures of the region stability, the pattern matching stage matches the splitted region with some pre-defined patterns using dynamic variance parameters while the region stability concepts and the parameters of dynamic variance and connectivity are applied during the merge stage to make the algorithm robust and efficient.

3.2 The Split Stage

The proposed ROSP algorithm performs multistage splitting of images, while the region stability test is applied to each region resulted from the splitting. After each splitting operation on an image, the image is divided into a number of homogeneous or heterogeneous image segments or regions. These homogeneous or heterogeneous regions are classified into one of the following three classes: 1) Background region, 2) Object region, and 3) Mixed region.

The splitted background region does not contain any object or a part of it except the background defined as the region out of interest. The object region is defined as the region of which, all the pixels inside it constitutes an object as a whole or part of it. The object region does not include any background. It is assumed that all the pixels inside the object region belong to only one object. The mixed region contains both the object(s) or a part of it and the background. The mixed region therefore may have 1) only one entire object; 2) only a part of an object; 3) only parts of multiple object; 3) multiple entire object; 4) one entire object and a part of another object; 5) multiple entire object and a part of another object; 6) multiple entire object and parts of multiple object. To identify the exact scenario for evaluating the image for segmentation further splitting is required. Hence, the splitting of image stage in ROSP algorithm works recursively until the region stability based terminating condition is achieved. There are three sub-stages in the splitting stage of ROSP algorithm: the resizing of the image, the initial split stage and the recursive split stage.

3.2.1 Resizing the Image

There are various resolutions used in different images. The ROSP algorithm proposed in this Chapter works either in adaptive way by taking images with any resolution and dividing it into a number of squared regions with each dimension is a multiple of 16 before applying any test on each region, or set to a specific set of parameters with specific resolution and number of regions in first splitting operation. To apply prefixed specific parameters for the algorithm, the image has to be resized into one of the standard resolution, say 1024×768, which may have 12 regions in initial split operation.

3.2.2 The Initial Split Stage

The initial split operation of the ROSP algorithm is performed based on the resolution of the image taken or resized. For an image having a resolution of 1024×768, the initial splitting operation divides the image into 12 regions as shown in Figure 3.1 so that each region's becomes N×N, where $N = 16 \times 2^n$ and n = 0, 1, ... If patterns are taken in M×M pixel-blocks, then N must be multiplied by M in the initial split stage to use the algorithm directly on it. For an image with resolution 1024×768 and 12 initial regions, the dimension or resolution of each initial region becomes 256×256. The term 16 is used since the ROSP algorithm is using patterns reside in 16×16 pixel-blocks. The main object of the initial split stage is to divide the image into squared regions where dimensions are multiple times of pattern-block or pixel block dimension.

R_1	**R_2**	**R_3**	**R_4**
R_5	**R_6**	**R_7**	**R_8**
R_9	**R_{10}**	**R_{11}**	**R_{12}**

Figure 3.1: The initial splitting stage of ROSP algorithm divides a 1024×768 image into 12 regions.

3.2.3 The Recursive Split Stage

The initial splitting stage of the ROSP algorithm generates a number of squared regions each of which is either the background region, the object region or the mixed region. Once the initial squared regions $\{R_i : i = 1, 2, \ldots\ldots, n\}$ are found, two measures are obtained for each region: 1) the average pixel value; and 2) the fiducially limit. The fiducially limit can be obtained by applying a stability test, called T-test, to each of the regions R_i. If the fiducially limit is above a threshold, K_F, say 95%, then the region is considered as the stable region. A region stability predicate can, therefore, be defined based on the region stability as follows:

$$P(R_i) = \begin{cases} 1, & \text{if fiducially limit} \leq K_F \\ 0, & \text{Otherwise} \end{cases} \tag{3.1}$$

The average pixel value of a region having resolution N×N is calculated using the equation,

$$A(R_i) = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y) \tag{3.2}$$

where $f(x, y)$ is the pixel intensity at location (x, y).

The average pixels' value and the region stability predicate value of a region is, then, used to classify the region as follows:

1. B (background) region: the average pixel value $< K_{th}$ and $P(R_i) = 1$.
2. O (object) region: the average pixel value $\geq K_{th}$ and $P(R_i) = 1$.
3. M (mixed) region: $P(R_i) = 0$.

where K_{th} is a predefined threshold used to separating an object from its background.

For classifying regions, at first the predicate value $P(R_i)$ is determined. If $P(R_i) = 0$, then the region is classified as the mixed region, called M region, containing both object/object portion and background. If a mixed region is obtained, one of the two following decisions is taken:

1) If the region has the resolution of {N×N: N=16}, a close match pattern are identified to replace the region.
2) Otherwise, the region with resolution $\{N \times N : N = 16 \times 2^n \text{ and } n = 0,1,2,\ldots\}$ is sub-divided into four equal region each of which will have resolution of $\frac{N}{2} \times \frac{N}{2}$.

In each stage of the recursive splitting, a mixed region is subdivided into four regions. It is continued until either the region becomes stable or a small region with the resolution of $\frac{N}{2} \times \frac{N}{2}$ is obtained. Figure 3.2 shows the recursive sub-division process of a region in recursive splitting stage.

The stability predicate value $P(R_i) = 1$ implies the region is stable and no further splitting is necessary. The stable region is further be classified as the object/object portion or the background. The average pixels' value is then used to mark the region as either O region or B region based on whether the value is above or below a predefined threshold K_{th}, respectively.

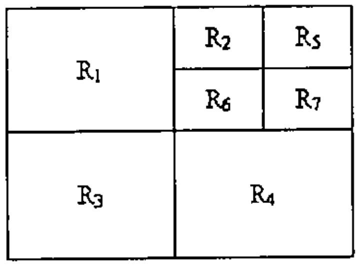
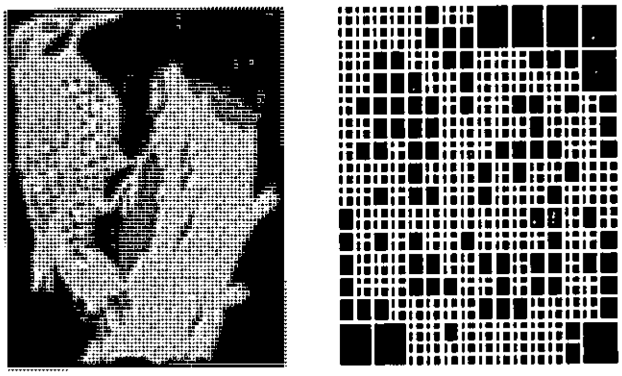
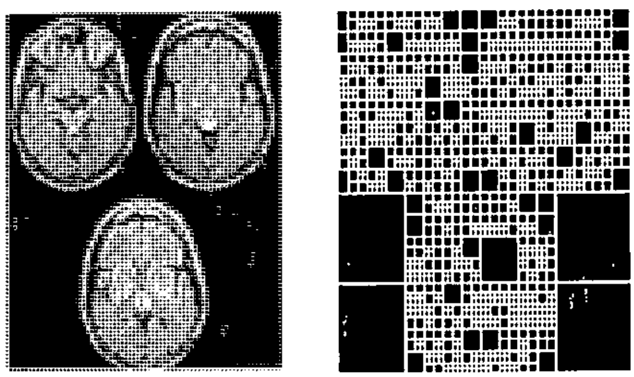


Figure 3.2: Images are splitted recursively in recursive splitting stage of the ROSP algorithm.



(a) (b)

Figure 3.3: The region splitting process in the wood-bird image: (a) Original Image (b) Splitted Image.



(a) (b)

Figure 3.4: The image splitting process in the brain image: (a) Original Image (b) Splitted Image

Figure 3.3 and Figure 3.4 show the region splitting process in details for two real-life examples using two gray scale images named, the wood-bird image and the brain image. The examples shows that some larger region is not required to be splitted further and reduced the segmentation complexity.

3.2.4 Region Information Table

To track the progress of splitting process and maintaining the properties of each region, a table is maintained as shown in Table 3.1. Each row of the table carries information about a splitted region.

Table 3.1: Region information table.

Region Number (i)	Initial Coordinate (S_x, S_y)	Region size (N)	Region Type	Stability $P(R_i)$	Avg Pixel Value	Object ID	Matched Pattern Number
1	(0,0)	256	B	1	168		-
2	(0, 256)	256	O	1	102		-
3	(0, 512)	256	M	0	140		-
4	(0, 512)	64	M	0	124		-
--------	-------	------	------	---	----	------	-------
n	(256, 256)	16	M	0	67		20

After initial splitting of each image into regions, the regions are numbered following a specific sequence, say from top to bottom and left to right. The region number, i, is used to identify each of these regions. When a region is further divided, it region will be deleted from the table and newly generated regions will be added such that the first one will get the previous number and the remaining three will get new region number. The initial coordinates indicates the starting coordinates in monitor space for each region. The region size N in table implies that the region has a resolution of N×N. The 32 patterns as discussed in Section 2.6 are number from 0 to 31. These numbers are used in the column of pattern number.

3.3 Pattern Matching Technique

The patterns shown in the Section 2.6 are used in segmentation to match the splitted regions that are accepted as the foreground. If the size of the M region is $a \times b$, say $a = b = 16$, the percentage of matching of a region R_i with any pattern P_j can be calculated using the following equation:

$$\eta = \frac{\sum_{x=1}^{a}\sum_{y=1}^{b} f(x,y)}{a \times b}, \quad \text{where } f(x,y) = \begin{cases} 1, & \exists i,j : R_i(x,y) \neq 0 \text{ and } P_j(x,y) = 1 \\ 0, & \text{Otherwise} \end{cases} \tag{3.3}$$

If $\eta \geq 95$, the pattern P_j is said fully matched with the region R_i while $60 \leq \eta < 95$ represents the partially-matched of the pattern P_j with the region R_i and $\eta < 60$ being the unmatched of P_j with the region R_i [3]. Any region which is unmatched with all the patterns in the pattern codebook is marked as background. If a pattern is fully matched, then the pattern number is added into the table. For any partially matched patterns, the algorithm searches for more patterns to find the best match. If any best matched pattern is not found, the region is considered as random or scattered region and mark it as background.

3.4 Merge Stage

A number of squared shaped regions $\{R_i : i = 1...n\}$ with different sizes and types are obtained in the region information table after the completion of split and pattern matching operations. Some of the mixed region with minimum size contains a pattern number from the list of patterns which is used to replace the region. The merging stage constructs entire objects through merging the O regions and the patterns based on matrices of following three criteria: 1) the binding or initial point; 2) the stability mapping threshold; and 3) the connectivity. The initial or binding point is the starting point of any region. When regions are merged together to get the entire object, the selection of region are performed based on this initial or binding point as any object is assumed to comprises of connected regions. The stability mapping threshold determined whether two connected O regions or patterns are belong to the same object or not. If the difference between the average pixel values of two connected region is under a threshold, the regions are assumed to be under the same object. The connectivity test ensures that misplaced patterns and O regions are not taken within the object. For the first time in this thesis, a 8-way connectivity method (instead of 4-way connectivity) is used to analyse the connectivity efficiently to find out the connected region of an object.

The proposed ROSP algorithm uses multistage merging as the regions were constructed using recursive splitting. The importance and performances of multistage merging [15] is discussed in Section 2.3.7. Incorporating the multistage merging, the ROSP algorithm merges O regions and patterns based on matrices of three measurement parameters:

(i) The region stability;
(ii) The selection patterns; and
(iii) The connectivity.

3.4.1 Merging on the Basis of Region Stability

After the splitting and pattern matching operations, all the entries in the region information table becomes stable, which is indicated by the region stability predicate $P(R_i)$. If any region fails to become stable, then the region is discarded from consideration of being a portion of any object and regarded as the B region. Since the merging operation combines two parts of an object which are unexpectedly separated by hard partitioning in the split stage, these parts, whether the O regions or patterns, must be correlated with each other by the minimum inter variance, determined by a variance threshold. After merging two regions, the stability test is also carried out to ensure that they are under the same object. Merging of two regions or patterns or region with pattern may lead to the entire object or still a portion of object. The recursive implementation of merging operations on all the regions and patterns of the region information table is necessary to constituent all of the entire objects within the image.

3.4.2 Merging on the Basis of Selected Patterns

Like O regions, patterns selected for a specific smallest region is also merged with the larger region already accumulated for obtaining an object. The squared region containing a pattern is called pattern region. As shown in Section 2.6, each pattern within a pattern region has two or three special sides, called the object sides, through which the pattern can be connected with the remaining object. The object side of a pattern region is important while merging a selected pattern with the accumulated regions. All of the following properties must be satisfied for merging a pattern region with the target region:

1. The object side of the pattern must match with the connected side of target region. This match is computed through pixel mapping and the match is satisfied when a predefined, say, 80% of the boundary points are the same.
2. The mapping region of the object with the pattern must satisfy the stability predicate.
3. The intra-variance of the pattern must be within the acceptable threshold and the inter-variance between the pattern and the immediate region at the pattern side must be within the threshold.

Note that a pattern region cannot be connected with any target region or another pattern region along without the object side of the pattern. Including a pattern into an object implies that the object is terminated by the non-object side of the pattern.

3.4.3 Merging on the basis of the Connectivity

The stability test ensures that the O regions and patterns having inter-variance under a certain threshold are combined together to form a larger region which satisfy the stability predicates and ensures a minimum intra-variance acceptable for an object. Although this merging process produces the target object or part of it, it is highly likely that some portions of object or regions or patterns may still be missing (or added in some cases) making the object distorted. The connectivity operation ensures that connected regions or patterns are tested for its suitability of being part of an object by measuring the stability and variances of that regions or patterns with the already extracted portion of object. The existing 4-way connectivity [3] may still distort the objects at the diagonal of the region of interest, this thesis considers 8-way connectivity to ensure the efficiency of the merging process and generate better quality objects. Figure 3.5 shows the differences between the 4-way and 8-way connectivity operations. In the 4-way connectivity operation, only the regions or patterns on the left, right, top and bottom are considered, while the 8-way connectivity operation considers all the regions or patterns around the region of interest including the regions or pattern in the diagonals. The 4-way connectivity in Figure 3.5(a) indicates that the connectivity with the region of interest, say region 1 is tested with the four regions numbered 2, 3, 4 and 5. The 8-way connectivity as shown in Figure 3.5(b) indicates that the connectivity with the region 1 is tested with the regions 2 to 9. The merging of connected regions continues repeatedly until any connected regions or patterns are found which are classified as B regions or lack of intra- or inter-variance correlation is reached.

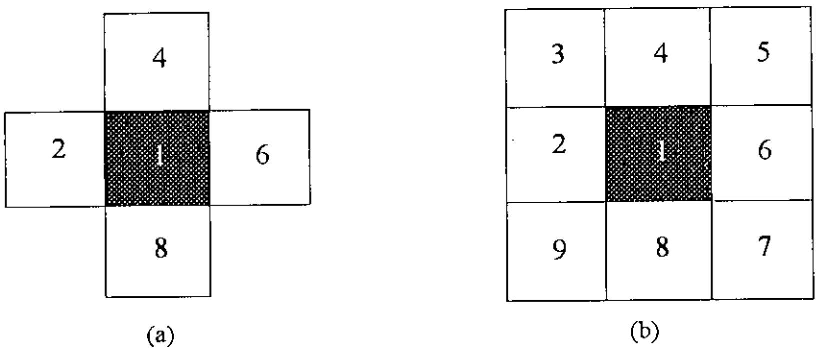
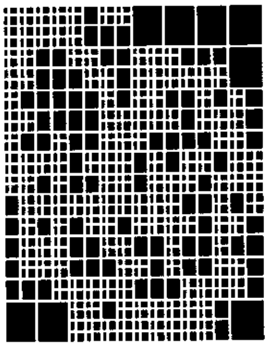


Figure 3.5: Differences between (a) 4-way connectivity, (b) 8-way connectivity of merging.

The concept of merging by connectivity is applied at the last step of merging stage so that the smaller regions are connected with their neighbouring connected larger object part or region to avoid hedge on the boundary of the identified objects. This gives better shape of the objects and the smoothness at the boundary line of it. This connectivity test also rejects the misleading or false positively identified region from the objects so that the actual object is separated leaving the connected noise.

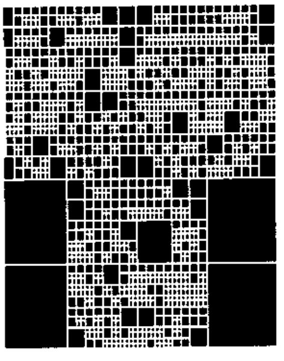
Figure 3.6 and Figure 3.7 shows the extracted objects from the splitted regions and patterns for the wood-bird and the brain images.



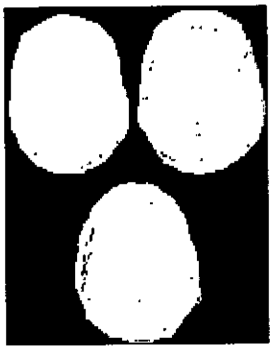
(a)

(b)

Figure 3.6: Merging of O regions and patterns for the wood-bird image: (a) splitted image and (b) image after merging.



(a)



(b)

Figure 3.7: Merging of O regions and patterns for the brain image: (a) splitted image (b) image after merging.

3.5 Proposed Robust Object Segmentation based on pattern matching (ROSP) Algorithm

This section presents a novel image segmentation algorithm, namely *robust object segmentation based on pattern matching* (ROSP), which enhances the segmentation process by reducing the complexity and increasing image quality. Three main constituent parts of this algorithm are: (i) split stage, (ii) pattern matching, and (iii) merge stage.

In the split stage of the algorithm, the image is resized, if necessary, so that it can be divided into a number of regions each of which has a resolution of $\{N \times N : N = 16 \times 2^n\}$. Then the stability test and average pixels' value are estimated to classify the region into O region, B region and M region. The M region is then further splitted into four equal squared regions until all the regions are classified as either O region or B region or the smallest size for any region say 16×16 is achieved. If the minimum sized region is still classified as mixed region, then a suitable pattern is selected from the pattern codebook of 32 predefined patterns to replace the mixed region with a pattern. In the pattern matching phase, a similarity parameter η is calculated to find the best matched pattern from the pattern list. If the minimum region does not match any of the patterns, then the region is considered as scattered region or background. Once the regions are classified and patterns are selected, then the regions and patterns are merged together using the stability measure, edge similarity and intra- and inter-variance correlation to obtain the objects within the image. The detail algorithm is given below:

Algorithm 3.1: *Robust Object Segmentation using Pattern Matching* (ROSP) algorithm.

Pre condition: Image with objects to be segmented, pattern codebook.

Post condition: Final segmented regions $\Re$.

1. Resize the image with resolution 1024×768 pixels
2. Split the image into 12 regions each of which have a resolution of 256×256 pixels.
3. While there is any region i not mark yet do

```
        if P(R_i) = 1 then
            if A(R_i) ≥ K_th then
                mark the region_type with O
                region(i).pattern_id =-1;
            else  mark the region_type with B
            endif
        else
            if the region resolution is 16×16 then
                mark the region_type with M
            else  split the region into four equal regions
            endif
        endif
```

endwhile

4. While there is any region i mark with M do

    region(i).patternid = -1.

    similarity = 0

    For each pattern with pattern ID j in the pattern codebook do

    Calculate $\eta = \frac{\sum_{x=1}^{a}\sum_{y=1}^{b} f(x,y)}{a \times b}$

    if $\eta > 95$ then

        region(i).patternid = j; break for loop;

    else if $\eta > 60$ then

        if $\eta >$ similarity then

            similarity = η

            region(i).patternid = j

        endif

    endif

    endfor

    mark the region_type with O

    region(i).object_id = 0

    if region(i).patternid = -1 then

        mark the region_type with B

    endif

endwhile

5. Obj_id =0

While there is any region with object_id 0 do

    Obj_id = Obj_id + 1

    Take an region i with object_id 0 and call the Find Object Algorithm with the connected region and *obj_id*

endwhile

Algorithm 3.2: *Find Object Algorithm*

Pre condition: Region information table, region i, obj_id, pattern codebook.

Post condition: Final segmented regions $\Re$.

set region(i).object_id = *obj_id*

while there is any connected O region around the region i do

  calculate the stability predicate value $P(R_i)$ and inter-variance between the region i and the connected region.

  if region(i).pattern_id = -1 and its connected region is O type and satisfy $P(R_i)$ and inter-variance then

    call Find Object Algorithm with the connected region and *obj_id*

  else if region(i).pattern_id > 0 and its connected region is O type and satisfy $P(R_i)$ and inter-variance then

    if the connected region is in the object side then

      call Find Object Algorithm with the connected region and *obj_id*

    endif

  endif

endwhile

3.6 Summary

This chapter has proposed a new image segmentation approach called *robust object segmentation based on pattern matching* (ROSP) algorithm to address some of the limitations inherent with the well known image segmentation algorithms, including the *split and merge* (SM) algorithm and fuzzy clustering algorithms. The ROSP algorithm firstly splits the images into several regions until the image stability is achieved or the region size becomes 16×16 which is regarded as the minimum region size. If the minimum sized region still contains both foreground and background, these regions are replaced, for the first time, by the best matched predefined patterns. The algorithm also considers the region stability and both intra- and inter-variances to incorporate the visual perception properties of human observations. This algorithm, for the first time, able the segment any image with any number of objects correctly without asking for any prior knowledge about the number of objects. Hence, the algorithm provides a generalized approach for segmenting any types of images and resolves the problem of all previous algorithms each of which is applicable to segment a specific type of images.

Experimental Results and Analysis

The performance study for the proposed *robust object segmentation based on pattern matching* (ROSP) algorithm presented in Chapter 3 has been carried out through experiments using MATLAB 7.1 on different types of images having different numbers, shapes and sizes of objects. The experiments use gray-scale images considering the intensity and location of pixels as the feature of extraction for comparing the proposed algorithm with the *suppressed fuzzy c-means*, and the shape-based fuzzy clustering algorithm namely *object based image segmentation using fuzzy clustering*. The qualitative analysis on the experimental data proves the superiority of the proposed ROSP algorithm and hence upholds its suitability for applying it in various applications including segmentation based low bit rate video coding.

4.1 Introduction

There are several philosophical strategies for segmenting objects from an image. The split and merge and fuzzy techniques are the most popular among them. Despite their popularity, none of these techniques can provide satisfactory performance by segmenting all types of images, nor can they be applied to an unknown image. This creates motivation for moving towards new and generalized approaches for image segmentation, which will be robust at the same time efficient. The proposed novel *robust object segmentation based on pattern matching* (ROSP) algorithm is resulted from this motivation. This algorithm for the first time has incorporated patterns in the segmentation process working based on the philosophy of the split and merge.

The segmentation algorithm separates the objects, i.e., the region bounded by a non-linear line, despite of the colours of images inside it. Most of the colour images when converted to the gray-scale images can still signify the object-boundary by the intensity of the pixels within and outside the object, although for some cases the intensity of gray-scale image pixel may be same for two different colours. Keeping the difficulties and progress of image segmentation in mind, analysing the performances of a new method based on gray-scale images is almost equally acceptable as the algorithm can then be applied with very minor modification for colour images. Moreover, in most cases including the medical image diagnosis, both the colour and gray-scale images provide the same visual perception while detecting objects within the images. Hence, without loss of generality, it is assumed in this thesis that all the images will be converted to gray-scale at first before being used for the performance analysis of different algorithms.

Though the intensity values of pixels within an image vary continuously between the maximum and the minimum, the pixels are classified or clustered in two parts: objects or the foreground and the background. To separate these two parts, this thesis carried out analyses of two tests, the stability test and the analysis of average pixels' values. For the sake of implementation efficiency, once a region is identified as background, all the pixels' values in that region are set to 0 (zero) in the experiments of this thesis. Similarly if necessary, the background of an image can be retrieved by setting all pixels in foreground to 0. The concepts can be extended for colour images if a mapping table is generated to represent the foreground through replacing the pixels with 1. The experimental results are produced by generating mapping tables for reviving the generality.

There are several algorithms available in literature for image segmentation. Among them the *object based image segmentation using fuzzy clustering* (OSF) and the *suppressed fuzzy c-means* (SFCM) algorithms developed recently outperforms all of the existing algorithms. Therefore, the ROSP algorithm proposed in this thesis is compared with these two algorithms to prove its superiority in performances.

4.2 Generality Analysis

There is no single unified definition of what exactly constitutes an object as there typically exist a large number of objects and their definition very much depends on the user's perception and purpose of the application. Hence, a major consideration in any clustering algorithm, especially in object-based image segmentation, is how to determine the actual number of clusters (i.e., objects) either directly from image data or using a priori knowledge. In any clustering algorithm, the number of objects can either be 1) provided manually; or 2) determined automatically from image data. In former case, there should have sufficient prior knowledge about the image and number of objects so that the number can be inserted in the prior of running the image segmentation algorithm. Therefore, this algorithm cannot be applied for an unknown image, limiting the real-world applications of it. There are numerous applications, for instance in the manufacturing and medical imaging [58] domains, where the number of objects to be segmented is known a priori, and so for all the clustering algorithms which are considered in Chapter 2 of this thesis, the number of objects is provided manually. In the latter case, the standard approach adopted is to use validity measures to determine an optimal number of objects, with examples provided in [12, 21, 24, 25]. There are many situations however, particularly in object-based segmentation where such validity algorithms fail to generate the correct (optimal) number of clusters [21, 23, 45] because they tend to focus on the homogeneous regions of interest in an image, which can contradict with the human perception of an object and hence degrading the overall performance of the segmentation algorithm.

The main feature of the proposed ROSP algorithm is that it can determine the object correctly and automatically in most of the cases without any prior knowledge about the number of objects in the given images and hence the algorithm can be applied to any object whether it is known or unknown.

4.3 The Complexity Analysis

One of the main features of the proposed ROSP algorithm is to reduce the complexity by using patterns. The algorithm has two parts: linear part and recursive part. The algorithm 3.1 executes its operation linearly and referenced a recursive algorithm (Algorithm 3.2) for finding the entire object within it. Therefore, to study the complexity of Algorithm 3.1, it is required to study the complexity of Algorithm 3.2 at first as it is embedded within it. The detail analysis of the complexity measures for Algorithm 3.2 and then Algorithm 3.1 is given below:

4.3.1 The Computational Complexity for Algorithm 3.2

The Algorithm 3.2 calculates the stability measures and inter-variances of each region. If there are n pixels in the image the algorithm need to use all the n pixels, in worse case, when predicate stability and inter-variances are estimated each time. So, the complexity required for estimating the stability and inter-variance is O(n). The Algorithm 3.2 is a recursive one executing O(n) times. Hence, the overall complexity of the Algorithm 3.2 is O(n)×O(n) = O(n^2).

4.3.2 The Computational Complexity for Algorithm 3.1

There are five steps in Algorithm 3.1. The computational complexity in each stage is given below:

***Step* 1:** this step resizes the image in the resolution of 1024×768. This resizing needs to analyse all the n pixels in the image for mirroring, resulting the computational cost of O(n).

***Step* 2:** dividing the image into 12 regions requires the computational complexity of of O(1).

***Step* 3:** computational complexity to be required for performing the stability test is O(n), estimating the average pixel value is O(n) and assigning region type for each region is O(log$n$). So, the overall complexity required in this step is O(n) + O(n) + O(log$n$) = O(n).

***Step* 4:** computational complexity required to calculate the pattern matching parameter η is O(n) and checking all blocks as well as assigning region type and pattern id requires the complexity cost of O(1). The process is repeated O(log$n$) times. This provides the total complexity of O(log$n$) × (O(n) + O(1)) = O($n$log$n$).

***Step* 5:** in this step, the Algorithm 3.2 is referenced at O(log$n$) times. Since the complexity of the algorithm 3.2 is O(n^2), the overall complexity of this step is O(n^2) × O(log$n$) = O(n^2log$n$).

Hence the total complexity required for this algorithm is O(n) + O (1) + O(n) + O($n$log$n$) + O(n^2log$n$) = O(n^2log$n$).

4.4 Qualitative Analysis

The subject or qualitative evaluation for the proposed ROSP algorithm has been carried out in two steps. In first step, it is analysed that the proposed ROSP algorithm can segment and identify the object; while in the second step the proposed algorithm is compared with the SFCM algorithm and the OSP algorithm.

For the simplicity, all the experiments in this Chapter have been carried out using eight patterns as shown in

Figure 4.1 and the results produced are discussed below:

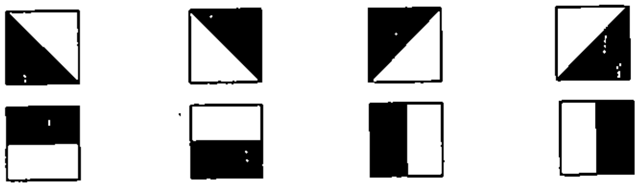
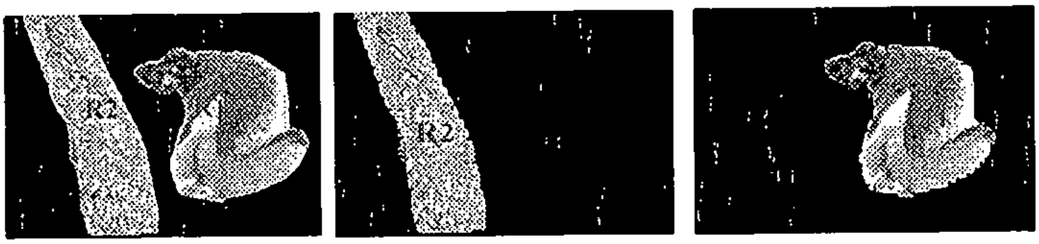


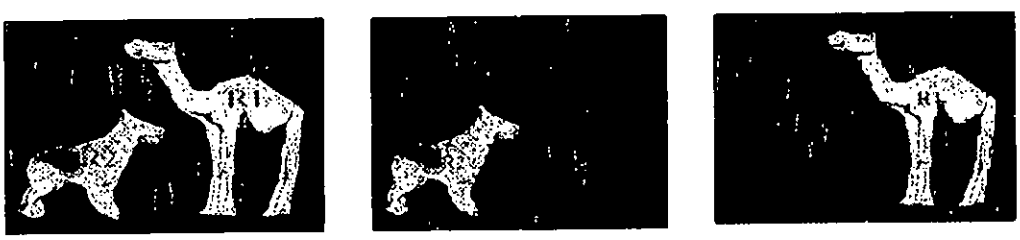
Figure 4.1: Eight patterns used for image segmentation in this chapter.

Figure 4.2 shows the segmented results of the wood-squirrel image where objects such as the wood and the squirrel are separated using the ROSP algorithm. The figure shows that the objects are separated perfectly, but due to using eight patterns the edges of the objects have been distorted slightly. The distortion does not change the shape of the object and can be improved by using more patterns. Similarly, Figure 4.3 and Figure 4.4 show the segmentation results produced for the images of the dog-camel and the fish-jar images respectively.



(a) (b) (c)

Figure 4.2: Segmentation of the wood-squirrel image: (a) the original image; (b) the first segmented object; and (c) the second segmented object.



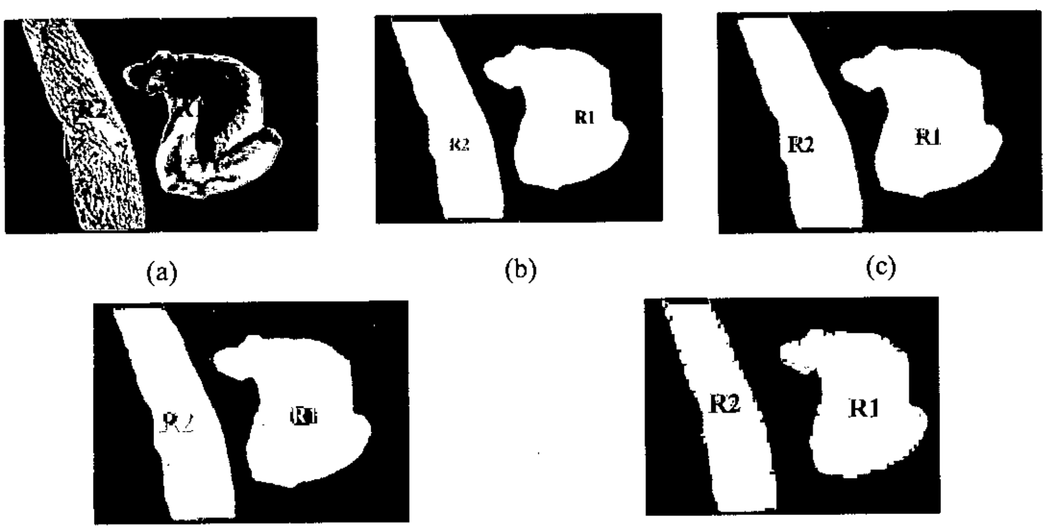
(a) (b) (c)

Figure 4.3: Segmentation of the dog-camel image: (a) the original image; (b) the first segmented object; and (c) the second segmented object.

(a) (b) (c)

Figure 4.4: Segmentation of the fish-jar image: (a) the original image; (b) the reference image; (c) segmentation of the first image; and (d) segmentation of the second image.

The subjective performances of the proposed ROSP algorithm compared with the SFCM and OSF algorithms are carried out for three images the wood-squirrel, the man-dog and the brain images respectively. Since the OSF algorithm outperforms the SM algorithm and the SM algorithm produced poorer segmentation results, analysis has been conducted on the results of the OSF and the ROSP algorithms. Reference images represent different objects with different colours and numbers for easy assimilation. The experimental results are compared with that of reference images.



(a) (b) (c)

(d) (e)

Figure 4.5: Segmentation of the wood-squirrel image: (a) the original image; (b) the reference image; (c) the segmentation mapping produced by the SFCM algorithm; (d) the segmentation mapping produced by the OSF algorithm; (e) the segmentation mapping produced by the ROSP algorithm.

The segmentation results for the *wood-squirrel* image for different algorithms have been analysed and the segmented results compared with the reference image is given in Figure 4.5. The figures clearly indicate that both the SFCM and the OSF algorithms misclassified the images as some portions of the squirrel have got the same colour of the wood. Although there are some distortions in the edges of the objects obtained in the ROSP algorithm, the result is undoubtedly better while it does not misclassify any object portion. Moreover, the distortion can also be minimized by using 32 patterns as explained in Chapter 2 rather than 8 patterns used.

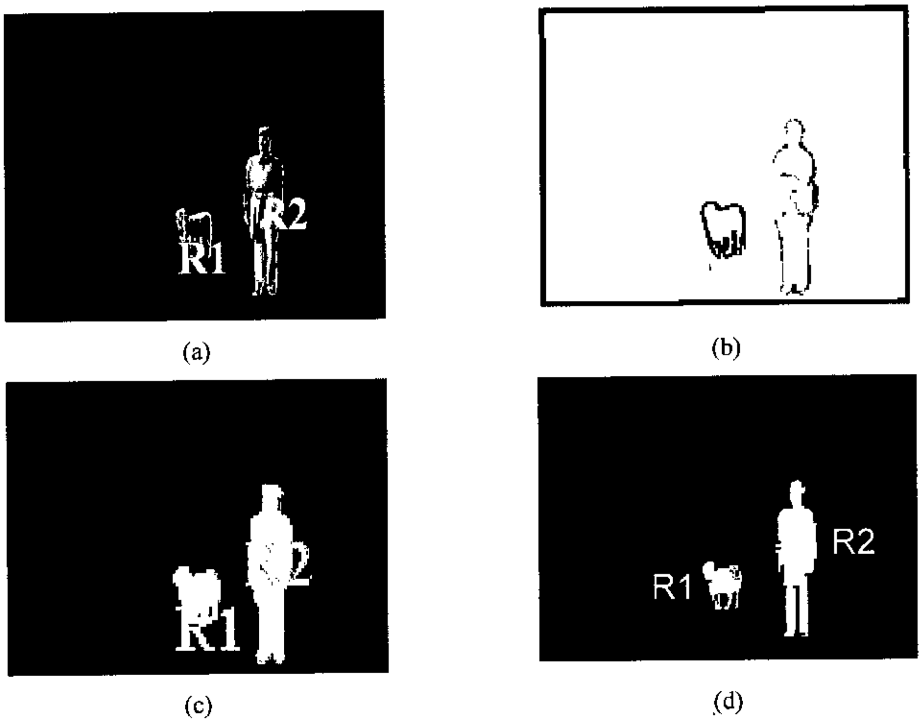


Figure 4.6: Segmentation of the man-dog image: (a) the original image; (b) the reference image; (c) the segmentation mapping produced by the OSF algorithm; and (d) the segmentation mapping produced by the ROSP algorithm.

Figure 4.6 presents comparative results between the OSF algorithm and the proposed algorithm using the *man-dog* image. As like the split and merge (SM) algorithms, the SFCM algorithm produces poorer segmentation performance of this image and hence the segmentation results of this algorithm has been omitted. The comparison between the reference image and the segmentation mapping between produced by the OSF algorithm clearly indicates that the OSF algorithm misclassified some portion of human as the dog due to improper scaling of objects during iterative process. The ROSP algorithm, on the contrary, classified all the objects correctly although some distortions are observed in the edges of the object. The reason for this distortion is that the original splitted blocks of the image are replaced with the given patterns which are not containing same shape as in the original image. This distortion can be reduced by including more patterns. Despite this, as the distortion is more acceptable than misclassifying the connected object portion, the performances of proposed ROSP algorithm is considered significantly better than the OSF algorithm.

The experiments on the *brain-image* shown in Figure 4.7 indicates that, compared with the reference image generated manually, the SFCM algorithm using combinations of pixel intensity normalized pixel locations fails to classify the objects due to their variations in pixel intensities and two objects being very close. The OSF algorithm can segment the objects almost correctly as shown in Figure 4.7(d) although it fails to retrieve the original shape observed in the original image. The correct classification of the images obtained by the OSF algorithm actually depends on the initialization and in case of any wrong initialization the result could be different. The proposed ROSP algorithm, on the other hand, classifies the objects correctly by using connectivity and the pattern matching concepts. From the results generated by the ROSP algorithm, it is visually apparent that there are some shape distortions of objects due to the mismatch of the shape of the pattern blocks with the splitted original blocks, which is also observed in other images.

(a) (b) (c) (d) (e)

Figure 4.7: Segmentation of the brain image: (a) the original image; (b) the reference image; (c) the segmentation mapping produced by the SFCM algorithm; (d) the segmentation mapping produced by the OSF algorithm; (e) the segmentation mapping produced by the ROSP algorithm.

4.5 Evaluation

The proposed ROSP algorithm is compared with the OSF and the SFCM algorithms developed recently. The OSF algorithm is developed addressing the problems of the SM algorithm and the SFCM algorithm by addressing the problems of fuzzy algorithms, covering both the domains. The superiority of the proposed ROSP algorithms compared with the OSF and the SFCM algorithm is listed below:

1. Both the OSF and the proposed ROSP algorithms use multi-stage merging after splitting the images. However, the OSF algorithms has the computational complexity of $O(n^3)$, which is reduced to $O(n^2 \log n)$ in the proposed ROSP algorithm.

2. The OSF algorithm highly depends on the initialization value and any wrong initialization may lead this algorithm to poor performances. The ROSP algorithm depends on the region stability, connectivity and pattern matching, and hence does not require any initialization.
3. The OSF algorithm suffers from scaling problems but the ROSP algorithm shows robust result due to the use of T-test for object stability and pattern matching.
4. The SFCM algorithm, in many cases, cannot classify objects having high pixel variations, while both the OSF and ROSP algorithms can segment such an object.
5. Both the SFCM and the OSF algorithms misclassify objects due to its failure of stability and connectivity test. The proposed ROSP algorithm solves this misclassification problem.
6. For both the OSF and the SFCM algorithms, the number of objects needs to be known in prior, while the ROSP algorithm can separate objects from any unknown image.
7. The main limitation of the proposed ROSP algorithm is that it distorts the edges of the retrieved object if the number of patterns is not sufficient. This problem requires some future works for being resolved.

4.6 Summary

This Chapter shows the performance analysis of the proposed *robust object segmentation based on pattern matching* (ROSP) algorithm compared with the *suppressed fuzzy c-means* (SFCM), and the *object based image segmentation using fuzzy clustering* (OSF) algorithms. The experiments show that the ROSP algorithm can segment any type of image including unknown images for the first time and exhibits superior performances. The main limitation of this algorithm is that it generates distorted edges of an object due to replacing the original region of objects by patterns. This requires further study and research in the field. The ability of the ROSP algorithm to segment object from an unknown image makes the algorithm robust to be applied in any domain including object-based video streaming.

Conclusions and Future Works

Object-based image segmentation is widely used in many different practical applications from medical imaging to robotic vision, but is very challenging because in most real world images there are large number of perceptual objects and wide variations amongst them. To achieve superior segmentation results to meet the demand from leading edge multimedia applications, it is mandated to segment objects by classifying the splitted regions into different categories and segment them accordingly.

In this thesis, a novel concept of image segmentation is provided by proposing a new algorithm called the *robust object segmentation based on pattern matching* (ROSP) algorithm. The ROSP algorithm segments objects based on region stability test or t-test, connectivity and pattern matching which were not considered previously. The T-test actually classifies the splitted regions as the background region, the object region and the mixed region. The most interesting part of this object segmentation process is the mixed region which contains both the object and background and lies at the contour of an object. The mixed region is recursively splitted until a small region containing 16x16 pixels is obtained, otherwise being classified to either the background or the object region.

The merging, splitting and pattern matching techniques used in the algorithm makes the algorithm robust to segment all types of objects in an image including the scattered pixel on it. Moreover, the beauty of the ROSP algorithm lies on the fact that it does not require any prior knowledge about the number of objects within the image, a burning issue of suffering for all previous image segmentation algorithms.

The experimental results show that the ROSP algorithm outperforms the *suppressed fuzzy c-means* (SFCM), and a newly developed *object based segmentation using fuzzy clustering* algorithm named as OSF algorithms in terms of its complexity and ability to segment the objects.

The most unique success of this algorithm is that it can segment both gray-scale and colour images under the uniform background condition. As such this technique may be used in object-based video transmission that can save huge amount of bandwidth. During initial transmission only the pattern will be sent to the recipient and for onward transmission only the pattern number requiring few bits to be sent to construct the object in video streams. Thus reduces huge communication cost.

The ROSP algorithm, in the larger domain, proposed based on the concepts of SM technique through providing the novel idea of pattern matching. Thus this algorithm emphasizes its ancestor of the taxonomy the concept of SM, making it the most acceptable algorithm for image segmentation. This also gives the region based segmentation technique a dominating boost over the use of parameter human perception, inter and intra-variance, and clustering technique. The analysis of the existing SM algorithm along with its modified versions available in the literature carried out in this thesis along with the proposed ROSP algorithm have formulated a strong base of knowledge on segmentation techniques. The overall thesis has enlightened a direction of the SM algorithm along which the researcher will peep through their wisdom and creativity in future.

Despite of all lucrative success of the ROSP algorithm, it suffers from the selection of appropriate patterns to replace its boundary regions. The quality of objects obtained through splitting and merging are highly depended on the size and shape of the patterns. The smaller the size of patterns more will be complexity of the algorithm, while better quality of the image will be obtained. Thus reducing the pattern size is good enough for solving the distortion at the object contour, but requires more computational time. A compromise is made in this thesis by selecting patterns under the 16x16 pixel blocks. On the other hand, selecting patterns from more number of options will lead to better quality of object contour but increases complexity. The shape of patterns is also a candidate part for getting smooth edges of an object. The smoothness of object's contour can also be obtained through using one of many existing smoothing functions such as splines, B-splines and so on, which is left for future studies.

Bibliography

[1] Aach, T., Franke, U. and Mester, R., "Top-down image segmentation using object detection and contour relaxation", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Glasgow, UK, pp 1703–1706, 1989.

[2] Ali M., and Murshed, M. M., "Lossless compression of correlated images/data with low complexity encoder using distributed source coding Techniques", Lecture Notes in Computer Science, pp 3656, 302–309, 2005.

[3] Ameer Ali, M., "Generic shape-based fuzzy clustering framework for image segmentation", in PhD Thesis Dissertation, Monash University: Australia, 2006.

[4] Ameer Ali, M., Dooley, L. S. and Karmakar, G. C., "Automatic feature set selection for merging image segmentation results using fuzzy clustering", International Conference on Computer and Information Technology (ICCIT), pp 337-342, 2005.

[5] Ameer Ali, M., Dooley, L. S. and Karmakar, G. C., "Object based Image Segmentation Using Fuzzy Clustering", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 67-76, 2006.

[6] Ameer Ali, M., Karmakar, G. C. and Dooley L. S., "Fuzzy image segmentation of generic shaped clusters", IEEE International Conference on Image Processing (ICIP), pp 512-525, 2005.

[7] Andrey, P. and Tarroux, P., "Unsupervised segmentation of Markov random field modeled textured images using selectionist relaxation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-20, pp 252-262, 1998.

[8] Babuska, R., Van der Veen, P. J. and Kaymak, U., "Improved covariance estimation for Gustafson-Kessel clustering", IEEE International Conference on Fuzzy Systems, Vol-2, pp 1081-1085, 2002.

[9] Baker, R. L., "Vector quantization of digital image", PhD. dissertation, Stanford University Stanford, Canada, 1984.

[10] Barnes, C. F., Rizvi S. A. and Nasrabadi, M., "Advances in residual vector quantization", IEEE Transactions on Image Processing, Vol-5, No. 2, pp 226-262, 1996.

[11] Baxes, G. A., "Digital image processing: principles and applications", USA: John Wiley & Sons, Inc., 1994.

[12] Bezdek, J.C., "Pattern recognition with fuzzy objective function algorithm", New York: Plenum Press, 1981.

[13] Bjontegaard, G. and Lillevoid, K., "Context-adaptive VLC coding of coefficients", JVT document JVT-028, JVT of ISO MPEG & ITU VCEG, 3rd Meeting, Rairfax, Virginia, USA, 2002.

[14] Braquelaire, J. P. and Brun, L., "Image segmentation with topological maps and inter-pixel representation", Journal of Visual Communication and Image Representation, Vol- 9, No. 1, pp 62-79, 1998.

[15] Brox, T., Firin, D. and Peter, H. N., "Multistage region merging for image segmentation", 22nd Symposium on Information Theory, NL: Enschede, Benelux, pp 189-196, 2001.

[16] Canny, J. F., "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-8, No. 6, pp 679–698, 1986.

[17] Chakraborty, A. and Duncan, J. S., "Game-theoretic integration for image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligenee, Vol-21, No. 1, pp 12–30, 1999.

[18] Chan, Y. L. and Siu, W. C., "Variable temporal-length 3-D discrete cosine transform coding", IEEE Transaction on Image Processing, Vol-6, No. 5, pp 758-763, 1997.

[19] Chaudhuri, D., Chaudhuri, B. B. and Murthy, C. A., "A new split-and-merge clustering technique", Pattern Recognition Letters, Vol-13, No. 6, pp 399-409, 1998.

[20] Cheevasuvit, F., Maitre, H. and Madjar, D. V., "A robust method for picture segmentation based on a split-and-merge procedure", Computer Vision, Graphics and Image Processing, Vol-34, pp 268-281, 1986.

[21] Chong, A., Gedeon, T. D. and Koczy, L. T., "A hybrid approaches for solving the cluster validity problem", International Conference on Digital Signal Processing, pp 1207-1210, 2002.

[22] Cohen, L. D., "Active contour models and balloons", Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding, Vol-53, No. 2, pp 211–218, 1991.

[23] Dave, R. N., "Boundary detection through fuzzy clustering", IEEE International Conference on Fuzzy Systems, pp 127-134, 1992.

[24] Dave, R. N., "Fuzzy shell-clustering and applications to circle detection in digital images", International Journal of General Systems, Vol-16, No. 4, pp 343-355. 1990.

[25] Derin, H. and Elliot, H., "Modeling and segmentation of noisy and textured images using Gibbs random fields", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-9, No. 1, pp 39-55, 1987.

[26] Duda, R. O. and Hart, P. E., "Pattern classification and scene analysis", USA: John Wiley & Sons, Inc., 1973.

[27] Fan, J. L., Zhen, W. Z. and Xie, W. X., "Suppressed fuzzy c-means clustering algorithm", Pattern Recognition Letters, Vol-24, pp 1607-1612, 2003.

[28] Faruquzzaman, A. B. M., Paiker, N. R., Arafat, J. and Ameer Ali, M. A, "Survey report on image segmentation based on split and merge algorithm", Submitted in the IETECH International Journal on Information Systems, 2008.

[29] Faruquzzaman, A. B. M., Paiker, N. R., Arafat, J., Ali, M. A. and Sorwar, G., "Literature on image segmentations based on split-and-merge techniques", The 5th Internatioal Conference on Information Technology and Applications (ICITA 2008), Australia., 2008.

[30] Faruquzzaman, A. B. M., Paiker, N. R., Arafat, J., Karim, Z. and Ameer Ali, M., "Object segmentation based on split and merge algorithm", IEEE TENCON, IEEE Region 10 Conference, pp 1-4, 2008.

[31] Francis, S. H., Computer Graphics, New Jersey: Prentice Hall, 1994.

[32] Gath, I. and Hoory, D., "Fuzzy clustering of elliptic ring-shaped clusters", Pattern Recognition Letters, Vol-16, No. 7, pp 727-741, 1995.

[33] Geiger D. and Yuille, A., "A common framework for image segmentation", International Journal on Computer Vision, Vol-6, No. 3, pp 227–243, 1991.

[34] Geman, S. and Geman, D., "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-6, pp 721-741, 1984.

[35] Germond, L., "A cooperative framework for segmentation of MRI brain scans", Artificial Intelligence in Medical Science, Vol-20, pp. 77–93, 2000.

[36] Gonzalez, R. C. and Woods, R. E., "Digital image processing", New Jersey: Prentice Hall, Inc., 2002.

[37] Gosh, A., Pal, N. R. and Pal, S. K., "Self-organization for object extraction using multilayer neural networks and fuzziness measure", IEEE Transactions of Fuzzy Systems, Vol-1, No. 1, pp 54-68, 1993.

[38] Gupta, S. P., "Advanced practical statistics", New Delhi, S. Chad and Company Ltd., pp 632-633, 2001.

[39] Gustafson, D. E. and Kessel, W. C., "Fuzzy clustering with a fuzzy covariance matrix", IEEE Conference on Decision and Control, Vol-2, pp 761-766, 1979.

[40] Haris, K., "Hybrid image segmentation using watersheds and fast region merging", IEEE Transactions on Image Processing, Vol-7, No. 12, pp1684–1699, 1998.

[41] Hearn, D. and Baker, M. P., Computer Graphics, New Jersey: Prentice Hall, 1994.

[42] Horowitz, S. L. and Pavlidis, T., "Picture segmentation by a directed split-and-merge procedure", 2nd International Joint Conference on Pattern Recognition, Denmark: Copenhagen, pp 424–433, 1974.

[43] Huang, J. J., Li, S.Y. and Ban, X. J., "A fast approach to building rough data model through G-K fuzzy clustering", Second International Conference on Machine Learning and Cybernetics, pp 1559-1564, 2003.

[44] Hunter, G. M. and Steiglitz, K. S., "Operations on images using quad-tree", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-1, No. 2, pp 145–153, 1979.

[45] Jahne, B., Digital Image Processing, New York: Springer-Verlag, 1997.

[46] Jain, A. K. and Dubes, R. C., Algorithms for Clustering Data, New Jersey: Prentice Hall Inc., 1988.

[47] Jain, A. K., "Advances in mathematical models for image processing", IEEE Proceedings, Vol-69, No. 5, pp 502-528, 1981.

[48] Jain, A. K., Fundamentals of Digital Image Processing, New Jersey: Prentice Hall, Inc., 1989.

[49] Karmakar, G. C., "An integrated fuzzy rule-based image segmentation framework", in PhD Thesis Dissertation, Monash University: Australia, 2002.

[50] Kass, M., Witkin, A. and Terzopoulos, D., "Snake: Active contour models", International Journal on Computer Vision, Vol-1, pp 312–331, 1988.

[51] Krishnapuram, R. and Keller, J. M., "A possibilistic approach to clustering", International Journal of Fuzzy Systems, Vol-2, No. 2, pp 98-110, 1993.

[52] Krishnapuram, R. and Kim, J., "A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms", IEEE Transactions on Fuzzy Systems, Vol-7, No. 4, pp 453-461, 1999.

[53] Levine, M. and Shaheen, S., "A modular computer vision system for picture segmentation and interpretation", IEEE Transaction on Pattern Analysis and Machine Intelligent, Vol-3, No. 5, pp 540-556, 1981.

[54] Man, Y. and Gath, I., "Detection and separation of ring shaped clusters using fuzzy clustering", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol-16, No. 8, pp 855-861 1994.

[55] Montoya, M. D. G., Gil, C. and Garcia, I., "The load unbalancing problem for region growing image segmentation algorithms", Journal of Parallel and Distributed Computing, Vol-63, No. 4, pp. 387-395, 2003.

[56] Murshed, M. and Paul, M., "Pattern identification VLC for pattern-based video coding using co-occurrence matrix", International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications (CSITeA), Egypt, 2004.

[57] Ohlander, R., Price, K. and Reddy, D. R., "Picture segmentation using recursive region splitting method", Computer Graphics and Image Processing, Vol-8, No. 3, pp 313–333, 1978.

[58] Pal, N. R. and Pal, S. K., "A review on image segmentation techniques", Pattern Recognition, Vol-26, No. 9, pp 1277-1294, 1993.

[59] Paul, M. and Murshed, M., "An efficient similarity metric for pattern based VLBR video coding", Journal of Internet Technology, 2005.

[60] Paul, M. and Murshed, M., "Impact of similarity threshold on arbitrary shaped pattern selection very low bit-rate video coding algorithm", International Pacific-Rim Conference on Multimedia (PCM), Vol-3, pp 184-191, 2004.

[61] Paul, M., Murshed, M. and Dooley, L. S., "A new efficient similarity metric and generic computation strategy for pattern-based very low bit rate video coding", IEEE International Conference of Acoustics, Speech, and Signal Procssing (ICASSP), Canada, 2004.

[62] Paul, M., Murshed, M. and Dooley, L. S., "A new real-time pattern selection algorithm for VLBR video coding focusing on moving regions", IEEE International Conference of Acoustics, Speech, and Signal Processing (ICASSP), Hong Kong, Vol-3, pp 397-400, 2003.

[63] Paul, M., Murshed, M. and Dooley, L. S., "A real-time pattern selection algorithm for VLBR video coding using Relevance and Similarity Metrics", IEEE Transaction on Circuits and Systems on Video Technology, 2004.

[64] Pavlidis, T., Structural Pattern Recognition, Springer Verlag, New York: Berlin Heidelberg, 1977.

[65] Prewitt, J. M., Object Enhancement and Extraction, New York: Academics Press Ltd., 1970.

[66] Revol, C., and Jourlin, M., "A new minimum variance region growing algorithm for image segmentation", Pattern Recognition, Letter 18, pp 249–258, 1997.

[67] Ronfard, R., "Region-based strategies for active contour models", International Journal on Computer Vision, Vol-13, No. 2, pp 229–251, 1994.

[68] Sahoo, P. K., "A survey of thresholding techniques", Computer Vision, Graphics Image Processing, Vol- 41, pp 233–260, 1988.

[69] Tilton, J. C., "Image segmentation by iterative parallel region growing with applications to data compression and image analysis", 2nd Symposium on the Frontiers of Massively Parallel Computation, Annapolis, MD, IEEE Computer Society, pp 357, 1988.

[70] Ueda, N., Nakano, R., Ghahramani, Y. Z. and Hiton, G. E., "SMEM algorithm for mixture models", Neural Computation, Vol-12, No. 10, pp 2109–2128, 2000.

[71] Veeman, C. J., Reinders, M. J. T. and Backer, E., "A cellular convolutionary algorithm for image segmentation", IEEE Transaction on Image Processing, Vol-12, No. 3, pp 304-316, 2003.

[72] Yang, M. S., "A survey of fuzzy clustering", Mathematical and Computer Modeling, Vol-18, No. 11, pp 1-16, 1993.

[73] Zhang, Z., Chen, C., Sun, J. and Chan, K. L., "EM algorithms for Gaussian mixtures with split-and-merge operation", Pattern Recognition, Vol-36, pp 1973-1983, 2003.

[74] Zhu, S. C. and Yuille, A., "Region competition: unifying snakes, region growing and Bayes/MDL for multi-band image segmentation", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol-18, No. 9, pp 884–900, 1996.

===============