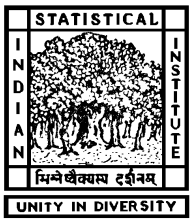
On Edge Detection And Object Recognition In Color Images

Thesis Submitted to
Indian Statistical Institute



2006

by

Sarif Kumar Naik

Machine Intelligence Unit, Indian Statistical Institute
203 B. T. Road, Kolkata, India

On Edge Detection And Object Recognition In Color Images

A thesis submitted to the Indian Statistical Institute in partial fulfillment

of the requirements for the award of degree of

Doctor of Philosophy

2006

by

Sarif Kumar Naik

Machine Intelligence Unit, Indian Statistical Institute

203 B. T. Road, Kolkata

E-mail : sarif_r@isical.ac.in

under the supervision of

Prof. C. A. Murthy

Machine Intelligence Unit, Indian Statistical Institute

203 B. T. Road, Kolkata

E-mail : murthy@isical.ac.in

Indian Statistical Institute

203 B. T. Road, Kolkata, India

To my parents

Acknowledgments

I express my sincere thanks to my supervisor Prof. C. A. Murthy who introduced me to the world of images, vision and patterns. With a deep sense of gratitude I remain obliged for his unconditional support and guidance during the course of my research work. I greatly acknowledge his valuable suggestions during the course of this work. It is my privilege to work under the supervision of a person like Prof. Murthy of profound knowledge and, caring and humble persona.

I express my sincere thanks to Prof. Sankar K. Pal and Prof. Malay K. Kundu for their valuable suggestions, moral support and encouragement time and again during my stay at ISI. I would like to thank my teachers at Sambalpur University, especially to my teacher Prof. S. Pattanaik who inspired me to pursue research.

My sincere thanks are due to Mr. B. Uma Sankar, Dr. D. P. Mandal, Dr. Ashish Ghosh, Dr. R. K. De, Dr. S. N. Biswas, Prof. Sushmita Mitra, Dr. Sanghamitra Bandyopadhyay, Mrs. Minakshi Banerjee, Dr. Swati Choudhuri and other members of the Machine Intelligence Unit for their encouragement and moral support. I would like to thank Prof. B. Chanda for his constructive criticism on several aspects of my research. Special note of thanks to Joydev da, Indra da, Maya di, Niyati di, Sanjay da and Porel da for providing me all kinds of facility whenever needed.

I would like to thank Dr. J. Burianek, Prof. Joseff Kittler and Dr. A. Ahmadyfard for creating the SOIL-47 database and making it available for research purpose. I would like to thank Prof. S. K. Nayar for giving me the approval to use the COIL-100 dataset for research purpose. I would also like to thank Dr. J.-M Geusebroek for making the ALOI

database available for research purpose.

I would never forget the company I had from my fellow research scholars and friends. In particular, I am thankful to Pabitra da, BLN (B. L. Narayan) and Lingaraj, with whom I had several fruitful academic and non academic discussions. I enjoyed my work and had fun with the company of my dear friends Muni, Praveen, Vikal, Vikrant, Prem, Sanjaya and, particularly my seniors Sounaka da, Pradeep (Mishra) da and Pradeep (Giri) da during this period. I must thank all my friends who co-operated with me in all possible ways and stood by me in all times.

Lastly, I thank my parents and elders in my family, who taught me the value of hard work and patience. I would like to share this moment of happiness with all my family members and well wishers.

(Sarif Kumar Naik)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recognizing objects through vision is a common task performed by human beings in day to day life. At the time of performing any kind of task we have to identify different types of objects. For instance we have to identify keys while opening the door, identify faces to talk with different people, identify shoes in the rake etc. While human beings do these tasks accurately and effortlessly, there is no particular way in which a computer would perform the same tasks. Lots of theoretical and practical problems arise while making the recognition process automated. This has been one of the fundamental and challenging problems in the field of computer vision. This thesis addresses the problem of object recognition and the problems associated with it such as edge detection and enhancement for color images.

The object recognition problem can be classified in many ways based on the kind of images it classifies, kind of algorithms it uses and the kind of representations it uses to represent an object and etc. In general, an object identification system seeks the answers to the questions of following types :

1. Identify the objects in an image or a sequence of images.
   Here the task is to differentiate the objects from the background irrespective of the type of the objects. The input to the system is an image and output is either an image containing only the segmented regions of interest or a description of the objects in

the image. This is to some extent an image segmentation problem but generally a first step towards the solution of the more challenging task of identifying and recognizing objects of different shapes.

2. Identify the object X in the image where X is a given shape.
   Here X can be an example image of the desired object. Here, the task is to find out the regions of interest, i.e., the sets of pixels constituting different objects and, then extract characteristic features from these regions and identify the object using the knowledge supplied to the system in the form of X. Identification is the problem of matching two sets of characteristic features based on a dissimilarity measure.

3. Identify the object X in the image where X is a concept.
   Here X is a concept such as a chair, a bicycle, a car, a ball, a balloon, a computer etc. When the query is in terms of a description instead of a direct example in terms of image, we are calling it a concept. Point to be noted here is that in this class, query objects do not have a unique shape. For instance, the term "chair" represents a wide variety of shapes. A chair can have four legs with two arm rests or without two arm rests; instead of four legs it can have wheels; the seat and the back rest may have different shapes and sizes.

In a broader sense the above mentioned problems are called Object Recognition problems. Object recognition is all about modeling the object structure from a set of example data (object images) so as to obtain a stable representation of the objects and then to obtain a method for correspondence of the object structure of an unknown object with the models of the known objects. The object structure should be a good representative of the objects. The features should be distinctive enough to preserve the uniqueness property of the object.

Finding the answer when the query is a concept is more difficult than the case when the query is an example image. In the present work we are not considering the problem of object recognition when the query is a concept. The problem of identifying an object

based on an example query is considered for the study. We pose the object recognition problem as follows :

Let $\Omega = \{M_1, M_2, \cdots, M_n\}$ be a collection of n different classes of objects known to the system and Q is a query image containing the picture of the object of interest. Each of the entities $M_1, M_2, \cdots, M_n$ of the set Ω represents a particular type of object. The problem of object recognition under study is to determine the class of the query image corresponding to the known classes.

Many times, certain preprocessing of the images may be necessary before performing the object recognition task. The processing may include enhancement, noise removal etc. Doing these tasks automatically on a variety of images may not necessarily produce the desired result without damaging some important information in the images. This thesis deals with these processing operations also. It deals with the problem of hue preserving image enhancement without gamut problem in *Chapter 2*. The problem of automatic tuning of threshold values for edge detection in color images is tackled in *Chapter 3*. Two methods for object recognition are provided in *Chapter 4* and *Chapter 5*. One of them utilizes the edge detection method of *Chapter 3* and the other utilizes clustering. Finally, in *Chapter 5*, contrast enhancement principle, developed in *Chapter 2*, is utilized for object recognition. The problem of noise removal is not considered in this thesis. The thesis aims at providing solutions to problems encountered during various stages of an object recognition scheme.

The challenges involved in the object recognition problem are mainly the representation of the objects and matching between two objects through their representations. Various strategies can be adopted for the representation of objects. The two main strategies are selection of features globally and the other one is the selection of features locally. Both the approaches have advantages as well as disadvantages. The models using global object representation are based on the assumption that the appearance of the object as a whole does not change significantly in different views. But, methods using local object representation rely on the fact that appearance of an object may be only locally similar.

Features for representing the objects are either geometric or non-geometric. Geometric features are the features like area, centroid, eccentricity of associate region. Generally, these features represent the geometrical shape of the objects. In the second kind, we consider those features, which use the grey values or the color values of the pixels more directly or in the form of coefficients of certain kinds of transformations. Here, we call these methods as color based methods. In the following section we will review some of the works on Object Recognition appeared in the last decade.

1.1 A Survey of the Related Works

An object recognition task is performed mainly in three steps :

1. Feature extraction : In this step interest points of the objects are located. These can be a collection of regions, a collection of pixels or geometric elements of the objects such as edge, boundary or corner.

2. Object representation : A meaningful representation of the extracted features of the objects. The objective of this step is to represent the object in such a way that the signature of the object contains most of the discriminating features.

3. Object Matching : A dissimilarity measure is employed to check the dissimilarity between two objects through the adopted representation and based on the dissimilarity measure, the object most likely to be the query object is determined.

The existing object recognition methods can be classified into two different categories based on the object representation approach used. Many earlier methods proposed for object recognition mainly use global based object representation. Among the popular color-based global image representation methods, color histograms [36, 49, 105, 132], eigenimages [10, 12, 51, 65, 66, 68, 90, 97–99, 112], color moments [46, 85] are prominent. These methods are popular because of the simplicity in representation and flexibility during matching. However, these methods are not well suited for images containing multiple

objects due to partial and self occlusions. Also, many of these methods can't take care of the changes in backgrounds.

1.1.1 Representation using Global Features

Histogram based Methods

Histogram based approach for image representation is an attractive method for object recognition as well as image retrieval because of its simplicity, speed and robustness [123]. The uses of color histograms for image retrieval are given in several articles [52, 130, 132]. One of the first histogram based representations was proposed by Swain and Ballard [132]. They proposed to represent an object by its color histogram. The advantage of this approach was its robustness to changes due to object orientation, scale, and view points. Stricker [130] introduced indexing technique based on boundary histogram of multi-colored objects.

Although histogram based methods are simple, robust and fast, the drawbacks are their sensitivity to lighting conditions, and the usage of only color information for distinguishing the objects. Sensitivity to lighting condition is a problem in any color based representation. But, the use of only color information limits the discrimination ability of the histogram based methods. There are many objects in the real world, which can't be described in terms of color only. Histogram based representations do not incorporate spatial adjacency of pixels in the image and may lead to inaccuracies in the retrieval [52].

In order to overcome its sensitivity to illumination changes, researchers started developing techniques which are invariant to illumination and color. Healey and Slater proposed representation of an object through color moments of the entire color histogram assuming a constant intensity change over the entire image [46]. They have shown that some moments of the color distribution are invariant to changes in illumination. Derivatives of the logarithms of the color channel are used by Funt and Finlayson [33]. Gevers *et al.* [37] proposed a variable kernel density estimation to construct robust color invariant

histograms for object recognition. The variable kernel density estimation is derived from a theoretical framework for noise propagation through color invariants.

Nature of most of the histogram based methods is global. It is known that global color distribution may change with change in view angle, illuminations, occlusion etc [41]. Ennesser *et al.* [30] proposed a local color histogram method in this regard. Das *et al.* [25] also used histogram based feature representation. Peaks of the histogram in HSV color space are used for object representation here. A more efficient representation of color histogram was developed by Hafner *et al.* [43]. Gevers *et al.* [35] proposed a method for content-based image retrieval where, features are selected by combining both color and shape features. Various similarity functions including cross correlation were compared for color-based histogram matching. They concluded that retrieval accuracy of similarity functions depends on the presence of object clutter in the scene. The histogram cross correlation provides a good retrieval accuracy without any object clutter. Huang *et al.* [49] defined an image feature called the color correlogram and used it for image indexing and comparison. A color correlogram expresses how the spatial correlation of color changes with distance. It describes the global distribution of local spatial correlations of colors. It is a table indexed by color pairs, where the kth entry for (i,j) specifies the probability of finding a pixel of color j at a distance of k from a pixel of color i. This method improves the quality of representation of color histogram by incorporating spatial color information and carrying forward the advantages of color histogram based representation.

Eigenspace based Methods

Eigen spaces are also used for object representation. The standard procedure in an eigenspace based method is to represent the object by considering whole image as a vector and projecting it over a set of eigenvectors to achieve data compression as well as reduction of redundant information. In Principal Component Analysis (PCA), the eigen vectors corresponding to dominant eigen values are found for the dispersion matrix.

Some of the earliest works on object recognition using eigenspace based represen-

tation are by Murase *et al.* [90], Nayar *et al.* [97–99], and Turk and Pentland [141]. Leonardis *et al.* [65] developed a technique for eigenspace based representation of objects, which is capable of tackling the problem of occlusion. This is achieved by employing sub sampling, instead of computing the coefficients of eigenimages, by projecting the data onto the eigen images. Leonardis *et al.* [66] proposed a self-organizing framework to construct multiple low-dimensional eigenspaces from a set of training images. Bischof *et al.* [10] proposed an eigenspace based method for recognition. They incorporated a gradient based filter bank into the eigenspace recognition framework. They showed that the eigenimage coefficients are invariant to linear filtering. A robust procedure for coefficient recovery based on voting is proposed to achieve further illumination insensitivity.

Borotsching *et al.* [12] proposed an appearance-based object representation, namely the parametric eigenspace, and augments it by probability distributions. This helps the method to cope with possible variations in the input images due to changing imaging conditions. Lin *et al.* [68] presented a color image normalization method, called eigen color normalization for object recognition. It is completed in two steps, first, a compact representation of the image is obtained using the affine transform matrix computed from the image data and then the compact color image is further normalized by rotating the histogram to align with the computed reference axis. Other eigenspace representations used in object recognitions are the works of Retier *et al.*in [112], Huttenlocher *et al.* [51].

The methods using eigenspace representation are generally effective when the eigenspace captures the characteristics of the whole database. For example, when all the object images have uniform known background. If there is a large variation in the images, performance of the methods may deteriorate. Some success has been achieved by Leonardis *et al.* [65] and Bischof *et al.* [10] in this regard.

Graph Based Representation

In graph based representation, generally, regions with their corresponding feature vectors and the geometric relationship between these regions are encoded in the form of a graph.

Tu *et al.* [140] proposed a method which segments the image into regions of approximately constant color and the geometrical relationship of the segmented colored regions is represented by an attributed graph. Object matching, then, is formulated as an approximate graph-matching problem.

Matas *et al.* [83] proposed a representation for objects with multiple colors - the Color Adjacency Graph (CAG). Each node of the CAG represents a single chromatic component of the image defined as a set of pixels forming a unimodal cluster in the chromatic histogram. Edges in a CAG contain adjacency information of the color components and their reflectance ratios. The CAG is related to both the histogram and region adjacency graph representations. Nodes of CAG correspond to modes of the histogram. Attributed Relational Graph (ARG) based representation is used by Ahmadyfard *et al.* [3] to represent each model and the scene. In this method, each image region is first transformed to an affine invariant space. Then a multiple region representation is provided at each node of the ARG of the scene to increase the representation reliability. The matching between a scene ARG and a model graph is accomplished using probabilistic relaxation.

A shock graph representation of objects is proposed by Siddiqi *et al.* [128]. Macrini *et al.* [74] proposed a method of recognizing objects using shock graph representation of the object, which is invariant to within class shape deformations. Other shock object recognition techniques using shock graph representation are developed by Pelilo *et al.*in [106] and Sebastian *et al.*in [125].

Hybrid graph representations are proposed by Park *et al.* [104]. Lades *et al.* [63] presented an object recognition system based on Dynamic Link Architecture. Objects are represented by sparse graphs whose vertices are labeled by a multi-resolution description in terms of a local power spectrum and edges are labeled by geometrical distance vectors. Object recognition is performed by formulating it as an elastic graph matching problem. Kostin *et al.* [60] proposed an object recognition scheme using graph matching.

One advantage in graph based representation is that the geometric relationship can be used to encode certain shape information of the object and any sub-graph matching

algorithm can be used to identify single as well as multiple objects in query images. However, matching two such representations becomes a complicated process. Some of the issues in this regard are discussed in [60].

Other Object Representation Methods using Global Features

The initial methods for 3D object recognition from 2D object images used geometric features such as area, centroid and eccentricity of the associate regions, lines, edges, corners etc. These features are obtained as a result of the processes such as edge detection, corner detection and image segmentation. Many times features are obtained combining the results of multiple such processes. One of the main reasons behind using these processes for obtaining features is the availability of a number of effective and useful methods in the literature for finding them. However, the inherent problem in this approach is finding a meaningful relation between the obtained features to have a global object model. Another problem in using geometric features is the accurate/exact extraction of such features of the same object but in different images. Most of the methods can find such features effectively for a particular image by tuning the values of parameter set used for the methods. However, obtaining such features over a number of images consistently by the same set of parameter values has been a difficult task.

Li *et al.* [67] proposed an image retrieval system namely C-BIRD (content-based image retrieval in digital-libraries). Each image is represented using a feature descriptor and a layout descriptor. The feature vector consisting of (1) A color vector in the form of a 512-bin histogram, (2) centroids of the regions associated with the 5 most frequently occurring colors, (3) centroids of regions of the 5 most frequent edge orientations and (4) a 36-dimensional chromaticity vector. Along with this information, certain geometric information such as the area, centroid and eccentricity of the associate regions are also used. The layout descriptor is built using a color layout vector and an edge layout vector.

Mindru *et al.* [85] introduced a set of “Generalized Color Moments” to exploit the multi-spectral nature of the color images. These features are based on the moments of

powers of the intensities of the different color channels and their combinations. These features implicitly characterize the shape, intensity and the color distribution of the pattern in the images. Mel *et al.* [84] proposed a view-based high-dimensional feature-space recognition method namely SEEMORE. Objects are represented using color, texture and shape features. Kankanhali *et al.* [57] proposed a method of image representation using clustering in the RGB color space. Images are represented using the cluster centers and the fraction of the pixels in the cluster compared with the total number of pixels.

Learning free algorithms such as nearest neighbor classifier provide good recognition, but this kind of algorithms often suffer from generalization abilities in real-world conditions (Chapelle *et al.* [21] and Pontil *et al.* [108], Wallraven [143]). To overcome these problems Support Vector Machine classifiers have been proposed in the literature (Wallraven *et al.* [143], Pontil *et al.* [108], Roobaert *et al.* [115]). The class of SVM algorithms are based on a thorough mathematical foundation and have shown impressive learning and recognition performance over learning free algorithms such as nearest neighbor algorithm but, on the other hand, these are more computationally expensive than other matching algorithms. Another disadvantage of this class of algorithms is a proper selection of the kernel. There are a number of methods using Support Vector Machine(SVM) classifiers for three dimensional object recognition in the literature. These methods are used to classify both globally and locally obtained feature vectors of the objects. Pontil *et al.* [108] used SVM to recognize objects in a subset of COIL-100 dataset. Roobaert *et al.* [115] performed a number of experiments using SVM with three different representations namely, “Color only”, “Shape only” and “Shape&Color”, of the objects from COIL-100. Color cue is the average color value of the reduced image. For shape cue, it uses the reduced grey image of the original image. Roth *et al.* [116] proposed a view based algorithm for 3D object recognition using a network of linear units. Sparse Network of Winnows(SNoW) learning architecture is used to learn the representations of objects. Two experiments are carried out by them using pixel-based representation and edge-based representation of the objects separately.

Above discussed methods used global features for representation. Many of these methods achieved good recognition. However, although, global featurs yield good characterizations of isolated, segmented objects, they could be inappropriate for the wider spectrum of heterogeneous natural scenes [48]. Hence, image representation using local features became popular. In the following section, several object recognition methods which use local features for object recognition are discussed.

1.1.2 Representation using Local Features

To overcome the problems involved in global representation such as change in view angle, occlusion and problems due to storage requirement of storing high dimensional feature vectors, vision scientists started developing representation schemes which detect salient features of an object from the different regions of its image. Here, generally, salient features are extracted to represent the region or neighborhood surrounding a point. There are two steps in this approach. First, locate every such point and determine a region around each point of interest. A set of features based on the intensity values of the pixels are generally selected to represent the region. Over the years different schemes have been developed using local feature representation.

Lowe *et al.* [70] proposed an object representation namely "Scale Invariant Feature Transform (SIFT)", which is seen to be invariant to different types of image transformations. Keypoints are extracted in four stages in this scheme. The first stage of computation searches over all scales and all image locations. In the second stage, a detailed model is fit to determine key locations and scales. Keypoints are selected based on a measure of similarity. In stage three, dominant orientations for each keypoints are identified based on local image gradient direction. In stage four, local image gradients are measured at the selected scale in the neighborhood of the keypoints. Based on these, a 128 dimensional local image descriptor is constructed for each keypoint. Key *et al.* [58] proposed a modification, PCA-SIFT, of the Lowe's SIFT descriptor. The idea behind the modification is to reduce dimension of the feature vector to reduce redundant information and make it

more representative of the neighborhood. Principal Component Analysis is used in the stage four of the Lowe's algorithm to achieve dimensionality reduction. A family of new features is proposed by Brown *et al.* [14], which uses groups of interest points to form geometrically invariant descriptors of image regions. Interest points are located at the extrema of the Laplacian of the image in scale space. Feature descriptors are formed by resampling the image relative to canonical frames defined by the interest points.

Baumberg [7] used a Harris like feature detector [45] to find a fixed number of interest points for each image. He found the corner strength for each point in the image using the determinant and trace of the second moment matrix. The top 'n' points having maximum corner strength are selected as interest points. These image patches are normalized using the square root of the covariance matrix to achieve affine invariance. Finally, an image descriptor is obtained using a variant of the Fourier-Melin transformation on each of the image patches.

Matas *et al.* [80] [81] have proposed a method to find Distinguished Regions called "Maximally Stable Extremal Regions (MSERs)". Features are extracted from these MSERs to describe the objects. An object recognition method using Local Affine Frames(LAF) has been proposed by Obdrzalek *et al.* [103], which detects the distinguished regions of data-dependent shapes and establishes local affine frames using affine invariant constructions on these regions. Features obtained from these regions are used for comparison. Tuytelaars *et al.* [142] proposed two ways to find invariant regions to changing view points of object images. The first one starts from corners and uses the nearby edges and the second one is based on the intensity of the pixels around the point of interest. Schmid *et al.* [124] proposed a method of feature representation using color moments. The points of interest are detected using a Harris like interest point detector and local differential invariants are computed in a multi-scale fashion. Leibe *et al.* [64] carried out an experiment of object categorization to analyze the performance of different methods on a dataset. They have found that no single method is superior for all categories of objects.

Matas *et al.* [82] proposed Multi-modal Neighborhood Signature(MNS) for object

recognition and image retrieval. Color features are extracted from the different regions of the image having multi-modal color distribution. All the distinct pairs of modes are taken to form the MNS of the object. Performance of MNS is evaluated [61] using SOIL-47A dataset. Kadir *et al.* [56] proposed a method to detect salient regions based on the unpredictability in their local attributes and over all spatial scale. Shannon entropy is used to check the unpredictability of the local attributes. This method is invariant to the similarity group of geometric transformations and to photometric shifts but not affine invariant to geometric transformation. A generalization of this method to incorporate affine invariance to geometric transformations is proposed by Kadir *et al.*in [55]. Lindeberg *et al.* [69] developed a scale invariant interest point detector. It searches for 3D maxima of scale normalized differential operators.

Basri *et al.* [6] presented a method for recognition that uses region information. In their approach the model and the image are divided into regions. Given a match between subsets of regions (without any explicit correspondence between different pieces of the regions) the alignment transformation is computed. The method applies to planar objects under similarity, affine, and projective transformations and to projections of 3-D objects undergoing affine and projective transformations.

High-dimensional global invariants are employed by Califano *et al.* [17] to implement a 2D shape recognition system. It is based on a two step table look-up mechanism. In the first stage, local curve descriptors are obtained by correlating image contour information at short range and then, seven-dimensional global invariants are computed by correlating triplets of local curve descriptors at longer range. Rothganger *et al.* [117] proposed a representation for 3D objects in terms of local affine-invariant descriptors of their images and the spatial representation of the corresponding affine regions. The method by Hall *et al.* [44] is based on a sampling of a local appearance function at discrete viewpoints by projecting it onto a vector of receptive fields which have been normalized to local scale and orientation.

Agarwal *et al.* [2] developed a sparse, part-based representation of the objects. A

vocabulary of distinctive object parts is constructed from a set of sample images of the object class of interest automatically. Images are then represented using the parts from this vocabulary together with the spatial relations observed among the parts. A feature-efficient machine learning algorithm is employed to automatically learn to detect instances of the object class in new images. A method for selecting discriminative scale-invariant object parts is proposed by Dorko *et al.* [28]. At first, the scale invariant interest points and then the rotation-invariant descriptors for each region are detected. Clustering is performed to obtain a set of parts.

Recently, Maree *et al.* [76] have proposed a generic approach to image classification based on decision tree ensembles and local sub-windows. Their methods directly operate on pixel values and do not require any task specific feature extraction. Maree *et al.* later extended this method [77] by introducing randomness in the process of selection of the sub-windows and the features are represented using HSV color space instead of RGB color space.

1.1.3 Methods integrating Color and Shape Features

Nature of most of the algorithms discussed above is that they either use shape features or color features. There are some algorithms which use both color and shape featues [5, 29, 52, 54, 91, 115, 129, 148], which are described below.

Slater *et al.* [129] proposed a method to combine geometric and color features extracted from local regions of the images. Nagao [91] used the centroid of the 2D image geometric features of the object and a vector formed from the ratios B/R and G/R of the image channels. Finally, the object was represented by the concatenation of these two vectors. A kernel based method which combines color and shape information for appearance based object recognition is proposed by Caputo *et al.* [19]. Individual color and shape cues of objects are combined using kernels in a spin glass and Markov random field framework for object representation.

Dubuisson *et al.* [29] proposed a method for object matching using distribution of

color and the edge map of the image. Matching score was obtained by a linear combination of the scores obtained by comparing color and edge features individually. Image retrieval method by Jain *et al.* [52] used the normalized histogram of edge directions to represent shape attribute. Three individual one dimensional histograms from three color bands are considered as the color features. An integrated simlarity measure that is a normalized weighted average of the scores obtained by individual comparison of shape only and color only feature vectors is employed for matching.

The work by Zhong *et al.* [148] used color, shape and texture information for object localization. Color and texture features are extracted from the coefficients of the Descrete Cosine Transform(DCT) of the image blocks. Their method operates in two stages, In the first stage, color and texture features are used to find out the candidate images from the database, and identify regions in the candidate images whose color and texture feature match with the query. In the second stage, a deformable template matching method proposed in [53] is used to match the query shape to the edges at the locations detected in the first stage. Shape and color features have also been used for automatic fruit detection [54].

Although, these methods use both color and geometric features, the matching scores are found individually and they are combined. In most of the above algorithms color and shape features are extracted separately and later they are combined together at the time of matching. Thus, here, a method of image representation is proposed, which inherently keeps the geometrical shape of the regions of interest in an object image as well as the color information of that region simultaneously. It has been called as Multi-Colored Region Descriptor (M-CORD).

1.2 Proposed Approach of Object Recognition

The proposed Multi-Colored Region Descriptor (M-CORD) is a model-based object recognition method. Object modeling is done using the local color structure appearing on the object sureface. Two different methods have been proposed to determine the region of

interest As mentioned in the above section, M-CORD descriptor inherently keeps shape and color information of regions of interst of the objects.

In order to determine the regions of interest two different methods have been adopted. These two methods are : (1) clustering in RGB color space of the 3-dimentional color vectors of pixels within the region of interst (method iscalled M-CORD-Cluster) (2) finding the colors from the different smaller regions partitioned by the edge map of the region of interest (method is called M-CORD-Edge). To obtain the multi-colored regions, a simple and fast clustering algorithm has been proposed. It can eliminate the regions possessing uniform color quickly and determines the regions of interest i.e., the regions possessing multiple colors. In the second method, edge maps of the regions are employed to determine the nature of the regions. For this purpose a new edge detection method has been proposed, which is capable of finding uniformly acceptable edge maps from all the images without tuning the parameters used in the algorithm individually for each of the image.

In many databases, quality of the image contrast may not be adequate. Contrast of the images of such databases is needed to be enhanced before starting the recognition process, to obtain good representative features of the objects. Thus a hue preserving color image enhancement technique is proposed, which is used to enhance the images before finding the regions of interest. However, the enhancement process is not mandatory. Image enhancement is to be performed when the images are of poor contast.

This thesis consists of six chapters. A chapter-wise abstract of the thesis is provided in the following section.

1.3 Outline of the Thesis

This thesis consists of four contributed works and distributed in four different chapters apart from the Introduction (*Chapter 1*) and the Conclusions, Discussion and Scope for Further Works (*Chapter 6*) chapters. *Chapter 2* addresses the problem of image enhancement which is later used in *Chapter 4* and *Chapter 5* for object representation. *Chapter*

3 proposes an edge detection method which standardizes the edge magnitudes to obtain uniformly acceptable edge maps from all the images. This methodology has been used to obtain salient regions from color images, when the selected features are used for object representation. *Chapter 4* describes a method of object representation using local color structure of the image and in *Chapter 5* the performance of the proposed representation technique has been tested on different object image databases. This thesis is concluded in *Chapter 6*. An appendix is provided after *Chapter 6* containing additional results of the methods of *Chapter 3*. All the images of appendix are given only as a soft copy in the attached CD to the thesis. The attached CD also contains (i) the whole thesis and (ii) the synopsis of the thesis. A brief descriprion of the thesis from *Chapter 2* to *Chapter 6* is given below for a quick appraisal.

1.3.1 Chapter 2 : Hue-Preserving Color Image Enhancement Without Gamut Problem [92]

Color plays a critical and crucial role in color image enhancement which is a combination of both chrominance and luminance information. Chrominance information is the information regarding the hue and saturation of the color, and luminance is the perceived intensity. From the image enhancement perspective chrominance information in the color needs careful attention. Mainly, undesirable shift in hue value may deteriorate the quality of the image drastically. Most of the image data available are in RGB color space. Thus a color image is available in three different channels (R, G and B) of information which can be viewed and seen as three gray scale images individually. This suggests that the direct application of usual gray level image enhancement techniques to individual channels independent of others and reunion of the enhanced channels would give enhancement. Unfortunately, reunion would not give a satisfactory result and sometimes it may become worse, because, it is seen that the R, G and B channels are highly correlated with respect to the chrominance and luminance information in the color. Thus individual processing of

the channels may shift the hue and saturation of a pixel considerably for some pixels. This generates visual artifacts. To avoid this problem, most of the methods first transfer the image data to a color space which de-correlates the chrominance and luminance information of the color. Then leaving one or both the chrominance components intact, luminance is modified to achieve good contrast. There are two notable problems in this approach. (1) Transforming from RGB space to other spaces may need large number of computations. These transformations are many times prone to noise. (2) After the enhancement, when the data is again transformed back to RGB space, many values go beyond the range of the RGB space. The second problem is commonly known as gamut problem. Thus either it is rescaled or truncated to the bounds of the RGB space. Rescaling decreases the achieved contrast and truncation changes the hue component of the affected pixels.

This chapter addresses these problems and suggests a principle to overcome these problems so that the existing knowledge of contrast enhancement in gray scale images can be used to color images. Using the suggested principle, the well known image enhancement techniques such as S-type enhancement and histogram equalization, are generalized to achieve hue-preserving color image enhancement. The proposed method is also seen to be gamut problem free. The proposed method is compared with two different hue-preserving color image contrast enhancement techniques and superiority of the proposed method over these methods is shown.

1.3.2 Chapter 3 : Standardization of Edge Magnitudes in Color Images [95]

Edge detection is a useful task in low-level image processing. The efficiency of many image processing and computer vision tasks depends on the perfection of detecting meaningful edges. To get a meaningful edge, thresholding is almost inevitable in any edge detection algorithm. Many algorithms reported in the literature adopt ad hoc schemes for this purpose. These algorithms require the threshold values to be supplied and tuned

by the user. There are many high-level tasks in computer vision which are to be performed without human intervention. Thus there is a need to develop a scheme where a single set of threshold values would give acceptable results for many color images. In the present work, an attempt has been made to device such an algorithm. Statistical variability of partial derivatives at each pixel is used to obtain standardized edge magnitude and is thresholded using two threshold values. The advantage of standardization is evident from the results obtained. The principle of edge detection proposed in this chapter is used in the subsequent chapters.

1.3.3 Chapter 4 : Multi-Colored Region Descriptor [94, 96]

There are two approaches to object recognition. One uses only shape features and the other uses only color features. Some methods are also available which combines these two types of features. There are limitations to any algorithm which uses only one type of features. There are many objects which are indistinguishable in terms of their shape and there are many which can't be distinguished just from the colors. Still, to some extent, these types of objects can be distinguished from the patterns on them. Thus, there is a need of a scheme to describe an object which contains both shape and color information. In other words, the representation scheme should carry the color information and its pattern of appearance on the object surface. This study proposes a scheme to describe an object in such a way that the description contains the color information as well as the patterns of colors on the object surface. Note that, in most of the cases, wherever there is a shape or structural information in the object, the corresponding patterns in the image possess discontinuities in colors. Thus extraction of information regarding patterns of colors automatically lead to extracting shape and structural information of the object. These obtained features with the proposed representation carry the pattern information that indirectly keeps the shape information regarding the object.

Here, the shape and structural information of an object is extracted from the appearance of colors in different local regions of the image which we call "Multi-Colored Neighborhood (MCN)". The relevant cues from these MCNs are clubbed together to form the

descriptor of the object. We call this descriptor as "Multi-Colored Region Descriptors (M-CORD)". Two different methods have been proposed here to identify the MCNs in an object image. In the first method, regions with multiple colors are detected using a simple and fast clustering technique and each region is represented using the mean value of the different clusters. This descriptor using clustering is called M-CORD-Cluster. In the second method, edge map of the color images is used to locate the regions with multiple colors. Those regions which are divided into multiple segments by the edge maps are selected. The edge detection scheme described in *Chapter 3* is used to find the edge maps of the object images to find the MCNs and hence the M-CORD. This descriptor using edge map of the object image is called M-CORD-Edge. Either using M-CORD-Cluster or M-CORD-Edge, several MCNs are detected on the object surface. However, the information from all these MCNs is not required. Thus, only those MCNs which are distinctly different from others are selected using a simple elimination technique. Two MCNs are said to be distinct if the Hausdorff distance between them is greater than a threshold value. Performance of the M-CORD in the context of object recognition is evaluated in the next chapter.

1.3.4 Chapter 5 : Object Recognition using Multi-Colored Region Descriptors [94, 96]

In this chapter, the performance of the object representation scheme (M-CORD) is evaluated in the context of object recognition. M-CORD of each of the object images in the dataset is found as described in the previous chapter. It is then divided into two sets namely model set and test set. Each M-CORD in the test set is compared to each of the M-CORD in the model set and the rank of the correct matches are noted down. Percentage of recognition rates for ranks one, less than or equal to two, and less than or equal to three are used to evaluate the performance of the M-CORD.

Comparison between two object images through their M-CORDs are performed using

two proposed dissimilarity measures

(1) The first dissimilarity measure compares two MCNs, one each from two different M-CORDs.

(2) The second dissimilarity measure compares two M-CORDs of two different object images.

The methods have been implemented on COIL-100, SOIL-47 and ALOI-VIEW object datasets. The performance of the methods is evaluated using different number of training views per object. Performance of the proposed methodology is significantly better compared to other existing methods when one, two and four training views per objects are considered and better results are obtained when more than four views per objects are considered.

In many datasets, due to poor contrast, the difference between object pixels and background pixels may not be prominent. This leads to poor output of object recognition methodology. The hue-preserving color image enhancement procedure described in *Chapter 2* is used to enhance the images in the database using a contrast enhancement function. Application of the proposed enhancement principle along with the M-CORD-Edge descriptor and the object recognition scheme provides better results. This has been tested on 100 objects of ALOI-VIEW database. However, it is not always necessary to use image enhancement before the construction of object descriptor. This should be used when the images in the dataset possess poor contrast.

1.4 Conclusions, Discussion and Scope for Further Works

The last chapter (*Chapter 6*) of the thesis deals with the conclusions and discussion.

1.5 Appendix

This contains results of the methods stated in *Chapter 3*, when applied to several images.

Chapter 2

Hue-Preserving Color Image Enhancement Without Gamut Problem

Image enhancement is a first step in many image processing tasks such as edge detection, image segmentation and other high level tasks in computer vision such as object recognition, object extraction etc. The objective of image enhancement is to improve the quality of an image for further processing. In many cases, image enhancement is used to improve the quality of an image for visual perception of human beings. However, enhancement of visual quality of the image is not the sole purpose of image enhancement. Feature extraction from images is a common task in computer vision and image processing. In many cases, due to poor quality of the images, feature detectors fail to perform upto their potential and hence, compromising the quality of the final result. Image enhancement is essentially required for such images before performing feature detection. In general, image enhancement is a task in which the set of pixel values of one image is transformed to a new set of pixel values so that the new image formed is either visually pleasing or is more suitable for analysis. It is a widely studied topic of image processing for grayscale images. The main techniques for image enhancement such as contrast stretching, slicing, histogram equalization for grayscale images are discussed in many books. The problem of image enhancement in color images is a difficult task compared to grayscale images be-

cause of several factors. The generalization of grayscale image enhancement techniques to color images is not a trivial work [127]. Unlike grayscale images, there are some factors in color images like hue, which need to be properly taken care of for enhancement. These are going to be discussed below.

Hue, saturation and intensity are the attributes of a color [9]. Hue is that attribute of a color which decides what kind of color it is, *i.e.*, a red or an orange. It is the quality of color, which may be characterized by its position in the whole visible spectrum through red, yellow, green, cyan, blue and magenta. In the spectrum each color is at the maximum purity (or strength or richness) that the eye can appreciate, and the spectrum of colors is described as fully saturated. If a saturated color is diluted by being mixed with other colors or with white light, its richness or saturation is decreased [20]. It is that attribute of a color, which describes the degree to which a pure color is diluted with white or grey. For the purpose of enhancing a color image, it is to be seen that hue should not change for any pixel. If hue is changed then the color gets changed, thereby distorting the image. Modification of hue may lead to results that are unpleasant to human observer, since it is known that human visual system is extremely sensitive to shifts in hue [16]. One needs to improve the visual quality of an image without distorting it for image enhancement. As it is already said, visual enhancement is not the only purpose of enhancement and it may be used for other purposes. The method of image enhancement described in this chapter will be used for feature extraction for object recognition in *Chapter 4*. A hue preserving color image enhancement scheme is proposed in this chapter which can be used to generalize many existing grayscale image enhancement techniques.

For many images, increasing the contrast would result in an improvement in the visual quality of the image. Several algorithms are available for contrast enhancement in grayscale images, which change the gray values of pixels depending on the criteria for enhancement. On the other hand, literature on the enhancement of color images is not as rich as grayscale image enhancement. Many authors transformed the original RGB images to other color spaces for the purpose of enhancement. Usually such transformations

are computationally expensive since additional calculations are needed to get the hue, saturation and intensity values of pixels. Some of the existing algorithms are described below in this regard for color image enhancement.

2.1 A Survey of Color Image Enhancement

Image enhancement is one of the fundamental image analysis tasks. Hence, it has drawn the attention of several researchers in the field of image processing. Over the years several methods have been proposed for image enhancement for grayscale images such as S-type enhancement and Histogram Equalization. However, a limited number of color image enhancement techniques are available in the literature. The task of image enhancement can be any such process by which an image is made to be suitable for further analysis for certain pre-determined purpose. We shall not discuss tasks like noise removal here. We shall be discussing contrast enhancement here. A brief literature survey on color image contrast enhancement techniques is presented here.

The color equalization method proposed by Bockstein [11] is a method based on both saturation and brightness of the image. The color triangle (which cuts equal segments at axes R, G and B) is divided into 96 disjoint hue regions. A computationally efficient method is given to divide the color triangle into different hue regions and to compute the maximum realizable saturation for each of these regions. Saturation is separately equalized once for each region within the bounds 0 and the maximum realizable saturation of that region, but brightness is equalized once for the whole space. After the equalization some of the R, G and B values exceed allowable bounds. The author suggested the use of normalization coefficients to reduce R, G and B equally should any of them go out of bounds.

Strickland *et al.* [131] proposed an enhancement scheme based on the fact that objects can exhibit variation in color and saturation with little or no corresponding luminance variation. In their scheme edge information from the saturation data is combined

with edge information from luminance data to construct a new luminance component. Then the ratio of new luminance data L' with the original luminance data L is multiplied with the R, G and B individually to get the enhanced R', G' and B' values respectively. Thomas *et al.* [134] proposed an improvement over this method by considering the correlation between the luminance and saturation components of the image locally. Toet [135] extended Strickland's method [131] to incorporate all spherical frequency components by representing the original luminance and saturation components of a color image at multiple spatial scales.

Four different methods of enhancement for highly correlated images have been proposed by Gillespie *et al.*in two parts. In Part I [38], a method named "decorrelation stretching" is suggested in which the image data is stretched along its principal axes. Method two in the same article suggests the individual stretching of the components in HSI color space. In Part II [39] : two methods are discussed based on ratioing of data from different image channels. These methods are mainly applicable to satellite images. These transformations are not hue preserving since the stretching is not same in each component.

A 3-D histogram specification algorithm in RGB cube with the output histogram being uniform is proposed by Trahanias *et al.* [137]. This method computes the 3-D cumulative distribution function (cdf) $C(R_x, G_x, B_x)$ of the original image and a 3-D uniform cdf $\overline{C}(R_y, G_y, B_y)$. For a triple (R_x, G_x, B_x) the smallest (R_y, G_y, B_y) is determined for which $\overline{C}(R_y, G_y, B_y) - C(R_x, G_x, B_x) > 0$. Since this condition does not provide unique solution, a sequentially incrementing algorithm to determine the smallest possible (R_y, G_y, B_y) is proposed. This transformation is not hue preserving.

Yang *et al.* [145] proposed two hue preserving techniques, namely, scaling and shifting, for the processing of luminance and saturation components. To implement these techniques one does not need to do the color coordinate transformation. Later, the same authors have developed clipping techniques [146] in LHS and YIQ spaces for enhancement to take care of the values falling outside the range of RGB space. Clipping is per-

formed after the enhancement procedure is over. A high resolution histogram equalization of color images is proposed in [114]. The effect of quantization error for the luminance component in histogram equalization is also studied.

Mlsna *et al.* [86] proposed a multivariate enhancement technique "histogram explosion", where the equalization is performed on a 3-D histogram. This algorithm finds the centroid of the histogram and considers it as an operating point. For each triplet a ray starts from the operating point and passes through the triplet. A histogram is built for each such ray. First order interpolation is used to decide which triplets are falling on the ray while building the histogram. Explosion is determined by equalizing the ray histogram. The objective is the development of a method for greatest possible contrast enhancement procedure rather than preserving perceptual attributes. Another version of this paper is proposed in CIE LUV space [87]. The same authors later proposed a recursive algorithm for 3-D histogram enhancement scheme for color images [147].

Weeks *et al.* [144] proposed a hue preserving color image enhancement technique which modifies the saturation and intensity components in color difference (C-Y) color space. Their algorithm partitions the whole (C-Y) color space into n $\times$ k number of subspaces, where n and k are the number of partitions in luminance and saturation components respectively. The maximum realizable saturation in each subspace is computed and stored in a table. Saturation is equalized once for each of these n $\times$ k subspaces within the maximum realizable saturation of the subspace. Later the luminance component is equalized considering the whole image at a time. To take care of the R, G and B values exceeding the bounds, Weeks *et al.*suggested to normalize each component using $\frac{255}{max(R,G,B)}$.

Pitas *et al.* [107] proposed a method to jointly equalize the intensity and saturation components. It has been reported that histogram modification of intensity enhancement gives the best result. However, it is also reported that the modification of the saturation or joint modification of the saturation and intensity, though mathematically correct, usually lead to large saturation values which are not present in the natural scenes.

Gupta *et al.* [42] proposed a hue preserving contrast stretching scheme for a class of color images. Their method assumes that the images are taken from a microscope which looks at a three dimensional micro structure with a beam of reflected light. They studied the dichromatic reflection model and have used it in contrast enhancement. They dealt with the chromatic and achromatic pixels separately.

A genetic algorithm (GA) approach in which the enhancement problem is formulated as an optimization problem is suggested by Shyu *et al.* [127]. A set of generalized transforms for color image enhancement is formed taking linear combination of four types of non-linear transforms. The fitness function for GA is taken to be a combination of four performance measures. GA is used to determine an optimal set of generalized transforms. Normalization method is suggested to accommodate all the enhanced image data into the RGB space.

Buzulois *et al.* [16] proposed an adaptive neighborhood histogram equalization algorithm. This algorithm modifies only the intensity of the image. Intensity component is equalized both globally and locally in a neighborhood of each pixel. A neighborhood of each pixel is found using a region growing method. On the basis of the variance and mean of the intensity of these neighborhood pixels, local histogram equalization is performed. Result of this local histogram equalization is combined with the global histogram equalization to produce the final result. In this process, only the intensity value of the seed pixel is modified. This procedure is repeated for all the pixels in the image. This method is hue preserving but suffers from the gamut problem.

Lucchese *et al.*'s [71] method for color contrast enhancement works in xy-chromaticity diagram, which consists of two steps : (1) transformation of each color pixel into its maximally saturated version with respect to a certain color gamut and (2) desaturation of this new color towards a new white point.

Though the algorithms reported above are interesting and effective for enhancement, most of them do not effectively take care of the gamut problem – the case where the pixel values go out of bounds after processing. Due to the non-linear nature of the uniform color

spaces, conversion from these spaces with modified intensity and saturation values to RGB space generates gamut problem. In general this problem is tackled either by clipping the out of boundary values to the bounds or by normalization [11, 127, 144]. Clipping the values to the bounds creates undesired shift of hue. Strickland has discussed the problem of clipping in [131]. Normalization reduces some of the achieved intensity [127] in the process of enhancement which is against its objective.

In this chapter we have suggested a novel and effective way of tackling the gamut problem during the processing itself. Hue preserving color image enhancement is analyzed theoretically in this chapter. It does not require the property of bringing back the R, G and B values to its bounds after the processing is over. Proposed algorithm does not reduce the achieved intensity by the enhancement process. The enhancement procedure suggested here is hue preserving. It generalizes the existing grayscale image enhancement techniques to color images. The processing has been done in RGB space and the saturation and hue values of pixels are not needed for the processing.

This chapter is organized in the following way. In Section 2.2, a general hue preserving transformation is proposed for image enhancement. In Section 2.3, a linear contrast enhancement transformation is described. In Section 2.4, a general principle for gamut problem free, hue preserving, non-linear transformation is developed and its consequences are discussed. Section 2.5 provides the comparison of the results of the proposed method only with existing methods. This chapter is concluded in Section 2.6 with a discussion on the proposed scheme and concluding remarks.

2.2 Hue Preserving Transformations

Hue preservation is necessary for color image enhancement. Distortion of image characteristics may occur if hue is not preserved. The hue of a pixel in the scene before the transformation and hue of the same pixel after the transformation are to be same for a hue preserving transformation. In this chapter, the aim is the development of a general hue

preserving transformation for contrast enhancement.

A common approach for hue preserving color image enhancement involves the enhancement in a color space such as LHS, HSI,YIQ, *etc.*, thereby separating the luminance and chrominance components of color. Some authors modified the luminance component keeping the chrominance components unchanged for enhancement. Some others modified saturation. In general, color images are stored and viewed using RGB color space. To process an image for enhancement in any of the above mentioned spaces (*i.e.*, LHS, HSI, YIQ, *etc.*), the image needs to be transformed to that space. The transformations involved in changing a color image from RGB space to other mentioned spaces are, generally, computationally costly [145] and again the inverse coordinate transformation has to be implemented for displaying the images. Two operations, *scaling* and *shifting*, are introduced in [145,146] for luminance and saturation processing. Scaling and shifting are hue preserving operations [145,146]. Using these two operations hue preserving contrast enhancement transformations are developed in this section.

To explain scaling and shifting in a mathematical fashion let us denote a pixel of an image $\mathbf{I}$ by $\boldsymbol{p}$ and the corresponding RGB color vector of $\boldsymbol{p}$ by $\tilde{\boldsymbol{x}}$, where, $\tilde{\boldsymbol{x}} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3})$, and $\boldsymbol{x_1},\ \boldsymbol{x_2},\ \boldsymbol{x_3} \in [\mathbf{0}, \mathbf{1}]$ correspond to the normalized *red*, *green* and *blue* pixel values respectively.

Scaling: Scaling the vector $\tilde{\boldsymbol{x}}$ to $\tilde{\boldsymbol{x}}'$ by a factor $\boldsymbol{\alpha} > \mathbf{0}$ is defined as

$$\tilde{\boldsymbol{x}}' = (\boldsymbol{x_1} \cdot \boldsymbol{\alpha},\ \boldsymbol{x_2} \cdot \boldsymbol{\alpha},\ \boldsymbol{x_3} \cdot \boldsymbol{\alpha}).$$

Shifting: Shifting a vector $\tilde{\boldsymbol{x}}$ to $\tilde{\boldsymbol{x}}'$ by a factor $\boldsymbol{\beta}$ is defined as

$$\tilde{\boldsymbol{x}}' = (\boldsymbol{x_1} + \boldsymbol{\beta},\ \boldsymbol{x_2} + \boldsymbol{\beta},\ \boldsymbol{x_3} + \boldsymbol{\beta}).$$

A transformation which is a combination of scaling and shifting can be written as

$$\tilde{\boldsymbol{x}}' = (\boldsymbol{\alpha} \cdot \boldsymbol{x_1} + \boldsymbol{\beta},\ \boldsymbol{\alpha} \cdot \boldsymbol{x_2} + \boldsymbol{\beta},\ \boldsymbol{\alpha x_3} + \boldsymbol{\beta})\,. \tag{2.1}$$

Note that in equation (2.1), x'_k is linear in x_k $\forall$ k, and α and β are not dependent upon $\tilde{x}$. Hue of a color vector in HSI (*hue, saturation, intensity*) space is defined to be an angle θ [40] where

$$\theta = cos^{-1}\left(\frac{[(x_1 - x_2) + (x_1 - x_3)]\frac{1}{2}}{\sqrt{(x_1 - x_2)^2 + (x_1 - x_3)(x_2 - x_3)}}\right). \tag{2.2}$$

When $x_1 = x_2 = x_3$, θ is undefined. *i.e.* the hue of all the points on the gray line in the RGB cube is undefined. Therefore the black pixel (having intensity 0) and white pixel (having intensity 3) have no hue or hue is undefined.

It can be shown that the transformation, as given in equation (2.1), is hue preserving. That is the θ value of $\tilde{x}$, as given in equation (2.2), is same as the θ value of $\tilde{x}'$. It is known in the literature [145, 146] that linear transformations are hue preserving. However for the purpose of clarity and continuity of the investigation, the above material is provided here.

Note that in equation (2.1), α and β are not dependent upon $\tilde{x}$. A general transformation in which α and β vary with each $\tilde{x}$ but same $\forall$ $k = 1, 2, 3$, is defined as

$$x'_k = \alpha(\tilde{x})x_k + \beta(\tilde{x}),\ k = 1, 2, 3. \tag{2.3}$$

Where $\alpha : \mathbb{R}^3 \longrightarrow (0, \infty)$ and $\beta : \mathbb{R}^3 \longrightarrow (-\infty, \infty)$. Note that, in the above equation (2.3), α and β are functions of $\tilde{x}$. It can be shown from elementary mathematics that the color vector $\tilde{x}'$ as defined in equation (2.3) possesses the same hue as the color vector $\tilde{x}$ for any two functions α and β. In other words, one can obtain several (in fact uncountable) non-linear hue preserving transformations by varying the functions α and β, as defined in equation (2.3). In the next section we shall study the two functions α and β involved in equation (2.3) for obtaining the mathematical expression of linear transformation for enhancement.

2.3 Linear Transformations

In this section, linear transformations are studied for enhancing the color images. Note that, for grayscale images, in many situations, the gray values are stretched to the max-

imum possible extent for achieving contrast. The same principle is extended to color images in this section. The intervals for each of the components R, G and B are stretched to the maximum possible extent without encountering the gamut problem.

Linear transformations are common for grayscale image enhancement. If we take $\boldsymbol{\alpha}(\tilde{\boldsymbol{x}})$ and $\boldsymbol{\beta}(\tilde{\boldsymbol{x}})$ as constant functions in equation (2.3), it will reduce to a linear transformation as stated below.

$$\begin{aligned} \boldsymbol{x'_k} &= \boldsymbol{\alpha} \cdot \boldsymbol{x_k} + \boldsymbol{\beta} \text{ or} \\ \boldsymbol{x'_k} &= \boldsymbol{\alpha_1} \cdot (\boldsymbol{x_k} + \boldsymbol{\beta_1}),\ \forall\, \boldsymbol{p} \in \mathbf{I},\ \boldsymbol{k} = \mathbf{1, 2, 3.} \end{aligned} \tag{2.4}$$

where $\boldsymbol{x_k}$ is the gray value of the $\boldsymbol{k^{th}}$ component of the pixel $\boldsymbol{p}$, $\boldsymbol{x'_k}$ is the modified gray value of the $\boldsymbol{k^{th}}$ component of the pixel, $\boldsymbol{\alpha_1} = \boldsymbol{\alpha}$, and $\boldsymbol{\beta_1} = \frac{\boldsymbol{\beta}}{\boldsymbol{\alpha}}$.

Note that $\boldsymbol{x_k}$'s are normalized pixel values. Hence $\boldsymbol{x_k} \in [\mathbf{0, 1}]\ \forall\ \boldsymbol{k} = \mathbf{1, 2, 3}$. We would like to make $\boldsymbol{x'_k} \in [\mathbf{0, 1}]\ \forall\ \boldsymbol{k}$. Then $\boldsymbol{\alpha_1}$ must be non negative since $\boldsymbol{x'_k} \geq \mathbf{0}\ \forall\ \boldsymbol{k}$. To achieve maximum contrast, the pixel values, after the transformation, should spread over the whole range *i.e.* from 0 to 1. It can be clearly seen that if

$$\boldsymbol{\beta_1} = -\min_{\boldsymbol{p} \in \mathbf{I}} \{\min_{\boldsymbol{k}} \boldsymbol{x_k}\} \text{ and } \boldsymbol{\alpha_1} = \frac{\mathbf{1}}{\max\limits_{\boldsymbol{p} \in \mathbf{I}} \{\max\limits_{\boldsymbol{k}} \boldsymbol{x_k}\}},$$

equation (2.4) provides a hue preserving linear transformation with maximum possible contrast. Equation (2.4) not only enhances the intensity of the image but also modifies the saturation. $\boldsymbol{\beta_1}$ is responsible for the increase or decrease in saturation where as $\boldsymbol{\alpha_1}$ is responsible for maximizing the range of the intensity and hence the enhancement in contrast of the image. This method basically provides a linear stretching of pixel values by the same amount for each component.

Two real life images namely, [Lenna Fig. 2.2(a)], Dancer [Fig. 2.2(b)] and one artificial image [Fig. 2.2(c)] are considered for showing the results and comparison with other methods. Each is a 256 $\times$ 256 image with each of R, G and B values ranging from 0 to 255. Lenna image has good contrast. Dancer image is a noisy image because of the poor lighting. In the artificial image, there are eight rectangular colored boxes with different

hues and different luminance. Each rectangular box is divided into two square boxes with different saturations but same hue and same luminance values.

For each image and for each of R, G and B, the pixel values are normalized (*i.e.*, division by 255) and the processing is performed with the normalized values. The obtained normalized values after processing are converted to the usual pixel values for each of R, G and B (*i.e.*, multiplication by 255) and the images thus obtained are shown. Same is the case with applying the other enhancement techniques (developed in section IV). It is to be noted from the obtained results of linear enhancement that for the Lenna [Fig. 2.2(a)] and artificial [Fig. 2.2(c)] images, contrast change is not significantly visible between the original and the transformed images [Fig. 2.3(a) and Fig. 2.3(c)]. This may be due to the fact that the original Lenna and artificial images are images with good contrast. The hue of the colors in the images is preserved. On the other hand, in Dancer image [Fig. 2.3(b)], enhancement is sharply visible and the hue is also preserved.

In this section we have established a linear transformation without gamut problem and it stretches the length of the interval for each of R, G and B to the maximum possible extent. In the next section we shall discuss non-linear hue preserving transformations.

2.4 Non-Linear Transformations

In this section we shall generalize the usual grayscale contrast enhancement techniques to color images in such a way that they are hue preserving. The objective is to keep the transformed values within the range of the RGB space *i.e.*, the transformations are free from gamut problem. Some general and widely used contrast enhancement techniques for grayscale images are S-type enhancement, piecewise linear stretching, clipping, gamma correction function *etc.* The methodologies of these techniques can be found in the literature. Some of these transformations are listed below for grayscale images.

Let $\boldsymbol{f}$ denote the enhancement function and $\boldsymbol{x}$ represents the gray value of a pixel. Thus $\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}) \in [\boldsymbol{\delta_1}, \boldsymbol{\delta_2}]$, where $\boldsymbol{\delta_1 = 0,\ \delta_2 = 1}$.

Contrast Stretching :

$$f(x) = \begin{cases} \gamma_1 x, & \delta_1 \leq x < a \\ \gamma_1 a + \gamma_2(x-a), & a \leq x < b \\ \gamma_1 a + \gamma_2(b-a) + \gamma_2(x-b), & b \leq x \leq \delta_2 \end{cases} \tag{2.5}$$

Where $\gamma_1, \gamma_2, \gamma_3$ are suitably chosen constants and $\delta_1 \leq a < b \leq \delta_2$.

S-type :

$$f_{m,n}(x) = \begin{cases} \delta_1 + (m-\delta_1)\left(\dfrac{x-\delta_1}{m-\delta_1}\right)^n, & \delta_1 \leq x \leq m \\ \\ \delta_2 - (\delta_2 - m)\left(\dfrac{\delta_2 - x}{\delta_2 - m}\right)^n, & m \leq x \leq \delta_2 \end{cases} \tag{2.6}$$

m and n are two constants, $n \in (0, \infty)$, $m \in [\delta_1, \delta_2]$. This transformation is written in most general form. If $\delta_1 = 0$, $\delta_2 = 1$, $n = 2$ and $m = 0.5$, it provides the standard S-type contrast enhancement.

Gamma Correction :

$$f(x) = x^{\frac{1}{\gamma}} \tag{2.7}$$

where γ is a positive real number.

The above listed transformations are one dimensional but a pixel vector in a color image is represented by a three dimensional vector. The procedure followed for generalizing grayscale contrast intensification to color images is discussed below.

A generalized transformation changing $\tilde{x}$ to $\tilde{x}'$ is given as

$$x'_k = \alpha(x_k)x_k + \beta(x_k) \; \forall \; p \in \mathbf{I}, \; k = 1, 2, 3. \tag{2.8}$$

In the above equations $\alpha : [0, \infty) \longrightarrow [0, \infty)$ and $\beta : [0, \infty) \longrightarrow (-\infty, \infty)$.

This transformation is a most generalized transformation, where α and β can be any two functions. Looking into the increasing complexity of the problem, the linear transformation as described in previous section, is applied initially. Thus, the quantity $\beta(x_k)$ can

taken to be zero for all $\boldsymbol{p} \in \mathbf{I}$ and $\boldsymbol{k} = \mathbf{1, 2, 3}$. Non-linear transformation is applied on the changed color vectors. Applying initially the said linear transformation stretches the intervals of the color levels linearly so that the pixel values will be spread to the maximum possible extent for each of the intervals for R, G and B. After taking $\boldsymbol{\beta}(\boldsymbol{x_k})$ to be zero for all $\boldsymbol{p} \in \mathbf{I}$ and $\boldsymbol{k} = \mathbf{1, 2, 3}$ in the equation (2.8), the transformation would become

$$\boldsymbol{x'_k = \alpha(x_k)x_k \ \forall \ p \in \mathbf{I}, \ k = 1, 2, 3.} \tag{2.9}$$

In the above equation $\boldsymbol{\alpha}$ is a function of $\boldsymbol{x_k}$ *i.e.*, it modifies the three components of the color vector by three different scales. This leads to change in hue of the color vector, which is against our aim. A way of making this transformation hue preserving is to have the same scale for each of the three components of the vector. It is already shown that this type of scaling is hue preserving. In particular, $\boldsymbol{\alpha}$ can be taken as a function of $\boldsymbol{l(\tilde{x})}$, where $\boldsymbol{l(\tilde{x}) = \sum_{k=1}^{3} x_k}$. Then the transformation will be of the form

$$\boldsymbol{x'_k = \alpha(l(\tilde{x}))x_k \ \forall \ p \in \mathbf{I}, \ k = 1, 2, 3.} \tag{2.10}$$

Initially, we define

$$\boldsymbol{\alpha(l(\tilde{x})) = \begin{cases} \dfrac{f(l(\tilde{x}))}{l(\tilde{x})}, & l(\tilde{x}) \neq 0 \\ 0, & otherwise \end{cases}} \tag{2.11}$$

where $\boldsymbol{f(l(\tilde{x}))}$ is a non-linear transformation used in contrast enhancement for grayscale images. For example, S-type transformation is listed earlier in this section. In the present case we can take $\boldsymbol{\delta_1 = 0}$ and $\boldsymbol{\delta_2 = 3}$. *i.e.*, $\boldsymbol{l(\tilde{x}), \ f(l(\tilde{x})) \in [0, 3]}$. As $\boldsymbol{\alpha(l(\tilde{x}))}$ is a ratio of $\boldsymbol{f(l(\tilde{x}))}$ and $\boldsymbol{l(\tilde{x})}$, when $\boldsymbol{f(l(\tilde{x})) > l(\tilde{x})}$, value of $\boldsymbol{\alpha(l(\tilde{x}))}$ will be greater than 1. In such a case value of $\boldsymbol{x'_k}$ may exceed 1 and thus resulting in gamut problem. A possible solution to this is to transform the color vector to CMY space and process it there. This will be dealt with in two separate cases.

Case 1 : $\boldsymbol{\alpha(l(\tilde{x})) \leq 1}$

$$x'_k = \alpha(l(\tilde{x}))\, x_k \;\; \forall\, k = 1, 2, 3.$$

Case 2 : $\alpha(l(\tilde{x})) > 1$

1. Convert RGB vector $\tilde{x}$ to CMY vector $\tilde{y}$ by
   $\tilde{y} = (1 - x_1, 1 - x_2, 1 - x_3)$
2. $l(\tilde{y}) = \sum_{k=1}^{3} y_k = \sum_{k=1}^{3}(1 - x_k) = 3 - l(\tilde{x}).$
3. Take $g(\tilde{y}) = 3 - f(l(\tilde{x}))$
4. Take $\alpha(l(\tilde{y})) = \dfrac{g(l(\tilde{y}))}{l(\tilde{y})} = \dfrac{3 - f(l(\tilde{x}))}{3 - l(\tilde{x})},$
   Here, it is to be noted that $\alpha(l(\tilde{y})) < 1.$
5. $y'_k = \alpha(l(\tilde{y}))\, y_k \;\; \forall\, k = 1, 2, 3.$
   Note, $\alpha(l(\tilde{y})) < 1$ implies $y'_k \leq 1 \;\; \forall\, y_k \leq 1$.
6. Convert CMY vector $\tilde{y}_e$ to RGB vector $\tilde{x}'$ by
   $\tilde{x}' = (1 - y'_1,\; 1 - y'_2,\; 1 - y'_3)$ □

2.4.1 Salient Points of the Proposed Scheme

Some of the salient points regarding the principle established above for hue-preserving color image enhancement are stated below.

1. In the proposed principle, intensity of a pixel is getting modified with the help of the contrast enhancement function f. Different contrast enhancement functions provide different importance to pixels. But α need not always be a function of intensity. There can be several hue preserving α functions whose functional forms are not dependent upon the intensity, $l(\tilde{x}) = \sum_{k=1}^{3} x_k$ of the pixel.
2. Any grayscale image enhancement scheme can be generalized to color images with the above principle.

3. Basically, in this procedure, a gray level contrast enhancement function is applied to intensity values and the transformed intensity value is divided proportionately among the three components in RGB space or CMY space.

4. The above method also possesses another desirable property *i.e.* if the enhancement function f preserves the orderings of $l(\tilde{x})$ then above method also preserves the same ordering.

   To explain it mathematically, we can express the proposed enhancement scheme as

$$\tilde{x}' = \begin{cases} \dfrac{f(l(\tilde{x}))}{l(\tilde{x})}\tilde{x} & if \quad f(l(\tilde{x})) \leq l(\tilde{x}) \\ \\ \tilde{1} - \dfrac{3 - f(l(\tilde{x}))}{3 - l(\tilde{x})}(\tilde{1} - \tilde{x}) & if \quad f(l(\tilde{x})) > l(\tilde{x}) \end{cases} \tag{2.12}$$

   where. $l(\tilde{x}) = \sum_{k=1}^{3} x_k$ for any RGB color vector $\tilde{x}$ and $\tilde{1} = (1, 1, 1)$.
   Let us assume that f is a contrast modification function, as used in the proposed principle, such that for any two color vectors $\tilde{x}$ and $\tilde{z}$,

$$l(\tilde{x}) \leq l(\tilde{z}) \Rightarrow f(l(\tilde{x})) \leq f(l(\tilde{z})) \tag{2.13}$$

   Then, our claim is

$$l(\tilde{x}') \leq l(\tilde{z}'),$$

   where $\tilde{x}'$ and $\tilde{z}'$ are the enhanced vectors of $\tilde{x}$ and $\tilde{z}$ respectively using equation (2.12).

   From equation 2.12 it can be seen that for both the cases i.e, for $f(l(\tilde{x})) \leq l(\tilde{x})$ and $f(l(\tilde{x})) > l(\tilde{x})$, and for any RGB color vectors $\tilde{x}$ and $\tilde{z}$,

$$\left.\begin{aligned} l(\tilde{x}') &= \sum_{k=1}^{3} x'_k = f(l(\tilde{x})) \\ l(\tilde{z}') &= \sum_{k=1}^{3} z'_k = f(l(\tilde{z})) \end{aligned}\right\} \tag{2.14}$$

Hence, from (2.13) and (2.14)

$$l(\tilde{x}') = f(l(\tilde{x})) \leq f(l(\tilde{z})) = l(\tilde{z}'). \qquad \square$$

5. If the function f used for contrast enhancement is a continuous function of $l(\tilde{x})$ then proposed principle preserves the continuity property i.e, to say that if f is continuous then the transformation $x \mapsto x'$ is continuous. To prove this consider equation (2.12). It can be seen that individually the two terms

$$\frac{f(l(\tilde{x}))}{l(\tilde{x})}\tilde{x}_k \text{ and } \tilde{1} - \frac{3 - f(l(\tilde{x}))}{3 - l(\tilde{x})}(\tilde{1} - \tilde{x})$$

are continuous since $f(l(\tilde{x}))$ and $l(x)$ are continuous. It can also be seen that they attain the same value when $f(l(\tilde{x})) = l(\tilde{x})$. Hence the transformation $\prime$ is continuous.

6. Note that, initially, linear transformation, as developed in the previous section, is applied on each of the pixels to extend the interval for each of R, G and B to the maximum possible extent. We believe that non-linear hue preserving transformation without gamut problem can be developed without initially applying the linear transformation, though, we have not attempted to find one such transformation here. The block diagram for the proposed scheme is shown in Fig. 2.1.

7. For S-type enhancement function, the values of δ_1 and δ_2 (equation (2.6)) are taken to be 0 and 3 respectively because maximum possible value of $l(\tilde{x})$ is 3 and minimum possible value is 0. Values of δ_1 and δ_2 may be chosen accordingly if $l(\tilde{x})$ is other than $\sum_{k=1}^{3} x_k$.

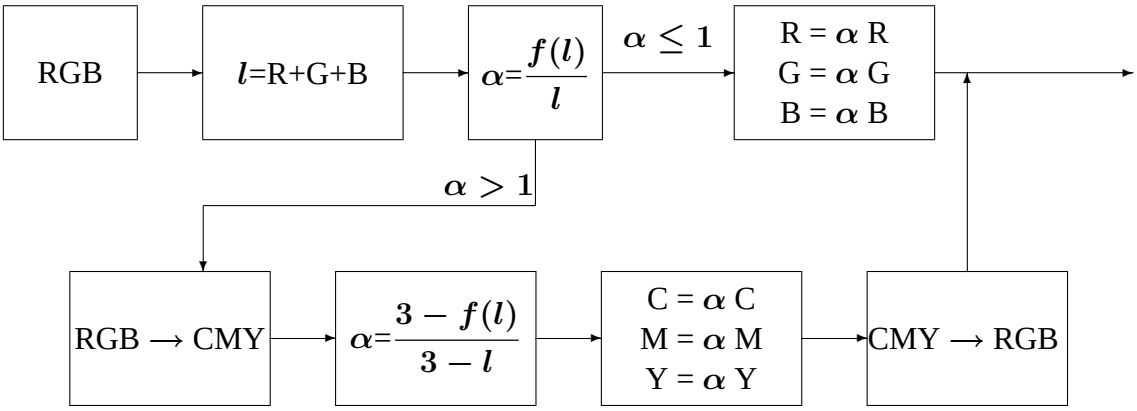


Figure 2.1: Block diagram of the proposed enhancement scheme

8. A good hue preserving transformation should not map a pixel having nonzero saturation to a pixel having saturation zero. Otherwise, hue of the pixel becomes undefined. Using the same principle developed in this section, such a case will not arise unless it is a digitization error or the enhancement function f maps a $l(\tilde{x}) \neq 0$ or 3 to either 0 or 3.

9. The suggested non-linear transformation for image enhancement is hue preserving and free from gamut problem. The gamut problem is tackled successfully by transforming the corresponding color vectors of the pixels for which $\frac{f(l(\tilde{x}))}{l(\tilde{x})}$ to CMY space. This is a novel technique because we can avoid the gamut problem keeping the hue unchanged. Note that in CMY space also hue is scaling and shifting invariant.

10. It is to be noted that costly co-ordinate transformation from one space to another space is not carried out in the proposed scheme, though the knowledge regarding the RGB and CMY spaces is used for developing the above principle.

2.4.2 Histogram Equalization

Histogram equalization is a very powerful scheme for contrast enhancement in grayscale images. In grayscale histogram equalization, the method rearranges the gray values in such a way that the modified histogram resembles the histogram of uniform distribution. Intuitively, the same technique should work on a color image also *i.e.* by equalizing the image in three dimensional space where the three dimensions are the three primary components of the color space. But this causes unequal shift in the three components resulting in change of hue of the pixel [127]. A solution is to equalize only the luminance/saturation or both the components in the color spaces such as LHS, HSI or YIQ. This sometimes leads to gamut problem when the processed data is again transformed back to RGB space. Using the principle mentioned previously this problem can be avoided *i.e.,* equalize the intensity $l(\tilde{x})$ in $[\delta_1, \delta_2]$ and let $f(l(\tilde{x}))$ be the equalization function and then follow the steps of the algorithm as in the case of non-linear enhancement. The same principle works for any histogram specification problem. To eliminate the cases of undefined hue, values of δ_1 and δ_2 are to be chosen properly.

2.5 Results and Comparisons

In this section, results using S-type Enhancement and Histogram equalization with proposed principle is compared with three different methods. It is evident from the previous sections that hue-preservation is essential for image enhancement. Hence, we have not considered any algorithm, which is not hue preserving, for comparison . Histogram equalization is the most popular and widely used technique for image enhancement. Thus we have considered three histogram equalization algorithms for comparison. Out of these three algorithms, two of them use clipping technique to cope with the gamut problem and the other one uses normalization for this purpose. The two methods proposed by Yang *et al.* [146] used clipping techniques using the knowledge of LHS and YIQ color spaces respectively. The other considered algorithm is the histogram equalization technique pro-

posed by Weeks *et al.* [144], which uses normalization to counter the gamut problem.

The proposed principle for enhancement is implemented on several color images successfully. The images of Fig. 2.2 are considered for showing the results of the techniques developed in the previous section. Figs. 2.4 and 2.7 provide results of applying the proposed principle on the images shown in Fig. 2.2, for the S-type enhancement function with $n = 2,\ m = 1.5,\ \delta_1 = 0$ and $\delta_2 = 3$ and for histogram equalization respectively. The proposed method is compared with Yang *et al.* [146] and Weeks *et al.*'s [144] methods. Figs. 2.5, 2.6 show the clipped histogram equalized images obtained by Yang *et al.*'s method. Fig. 2.8 shows the results of Weeks *et al.*'s method.

Many existing algorithms preserve hue and aim to achieve high contrast. In the process, these algorithms face gamut problem [42, 127, 131, 134, 135]; *i.e.* the transformed R, G and B values may not always lie in their designated intervals. A solution to this is to rescale the R, G and B values of the color vectors [11, 127, 144]. But rescaling the R, G and B values of the color vectors may reduce some luminance enhancement effect [127]. Weeks *et al.* [144] also mentioned that the luminance may be modified due to rescaling. Clipping the processed luminance values before transforming back to RGB space is also a solution to retain the transformed R, G and B values within their designated intervals [146]. But this may reduce the effectiveness of the image enhancement. For example, this may result in less contrast when doing histogram equalization [146]. To preserve the achieved contrast, Yang *et al.* [146] have suggested clipping of saturation component instead of luminance component.

In [131], it is mentioned that objects can exhibit variation in color saturation with little or no corresponding luminance variation. From this fact it can so happen that there can be an edge due to the variation in saturation and having little or no variation in spatial intensity. If such type of pixels are severely effected by clipping then edges may be lost or direction of the edges may be changed, which is certainly against the principles of enhancement. Keeping this fact in mind the artificial image Fig. 2.2(c) is created in which different hues with different saturations are shown.

The results from Yang *et al.*'s method and our methods are comparable for Lenna and Dancer images. Note that the effect of clipping is not distinctly visible in these images. Let us examine the artificial image [Fig. 2.5(c)] enhanced by Yang *et al.*'s method. The edge between the two square boxes in the rectangular boxes of first column and fourth rectangular box of second column in Fig. 2.5(c) is lost. This is because of the clipping of saturation component. Similarly, in Fig. 2.6(c), edges between the square boxes within last three rectangular boxes of the first column and the fourth rectangular box of the second column are lost. In the image Fig. 2.7(c), the edges between the consecutive square boxes are preserved. Note that the proposed method preserves the order of occurrence of intensity. It may also be stated that, visually, edges are not deleted in the enhanced versions of the artificial image, using the proposed histogram equalization method and the S-type function based scheme. Weeks *et al.* applied normalization to bring back the out of bounds values to within the bounds. Effect of it is that the images in Figs. 2.8 are not as bright as in the images in Figs. 2.7, 2.5 and 2.6. Equalization of saturation sometimes degrades the quality of the image since it leads to very large saturation values that are not present in the natural scenes [107]. Sometimes, increase in both luminance and saturation cause unnatural color. This can be observed in the case of Lenna image [Fig. 2.8(a)]. The feathers in the hat of Lenna look different from the original. In the original these have low luminance, thus look black. After the enhancement, with the increase in saturation as well as luminance, the colors of these pixels have been changed. Results on two more images are shown in Fig. 2.10 and Fig. 2.11. The output obtained indicates that the proposed scheme provides acceptable enhancement with all the images. It may be noted that the results of Weeks *et al.*'s method on the artificial image looks better than output of the proposed scheme though the output of the proposed scheme is also an acceptable one.

2.6 Conclusion and Discussion

The main contribution here is the scheme to generalize any grayscale image enhancement method to color images without encountering gamut problem. The overall enhancement obtained by the proposed scheme is mainly dependent on the already existing different contrast enhancement functions for grayscale images. These contrast enhancement functions for grayscale images are generalized to enhance the intensity of the color images, keeping the hue intact. A novel scheme is proposed to avoid gamut problem arising during the process of enhancement. This scheme is used to enhance the intensity of color images using a general hue preserving contrast enhancement function. The transformation is general in the sense that the function $\boldsymbol{f}$ can be any contrast enhancement function used for grayscale images and is gamut problem free irrespective of the nature of the function $\boldsymbol{f}$.

Linear stretching on intensity here is very similar to the linear stretching done on gray levels in grayscale images. But it may be noted that the three components are not stretched separately. This helps in preserving the hue.

While doing the non-linear enhancement, we have suggested that linear stretching is to be applied prior to the non-linear transformation. Here, it may be noted that non-linear enhancement can also be done independently. The general hue preserving non-linear transformation is of the form :

$$\boldsymbol{x'_k = \alpha(\tilde{x})x_k + \beta(\tilde{x}).}$$

The problem here is the choice of the functions $\boldsymbol{\beta(\tilde{x})}$ and $\boldsymbol{\alpha(\tilde{x})}$.

In the theory of proposed histogram equalization method, the maximum possible interval for $\boldsymbol{l(\tilde{x})}$ is taken to be $\boldsymbol{[0, 3]}$. But sometimes this causes over enhancement. To avoid over enhancement it can be equalized within $\boldsymbol{[a, b]}$ for some $\boldsymbol{a > 0}$ and $\boldsymbol{b < 3}$. In Section 2.5, an example has been provided which shows the over enhancement. In Fig. 2.7(c), due to this over enhancement, the square box with yellow color becomes almost white, because $\boldsymbol{l(\tilde{x})}$ of this box is the highest in the image, and after equalization it

becomes close to 3 which is almost white.

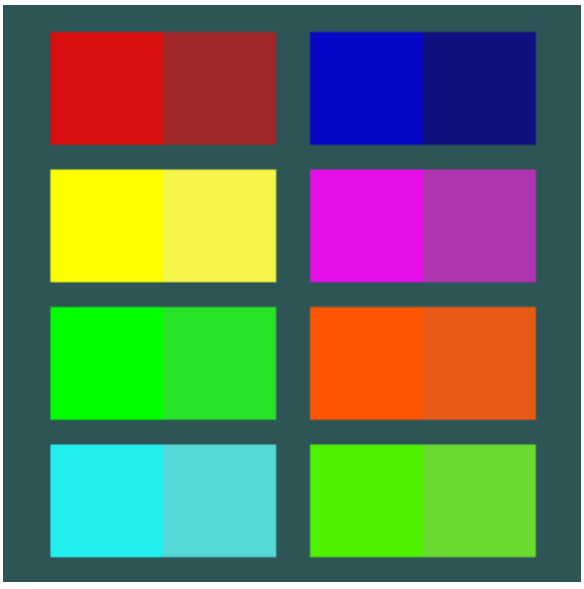
It is to be noted that, image contrast enhancement is not the panacea of all the problems associated with better visual quality of images. For example, noise in an image will not be removed by contrast enhancement. The proposed histogram equalization technique for color images indeed enhances the image but the clarity of the enhanced image may still be poor in a noisy image (See Fig. 2.2(b) and Fig. 2.7(b)). Note that, in Fig. 2.2(b), there are a few bright patches present in it and they dominated the whole enhancement procedure. If the bright patches are removed and the enhancement is performed in the rest of the image then the clarity would be much better. Such an experiment is performed with the help of the proposed histogram equalization method and the result is shown in Fig. 2.9. The white patches in the image are not considered while equalization is performed. Equalization of $l(\tilde{x})$ is done in the interval $\left[\frac{1}{255}, 2.3\right]$. The visual quality of Fig. 2.9 is indeed better than the results obtained by all the other methods. For an even better image quality for Dancer image, probably, some noise cleaning algorithm are to be incorporated before enhancement.

The proposed scheme always decreases the saturation whenever it is affected by the condition $\alpha(l(\tilde{x})) > 1$, which is not always desirable. Some pixels may require to saturate more whereas some may require a decrease in the saturation. This is possible if we consider general non-linear hue preserving transformation. One needs to be cautious in varying the saturation. The variation should be continuous. Otherwise, it may result in visual artifacts.

Even if $\beta(\tilde{x})$ is zero for all and the linear stretching operation is performed in the original image to increase the length of the interval in each of R, G and B to the maximum possible extent, $\alpha(\tilde{x})$ need not be a function of $l(\tilde{x})$. For example, one can take $\alpha(\tilde{x})$ to be a function of $\sqrt{\frac{x_1^2 + x_2^2 + x_3^2}{3}}$, which makes the enhancement function hue preserving. In fact, several other such hue preserving $\alpha(\tilde{x})$ can easily be defined in this way. Construction of generalized non-linear (both for α and β functions) hue preserving transformation without gamut problem is a matter for future research.

(a) Lenna

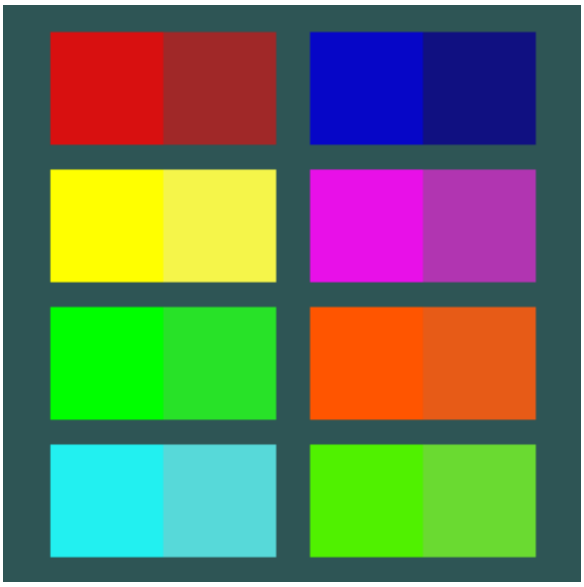
(b) Dancer



(c) Artificial

Figure 2.2: Original images considered for Enhancement

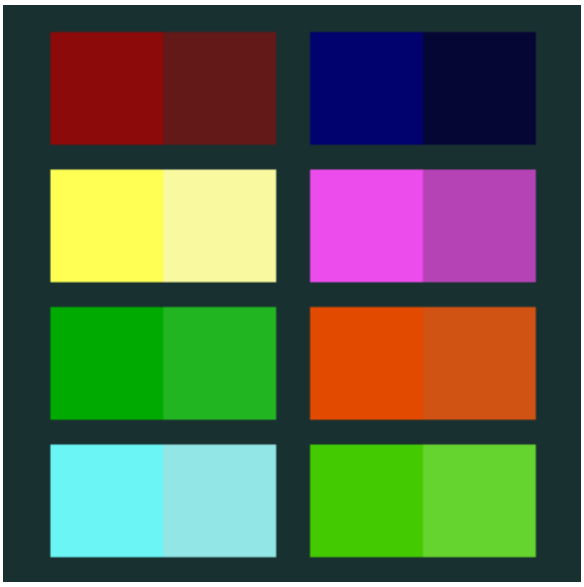
(a) Lenna (b) Dancer



(c) Artificial

Figure 2.3: Images Enhanced by Linear Stretching

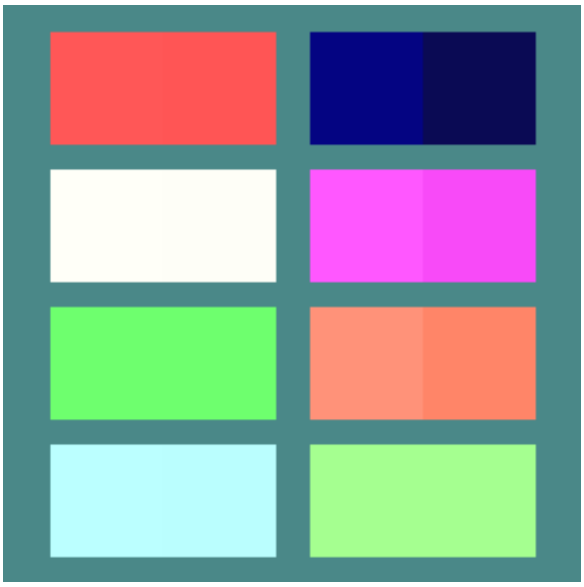
(a) Lenna (b) Dancer



(c) Artificial

Figure 2.4: Images Enhanced using S-type function with n=2 and m=1.5

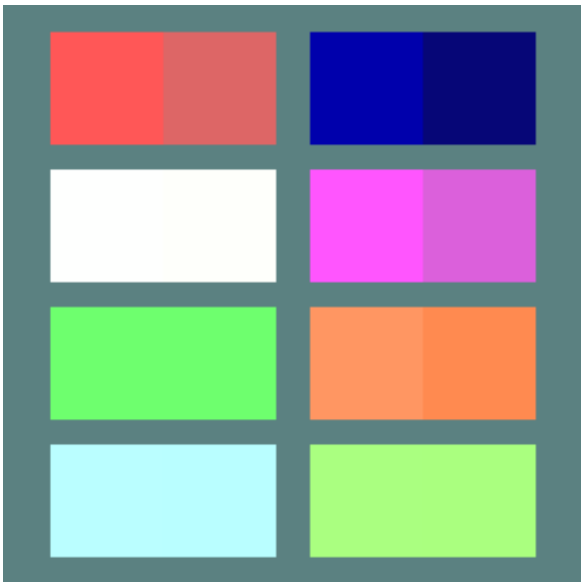
(a) Lenna (b) Dancer



(c) Artificial

Figure 2.5: Images Enhanced by Yang *et al.*'s Method in LHS system

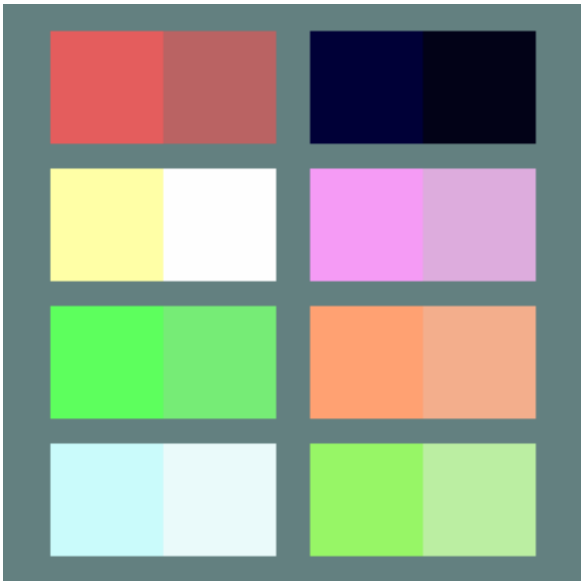
(a) Lenna (b) Dancer



(c) Artificial

Figure 2.6: Images Enhanced by Yang *et al.*'s Method in YIQ system

(a) Lenna (b) Dancer



(c) Artificial

Figure 2.7: Images Enhanced by proposed Histogram Equalization method

(a) Lenna (b) Dancer

(c) Artificial

Figure 2.8: Images Enhanced by Weeks *et al.*'s Equalization method

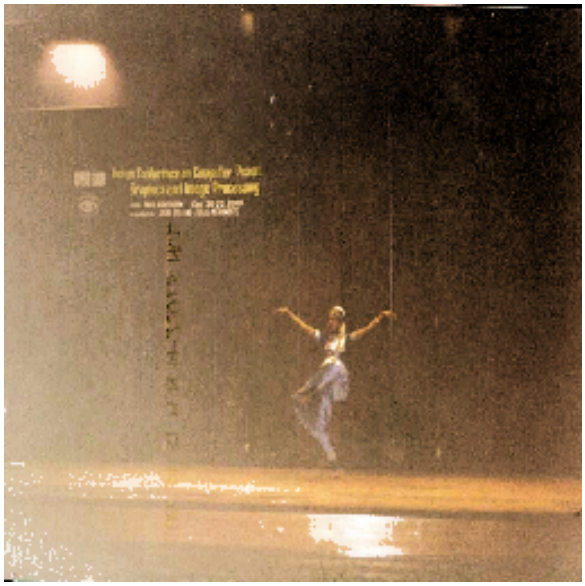
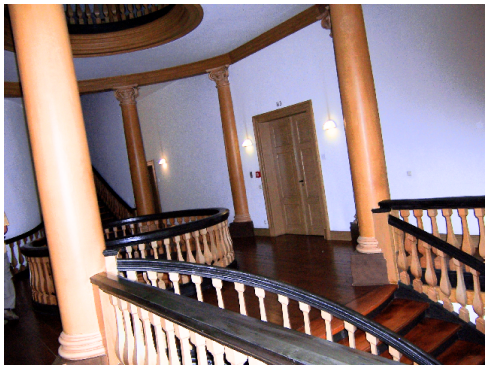


Figure 2.9: Equalized by the proposed method without considering the white patches.

(a) Original Image

(b) Proposed Histogram Equalization



(c) Yang *et al.*(LHS)

(d) Yang *et al.*(YIQ)

(e) Weeks *et al.*Histogram Equalization

Figure 2.10:

(a) Original Image

(b) Proposed Histogram Equalization

(c) Yang *et al.*(LHS)

(d) Yang *et al.*(YIQ)

(e) Weeks *et al.*Histogram Equalization

Figure 2.11:

Chapter 3

Standardization of Edge Magnitude in Color Images

Edge detection has been a challenging problem in low level image processing. It becomes more challenging when color images are considered because of its multidimensional nature. Color images provide more information than grayscale images. Thus more edge information is expected from a color edge detector than a grayscale edge detector [31, 59, 102]. In a grayscale image, edges are detected by detecting the discontinuities in the image surface i.e. the discontinuities in the intensity of a sequence of pixels in a particular direction called gradient direction. The discontinuities in grayscale are easy to determine because gray values are partially ordered, but in a color image this freedom is not there. The simple difference between color vectors does not give the true distance between them. Sometimes it is difficult to detect a low intensity [59] edge between two regions in grayscale, but in color image, the clarity is more because, without being much different in intensity there can be a substantial difference in hue. It is observed by Novak *et al.* [102] that almost 90% of edge information in a color image can be found in the corresponding grayscale image. Though, this is not a significant ratio in favor of color edge detection, the remaining 10% can still be vital in certain computer vision tasks. Further, human perception of color picture is perceptually richer than an achromatic picture [101].

Image edge maps provide crucial information regarding the image. It is one of the first steps in many high level computer vision and image processing tasks. For instance, local and global shape information of the images are often sought for representation of images for different purposes. In this thesis edge maps of images are sought for the purpose of object representation. There are several edge detection algorithms available in the literature for color images. However, they produce good results subject to the condition of tuning the parameters involved in the algorithm for each individual image. But, it becomes a difficult task when the algorithm is to be used for a large number of images and expecting uniformly acceptable results from all the images without tuning the parameters. Thus a new edge detection algorithm for color images is proposed here to find uniform and acceptable results for all the images. A short literature survey on color image edge detection methodologies is conducted in the following section.

3.1 A Survey of Edge Detection in Color Images

One of the earliest color edge detectors is proposed by Navatia [101]. The image data is transformed to luminance Y and two chrominance components $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ and Huckel's edge detector is used to find the edge map in each individual component independently, except for the constraint of having the same orientation. Shiozaki [126] found entropy in each component using a local entropy operator and merged the three values for color edge detection. Machuca *et al.* [73] transformed the image from RGB to YIQ and detected edge in the hue plane. Fan *et al.* [31] proposed a method where they find edges in YUV space. Edge magnitude is found individually and thresholded in each component and merged. They proposed an entropy based method to automatically detect a threshold value for each component. This method is simple and may be faster in computation but produces less accurate results [32].

A multi-dimensional edge detection method using differential geometric approach was proposed by DiZenzo [26]. He considered the multi-images as a vector field and

found the tensor gradient. Explicit formulas for the edge direction and magnitude in multi-spectral images are derived. He also shows that the earlier ways of finding edges by combining the output of difference operators in each component does not actually cooperate with one another. Cumani proposed an extension of the second-directional derivative approach to color images in [23]. He found the edge map by locating the zero crossings in image surface. Formulas for second order partial derivatives for finding zero crossings in multi-spectral images are derived. He defined the direction of maximal-contrast by the corresponding eigenvector of the largest eigenvalue of the $\mathbf{2} \times \mathbf{2}$ matrix formed from the outer product of the gradient vector in each component. Alshatti *et al.* [4] suggested a modification of the Cumani's approach to solve the problem of sign ambiguities and to reduce computational time involved in the selection of maximal directional contrast. Later Cumani [24] proposed an efficient algorithm to get rid of this problem. The multi-dimensional gradient method is also used by Saber *et al.* [121] for edge linking in image segmentation. Some of the other related works can be found in [27, 62, 113].

Trahanias *et al.* [138, 139] proposed a class of edge detectors using vector order statistics. Three different ways of finding edge magnitudes using dispersion of color vectors from the median of the set of vectors in the neighborhood of a pixel are proposed. Toivanen *et al.* [136] have pointed out that R-ordering sometimes orders two different spectra into the same scalar value. They have proposed a different ordering method of multi-spectral image pixels. They further used self-organizing map(SOM) for this purpose. A class of directional vector operators are proposed to detect the location and orientation of edges in color images by Scharcansk *et al.* [122]. A comprehensive analysis of color edge detectors can be found in Zhu *et al.*'s [149] work.

Ruzon *et al.* [120] have proposed an algorithm using compass operator. It considers a disc at each pixel location. The disc is divided into several pairs of opposite semi discs by rotating the diameter over 180 degrees with an interval of 15 degrees. The color distribution of the pixels in each such semi disc is found after doing vector quantization. The distance between two semi discs generated by a single diameter is the distance be-

tween their color distributions. The distance between two distributions is found using Earth Mover's Distance (EMD). The edge magnitude at the pixel is the maximum distance among all the distances found between each pair of opposite semi discs created by rotating the diameter.

Various types of edge detection algorithms have been discussed above. All of them have their advantages and disadvantages. Some of these are pointed out here. The early approaches to color edge detectors, which are extensions of achromatic color edge detectors failed to extract certain crucial information conveyed by color [149]. R-ordering in certain cases orders two different spectra into the same scalar value, and as a result, can miss some parts of edges [136]. The edge detection approaches which simply add the gradient magnitudes of all the color components may fail to detect some crucial edge information in certain cases as described by DiZenzo in [26]. Thus though several color edge detection algorithms are proposed in the literature several open problems still exist.

Most of the algorithms are concerned about finding the edge magnitude and direction. This is expected and also justified because the quality of edge map depends on these two quantities. But, none of the algorithms except the work by Fan *et al.* [31], addressed the selection of thresholding parameters, although, this is also an integral part in most of the algorithms. Edge detection algorithms discussed in this section have reported good results when the values of the parameters involved are adjusted suitably. Though, most of these algorithms produce good results, generally, none of them is designed to give uniformly acceptable results for all kinds of images with a fixed set of parameter values. Thus, there is a need to devise a method which can give uniformly acceptable results with a single set of parameter values. This is especially needed in the context of object recognition when the edge detection algorithm is to be applied to several images and manual tuning of parameter is not possible. In this study, this problem is addressed.

This chapter is organized in the following way. Section 3.2 gives a brief idea of the mathematical foundation of the method. In Section 3.3, results of the proposed method are compared with other methods. The chapter is concluded in Section 3.4 with concluding

remarks.

3.2 Edge Detection in Color Images

This section describes the theoretical foundation of the proposed edge detection method with a short note on an existing principle of finding edge magnitude and direction at each pixel location of a color image using image derivatives.

3.2.1 Theoretical Foundation

There are a number of well-established edge detection techniques available for grayscale images. Edges are generally detected using a local operator followed by a thresholding operator. The local operators are usually a digital approximation of the gradient or the derivatives of the image considering it as a digital 2-dimensional surface. The two important factors in a derivative based method are the edge magnitude at a pixel and its direction. There is existence of a strong theoretical foundation [18] behind derivative based edge detection. In short, it is a well established fact that for a grayscale image $\boldsymbol{f}$, the magnitude of the edge at a pixel location $(\boldsymbol{x}, \boldsymbol{y})$ is given by $\sqrt{\boldsymbol{f_x^2}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{f_y^2}(\boldsymbol{x},\boldsymbol{y})}$ and it is attained in the direction given by $\boldsymbol{\theta} = \mathbf{tan}^{-1}\left(\dfrac{\boldsymbol{f_y}(\boldsymbol{x},\boldsymbol{y})}{\boldsymbol{f_x}(\boldsymbol{x},\boldsymbol{y})}\right)$, where $\boldsymbol{f_x}$ and $\boldsymbol{f_y}$ are the partial derivatives of $\boldsymbol{f}$ along the x and y directions respectively. The finer aspects of the approach can be found in [18].

To carry forward this ideology to color images is challenging. It can be approached in several ways. Two such ways are (1) Output fusion method : obtain the edge maps individually for each component of color considering each of them as a grayscale image and integrate the obtained edge maps by following a principle (See Fig. 3.1), (2) Multi-dimensional gradient method (See Fig. 3.2): find the total edge magnitude (say $\boldsymbol{\lambda}$) and the corresponding direction (say $\boldsymbol{\theta}$), at each pixel location considering all the three components at a time and then find the edge map based on the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$. First approach deals with a straight forward extension of the grayscale edge detection techniques to color

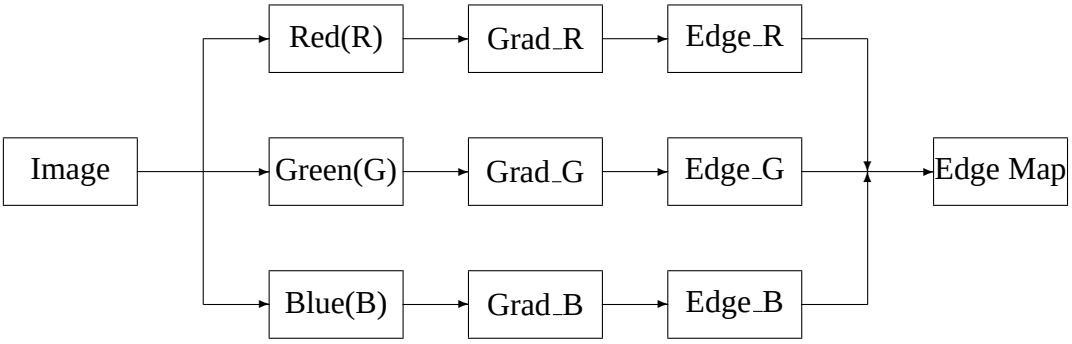


Figure 3.1: Output Fusion

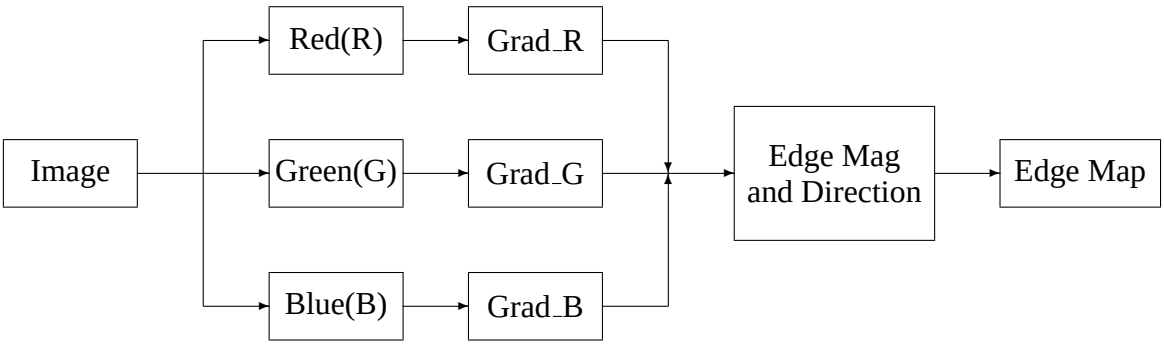


Figure 3.2: Multi-dimensional Gradient

images. This approach does not exploit the extra information contained in the image in the form of color vectors and their inter relationship. The second approach is an interesting extension of grayscale edge detection methods to color images. The challenges present in this approach is obtaining the total edge magnitude at a pixel combining the three components of color and the corresponding edge direction, and parameter tuning. A way of obtaining the combined edge magnitude is discussed below, which is extracted from the work by Cumani [23] and discussion is limited to color images instead of a general m-band image.

A color image can be represented as a function $\boldsymbol{f} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ that maps a point $\boldsymbol{P} = (\boldsymbol{x}, \boldsymbol{y})$ in the image plane to 3-dimensional vector $(\boldsymbol{R}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{y}))$. Thus an image value at a given pixel location is a vector in $\mathbb{R}^3$. Finding edge maps in images is to look for difference in the image values at two nearby points, say $\boldsymbol{P}$ and $\boldsymbol{Q}$, which is a 3-dimensional vector

$$\boldsymbol{\Delta f(P, Q) = f(Q) - f(P).}$$

When $\boldsymbol{Q - P}$ is an infinitesimal displacement $\boldsymbol{dP} = (\boldsymbol{d_x}, \boldsymbol{d_y})$, the above difference becomes the differential

$$\boldsymbol{df = f_x dx + f_y dy} \tag{3.1}$$

where $\boldsymbol{f_x}$ and $\boldsymbol{f_y}$ are the partial derivatives of $\boldsymbol{f}$ along x and y directions respectively.

The squared norm of $\boldsymbol{df}$ is given by

$$\boldsymbol{df = f_x \cdot f_x dx dx + 2 f_x \cdot f_y dy dx + f_y \cdot f_y dy dy} \tag{3.2}$$

where dot $(\cdot)$ indicates scalar product of vectors in $\mathbb{R}^3$.

It can be noted that for constant displacement of size $\|\boldsymbol{dP}\|$, $\boldsymbol{df^2}$ indicates how much the image value varies in the direction of $\boldsymbol{dP}$. Therefore, given a unit vector $\tilde{\boldsymbol{n}} = (\boldsymbol{n_x}, \boldsymbol{n_y})$ in $(\boldsymbol{x}, \boldsymbol{y})$ plane, the squared local contrast of $\boldsymbol{f}$ at $\boldsymbol{P}$ in the direction $\tilde{\boldsymbol{n}}$ can be defined as

$$\boldsymbol{S(P, \tilde{n}) = E n_x^2 + 2 F n_x n_y + G n_y^2} \tag{3.3}$$

where $\boldsymbol{E = f_x \cdot f_x}$, $\boldsymbol{F = f_x \cdot f_y}$ and $\boldsymbol{G = f_y \cdot f_y}$.

The extremal values of the quadratic form (3.2) coincide with the eigenvalues, represented by $\boldsymbol{\lambda_+}$ and $\boldsymbol{\lambda_-}$ of the 2×2 matrix

$$\begin{pmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{F} & \boldsymbol{G} \end{pmatrix}. \tag{3.4}$$

By elementary calculations the extreme values can be found as

$$\boldsymbol{\lambda_{\pm}} = \frac{\boldsymbol{E + G \pm \sqrt{(E-G)^2 + 4F^2}}}{\boldsymbol{2}} \tag{3.5}$$

and the corresponding eigenvectors are

$$\tilde{\boldsymbol{n}}_{\pm} = (\cos\boldsymbol{\theta}_{\pm}, \sin\boldsymbol{\theta}_{\pm}) \tag{3.6}$$

$$\boldsymbol{\theta}_{+} = \frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{tan}^{-1}\left(\frac{\boldsymbol{2F}}{\boldsymbol{E-G}}\right) + \boldsymbol{k\pi} \tag{3.7}$$

$$\boldsymbol{\theta}_{-} = \boldsymbol{\theta}_{+} \pm \frac{\boldsymbol{\pi}}{\boldsymbol{2}} \tag{3.8}$$

From the above derivation it is clear that the maximum contrast at a pixel $(\boldsymbol{x}, \boldsymbol{y})$ is given by the eigenvalue $\boldsymbol{\lambda_+}$ and the direction of maximum contrast is given by $\boldsymbol{\theta_+}$. Thus edge magnitude at a pixel can be considered as the value of $\boldsymbol{\lambda_+}$ along the direction of $\boldsymbol{\theta_+}$. The work by Cumani [23] can be referred for a general and detailed explanation of finding edge magnitudes and direction for m-band images.

In the proposed approach we have considered $\boldsymbol{\lambda_+}$ as the initial edge magnitude and $\boldsymbol{\theta_+}$ as the corresponding edge direction because of the following reasons. (1) Using this method combined edge magnitude and a single edge direction can be easily obtained, (2) it is supported by a strong mathematical reasoning for obtaining the edge magnitude and direction. However, other methods of obtaining edge magnitudes such as the root mean square [26] could have been used to obtain the initial edge magnitude. Here after the subscript '+' is dropped, and edge magnitude $\boldsymbol{\lambda_+}$ will be refereed as $\boldsymbol{\lambda}$ and direction $\boldsymbol{\theta_+}$ will be refereed as $\boldsymbol{\theta}$.

It is to be noted that although, the edge magnitude $\boldsymbol{\lambda}$ found using the partial derivatives of the three bands of the image is a good way of obtaining edge magnitude, it is not standardized. Hence, it becomes difficult to choose a stable set of parameter value for all the images. In the following section, this problem is discussed elaborately and a way of finding the standardized edge magnitude is proposed.

3.2.2 Standardization of Edge Magnitudes

The objective of an edge detection method is to determine and locate the pixels where there is abrupt change in image surface. In gray scale images it is determined by the variation in gray value of the pixels and in color images variation in color of the pixels is used. The nature of a pixel, whether it is an edge pixel or not, depends on its surrounding pixels. Hence, the edge magnitude found using the derivative vectors at each pixel should not be used straightway to test the eligibility of a pixel to be an edge pixel. The variation in the gray values or colors in the neighborhood of a pixel need to be incorporated to obtain the edge magnitude. Generally, the value of edge magnitudes vary from image to image based on the imaging condition. Hence, for different images, different threshold values are needed to obtain a good edge map. This task of selection of threshold values for each of the images becomes extremely difficult when the edge maps of the images are used as the input to a high level image processing or computer vision task such as object recognition or image retrieval, where thousands of images are to be processed. In such a context, an edge detection method, which is independent of parameter tuning becomes demanding. A solution to this problem is to obtain the standardized edge magnitude before thresholding. Hence a fixed set of parameter values would work for a wide variety of images. In order to standardize the edge magnitude, the variability of the gray values in the neighborhood of the pixel is successfully used by Rakesh *et al.* [110] for grayscale images. The same principle can also be used in the case of color images.

The proposed method basically follows the philosophy stated in [110] by Rakesh *et al.* and uses a different technique for finding the initial edge magnitude and the direction. Rakesh *et al.* have found the estimated image surface using Priestly-Chao [109] kernel smoother. Priestly-Chao kernel is used because, in statistics, it is a good estimator of a function when the independent variable is equally spaced which fits to the case of images as a function. Initial edge response and the direction of maximum contrast are found using the directional derivatives along x and y axes of the estimated image surface. Then non-maxima suppression [18] is performed on the initial edge response. The standardized

edge magnitudes at those pixels which are not suppressed by non-maxima suppression are found using the variability of the estimated image surface and the directional derivatives. In the end a two level thresholding is performed on the standardized edge magnitudes. In the proposed method Priestly-Chao kernel smoother is used to estimate the image surface in each component of the color image separately i.e.

$$f^c(x,y) = \frac{1}{\xi}\sum_{i=1}^{m}\sum_{j=1}^{n} K(x,i)K(y,j)\mathrm{I}^c(i,j), \tag{3.9}$$

$$\text{where,} \quad \xi = 2\pi mnh^2, \quad K(a,b) = \Psi\left(\frac{a-b}{h}\right),$$

$$\Psi(x) = e^{-x^2/2}, \quad c = R, G, B.$$

$\mathrm{I}^c(i,j)$ is the gray value at the $(i,j)^{th}$ pixel of the original image for the component c, h is the smoothing parameter, m and n are the number of rows and columns of the image respectively. From now onwards the super script indicates that the operation is done on the corresponding component (i.e, R or G or B) of the image.

The directional derivatives along x and y directions of the estimated image surface are

$$f_x^c(x,y) = \frac{1}{\xi}\sum_{j=1}^{n} K(y,j)\left\{\sum_{i=1}^{m} K'(x,i)\mathrm{I}^c(i,j)\right\}, \tag{3.10}$$

and

$$f_y^c(x,y) = \frac{1}{\xi}\sum_{i=1}^{m} K(x,i)\left\{\sum_{j=1}^{n} K'(y,j)\mathrm{I}^c(i,j)\right\}, \tag{3.11}$$

where $K'(x,i)$ and $K'(y,j)$ are the derivatives of $K(x,i)$ and $K(y,j)$ respectively.

The initial edge magnitude and corresponding direction are found as described in (3.5) and (3.7) using the partial derivatives f_x^c and f_y^c. After obtaining the initial edge magnitude non-maxima suppression and hysteresis are to be performed to obtain the final edge map [18]. In the proposed approach non-maxima suppression is applied on the initial edge magnitude λ to obtain the precise edge location. The problem occurs in thresholding at the time of hysteresis. The edge magnitude found is generally not uniform for both low intensity and high intensity regions in an image. Thus, the edge detectors are unable to

Algorithm 1 Edge_Detect(I), Input : I- RGB Image. Output : Edge - Binary Image

1: $h = 1.25$ {/* smoothing parameter or stiffness of gaussian */}
2: $T_u = 30$ {/* upper threshold */}
3: $T_l = 5$ {/* lower threshold */}
4: Find the directional derivatives f_x^R, f_y^R, f_x^G, f_y^G, f_x^B, and f_y^B from I as described in (3.10) and (3.11)
5: Find the initial edge magnitude (λ) and direction of maximum contrast (eigenvector corresponding to λ).
6: **for all** $(i, j) \in I$ **do**
7: apply non-maxima suppression (as described in Eq. [110]).
8: If it is not suppressed find S as in Eq. (3.12).
9: **if** $S > T_u$ **then**
10: Edge_Class(i, j) = 2 /* Edge pixel */
11: **else if** $S > T_l$ **then**
12: Edge_Class(i, j) = 1 {/* Edge pixel if any of its 8-neighbors is an edge pixel */}
13: **else**
14: Edge_Class(i, j) = 0 {/* Not an edge pixel */}
15: **end if**
16: **end for**
17: Initialize an array Edge(i, j) of the size of the input image to zero
18: **for all** $(i, j) \in I$ **do**
19: **if** Edge_Class(i, j) = 2 **then**
20: Track_Edge(i, j) {/* Track_Edge(i, j) is described in Algorithm 2, which is similar to the one described in [110] by Rakesh *et al.**/}
21: **end if**
22: **end for**

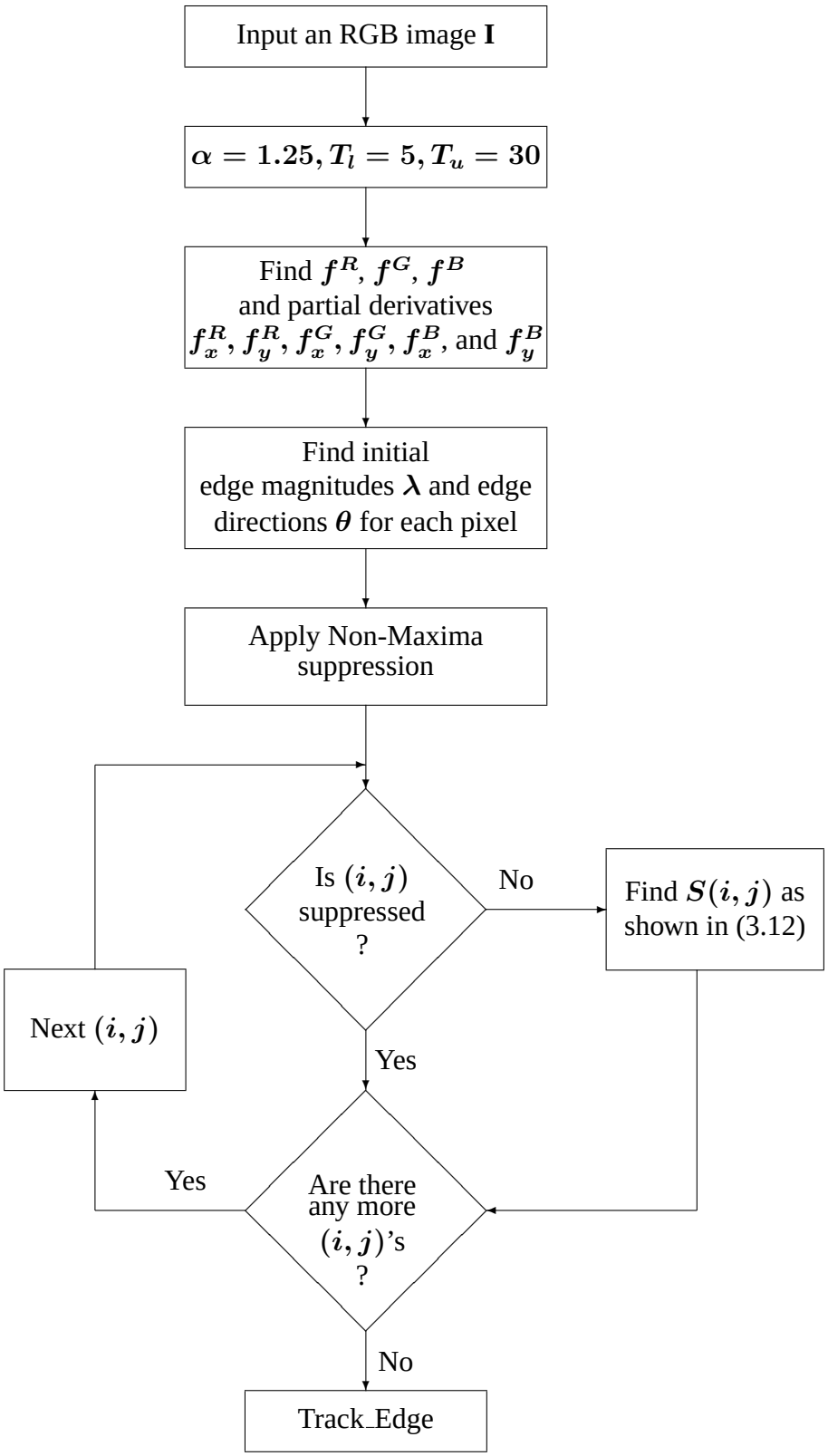


Figure 3.3: Diagram showing the main steps of the proposed edge detection method

Algorithm 2 Track_Edge(i, j)

```
1: if Edge(i, j) ≠ 1 then
2:    Edge(i, j) ← 1 {/* (i, j) is an edge pixel */}
3:    for all pixels (k, l) ∈ 8-neighbors of (i, j) do
4:       if Edge_Class(k, l) ≥ 1 then
5:          Track_Edge(k,l)
6:       end if
7:    end for
8: end if
```

extract, simultaneously, all the edges in an image having edges of different intensities. They can not extract all the edges in different images with a fixed set of threshold values. To get those edges, threshold values have to be tuned carefully for individual images. To overcome this problem a technique is proposed here to obtain the standardized edge magnitudes. The standardization of edge magnitude is done using the variability of the partial derivatives and estimated image surface.

It is to be noted here that in the context of obtaining a fixed set of threshold values for all the images, the quality of the edge map is not compromised by applying the non-maxima suppression on the non-standardized initial edge magnitude, because generally, thresholding is performed on the edge magnitude of those pixels which are not suppressed by the process of non-maxima suppression. Hence, standardization of edge magnitude is not required before non-maxima suppression. For those pixels, which are not suppressed by non-maxima suppression, a standardized edge magnitude is obtained for thresholding. The procedure for finding the standardized edge magnitude is discussed below :

Let us define the derivative vectors for each component of the smooth image $\boldsymbol{f}$ at a pixel $(\boldsymbol{x}, \boldsymbol{y})$ to be $\boldsymbol{v^c} = (\boldsymbol{f_x^c}, \boldsymbol{f_y^c})^{\boldsymbol{T}}$. Then the standardized edge magnitude at a pixel $(\boldsymbol{x}, \boldsymbol{y})$ is given by the statistic

$$\boldsymbol{S(x, y)} = \sum_{\boldsymbol{c=R,G,B}} \boldsymbol{(v^c)^T \left(\Sigma^c\right)^{-1} (v^c)}, \tag{3.12}$$

where

$$
\begin{aligned}
\Sigma^c &= \begin{pmatrix} \sigma_{11}^c(x,y) & \sigma_{12}^c(x,y) \\ \sigma_{12}^c(x,y) & \sigma_{22}^c(x,y) \end{pmatrix}, \\
\sigma_{11}^c(x,y) &= \left(\frac{\sigma^c}{\xi}\right)^2 \sum_{i=1}^{m}\sum_{j=1}^{n} K^2(y,j)K'^2(x,i), \\
\sigma_{22}^c(x,y) &= \left(\frac{\sigma^c}{\xi}\right)^2 \sum_{i=1}^{m}\sum_{j=1}^{n} K^2(x,i)K'^2(y,j), \\
\sigma_{12}^c(x,y) &= \left(\frac{\sigma^c}{\xi}\right)^2 \sum_{i=1}^{m}\sum_{j=1}^{n} K(x,i)K(y,j)K'(x,i)K'(y,j), \\
\text{and } (\sigma^c)^2 &= \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} (\mathrm{I}^c(i,j) - f^c(i,j))^2
\end{aligned}
$$

In the above expressions, c takes values R, G and B. $(\sigma^c)^2$ is the estimate of data variability of the estimated image surface $f^c(i,j)$, and $\sigma_{11}^c(x,y)$, $\sigma_{11}^c(x,y)$, $\sigma_{11}^c(x,y)$, $\sigma_{11}^c(x,y)$ are the co-variabilities.

Please note that the value $\sum_{c=R,G,B} (v^c)^T(v^c)$ can be considered as the edge magnitude at a pixel. Generally, in statistics, inverse of dispersion matrix is used for standardization. Mahalanobis [75] initially used it for obtaining distance between two populations and it is popularly known as Mahalanobis distance. It was also used by Rakesh *et al.* [110] in their article. We used the same here. Thus a simple way of standardization of magnitude is to consider the inverse of dispersion matrices Σ^c of respective colors and add the standardized magnitudes for each color, which has been done above. The proposed method use the value of $S(x,y)$ in (3.12) as the standardized edge response and is thresholded using two threshold values. Note that, it is not needed to estimate S for all the pixels in the image. S is estimated for those pixels which are not suppressed by non-maxima suppression.

Proposed edge detection method is described in an algorithmic form in Algorithm 1 and the flow of the steps is expressed in a block diagram in Fig. 3.3. The performance of the proposed method is compared to some of the existing methods and the process of

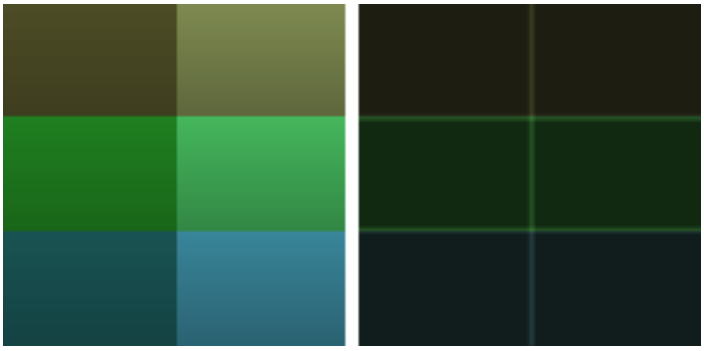


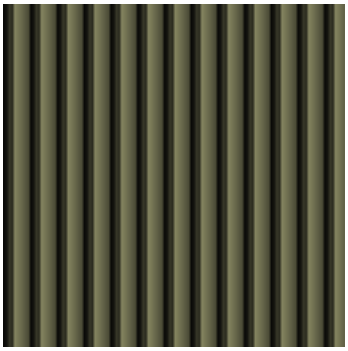
Figure 3.4: Two rectangles images, each of size 128×128 : Image on the left, say (a), has two horizontal edges in 43rd and 85th rows and one vertical edge in 66th column. Image on the right, say (b), contains 2 vertical edges in 64th and 66th columns and 4 other horizontal edges

obtaining the fixed set of parameter values is described in the following section.

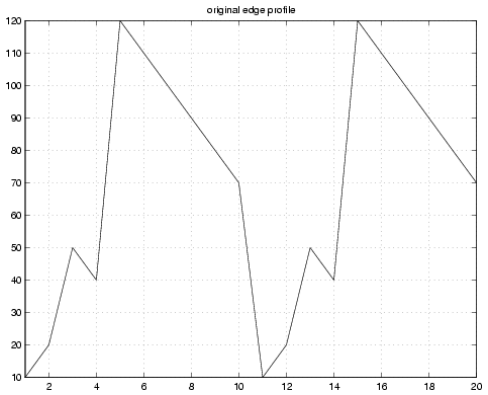
3.3 Results and Comparisons

A method has been proposed above to get standardized edge magnitudes at each pixel location and two level thresholding is performed on the standardized edge magnitudes to obtain the final edge map. This has been implemented on many artificially created images as well as natural color images. Four artificial images among these are two images shown in Fig. 3.4, vertical bars in Fig. 3.5(a) and circles in Fig. 3.6(a). The vertical bars image is constructed by varying the intensity along the rows with fixed values of saturation and hue so that the edge profile is only due to the intensity. As the gray values in this image are periodic with the period of 10 pixels, the edge profile of only the first twenty pixels of a row is shown in Fig. 3.5(b). A total of 200 images from the publicly available "Berkeley Segmentation Dataset and Benchmark" (BSDB) [1] [79] and six other images are considered to observe the performance of the proposed method. Out of these 206 images, 20 images are considered to obtain the threshold values for the proposed method. Three

[1] `http://www.cs.berkeley.edu/projects/vision/grouping/segbench/`



(a) Original vertical bars image



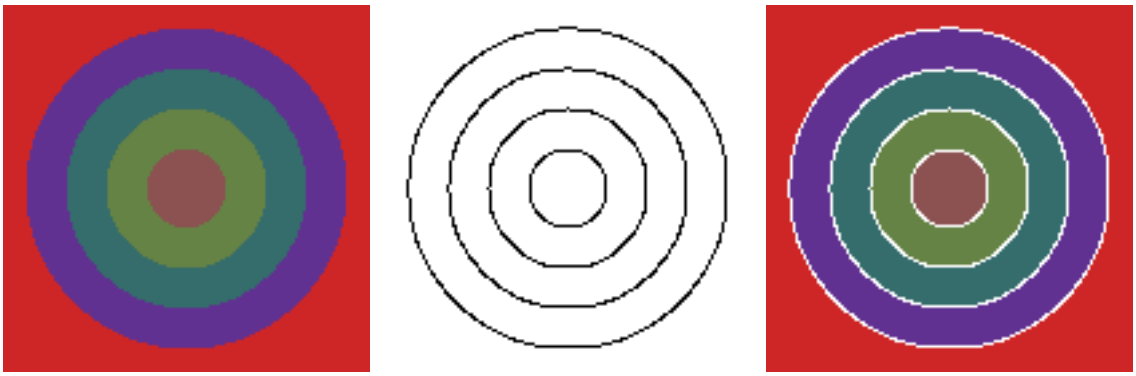
(b) Original edge profile of the first 20 pixels of a row of the image in Fig. 3.5(a)

Figure 3.5: Original vertical bars image and its edge profile

among the 20 images used for threshold analysis, namely Lenna [Fig. 3.8(a)], window [Fig. 3.9(a)] and balloon [Fig. 3.10(a)], are included here. As opposed to gray scale edge detection techniques, there is no standard edge detection technique, such as Canny's edge detection methods for gray scale, available in the literature for color edge detection. Here, we have considered two methods, which reported good results and produced single pixel width edge map, for comparing the over all performance of the proposed method. These two methods are (1) the work proposed by Cumani[2] [23], and (2) the work proposed by Ruzon[3] *et al.* [119, 120]. Cumani used second order image derivatives to locate edge pixels and Ruzon *et al.*used compass operator for determining edge magnitudes and edge direction. However, none of them used standardization of edge magnitudes.

[2]Code used for this method is taken from `http://www.ien.it/~cumani/mck04309.zip`

[3]Code used for this method is taken from his web page `http://robotics.stanford.edu/~ruzon/compass/`



(a) Original circles image

(b) Edge map obtained using proposed method with the fixed set of parameter values

(c) Edge map in Fig. 3.6(b) is super imposed on the original image

Figure 3.6: Results on circles image

3.3.1 Standardization of edge magnitude by proposed method

Proposed method obtains the standardized edge magnitude at the maxima of the initial edge response of the image. The variability of the directional derivatives of the estimated image surface is used to obtain the standardized edge magnitudes. To verify it in reality, the edge magnitudes of 42nd row of Fig. 3.4(a) and 64th column of Fig. 3.4(b) before and after the standardization are plotted in Fig. 3.7. From the plots of edge magnitudes in Fig. 3.7, it can be seen that the edge magnitudes are enhanced after standardization but the rate of enhancement is not uniform in both the cases. It can also be observed that the edge magnitudes before the standardization is in the range of 0 to 12 in one image and it is between 0 to 4 in the other image, whereas after standardization the edge magnitudes are enhanced and the corresponding values are in the range of 0 to 1400 for both the images. This shows that the procedure suggested here standardizes the edge magnitudes. This helps to obtain a stable set of parameter values for obtaining uniform acceptable results for a wide variety of images. If we observe the edge profile of Fig. 3.4(b) before standardization (shown in the right side column of Fig. 3.7), the edge magnitudes between

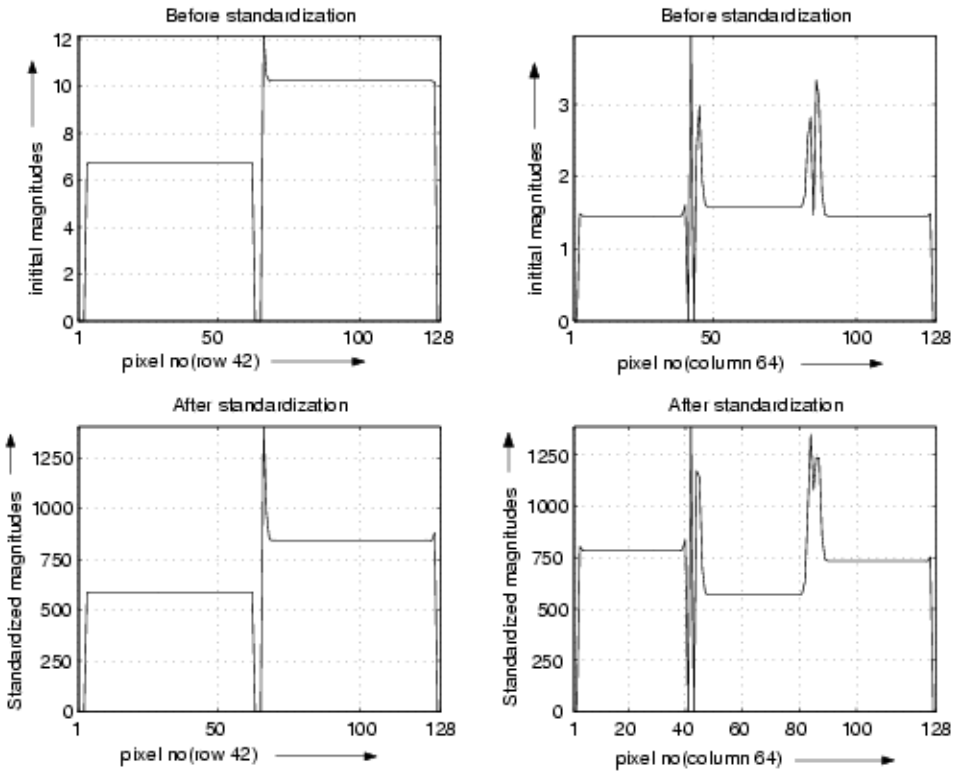


Figure 3.7: Plots shown in the left column are the edge magnitude plots corresponding to the horizontal edge at 42nd row of Fig 3.4(a) and plots shown in the right column are the edge magnitude plots corresponding to the vertical edge at 64th column of Fig. 3.4(b). The top and bottom rows of this figure show the edge magnitudes before and after the standardization respectively.

the two peaks are more than the edge magnitudes to the left of the left side peak and right of the right side peak. Actually, these edge magnitudes are due to the edge corresponding to the two green rectangles in the middle row of Fig. 3.4(b). But after standardization the middle edge magnitudes are less than both the right and left sides. This is happening because the variation of the color values in this region is more compared to the other two regions.

3.3.2 Analysis of parameters for the proposed method

The judgement of good performance on the obtained results is a matter of concern because there is no appropriate method to judge the quality with 100% confidence. Pratt's Figure of Merit (FOM), which is used for comparing two edge detectors, needs the knowledge of the ground truth of the edge map. It is a quantitative evaluation procedure. Due to the difficulty of obtaining the ground truth of the edge map for natural images, quantitative method of judgement is ruled out and a qualitative method of judgement by visual evaluation is adopted.

Criteria of selection for a set of parameter values

The criteria of selection of a set of parameter values for a given image is that the resultant edge map should satisfy the following :

- it should contain most of the prominent edges,
- it should not contain too many spurious edges and
- it should be visibly pleasing.

The selection of edge maps is based on a process of elimination among a set of edge maps obtained using different parameter values. The elimination of an edge map has been done visually.

Selection of parameter values

The proposed method takes $\boldsymbol{h}$ the stiffness of gaussian, and two threshold values $\boldsymbol{T_l}$ and $\boldsymbol{T_u}$ as input parameters. $\boldsymbol{T_l}$ and $\boldsymbol{T_u}$ are the lower and upper threshold values respectively. To find a set of parameter values, several edge maps are generated for each image by varying the parameter h between 0.8 and 2 with an increment of 0.05, $\boldsymbol{T_l}$ between 0 and 10 with an increment of 1 and $\boldsymbol{T_u}$ between 5 to 50 with an increment of 1 subject to the condition that $\boldsymbol{T_u}$ is always greater than $\boldsymbol{T_l}$.

There are several ways to obtain a fixed set of parameter values. A simple way is to examine the edge maps for a set of images and take that set of values which is producing acceptable edge maps for all the images in the set. When such an experiment is performed on the 20 considered images for threshold selection, the proposed method produced acceptable results with $\boldsymbol{h = 1.25}$, $\boldsymbol{T_l = 5}$, and $\boldsymbol{T_h = 30}$.

Some of the results using proposed method

Let us first judge the performance of the proposed method on the artificial images considered. The circles image is used to see that the edge detection method finds edges in all directions properly and in appropriate places. From the edge map of circles image [Fig. 3.6(b)], it can be seen that the proposed method has detected the boundaries of the discs properly in all the directions. It can be observed from Fig. 3.6(c), where the edge map in Fig. 3.6(b) is superimposed on the original image [Fig. 3.6(a)], that the edge pixels are detected in proper places. In the vertical bars image, ideally, there should be only two edge locations one at 4th pixel and another at 10th pixels [Fig. 3.5(b)] and there should not be any edge between 6th and 10th pixel because the transition of intensity between these pixels is very smooth compared to the transition in intensity between 4th and 5th pixels and 10th and 11th pixels. Proposed method has found the edges at these two locations i.e. at 4th and 10th pixels [Fig. 3.11(a)]. The results of applying the proposed method with the fixed set of parameter values ($\boldsymbol{h = 1.25}$, $\boldsymbol{T_l = 5}$, and $\boldsymbol{T_h = 30}$) on Lenna, window and balloon images are shown in Figs. 3.8(b), 3.9(b) and 3.10(b) respectively.

3.3.3 Analysis of parameters for Ruzon *et al.*'s method

Ruzon *et al.*'s method takes stiffness of gaussian $\boldsymbol{R}$, and two threshold values $\boldsymbol{low}$ and $\boldsymbol{high}$ as input parameters. For judging the performance of the algorithm, several edge maps are generated by varying $\boldsymbol{R}$ between 0.8 to 3 with an increment of 0.1, $\boldsymbol{low}$ between 0.0 to 0.60 with an increment of 0.01 and $\boldsymbol{high}$ between 0.1 to 0.60 with an increment of 0.1 subject to the condition that $\boldsymbol{high > low}$. Results on the circles image over a

wide range of parameter values are found to be appropriate, thus no figure is included here. When the vertical bars image [Fig. 3.5(a)] is considered, one extra edge is detected [Fig. 3.11(a)] between the 5th and the 10th pixels in reference to Fig. 3.5(b) for the parameter values $h = 1.5$, $0 \leq low < high$ and $0 \leq high < 0.46$. Ideally, there should not be any edge between these two pixels because the transition of intensity between these two pixels is very smooth which can be seen from Fig. 3.5(b). The correct edge map is obtained with $high = 0.46$ and above with $h = 1.5$ and $0 \leq low \leq high$. But results on balloon and window images with the same parameter values $h = 1.5$, $low = 0.1$ and $high = 0.46$ are found to be unsatisfactory since many prominent edges are missing [Figs. 3.12 and 3.13]. It is also found that when $high > 0.46$, some more edges are missing in window and balloon images. Thus, from the above observations, it is apparent that there does not exist a fixed set of parameter values in case of Ruzon *et al.*'s method providing acceptable results in all cases. Some more results are included in the Appendix A after applying these techniques on different images with different parameter values.

We made a subjective evaluation of the edge maps of the 20 natural images under consideration to obtain a fixed set of threshold values for Ruzon *et al.*'s method. It may be noted that a compromise is made in the quality of results for the selection of the parameter values. It is also to be noted that the selected set of parameter values can be different for different persons. The best fixed set of parameter values for Ruzon *et al.*'s method is found to be $R = 1.5$, $low = 0.1$ and $high = 0.3$.

3.3.4 Analysis of parameters for Cumani's method

Cumani's method takes stiffness of gaussian σ and a threshold value T as input parameters. For this method the edge maps with $0.8 \leq \sigma \leq 3$ with an increment of 0.1 and $0 \leq T \leq 50$ with an increment of 1 are observed to select the edge maps. Results on the artificial images over a wide range of parameter values using Cumani's method are found to be appropriate. Thus no edge map using this method on artificial images are included here. But results on many real life images by Cumani's method are either noisy or missing

important edges. This can be seen from the results on Lenna image [Fig. 3.8(c)]. There are many spurious edges present and at the same time many edges are disconnected. In particular the upper boundary of the hat is disconnected. To get these edges connected, $\boldsymbol{T}$ has to be decreased. With the decrease in the value of $\boldsymbol{T}$ more spurious edges will be introduced. When the threshold value is increased many important edges would be lost. Similar phenomenon is found with the window image [Fig. 3.9(c)] and balloon image [Fig. 3.10(c)]. Some more results in this regard are included in the appendix A in the attached CD.

3.4 Conclusions and Discussion

The proposed method along with the other two methods have been compared on several images. It is found that the proposed method consistently produces acceptable results for all the images. It is difficult to get a single fixed set of parameter values which would give a visibly pleasing edge map containing most of the prominent edges in the case of Ruzon *et al.*and Cumani's methods.

In the proposed method, the two fixed threshold values $\boldsymbol{T_l = 5}$ and $\boldsymbol{T_u = 30}$ have been used for detecting edges on more than 200 natural and artificial images. It is evident from the obtained results that the algorithm is producing results containing most of the important edges for all the images with these two threshold values. An intuitive reason for producing acceptable results is that it increases the edge response of all the detected edge pixels but not necessarily uniformly. This can be observed from two artificial images shown in Fig. 3.4 and from their edge profiles shown in Fig. 3.7.

In terms of computational complexity the proposed method is slower than Cumani's method but much faster than Ruzon *et al.*'s method which observed at the time of obtaining the edge maps from different images.

The main advantage of this algorithm is that the parameters do not need any fine tuning for getting acceptable results for an image. Thus, the proposed algorithm may

be handy for any computer vision task where extraction of edge maps is required for a large set of images for feature extraction or for any other work. In the above presented method, basically, a linear model with fixed effects is assumed. The performance of the edge detector may possibly be better if a generalized method is assumed considering the correlations between channels, though, the statistical analysis may be complicated. Probably, a similar experimentation for finding the performance of color edge detector, such as a work by Heath *et al.* [47], is needed to be done for an objective set of fixed threshold values.

(a) Lenna image (256 × 256)

(b) Proposed method with $h = 1.25$, $T_l = 5$, $T_u = 30$

(c) Cumani's method with $\sigma = 1.5, T = 15$

(d) Ruzon *et al.*'s method $R = 1.5$, $low = 0.1$, $high = 0.30$

Figure 3.8: Results on Lenna Image.

(a) Window image (256 × 256)

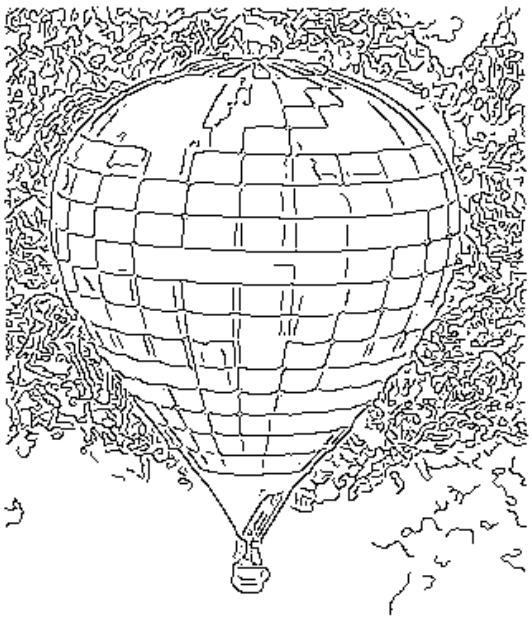
(b) Proposed method with $h = 1.25$, $T_l = 5$, $T_u = 30$

(c) Cumani's method with $\sigma = 1.5, T = 15$

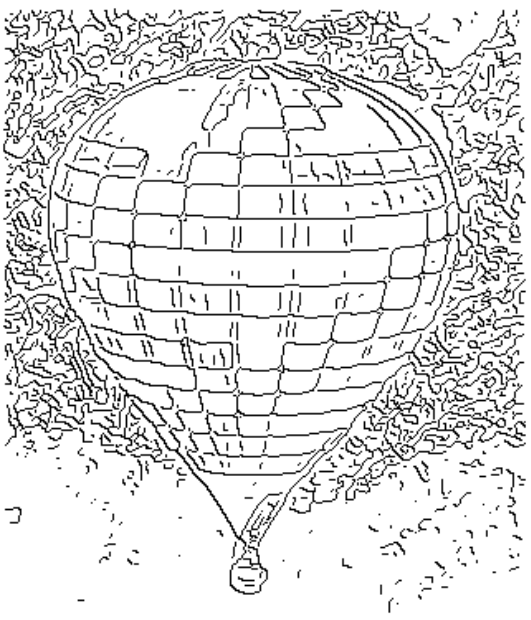
(d) Ruzon *et al.*'s method $R = 1.5$, $low = 0.1$, $high = 0.30$

Figure 3.9: Results on window image.

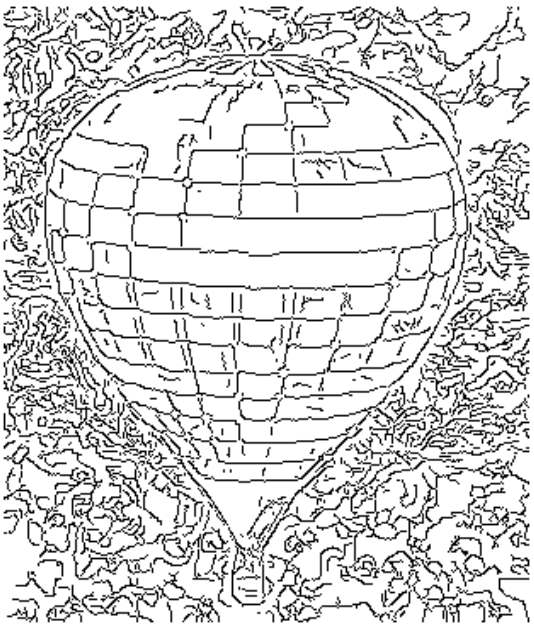
(a) Balloon image (320 × 373)



(b) Proposed method with $\boldsymbol{h = 1.25}$, $\boldsymbol{T_l = 5}$, $\boldsymbol{T_u = 30}$

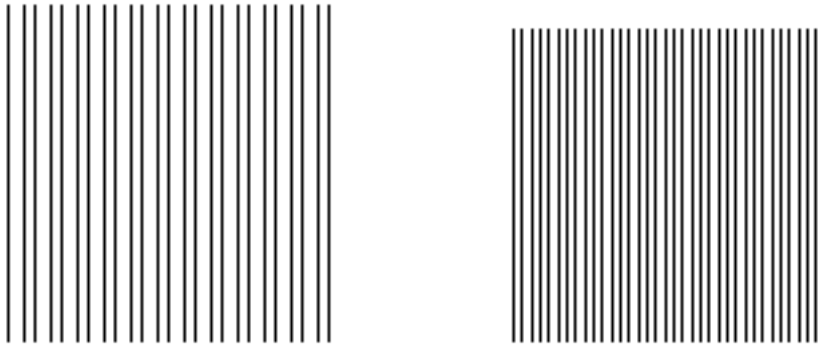


(c) Cumani's method with $\boldsymbol{\sigma = 1.5, T = 15}$



(d) Ruzon *et al.*'s method $\boldsymbol{R = 1.5}$, $\boldsymbol{low = 0.1}$, $\boldsymbol{high = 0.30}$

Figure 3.10: Results on balloon image.



(a) Proposed method with $h = 1.25$, $T_l = 5$ and $T_u = 30$

(b) Ruzon *et al.*'s method with $R = 1.5$, $0 < low < high$, $0 < high < 0.46$

Figure 3.11: Results on vertical bars image

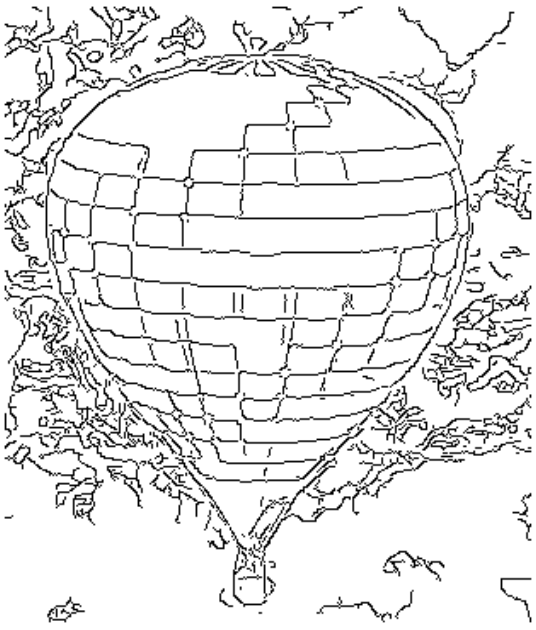


Figure 3.12: Result using Ruzon *et al.*'s method on balloon image shown in Fig. 3.10(a) with $R = 1.5, low = 0.1, high = 0.46$

Figure 3.13: Result using Ruzon *et al.*'s method on window image shown in Fig. 3.9(a) with $\boldsymbol{R = 1.5}$, $\boldsymbol{low = 0.1}$ and $\boldsymbol{high = 0.46}$

Chapter 4

Multi-Colored Object Descriptor

The challenges involved in object recognition are mainly the efficient representation and then the comparison of two objects through their representations. Broadly speaking, there are two types of approaches to object representation. While the first utilizes the knowledge gained from the spatial arrangements of the “shape features” such as the edge elements, boundaries, corners and junctions, the other uses the brightness or color features obtained more directly from the object images [8]. But, there are limitations to any algorithm which uses only either shape features or color features. There are many objects which are indistinguishable in terms of their shape [89]. For instance, if only shape features are considered then the recognition system can’t distinguish between a white car and a blue car when the cars are of the same model or their shapes are similar. Similarly, if only color features are considered, a blue bird may be wrongly matched with a blue car. Thus, there is a need of a scheme to describe an object which contains both shape and color information. In other words, the representation scheme should carry the color information and its pattern of appearance on the object surface. This study proposes a scheme to describe an object in such a way that the description contains the color information as well as the patterns of colors on the object surface. Note that, in most of the cases, wherever there is a shape or structural information in the object, the corresponding patterns in the image possess discontinuities in colors. Thus extraction of information regarding patterns

of colors automatically leads to extracting shape and structural information of the object. These obtained features with the proposed representation carry the pattern information that indirectly keeps certain shape information regarding the object.

4.1 Motivation for Proposed Method

Two important cues to distinguish between two objects are the overall shape and structure of the object, and the occurrence of different colors with respect to their spatial arrangements. Generally, human beings use both the cues for distinguishing objects in different stages. If we are interested to distinguish between a cow and an elephant, the shapes of these two objects are the main cues. To distinguish between two boxes of two different categories, the different colors appearing on the boxes become important. If the two boxes have more or less similar colors then their pattern of appearance or their spatial arrangements become more important. Thus, although it is difficult to imagine the actual shape of an object from the spatial arrangements of the different segments on its surface, it can be used as an important cue to represent the objects for classification. Several psychological studies regarding the representation of shape have been discussed in [78] and survey of literature in this regard is provided in [111]. It can be seen from object images shown in Fig. 5.1 (page 100) that objects of different categories can be distinguished from their colors and their spatial arrangements without explicitly comparing the corresponding shapes and structures, whether the comparison is between two boxes or between a cup and a box. Thus the relative positional information of the adjacent but differently colored segments must be preserved by the representation. A way of preserving the positional information of adjacent segments is to store their representing color vectors as a unit. This connected set which covers pixels from all the adjacent segments and contains the color information from these segments is the region of interest. Let us call such a region as a "Multi-Colored Neighborhood (MCN)". Six examples of such MCNs are shown in Fig. 4.1.

An object representation namely Multi-modal Neighborhood Signature(MNS), sim-

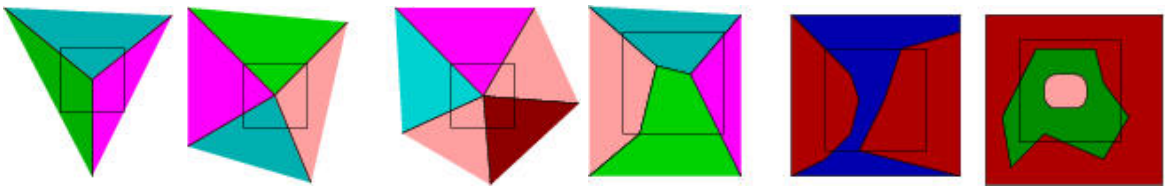


Figure 4.1: Examples of three types of junctions where multiple regions merge and three examples of presence of parts of different image segments in an image neighborhood. A rectangular window in each case shows the region of interest.

ilar to the proposed one, was carried out by Matas *et al.* [82]. Neighborhoods having multi-modal color distribution in RGB color space are located in the object image using a simplified mean-shift algorithm. Let the number of modes found in a neighborhood be $n(n \geq 2)$, and the set of modes be U=$\{\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_n\}$, where $\tilde{\mu}_i$ is three dimensional RGB vector of the i^{th} mode. Then $\binom{n}{2}$ color pairs $\{(\tilde{\mu}_i, \tilde{\mu}_j), i, j = 1, 2, \ldots, n$ and $i \neq j\}$ are formed from U. In this way all possible pairs of vectors are found from each multi-modal neighborhood of the image. Set of all such distinct pairs of vectors $(\tilde{\mu}_i, \tilde{\mu}_j)$ are defined as the MNS of the object. MNS contains the color information in the object. It also preserves the adjacency information of the colors on an object more or less. Each pair of vectors $(\tilde{\mu}_i, \tilde{\mu}_j)$ in the MNS contains the information that there is a region in the image where two segments of colors $\tilde{\mu}_i$ and $\tilde{\mu}_j$ are adjacent. However, this signature can not specify whether there are only two such adjacent segments or more than two adjacent segments in a region (Fig. 4.1). Thus, the neighborhoods having more than two-modal color distributions are not represented efficiently. Thus it lacks the crucial discrimination power regarding the neighborhoods having more than two segments. The discrimination power of the signature can be greatly enhanced if the properties of such type of neighborhoods could be preserved efficiently by the signature. This has been the motivation to propose one such representation which is capable of representing the neighborhoods having more than two modes. The proposed representation is described below.

4.2 Object Representation

An MCN is represented here as a unit consisting of the centers of the clusters. This unit of cluster centers contains the average color values corresponding to the different segments of MCN. Ultimately, the object is represented by the distinct sets of units of the cluster centers of the constituent MCNs. Let us call it as the "Multi-Colored Region Descriptor(M-CORD)" of the object.

4.2.1 Multi-Colored Region Descriptor

The objective of the M-CORD is to provide an adequately informative, discriminative and concise representation of the object. It should be discriminative enough to provide good classification and should be concise enough to save memory required to store.

The color values found from the cluster centers of an MCN are stored as a unit for each MCN, thereby keeping track of the structural information, especially when there are more than two clusters. Suppose $\boldsymbol{N}$ distinct MCNs are selected by the proposed algorithm. Then the signature of the object contains $\boldsymbol{N}$ units of cluster centers, and each unit of cluster center represents a single MCN. Let $\boldsymbol{k_i}$ be the number of clusters present in the $\boldsymbol{i^{th}}$ MCN. Then, the size of the signature in bytes with $\boldsymbol{N}$ number of MCNs and $\boldsymbol{k_i}$ clusters in $\boldsymbol{i^{th}}$ MCN is $\boldsymbol{2 + N + 3\sum_{i=1}^{N} k_i}$ assuming each mean vector takes 3 bytes of memory for its 3 color components.

This descriptor contains the information regarding each MCN of the image and the MCNs are either from the boundaries or junctions present in the image. Thus, it contains the information that if there is a unit of $\boldsymbol{k_i}$ clusters present in the descriptor then there is a patch of pixels which covers parts of $\boldsymbol{k_i}$ segments present in the image. This greatly enhances the discrimination power of the recognition system when same colors are present in two objects but in different alignments.

4.3 Detection of MCNs

The color distribution of each MCN is multi-modal [82]. Thus a clustering technique can be employed to find the number of colors present in a region. A simple and efficient clustering technique which detects the number of valid clusters present in a region is proposed here. Another way of detecting these regions is to find all the parts into which it is divided by the edge pixels present in the region. A special property of MCN is that it contains either a junction, or a part of boundary of the object, or simply an edge which divides the region into several parts. Each part of the region belongs to a different segment of the image. Thus edge maps of the images of the objects can also be used to locate such regions. These two approaches are explained separately below.

4.3.1 Detection of MCNs using Clustering

To obtain the different colors present in a neighborhood we propose a simple and fast clustering algorithm to find the cluster centers. It takes three parameters $\boldsymbol{V}$, $\boldsymbol{r}$ and $\boldsymbol{min\_clst\_size}$. Let $\boldsymbol{V} = \{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n\}$ be a set of color vectors. We call two color vectors $\tilde{\boldsymbol{v}}_i$ and $\tilde{\boldsymbol{v}}_j$ as similar if $\|\tilde{\boldsymbol{v}}_i - \tilde{\boldsymbol{v}}_j\| < \boldsymbol{r}$. Thus $\boldsymbol{r}$ is the dissimilarity parameter for two colors. $\boldsymbol{min\_clst\_size}$ is the parameter for checking the size of a cluster. A cluster is considered to be valid if its size is more than $\boldsymbol{min\_clst\_size}$. The steps of the algorithm are shown in Algorithm 3.

It can be seen that for a neighborhood with uniform color, this algorithm returns 1 cluster (step 9). It needs only $\boldsymbol{n-1}$ distance computations and $\boldsymbol{n-1}$ comparisons to detect a region with uniform color (steps 5-10) because in such a region, all the color vectors $\tilde{\boldsymbol{v}}_i$ are within a small disc of radius $\boldsymbol{r}$ centering $\boldsymbol{v_1}$. Hence, it eliminates such regions quickly which are of no interest to construct the descriptor. For the neighborhoods having more than one cluster, it needs $\frac{(n-1)n}{2}$ distance computations. However, the number of comparisons increases with the increase in the number of clusters. Let the set of vectors $\boldsymbol{V}$ be the union of $\boldsymbol{k}$ clusters $\boldsymbol{V_1, V_2, \cdots, V_k}$ i.e, $\boldsymbol{V = V_1 \cup V_2 \cup \cdots \cup V_k}$.

Algorithm 3 Cluster($V,\ r,\ min\_clst\_size$)

$c = 0, i_{max} = 1$ {/* $V = \{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n\}$ */}
while $n > min\_clst\_size$ **do**
  for all $i = 1 : n$ **do**
    if $c = 0$ **then**
      Find $d_{ji} = d_{ij} = \|\tilde{v}_i - \tilde{v}_j\| \; \forall \; j > i$
      $V_i = \{\tilde{v}_i\} \cup \{\tilde{v}_j \in V : d_{ij} < r \; \forall \; j \neq i\}$
      $U_i = V \setminus V_i$
      if $|U_i| < min\_clst\_size$ **then**
        return $\{c + 1\}$ {/* V is from a neighborhood of uniform color */}
      end if
    else
      $V_i = \{\tilde{v}_i\} \cup \{\tilde{v}_j \in V : d_{ij} < r \; \forall \; j \neq i\}$
      $U_i = V \setminus V_i$
    end if
    if $|V_i| > |V_{i_{max}}|$ **then**
      $i_{max} = i$
    end if
  end for
  if $|V_{i_{max}}| > min\_clst\_size$ **then**
    $c \leftarrow c + 1$
    $\tilde{\mu}_c = \dfrac{1}{|V_{i_{max}}|} \displaystyle\sum_{\tilde{v}_i \in V_{i_{max}}} \tilde{v}_i$
    $V = U_{i_{max}}, n = |U_{i_{max}}|$
  else
    return $\{c, \tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_c\}$
  end if
end while

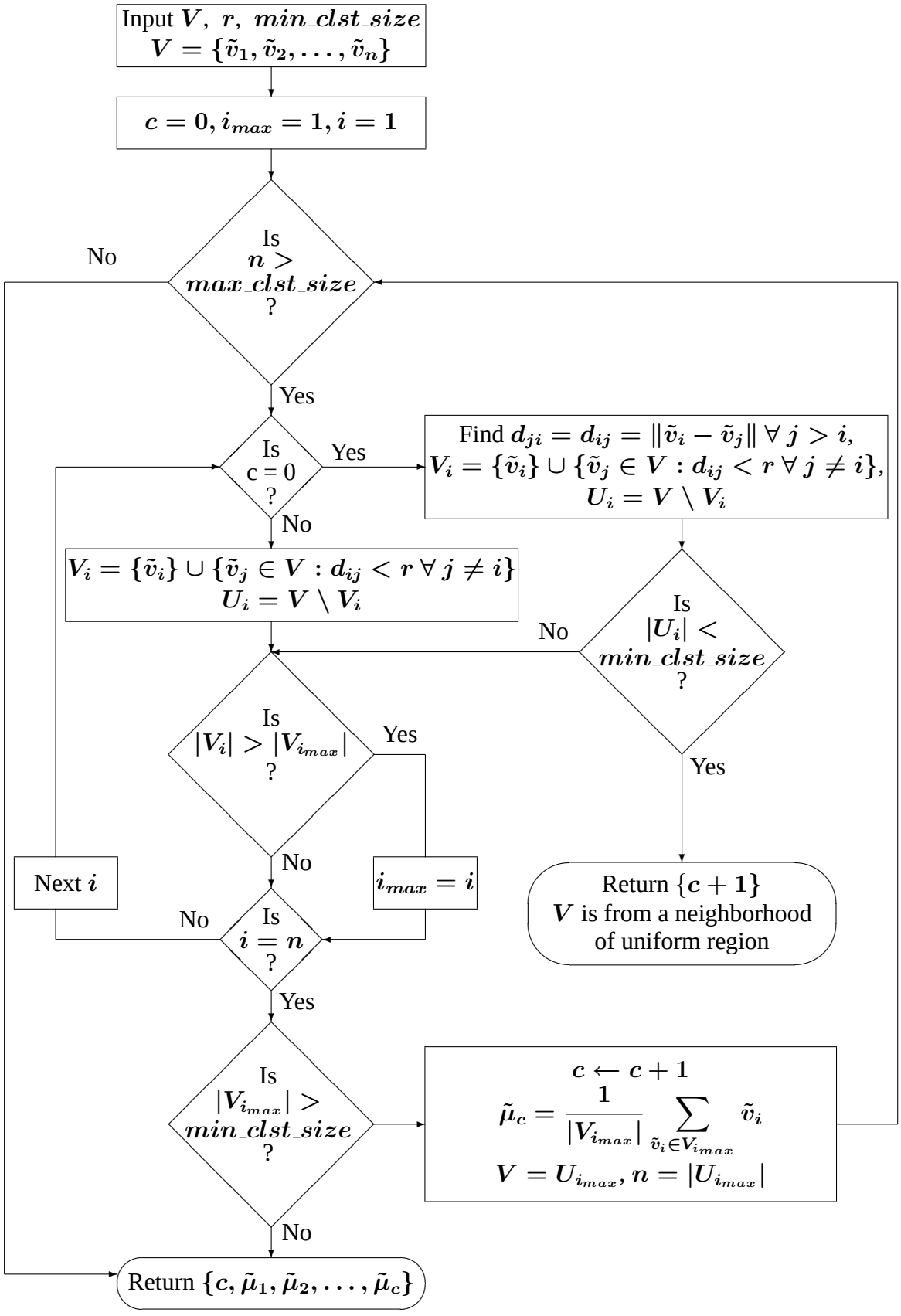


Figure 4.2: Diagram showing the main steps of the clustering algorithm used to detect multi-colored neighborhoods

Then the number of comparisons needed for the set of vectors $\boldsymbol{V}$ to be partitioned into $\boldsymbol{k} > \mathbf{1}$ clusters is $|\boldsymbol{V}|^{\mathbf{2}} + \sum_{\boldsymbol{j=2}}^{\boldsymbol{k}} \left(|\boldsymbol{V}| - \sum_{\boldsymbol{i=1}}^{\boldsymbol{j-1}} (|\boldsymbol{V_i}| - \mathbf{1}) \right)^{\mathbf{2}}$, ($|\boldsymbol{V_i}|$ denotes the number of elements in $\boldsymbol{V_i}$). Additionally, at most $\boldsymbol{n}$ vector additions and $\boldsymbol{k}$ divisions are needed for the computation of the cluster centers.

Philosophy behind Proposed Clustering Algorithm 3 : The proposed algorithm is mainly dependent on the value of the parameter $\boldsymbol{r}$. The idea behind selecting the value of $\boldsymbol{r}$ is that any color vector within a sphere of radius $\boldsymbol{r}$ centering a vector $\tilde{\boldsymbol{v}}$ are similar to $\tilde{\boldsymbol{v}}$ and two vectors withing a sphere should not look significantly different. Thus, when the region is a uniform region all the color vectors will be within the sphere of radius $\boldsymbol{r}$. Hence, that region can be declared as uniform region and the color information from this region is of limited interest to the representation scheme. When the number of color vectors in such a sphere is not significant (determined by the parameter $\boldsymbol{min\_clst\_size}$), we can safely say that the corresponding set of vectors are selected no where near the modes of the color distribution and these may be noise. Hence, these vectors can be simply ignored. If the number of vectors selected is significant and is maximum among all such spheres (note that a sphere of radius $\boldsymbol{r}$ is considered for each of the color vectors present in the region), it ensures that the selected set of vectors are from the most dense part of the color distribution. Hence, the average color value of these vectors is close to the modal value of the cluster formed by these color vectors. A pictorial representation of the clustering algorithm for a set of two dimensional vectors is given in Fig. 4.3. There are four types of vectors shown in the figure. It can be seen that there are three clusters present in the distribution and the fourth type (blue colored points) vectors are just distributed here and there. They do not form a cluster according a particular value of $\boldsymbol{r}$ chosen here.

To detect the M-CORD, $\boldsymbol{Cluster}()$ is performed at every considered neighborhood in the image. For simplicity of pixel manipulation, overlapping windows of size $\boldsymbol{w} \times \boldsymbol{w}$ are selected as neighborhoods. All the detected MCNs of an object image from SOIL-47A dataset are shown in Fig. 4.4(a). It can be seen that most of the edges are covered by number of MCNs. To construct the descriptor of the object, all the MCNs are not

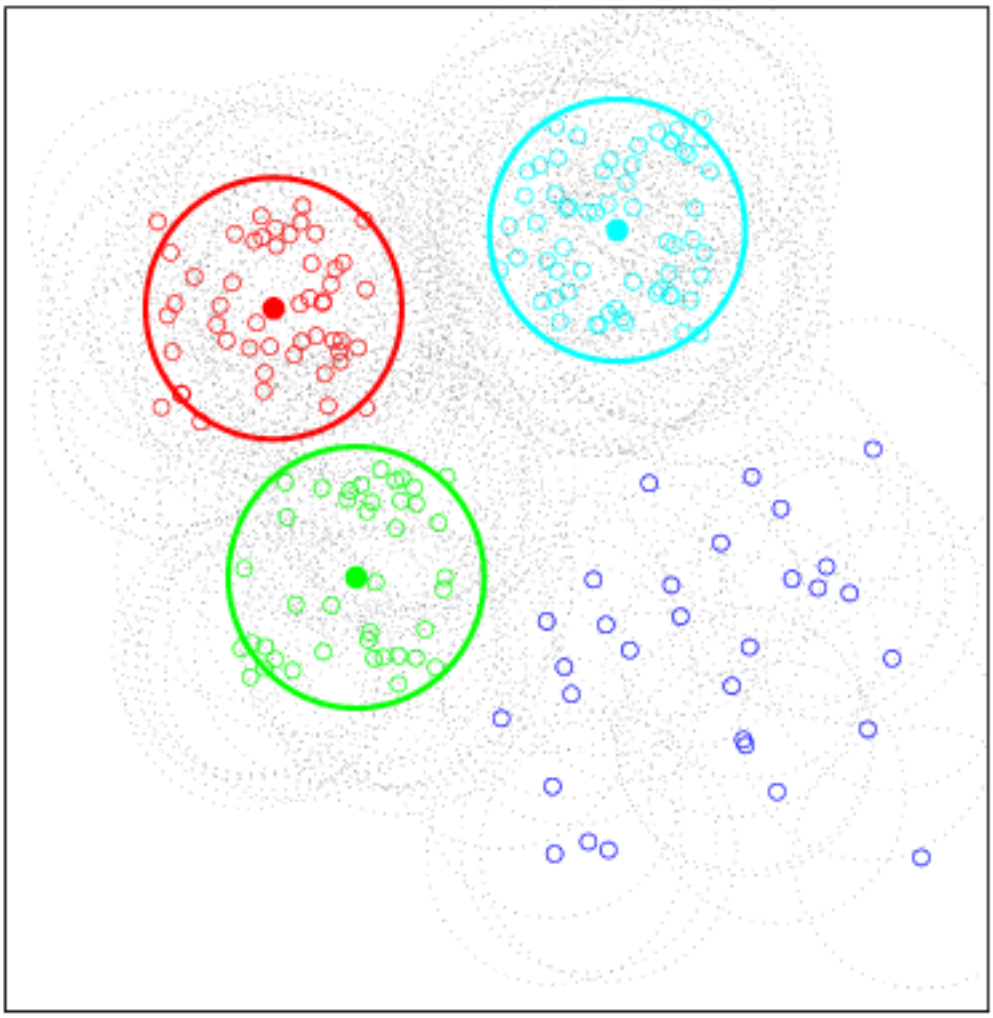


Figure 4.3: An example of clustering in 2-dimensional data set

needed because several MCNs detected over a stretch of boundary will have similar color distributions (See Fig. 4.4(a)). Thus, after finding an MCN, it is matched with all the previously considered MCNs and is included in the descriptor, if it is significantly different from all the previously considered MCNs. The amount of difference between two MCNs is determined by the dissimilarity value found using the Hausdorff distance (4.1) between the two sets of colors corresponding to two different MCNs that is explained in detail later in Section 4.4 of the article. In this way all the distinct MCNs are extracted from the object image to construct the M-CORD of the object (See Fig. 4.4(b)). Let us call this representation as M-CORD-Cluster. The matching algorithm for two different MCNs is described in Section 4.4. Finally, the M-CORD of each of the objects is stored in a separate file.

4.3.2 Detection of MCNs using Edge Map

Edge maps give crucial shape information of an object because connected edges are detected between every pair of adjacent segments. Thus every stretch of edge pixels gives information about two neighboring regions and any junction of more than two edges (as shown in Fig. 4.1) indicates the presence of multiple regions neighboring the surrounding point. The main problem in this approach is to use the "right" edge map for all the images. Most of the edge detectors fail to detect the correct edges with a fixed set of parameter values for all the images. Thus, it necessitates manual tuning of the parameters to obtain satisfactory results. This becomes an extremely difficult task while dealing with thousands of images.

A method of finding edge maps from color images is proposed in the previous chapter. This method is capable of finding uniformly acceptable results in all the images using a fixed set of parameter values. Hence, it is used here to find the edge maps of the object images with the default set of parameter values specified in the previous section. Square regions of size $w \times w$ are considered around the edge pixels. If the edges in a region divide it into disjoint smaller regions then the considered region is covering pixels from multiple

(a) All the MCNs present in the image (b) Selected MCNs

Figure 4.4: MCNs detected in obj39A of SOIL-47A dataset using clustering Algorithm 3.

image segments. In general, such regions are found over the boundaries where multiple image segments are present. If the number of connected components in the region is at least two, it is declared as an MCN. The average color values of each of the smaller regions in the MCN are found. Finally, all such distinct MCNs are clubbed together as described in Section 4.2.1 to construct the M-CORD of the object image and let us call it as M-CORD-Edge. Fig. 4.5 shows the MCNs detected using M-CORD-Edge in an object image. Each of the white rectangular windows is an MCN. Fig.4.5(a) shows only 10% of the MCNs among all the detected MCNs. Remaining 90% MCNs are not shown to avoid cluttering. All the distinct MCNs considered for the formation of the M-CORD of the

(a) Only 10% MCNs are shown among all the MCNs detected in the image to avoid cluttering

(b) Selected MCNs

Figure 4.5: MCNs detected in obj39A of SOIL-47A dataset using edge map of the image.

object are shown in Fig. 4.5(b).

Although, the idea of the representation using edge map of the object is same as the representation using clustering, the descriptors due to them are different because of the principles involved in them. For instance, in the third MCN from left in Fig. 4.1, M-CORD-Cluster will use 4 mean values whereas M-CORD-Edge will use 5 mean values to describe the neighborhood (the colors of the two smaller regions inside the MCN are same), because the edge map in this neighborhood divides it into 5 smaller regions due to five different segments of the image. In general, if there are n different colors present in the neighborhood then M-CORD-Cluster uses n means to represent the MCN irre-

spective of the spatial arrangements of the colors. But, in the case of M-CORD-Edge, if there are n types of color pixels and are divided into m (greater than n) smaller regions then the descriptor uses m mean values to represent the MCN. Thus, M-CORD-Edge representation is richer than M-CORD-Cluster.

4.4 Matching two MCNs of an Image

Matching between two MCNs of an image is performed to obtain the significantly different MCNs in the image. All the MCNs in an object image are detected either using clustering or the edge map of the object image as described in previous section. It can be observed from Fig. 4.4(a) that several MCNs are detected over a stretch of boundary of the object and most of them have similar color distribution. To represent the object, information from all of these MCNs generally is not needed. Only a few MCNs from a stretch of boundary having significantly different color distributions are enough for this purpose. Two MCNs are said to be significantly different if the dissimilarity between them is greater than a value δ_{max}. The dissimilarity between two MCNs, δ, is defined as follows.

Let $\boldsymbol{U} = \{\tilde{u}_1, \tilde{u}_2, \cdots, \tilde{u}_m\}$ and $\boldsymbol{V} = \{\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_n\}$ represent two different MCNs in an object image, where $\tilde{u}_i = (u_i^1, u_i^2, u_i^3)$ and $\tilde{v}_j = (v_j^1, v_j^2, v_j^3)$ are 3-dimensional color vectors from $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively. Then the dissimilarity between $\boldsymbol{U}$ and $\boldsymbol{V}$ is defined as

$$\delta = \max\left(\max_{\tilde{u}_i \in U}\{\min_{\tilde{v}_j \in V}\{\|\tilde{u}_i - \tilde{v}_j\|\}\}, \max_{\tilde{v}_j \in V}\{\min_{\tilde{u}_i \in U}\{\|\tilde{u}_i - \tilde{v}_j\|\}\}\right), \tag{4.1}$$

where $\|\tilde{u}_i - \tilde{v}_j\| = \sqrt{(u_i^1 - v_j^1)^2 + (u_i^2 - v_j^2)^2 + (u_i^3 - v_j^3)^2}$. Note that, in order that $\boldsymbol{U}$ and $\boldsymbol{V}$ are similar, each element in each set should have a similar element in the other set. If there is an element which does not have a similar element in the other set then these two sets are not similar. The expression (4.1) is the Hausdorff distance [50] between $\boldsymbol{U}$ and $\boldsymbol{V}$.

4.5 Summary

In this chapter we have proposed a novel object representation scheme M-CORD. It takes advantage of the shape features hidden in the color structure present in a local neighborhood. Two different methods are proposed to locate image neighborhood containing multiple colors using two different approaches. In the first approach we employed a clustering algorithm which efficiently determines the number of representing colors present in a neighborhood. The colors appearing in each neighborhood are stored as a unit to form the signature of the object images. Thus it keeps the structural information of the neighborhood which we call here as the shape feature of the neighborhood. Proposed clustering algorithm is computationally efficient but the no. of computations increases with an increase in the number of color vectors in a region or an increase in the size of the window. An alternative method to locate the multi-colored neighborhoods and the corresponding colors is proposed using the edge map information from that neighborhood.

Chapter 5

Object Recognition using Multi-Colored Region Descriptors

In the previous chapter the representation of an object using M-CORD has been described. In this chapter we shall describe a mechanism for comparing two objects. Performance of the proposed M-CORD representations of color images have been tested in the context of Object Recognition in this chapter. Subsequently, the proposed Object Recognition methodologies are compared with the existing recognition methodologies and it has been observed that the proposed methodologies have superior performance with respect to recognition rates. For those datasets where the images have low contrast, we applied the proposed contrast enhancement principle stated in *Chapter 2* before applying M-CORD. The results using contrast enhancement have also been found to be satisfactory on a recently created ALOI-VIEW dataset. Mainly there are two steps in the proposed object recognition methodology. First step is to obtain a suitable as well as a stable representation of the object images under consideration. The two methodologies M-CORD-Cluster and M-CORD-Edge described in the previous chapter are used for object representation. In step two, a similarity/dissimilarity measure is needed to make a comparison between two images through there representations. Following section describes a method to obtain a dissimilarity value between two object images through their M-CORD representation.

5.1 Matching two Objects

Matching two objects is performed by comparing their M-CORDs. Under ideal conditions, if the images under consideration for comparison are from the same object and same view then the procedures described in Section 4.3 should produce identical MCNs. But, in practice, when the images are taken under different conditions (i.e, different lighting conditions or from different views) the MCNs are not identical even if the two images are of the same object. Thus, the matching is not exact as it is in the case of matching two MCNs from same object image. Here, the matching operation assigns a score i.e., a dissimilarity value, based on the dissimilarity between the MCNs from the M-CORDs of the images under consideration.

Let $\boldsymbol{P = \{U_1, U_2, \cdots, U_M\}}$ $\boldsymbol{and}$ $\boldsymbol{Q = \{V_1, V_2, \cdots, V_N\}}$ be two M-CORDs from two different images. Here, $\boldsymbol{U_i = \{\tilde{u}_{i1}, \tilde{u}_{i2}, \cdots, \tilde{u}_{i\alpha_i}\}}$ and $\boldsymbol{V_j = \{\tilde{v}_{j1}, \tilde{v}_{j2}, \cdots, \tilde{v}_{j\beta_j}\}}$ are MCNs of $\boldsymbol{P}$ and $\boldsymbol{Q}$ respectively and, $\boldsymbol{\tilde{u}_{ik}}$ and $\boldsymbol{\tilde{v}_{jk}}$ are 3-dimensional color vectors. It can be seen that the number of elements in $\boldsymbol{P}$ and $\boldsymbol{Q}$ may not be same. Wallraven *et al.* [143] have proposed a distance function to find the distance between two feature vectors of different sizes which is used in a SVM classifier for image classification. The entities of their feature vectors are vectors of same size. But, in our case the entities $\boldsymbol{U_i}$ and $\boldsymbol{V_j}$ of the feature vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ respectively are also sets and they can have different number of elements. Thus, two dissimilarity measures are proposed, one is between two MCNs (5.1) and the other is between two M-CORDs (5.2).

$$\boldsymbol{\delta_{MCN}(U_i, V_j) = \frac{1}{\alpha_i}\sum_{k=1}^{\alpha_i}\min_{\tilde{v}_{jl}\in V_j}\{\|\tilde{u}_{ik} - \tilde{v}_{jl}\|\} + \frac{1}{\beta_j}\sum_{l=1}^{\beta_j}\min_{\tilde{u}_{ik}\in U_i}\{\|\tilde{u}_{ik} - \tilde{v}_{jl}\|\}} \quad (5.1)$$

$$\boldsymbol{\delta_{M-CORD}(P, Q) = \frac{1}{M}\sum_{i=1}^{M}\min_{V_j\in Q}\{\delta_{MCN}(U_i, V_j)\} + \frac{1}{N}\sum_{j=1}^{N}\min_{U_i\in P}\{\delta_{MCN}(U_i, V_j)\}} \quad (5.2)$$

The difference between the proposed dissimilarity measure and Wallraven's measure is that the proposed dissimilarity measure finds a dissimilarity value between two sets possessing different number of elements where each element is a set having different number of color vectors. Whereas, the measure proposed by Wallraven et al. [143] finds similarity between two sets possessing different number of elements where each element is a vector of fixed dimension.

The most similar MCNs $\boldsymbol{V_j} \in \boldsymbol{Q}$ for each $\boldsymbol{U_i} \in \boldsymbol{P}$ are found and the dissimilarity between these pairs of most similar MCNs are used to find the dissimilarity between two M-CORDs. The dissimilarity between a pair of MCNs ($\boldsymbol{U_i}$,$\boldsymbol{V_j}$) is found by finding the most similar $\tilde{\boldsymbol{v}}_{\boldsymbol{jl}} \in \boldsymbol{V_j}$ for each $\tilde{\boldsymbol{u}}_{\boldsymbol{ik}} \in \boldsymbol{U_i}$. It is to be noted that while the most similar color vector to $\tilde{\boldsymbol{u}}_{\boldsymbol{ik}} \in \boldsymbol{U_i}$ is some $\tilde{\boldsymbol{v}}_{\boldsymbol{jl}} \in \boldsymbol{V_j}$, the most similar color vector to $\tilde{\boldsymbol{v}}_{\boldsymbol{jl}} \in \boldsymbol{V_j}$ may not be $\tilde{\boldsymbol{u}}_{\boldsymbol{ik}} \in \boldsymbol{U_i}$. This implies that distance of $\boldsymbol{U_i}$ from $\boldsymbol{V_j}$ may not be same as the distance of $\boldsymbol{V_j}$ from $\boldsymbol{U_i}$. Hence, the distance is not symmetric. Thus, the summation of the average of the minimum distances of $\tilde{\boldsymbol{u}}_{\boldsymbol{ik}} \in \boldsymbol{U_i}$ to $\boldsymbol{V_j}$ and the average of the minimum distances of $\tilde{\boldsymbol{v}}_{\boldsymbol{lk}} \in \boldsymbol{V_j}$ to $\boldsymbol{U_i}$ is taken as the dissimilarity between $\boldsymbol{U_i}$ and $\boldsymbol{V_j}$. This is written mathematically in (5.1). Similarly, the distance between two M-CORDs is also asymmetric. Thus, the summation of the average of the minimum distances of $\boldsymbol{U_i} \in \boldsymbol{P}$ to $\boldsymbol{Q}$ and the average of the minimum distances of $\boldsymbol{V_j} \in \boldsymbol{Q}$ to $\boldsymbol{P}$ is taken as the dissimilarity between $\boldsymbol{P}$ and $\boldsymbol{Q}$.

5.2 Object Image Datasets

The performance of the proposed methods has been evaluated on two well-known datasets namely, Columbia University Object Image Library (COIL-100) [100] and Surrey Object Image Library (SOIL-47), corresponding to object recognition problem. In addition to these two datasets we have reported the experiments on a subset of Amsterdam Library of Object Images (ALOI) recently created by Geusebroek *et al.* [34]. Brief descriptions about each of these datasets are provided below.

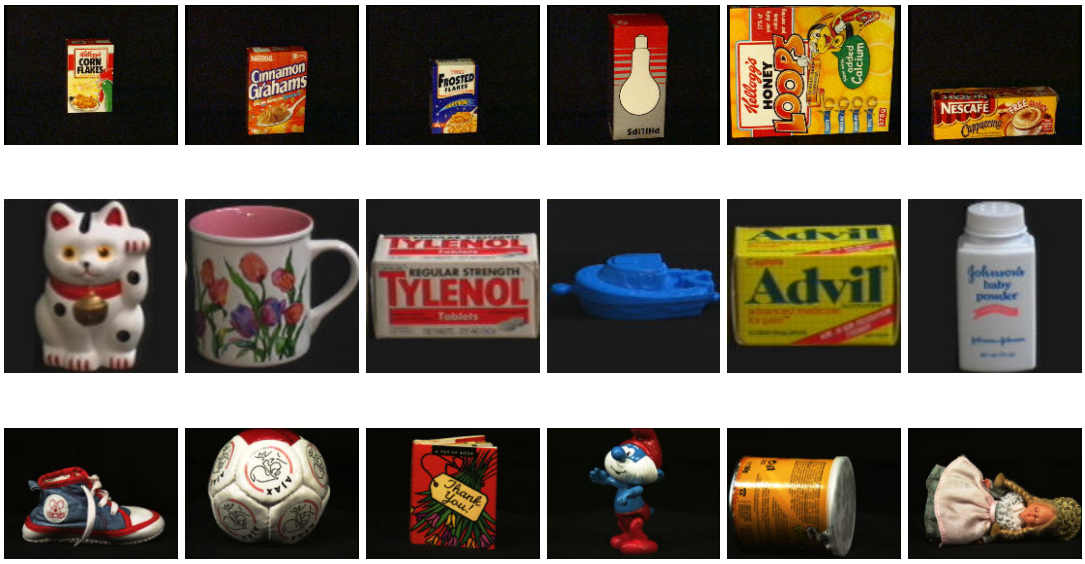


Figure 5.1: Rows 1 and 2 show the frontal view of eight objects from SOIL-47A and SOIL-47B datasets respectively. Row 3 shows frontal views of eight COIL-100 objects. Row 4 : shows the frontal views of eight ALOI-VIEW object images.

5.2.1 Surrey Object Image Library (SOIL-47)

The Surrey Object Image Library (SOIL-47)[1] [15] dataset is a collection of images of 47 different household objects from different view angles. There are 21 exposures of each of the 47 objects with 21 different view angles ranging from -90 to +90°. These 21 images are acquired by a robot arm moving around the object at intervals of approximately 9°. The size of each image is 288 $\times$ 360 except the frontal view i.e. the exposure of the object with 0° view angle. The images of the frontal view are of the size 576 $\times$ 720. This dataset is called SOIL-47A. Some of the objects of SOIL-47A are shown in Fig. 5.1. Another set of images namely, SOIL-47B is the collection of images of the same objects and same view angles as in SOIL-47A but under a different illumination set up.

[1] `http://www.ee.surrey.ac.uk/EE/VSSP/demos/colour/soil47/`

5.2.2 Columbia University Object Image Library (COIL-100)

The third dataset we have considered is the Columbia University Object Image Library (COIL-100)[2] [100]. COIL-100 is a dataset consisting of 100 different objects. There are 72 exposures of each object taken on a motorized turntable and images were captured by a fixed camera at pose interval of 5°. The size of each object image is 128 $\times$ 128. The COIL-100 has been widely used in object recognition. Some of the objects of COIL-100 are shown in the third row Fig. 5.1.

5.2.3 Amsterdam Library of Object Images (ALOI)

Amsterdam Library of Object Images (ALOI) [34] is a dataset consisting of images of different objects taken from different view points and illumination. It is recently created and introduced by Geusebroek *et al.* [34]. ALOI is a collection of images of 1000 different objects. These images are classified into four groups of image collections, each designed to a specific field of computer vision. The technical details of the experimental setup used to capture the images are described in [34]. The image collection relevant for our purpose is the collection of 72,000 images of the 1000 objects under in-plane rotation aim to describe object view (ALOI-VIEW) [3] with a quarter resolution. In this collection each of the objects has 72 exposures taken in an azimuth interval of 5° varying from 0° to 355°. The size of each object image in quarter resolution is $\mathbf{192 \times 144}$ pixels. This collection is too big for us to handle given the time and computational resources. Thus we have selected only 18000 images from 250 objects from among the 1000 objects for evaluating the performance of our methods. We believe a collection of 18000 images from 250 different objects provides sufficient variety to test the performance of the proposed methods.

Our main purpose is to compare the results of the proposed methods with the other methods on SOIL-47A and COIL-100 datasets only. The comparisons are provided in the

[2] `http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php`
[3] `http://www.science.uva.nl/~mark/aloi/aloi_red4_view.tar`

next section. Additionally, results on a part of ALOI-VIEW dataset is presented to test the performance of the proposed methods on a larger dataset. However, no comparison is made with other methods using this dataset.

5.3 Results and Comparisons

The performance of Multi-Colored Region Descriptor(M-CORD) with the two different approaches (i.e, using edge map of the objects (M-CORD-Edge) and using clustering (M-CORD-Cluster)) has been compared with MNS method proposed by Matas *et al.* [82] using SOIL-47A dataset, and some of the recently published methods on COIL-100 dataset. To compare the performance of the methods on SOIL-47A dataset, the results reported in [61, 103] as well as the results obtained using our implementation of MNS method have been considered for comparison. Similarly, results reported in [76, 103, 116] have been considered for the comparison of the performance of the methods on COIL-100 dataset. Additionally, recognition performances on ALOI using proposed methods have been included for better judgment.

The proposed object recognition methodologies possess resemblance with the MNS method. Thus a detailed performance evaluation between two methods have been conducted using the SOIL-47A dataset. Then the performance of the proposed methods have been compared with several other recently proposed object recognition methods on another widely used dataset COIL-100.

5.3.1 Performance evaluation on SOIL dataset

Let us consider the SOIL-47A dataset for evaluation of performance of the methods. The experiments have been done in the same way as described by Koubaroulis *et al.* [61]. Two experiments are conducted by them. In the first experiment, each of the 20 available views for each of the objects is matched with the frontal view of each of the 47 objects of the SOIL-47A dataset and the rank of the correct match is recorded. In the second experiment

only 24 objects (i.e., objects 1 to 19 and 24 to 29 excluding object number 26) with planar surface are considered.

To have a better understanding, we implemented the MNS method and parameters are tuned to obtain better results. The parameters used for MNS method are size of the window (w), Mean Shift kernel width (K), minimum mode size (C), threshold for the distance between two 6-dimensional vectors to be called similar used at the time of suppression (d) and the threshold value used at the time of MNS matching (T). Irrespective of the parameter values used by Matas *et al.* [82] we have tuned the parameters independently for L_1 and L_2 metrics. The chosen parameter values, after several experiments, are $w = 16$, $K = 17.3$, $C = 20$, $d = 50$, and $T = 57$ for L_2 and 80 for L_1. The obtained results are reported in fourth and fifth columns of Table 5.1. It can be seen that both for L_1 and L_2 metrics, we obtained better results than the ones reported by the authors. But, they are still far behind the results using proposed methods for SOIL-47A dataset.

Table 5.1: Recognition performance on SOIL-47A

Average recognition with rank 1					
	M-CORD		MNS		
			Reported	Our Implementation	
View Angle	Edge	Cluster	from [61]	L_1 metric	L_2 metric
Average time per match (ms)†	49	16	–	54	58
± 90 deg.	87.98	87.02	52.5	54.89	57.45
± 60 deg.	95.74	95.03	52.5	56.21	59.93
± 45 deg.	95.95	95.10	–	57.02	61.06
± 20 deg.	95.74	93.08	55.3	57.98	60.63
Average recognition with rank 1-3					
±90 deg.	96.17	95.21	78.0	82.87	96.88
±60 deg.	100.00	100.00	78.7	84.22	97.62
±45 deg.	100.00	100.00	–	83.62	96.67
±20 deg.	100.00	100.00	78.7	82.98	85.63

– indicates corresponding entry is not available

In the first experiment, it can be seen from Table 5.1 that the proposed methods

Table 5.2: Recognition performance on SOIL-24A

Average recognition with rank 1					
	M-CORD		MNS		
			Reported	Our Implementation	
View Angle	Edge	Cluster	from [61]	L_1 metric	$L_2 metric$
± 90 deg.	92.92	89.17	71.0	81.04	82.50
± 60 deg.	100.00	100.00	71.0	90.28	92.36
± 45 deg.	100.00	100.00	–	91.25	92.92
± 20 deg.	100.00	100.00	74.6	90.63	92.71
Average recognition with rank 1-3					
±90 deg.	97.92	96.25	85.6	97.08	82.50
±60 deg.	100.00	100.00	88.2	99.65	99.65
±45 deg.	100.00	100.00	–	99.58	99.58
±20 deg.	100.00	100.00	90.6	98.96	100.00

– indicates corresponding entry is not available

with the approach using Clustering (M-CORD-Cluster) and Edge Map (M-CORD-Edge), considerably out performed MNS based method. M-CORD-Edge has achieved average 87.98% and M-CORD-Cluster has achieved 87.02% accurate recognition with rank 1 while considering all the view angles from -90° to +90° as the test views. The recognition performance of both the approaches gradually increases when the test views considered are closer to the frontal view. If we include the rank 2 and rank 3 to the rank 1 recognition, both the methods achieved perfect (100%) recognition rate with the closer view angles onwards ±60°. M-CORD-Edge achieved 96.17% and M-CORD-Cluster achieved 95.21% recognition rate with all the views as test views.

In the second experiment i.e., only on 24 objects, M-CORD-Edge and M-CORD-Cluster have achieved 92.92% and 89.17% recognition rates respectively. The best result obtained for MNS method is 82.50% using L_2 metric, which is still far behind the performance of the proposed methods. Proposed methods achieved perfect (100%) recognition rate when views differing upto $\pm 60°$ from the frontal views are considered. Recognition rates touch 97.92% and 96.25% respectively for M-CORD-Edge and M-CORD-Cluster

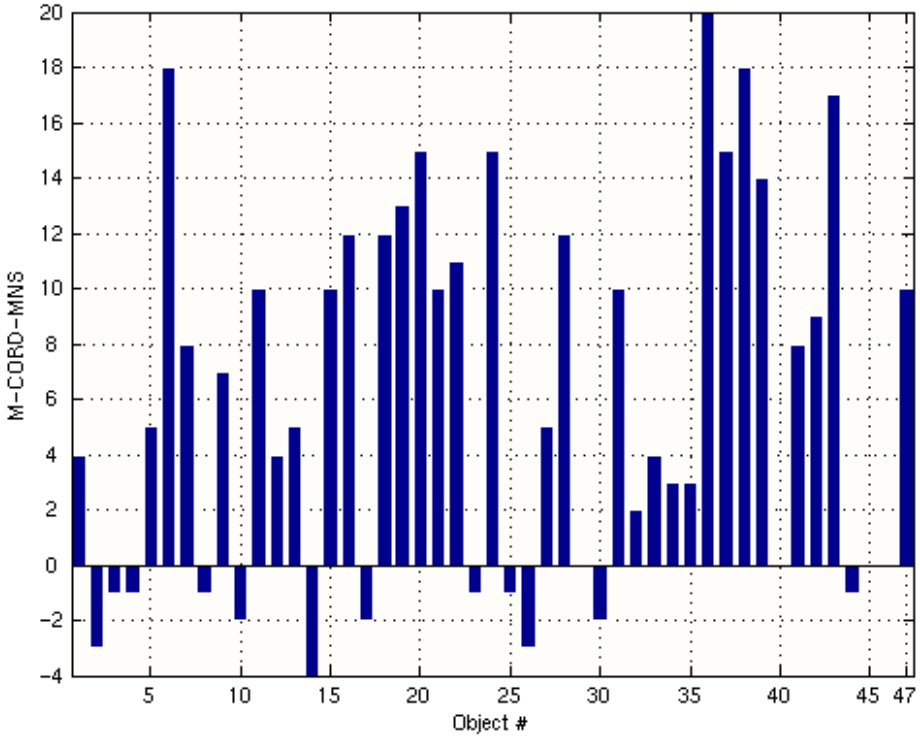


Figure 5.2: Improvement in object wise correct matches by M-CORD over MNS

when rank 2 and 3 are included and both the methods achieved 100% in the case of closer views differing upto $\pm\mathbf{60}°$. The best result on this dataset is reported by Obdrzalek *et al.* [103] and Kostin *et al.* [60] with 100% recognition rate using one training view per object and test view angles differing up to $\pm\mathbf{45}°$. Proposed methods also achieved 100% recognition with the same setup. Moreover, proposed methods achieved 100% recognition rate for the case of test views differing upto $\pm\mathbf{60}°$. The results of Obdrzalek *et al.*'s and Kostin *et al.*'s methods are not included in Table 5.1 and 5.2 since they have not mentioned their results for different view angles farther than $\pm$45°. They have however provided the results only for the test views ranging from -45° to +45°.

To better understand the advantage of M-CORD over MNS we looked for the objects for which the methods have performed poorly. As the performance of M-CORD-Edge and M-CORD-Cluster for SOIL-47A is similar, we have considered M-CORD-Edge for

comparison. To observe the object wise performance of the proposed method, no. of correct matches by MNS method for each object is subtracted from the no. of correct matches by the proposed method and the difference table is plotted in Fig. 5.2. It can be seen in Fig. 5.2 that M-CORD-Edge performed better than MNS for most of the objects. It can also be observed from Table 5.3 that none of the 20 views of three objects (# 20, 36 and 21) in SOIL-47A dataset are matched correctly by MNS method, whereas at least some views of every object are matched correctly by the proposed method. Only for object # 34 proposed method has performed poorly for which the correct no. of matches is just 4. For all other objects M-CORD-Edge has obtained at least 10 correct matches. Table 5.3 lists the number of mismatches by each object for both MNS and M-CORD-Edge, and, the object number of the mismatched objects. Proposed method mismatched object # 34 with object # 35 for sixteen test views. The two objects are very similar to each other and it is extremely difficult to distinguish one from other even for a human being. It is evident from the above mentioned results that M-CORD has superior performance over MNS.

For a better comparison between the two methods five more tables have been provided here. In Table 5.4 the number of correct matches for each object are listed in the ascending order of the object number. In Table 5.5, number of correct matches for each object is listed, and it is sorted with respect to the results obtained using MNS method. In Table 5.6, number of correct matches for each object is listed, and it is sorted with respect to the results obtained using M-CORD-Edge method. Tables 5.7 and 5.8 show the mismatched object numbers for each of the views for each of the objects of SOIL-47A dataset using MNS and M-CORD-Edge respectively.

In Table 5.6, it can be seen that proposed method failed to obtain satisfactory results only for three objects # 34, 23, 21 with number of mismatches 16, 10 and 10 respectively. Out of these three objects, MNS has performed marginally better than M-CORD for object # 23. These three objects along with the corresponding mismatched objects are shown in Fig. 5.3. If we consider row 1 and 2 of this figure it can be observed that the objects are visually similar to each other. The objects in row 3 are also similar. However, they are

Table 5.3: Object wise # of Mismatches and Corresponding List of Mismatched Objects†

Obj #	MNS		M-CORD-Edge	
	Total # of mismatches	List of Mismatched Objects ‡	Total # of mismatches	List of Mismatched Objects ‡
20	20	45(20)	5	19(4),45(1)
36	20	21(20)	0	–
21	20	45(10),47(10)	10	6(1), 7(1), 19(7), 24(1)
43	19	45(1),47(18)	2	46(2)
34	19	35(13),30(5),14(1)	16	35(16)
6	19	21(10),47(10)	1	41(1)
38	18	42(11),45(7)	0	-
16	17	13(1),20(13),24(3)	5	24(1),45(4)
19	16	13(2),20(13),45(1)	3	26(1),45(2)
24	15	43(1),45(14)	0	-
37	15	21(15)	0	-
39	14	47(14)	0	-
18	13	27(13)	1	28(1)
28	12	21(1),24(7),25(1),27(1),32(1),45(1)	0	-
11	12	21(9),27(2),40(1)	2	9(1),12(1)
15	12	1(1),13(1),20(4),21(5),39(1)	2	40(2)
22	11	27(2),40(4),44(5)	0	-
47	10	20(5),21(5)	0	-
31	10	42(10)	0	-
7	10	21(7),27(1),32(1),43(1)	2	27(1),40(1)
42	9	45(9)	0	-
9	9	21(9)	2	12(1),28(1)
23	9	3(3),34(5),40(1)	10	34(1),35(6),40(1)
41	8	21(8)	0	-
35	8	30(6),40(2)	5	34(5)
13	8	21(5),27(2),43(1)	3	46(3)
5	7	3(3),27(1),29(1),40(1),45(1)	0	-
1	7	13(6),20(1)	3	26(1),43(1),46(1)
8	6	7(1),21()4,30(1)	7	26(1),30(2),43(3), 46(1)
12	5	27(6)	1	40(1)
27	5	45(5)	0	-
33	4	9(2),21(2)	0	-
26	4	39(2),43(2)	7	46(7)
2	3	14(1),38(2)	6	3(5),38(1)
30	2	47(2)	4	40(4)
32	2	21(2)	0	-
17	1	4(1)	3	38(3)
10	1	14(1)	3	38(1),40(2)
14	0	-	4	23(1),38(1),40(2)
25	0	-	1	28(1)
44	0	-	1	40(1)
3	0	-	1	38(1)
4	0	-	1	38(1)

† The three objects # 29, 40 and 45 are not listed in the table because both the methods have correctly recognized them for all the 20 test views.

‡ The figures in brackets in column 3 (column 5) denote the no. of mismatches of the object in column 1 with the corresponding object no. shown in column 3 (column 5).
– All the views of the corresponding objects are correctly matched

Figure 5.3: Frontal views of three objects along with the corresponding mismatched objects for which M-CORD-Edge performed poorly. Row 1: object # 34 with the mismatched objects # 35, Row 2: object # 23 with the objects # 34, 35 and 40, and Row 3: object # 21 with mismatched objects # 6, 7, 19 and 24.

not as visually similar as the objects in row 1 & 2. It is difficult to state the exact reasons for this mismatch.

From Table 5.5, it can be seen that, the objects for which MNS failed most are # 20, 36, 21, 43, 34, 6, 38, 16, 19, 24, 37, 39, 18, 28, 11, 15, 22, 47, 31 and 7. MNS totally failed to obtain correct matches for the first three of them. These three objects along with the corresponding mismatched objects are shown in the Fig. 5.4. Figs. 5.5 and 5.4 clearly indicate the superior performance of the M-CORD method in comparison with MNS method. Note also that the similarity between object # 21 and objects # 6, 7, 19 and 24 in the case of M-CORD-Edge [See row 3 of Fig. 5.3] is visually more than the similarity between object # 21 and objects # 45 and 47 in the case of MNS [See row 3 of Fig. 5.4].

Table 5.4: A comparison of # of correct matches per object between MNS and M-CORD

Object #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# of correct Matches MNS	13	17	20	20	13	1	10	14	11	19	8	15	12	20	8	3
# of correct Matches M-CORD	17	14	19	19	18	19	18	13	18	17	18	19	17	16	18	15
Object #	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
# of correct Matches MNS	19	7	4	0	0	9	11	5	20	16	15	8	20	18	10	18
# of correct Matches M-CORD	17	19	17	15	10	20	10	20	19	13	20	20	20	16	20	20
Object #	33	34	35	36	37	38	39	40	41	42	43	44	45	36	47	
# of correct Matches MNS	16	1	12	0	5	2	6	20	12	11	1	20	20	20	10	
# of correct Matches M-CORD	20	4	15	20	20	20	20	20	20	20	18	19	20	20	20	

The different views in SOIL-47A data set are taken in the following way. The 0th view of the object is its frontal view. 11th view corresponds to 9° clockwise rotation of the object. 12th view corresponds to 18° clockwise rotation of it and so on. Thus 20th view corresponds to its 90° clockwise rotation. 10th view corresponds to -9° clockwise rotation, 9th view corresponds to -18° clockwise etc. Thus 1st view corresponds to -90° clockwise rotation. Only the 0th view of every object is in the training set. Thus, the dissimilarity is expected to increase as we move away from the frontal view. That is as we move from view 10 to view 1 and from view 11 to view 20, the dissimilarity is expected to increase. Hence, the recognition performance is expected to be poorer towards the boundary of the tables (Tables 5.7 and 5.8 given here). Table 5.7 (proposed method) shows this property but, Table 5.8 (MNS method), for a significant no. of rows, does not show it. This itself is a good indicator regarding the better performance of the proposed method.

5.3.2 Performance evaluation on COIL-100 datasets

COIL-100 is a widely used dataset for object recognition. We have considered eight different methods listed in Table 5.9 for comparison. Five different experiments have been conducted to evaluate the performance of the M-CORD methods and they are compared

Table 5.5: # of correct matches per object in the ascending order of # of correct matches obtained using MNS

Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches
20	0	37	5	42	11	27	15	44	20
36	0	39	6	9	11	33	16	3	20
21	0	18	7	23	11	26	16	45	20
43	1	28	8	41	12	2	17	40	20
34	1	11	8	35	12	30	18	46	20
6	1	15	8	13	12	32	18	25	20
38	2	22	9	5	13	17	19	14	20
16	3	47	10	1	13	10	19		
19	4	31	10	8	14	29	20		
24	5	7	10	12	15	4	20		

Table 5.6: # of correct matches per object in order of the ascending order of # of correct matches obtained using M-CORD

Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches	Obj. #	# of correct matches
34	4	30	16	43	18	39	20	22	20
23	10	17	17	7	18	31	20	36	20
21	10	13	17	25	19	40	20	45	20
8	13	10	17	44	19	33	20	27	20
26	13	1	17	12	19	41	20	46	20
2	14	19	17	6	19	28	20	38	20
35	15	15	18	3	19	42	20	47	20
20	15	5	18	4	19	37	20		
16	15	11	18	18	19	29	20		
14	16	9	18	24	20	32	20		

Table 5.7: Table describing the mismatched object for each view of each of the objects in SOIL-47A using MNS method (For example, from the 9th row we can say that the view no. 1 of object # 19 is mismatched with the frontal view of object # 13)†

Obj. #	# of mis-matches	View Numbers (1-20)																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20	20	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45
36	20	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
21	20	45	45	45	47	45	45	47	45	45	47	47	47	47	47	47	47	47	45	45	45
43	19	-	47	45	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47
34	19	35	30	30	30	35	35	30	35	-	35	35	35	35	35	35	35	35	35	30	14
6	19	21	21	21	21	21	47	47	47	-	47	47	47	47	47	47	21	47	21	21	21
38	18	42	42	45	45	45	42	42	42	45	45	42	45	42	42	45	42	42	42	-	-
16	17	24	24	20	20	20	20	-	-	-	20	20	20	20	20	20	20	20	20	13	24
19	16	13	-	-	20	-	20	20	45	-	20	20	20	20	20	20	20	20	20	13	20
24	15	43	-	-	-	45	45	45	45	45	45	45	45	45	45	45	45	45	45	-	-
37	15	-	21	-	21	21	-	-	21	-	21	21	21	21	21	21	21	21	21	21	21
39	14	-	-	47	47	47	47	47	47	47	47	47	47	47	47	47	47	-	-	-	-
18	13	27	27	27	27	27	27	27	-	-	-	27	-	-	-	-	27	27	27	27	27
28	12	24	24	24	27	-	-	-	-	-	21	45	-	-	24	24	24	-	25	24	32
11	12	27	-	-	27	21	-	21	-	-	21	21	21	21	21	-	21	21	-	-	40
15	12	39	1	-	-	-	-	-	21	-	21	20	21	21	20	20	20	-	-	13	21
22	11	44	40	44	40	44	-	44	-	-	-	-	-	-	-	44	40	27	27	-	40
47	10	21	21	20	20	21	20	-	-	-	20	-	-	-	-	-	20	-	21	-	21
31	10	-	-	-	42	42	-	42	42	-	42	42	42	42	42	-	42	-	-	-	-
7	10	43	21	-	21	-	-	-	-	-	-	-	-	21	21	-	21	21	32	21	27
42	9	-	-	-	-	-	-	45	45	-	45	45	45	45	45	45	45	-	-	-	-
9	9	-	-	-	21	-	-	-	-	-	21	21	21	21	21	21	21	21	-	-	-
23	9	40	3	-	-	-	-	34	-	-	-	-	34	34	34	-	34	-	3	-	3
41	8	-	-	-	-	21	-	-	-	-	21	21	-	21	21	21	-	21	-	21	-
35	8	30	30	30	-	30	-	-	-	-	30	30	-	-	-	-	-	-	-	40	40
13	8	43	-	21	-	21	-	-	-	-	-	21	-	21	21	-	-	-	-	27	27
5	7	40	3	3	27	-	-	-	-	-	-	45	29	-	-	-	-	-	-	-	3
1	7	13	20	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-	13	13	13
8	6	21	21	-	21	-	-	-	-	-	-	-	21	7	-	-	-	-	-	-	30
12	5	27	27	27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	27	27
27	5	-	-	-	-	-	-	45	45	45	45	45	-	-	-	-	-	-	-	-	-
33	4	9	9	-	-	-	-	-	-	-	-	-	-	-	21	-	-	-	-	-	21
26	4	39	-	39	-	-	-	-	-	-	-	-	-	-	-	-	-	-	43	-	43
2	3	38	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	14
30	2	-	-	-	-	-	-	-	-	-	-	-	-	-	47	-	47	-	-	-	-
32	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21	21
17	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4
10	1	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
46	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

† The figure in each of the boxes of row no. 1-47 and column no. 3-22 indicates the mismatched object number.

- indicates that the method has correctly recognized the object for that particular view.

Table 5.8: Table describing the mismatched object for each view of each of the objects in SOIL-47A using M-CORD method (For example, from the 2nd row we can say that the view no. 1 of object # 23 is mismatched with the frontal view of object # 40)†

Obj. #	# of mis-matches	View Numbers (1-20)																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
34	16	-	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	-	-	-
23	10	40	-	-	-	-	-	34	34	-	35	35	35	35	35	34	35	-	-	-	-
21	10	24	19	19	19	7	-	-	-	-	-	-	-	-	-	-	19	19	19	19	6
8	7	26	43	43	43	-	-	-	-	-	-	-	-	-	-	-	-	-	30	30	46
26	7	46	46	46	46	-	-	-	-	-	-	-	-	-	-	-	-	-	46	46	46
2	6	38	3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3	3
35	5	-	-	34	34	-	-	-	-	-	-	-	34	-	-	-	-	-	34	34	-
20	5	19	19	-	19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	19	45
16	5	45	45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24	45	45
14	4	40	40	23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38
30	4	40	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40	40
17	3	38	38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38
13	3	46	26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46
10	3	38	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40
1	3	46	43	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26
19	3	26	45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45
15	2	46	43	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	2	40	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	2	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
9	2	28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12
43	2	46	46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	2	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	27
25	1	28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40
12	1	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	1	41	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	1	38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	1	38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	28
24	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
33	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
41	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
42	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
46	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
47	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

† The figure in each of the boxes of row no. 1-47 and column no. 3-22 indicates the mismatched object number.

- indicates that the method has correctly recognized the object for that particular view.

Figure 5.4: Frontal views of those three objects along with the corresponding mismatched objects for which MNS method failed to correctly recognize even for a single view. Row 1: object # 20 with the mismatched object # 45. Row 2: object # 36 with the mismatched object # 21, and, Row 3: object # 21 with the mismatched objects # 45 and 47.

to other methods found in the literature for this dataset. The experiments are classified according to the number of training views considered for the experiments. All the test images are matched with training views and the ranks of the correct matches are found. The average values of rank 1 recognition are listed in Table 5.9. The results listed in Table 5.9 for other methods are taken from the cited papers except for the method Extra Tree + Random Sub-windows proposed by Maree *et al.* [77]. These recognition rates are obtained using the software PiXiT [4] provided by Maree and PEPITe. Maree *et al.* in their paper reported results using HSV coding. However, the results reported in row 3 of Table 5.9 are generated using RGB coding because proposed method too uses RGB

[4] `http://www.montefiore.ulg.ac.be/~maree/pixit.html`

values. It can be seen that the recognition performance increases with the increase in the number of training views of the objects. However, it is not always possible to have different training views available to obtain the model descriptor and the increase in the number of training views also increases the computational cost. Thus, a method should be judged better when it produces better results with less number of training views. Additionally, decreasing the number of training views increases demands on the method's generalization ability, and on the insensitivity to image deformations [103]. It can be seen from Table 5.9 that the rank 1 recognition rate obtained using proposed methods is better than other methods when one, two or four training views are considered. If eight training views are considered then the best result (99.40%) is reported for LAF [103] compared to 99.00% and 98.92% by M-CORD-Edge and M-CORD-Cluster respectively. In the case of 18 training views per object proposed method, M-CORD-Edge, achieved recognition rate (99.91%) which is equivalent to the best result reported in the literature for LAF. However, M-CORD-Cluster achieved slightly less percentage of recognition. When 36 training views are considered M-CORD-Cluster achieved 99.92% recognition which is slightly less compared to Sub-Windows [76]. But, proposed method M-CORD-Edge produces the best result with perfect (100%) recognition on this dataset. This is for the first time any method has obtained perfect recognition for COIL-100 dataset. Overall, M-CORD-Edge produces uniformly better results compared to other methods except for the case of eight training views per object and M-CORD-Cluster produces significantly better results compared to other methods when one, two or four training views are considered.

It is to be noted that, in the proposed approach no object modeling is done from the available training views to obtain a single M-CORD. The different training views of a particular object are treated as if they are separate object images. Also no particular method is used to select optimal training views for individual objects. Training views are selected in different regular intervals to have the knowledge of the object from all possible views. The different training views are selected as in [103] and [76], and are mentioned in Table 5.9.

Note : It is always a difficult task to implement one method exactly the way the authors have implemented their algorithm, especially when there is a fine tuning of parameters involved in the method. And, we believe authors are the best persons to tune the parameters involved in the algorithm. Hence, the results reported by the authors in their paper are taken for comparison and no attempt is made to implement their methods to replicate the results except in the case of MNS method. The results are not available for some of the table entries. However, we do not take unwarranted advantage to show the superior performance because for all such table entries for which results are available, proposed methods have performed better. Moreover, for instance, the results using one training view by proposed methods are better than the results using four and eight training views per object in the case of SNoW/intensity, Linear SVM and NN in Table 5.9. Here it is to be noted that the recognition rate increases with the increase in the number of training views. Performance of the proposed methods is similar or better than the LAF method when four or more number of training views are considered and for the case of 36 training views proposed M-CORD-Edge has obtained perfect recognition. Thus LAF can at most match the performance of the proposed method but can not better at least in the case of COIL-100. We implemented the MNS method [Section 5.3.1] and found that the results of our implementation are better than the results reported by the authors on SOIL-47A and SOIL-24A datasets. But, we have not got satisfactory results on COIL-100 dataset for the same method, probably due to poor tuning of parameters. It takes around 8 hours for obtaining results on COIL-100 dataset using MNS method, for one set of parameter values. Note that five parameters have to be tuned. The same parameter values used for SOIL-47A, provided poor results on COIL-100 dataset. Hence, no results are reported here using MNS method on COIL-100 dataset. It is difficult to do the tuning for all the methods and for all the data sets. Because of this difficulty with the other methods, we clearly specified the parameter values for the proposed methods.

Table 5.9: COIL-100: Rank 1 recognition performance

No. of Training Views/Obj.	36	18	8	4	2	1
No. of Training images†	3600	1800	800	400	200	100
Training Views in °	0+k10	0+k20	0+k45	45+k90	0,90	0
M-CORD-Edge	100	99.91	99.00	96.50	93.36	86.56
M-CORD-Cluster	99.92	99.87	98.64	96.46	92.74	86.93
Extra-Trees + Random Sub-Windows RGB [77]	99.86	99.50	97.67	92.43	88.36	79.58
LAFs [103]	–	99.90	99.40	94.70	87.80	76.00
Sub-windows [76]	99.94	99.61	98.47	95.06	88.00	75.17
Extra Trees [76]	99.67	97.96	92.45	87.64	75.09	63.90
SNoW/Edge [116]	–	94.13	89.23	88.28	–	–
SNoW/intensity [116]	–	92.31	85.13	81.46	–	–
Linear SVM [116]	–	91.30	84.80	78.50	–	–
NN [116]	–	87.50	79.50	74.60	–	–

† # of Test images in each case is 7200 - # of Training images

– indicates that results for the corresponding boxes are not available

5.3.3 Performance of the Proposed Method on ALOI

The last dataset considered is the ALOI-VIEW dataset. This is a huge dataset of 1000 objects and 72 different views for each of the object. Taking into consideration the computational time required to obtain the recognition performance for such a huge collection of images, only 250 objects (i.e, 18000 images) are selected from among 1000 objects (i.e., 72000 images). It has been observed that several object images in the dataset possess poor contrast. Additionally, many objects occupy very small area in the whole image. For some of the images a very few number of MCNs are selected due to this reason. Hence, a good representation of the object is not ensured. Thus, we have selected only 250 objects (i.e, 25% of the total no of images) based on the average number of MCNs selected for each object. These 250 objects are selected from the top of the sorted list of the 1000 objects based on the average number of MCNs. This ensures a good representation of the considered objects. Hence, it can also be assumed that the selected images possess good contrast to extract the desired features. Some of the images selected are shown in Fig. 5.1

Table 5.10: ALOI: Recognition Performance

# of Tr. Views/Obj.	8	4	2	1
Total # Tr. Images†	2000	1000	500	250
View Angles in °	0 + k45	45+k90	0, 90	0
Using M-CORD-Cluster				
Rank 1	98.89	95.28	86.67	75.15
Rank 2	99.67	97.55	90.46	81.10
Rank 3	99.84	98.12	92.27	83.48
Using M-CORD-Edge				
Rank 1	98.68	93.94	82.63	69.77
Rank 2	99.39	96.34	87.30	76.28
Rank 3	99.62	97.08	89.56	79.68

† # of Test images in each case is 18000 - # of Training images

and the recognition performance is summarized in Table 5.10. It can be seen that proposed method M-CORD-Edge and M-CORD-Cluster obtained moderately good recognition of 69.77% and 75.15% respectively, using one training view per object. Performance is not as good as in the case of COIL-100 dataset because the increase in the number of object classes increases the level of confusion between the objects and decreases the recognition performance. But, both the methods achieved good recognition when more number of views per objects are considered in the training set. This dataset is considered here to show the performance of the proposed method on a larger dataset.

5.4 Performance of M-CORD using Enhanced Images

In many databases, due to poor contrast, the difference between object pixels and background pixels may not be prominent. This leads to poor output from the object recognition methodology. In the ALOI-VIEW dataset there are several images which possess poor contrast between object and background. Some of the images are shown in Fig. 5.5.

Due to the poor contrast, the signature of several objects are not properly determined.

Table 5.11: List of the 250 objects from ALOI dataset used for the experiment

1	9	15	18	26	31	46	48	49	71	72	74
75	76	77	78	86	93	95	99	101	103	107	111
113	119	126	133	137	151	154	155	157	160	174	194
195	196	199	202	203	204	209	216	218	219	220	225
226	227	229	234	239	246	247	248	259	262	264	267
271	276	279	282	285	288	290	293	306	307	310	312
313	320	322	323	327	329	335	342	345	353	362	368
369	374	377	381	384	385	388	405	406	407	409	410
413	424	426	437	440	444	445	448	450	453	454	455
457	458	463	464	473	474	478	485	488	498	510	514
517	521	529	534	537	539	541	542	544	547	550	554
555	556	558	574	575	578	580	581	582	584	588	602
603	609	610	613	615	616	617	618	619	621	624	626
627	630	637	638	640	674	676	684	687	688	698	710
714	730	731	738	739	740	741	746	748	749	750	752
755	766	769	770	771	772	773	776	781	783	784	795
801	804	805	811	815	818	821	822	825	828	831	835
846	847	848	850	852	853	854	858	861	865	876	877
890	907	910	916	917	925	927	942	945	946	949	950
958	959	960	963	964	965	966	967	968	969	972	973
974	975	976	977	978	981	982	987	992	994		

Table 5.12: Average Time and Memory Utilization by Proposed Methods

M-CORD-Cluster					
	Window Size	# of MCNs	M-CORD size in bytes	Extraction time in ms	time taken per match in ms
SOIL-47A(View $\mathbf{0}°$)	20×20	371	4728	32700	16.9
SOIL-47A(View $\mathbf{0}°$)	10×10	113	851	1208	3.1
SOIL-47A(Other views)	10×10	45	338	241	–
COIL-100(All views)	16×16	37	346	999	1.2
COIL-100(All views)	10×10	18	135	94	0.2
M-CORD-Edge					
SOIL-47A(View $\mathbf{0}°$)	20×20	451	4781	5579	48.7
SOIL-47A(View $\mathbf{0}°$)	10×10	232	1849	2876	19.2
SOIL-47A(Other views)	10×10	79	1002	666	–
COIL-100(All views)	16×16	50	453	115	2.2
COIL-100(All views)	10×10	40	302	84	0.9

⁻ No comparison time is reported since comparisons are done only between the images of "View $\mathbf{0}°$" and "Other views" and no matching is performed between the images in "Other views" of SOIL-47A dataset.

Table 5.13: Performance of M-CORD-Edge on ALOI images with and with out Enhancement

# of Training Views/Object	36	18	8	4	2	1
Training View Angles in °	0 + k10	0 + k20	0 + k45	45+k90	0, 90	0
Average recognition without Enhancement						
rank 1	86.81	85.61	82.88	78.19	75.33	68.21
rank ≤ 2	87.69	87.04	85.25	80.99	79.47	74.49
rank ≤ 3	88.06	87.56	86.20	82.56	81.69	77.13
Average recognition with Enhancement						
rank 1	99.39	98.80	96.52	92.25	86.43	78.38
rank ≤ 2	99.86	99.63	98.38	95.79	91.31	86.34
rank ≤ 3	99.94	99.80	98.88	96.97	93.69	89.59

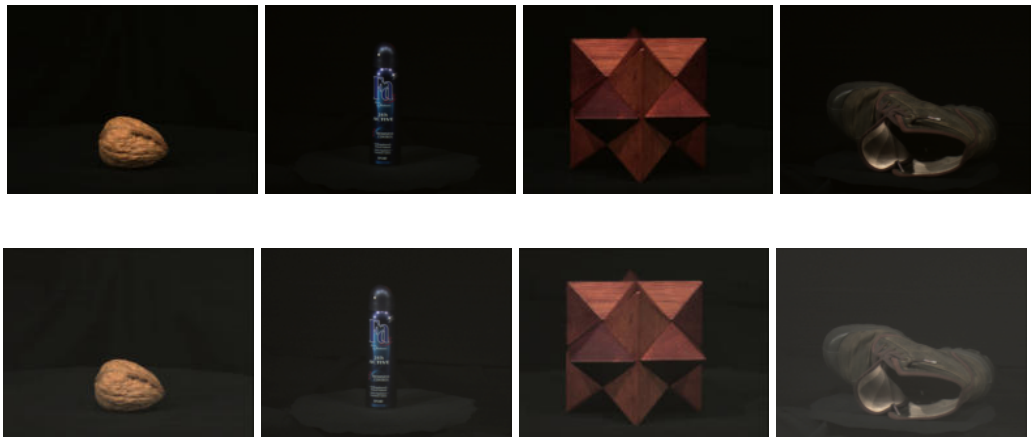


Figure 5.5: Row 1: Images possessing poor contrast from ALOI-VIEW dataset. Row 2: Corresponding enhanced images

The hue-preserving color image enhancement procedure described in *Chapter 2* is used to enhance the images in the database using a contrast enhancement function shown in Fig. 5.6.

To test the performance of the proposed representation scheme on enhanced images using the hue-preserving color image enhancement scheme proposed in *Chapter 2*, we have considered the M-CORD-Edge because, M-CORD-Edge produced good results on SOIL-47A and COIL-100 datasets. No experiment is conducted using using M-CORD-Cluster on enhanced images because in all other cases, both M-CORD-Edge and M-CORD-Cluster produced similar results. A collection of 100 images from ALOI-VIEW dataset is considered based on the number of MCNs in each object image. It is already seen in the cases of SOIL-47A and COIL-100 datasets that when the images possess good contrast, the proposed M-CORD-Edge produced a good representation. As our objective here is to test the performance of the proposed method on the object images with poor contrast, we have adopted a different procedure for selection compared to the case mentioned in the last section to select 250 objects from ALOI-VIEW dataset. Here, only 100 objects are chosen subjectively. We have randomly selected those 100 objects for which the number of MCNs for each of the images lies between 0 and max_mcns(say 30). It

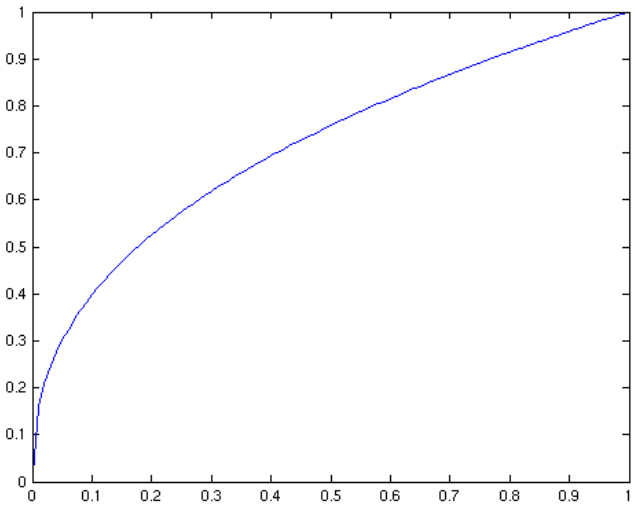


Figure 5.6: Enhancement function used to enhance the objects in ALOI-VIEW dataset with poor contrast. It is a function of type $\boldsymbol{f(x) = x^{\frac{1}{\gamma}}}$ with $\boldsymbol{\gamma = 2.5}$.

Table 5.14: List of the 100 objects from ALOI dataset used for the experiment

5	10	11	14	22	30	32	45	57	67
69	87	102	112	115	120	123	130	131	164
175	179	181	190	192	217	232	236	244	254
255	261	269	270	278	287	292	303	314	319
359	367	370	372	412	429	435	436	446	462
476	479	491	494	496	506	513	518	527	533
540	545	568	571	572	590	593	643	649	653
657	667	691	708	711	747	786	790	810	813
826	836	839	859	873	881	887	892	895	896
903	914	922	929	931	932	935	941	948	996

may be noted that if there is contrast in the image we shall have many MCNs. Thus we limited the no of MCNs so that the contrast in each of the 72 images of the selected objects is not high. Secondly, it is possible that, enhancement need not necessarily increase the no of MCNs. Thus the 100 objects are chosen in such a way that, on an average, the difference between no of MCNs before enhancement and no. of MCNs after enhancement is more than 5. Thus it was a tedious task to select 100 objects. The figure 100 has been considered from the point of view of ease in implementation. The 100 selected objects are listed in Table 5.4.

Application of proposed enhancement scheme along with the M-CORD-Edge descriptor and the object recognition scheme on the 100 selected objects provides better results. It can be observed from Table 5.13 that recognition performance has improved due to enhancement. However, it is not always necessary to use image enhancement before the construction of object descriptor. This should be used when the images in the dataset possess poor contrast.

It may be noted that the objects in COIL-100 and SOIL-47A possess good visual contrast. Hence, no contrast enhancement is needed for recognition. Thus, no experiments were conducted on them using the enhancement for object recognition. Our belief is that enhancement would work well for recognition if it is indeed needed. On the other hand, if enhancement is not needed, then the performance of the recognition methodology may deteriorate if an enhancement function is used in the processing stage.

5.5 Performance of M-CORD using Partially Occluded Images

Proposed methods of object recognition have been implemented on COIL-100 dataset to test the performance of the proposed representation schemes under partial occlusion. Occlusion of the objects has been simulated by removing one half of the object portion from the images before features are extracted. Some of the images are shown in Fig. 5.7.

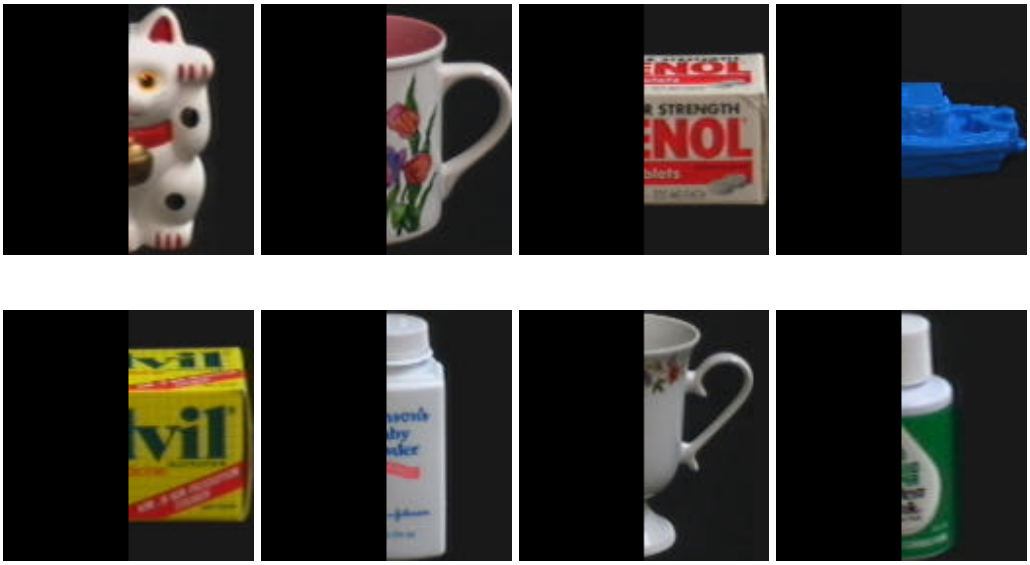


Figure 5.7: Example images from COIL-100 dataset. Half of the images is erased to create occlusion

Table 5.15: COIL-100 : Performance of M-CORD in partially occluded images

# of Training Views/Object	18	8	4	2	1
Training View Angles in °	0 + k20	0 + k45	45+k90	0, 90	0
M-CORD-Edge	88.80	85.89	82.65	73.39	68.10
M-CORD-Cluster	92.37	88.25	84.47	79.09	73.52
LAF [103]	92.6	89.1	82.6	69.9	63.3

The experiment is conducted as it is performed in the work by Obdrzalek *et al.* [103]. For the performance evaluation five different numbers of training views are considered. M-CORDs of the training images are obtained using the full size image without occlusion and all the M-CORDs for test images are obtained using the occluded images. The same set of parameter values, which are used to generate the results for the proposed methods in Table 5.9.

Rank 1 recognition rates are obtained by this experiment using proposed methods along with the results from LAF are listed in Table 5.15. It can be seen that the performance of the methods is affected by occlusion. The recognition performance is not as good as the results without occlusion. However, the result using proposed methods is much better than LAF when only one or two number of training images are considered. When the number of training images is increased performance of the proposed methods is similar to LAF. Comparison with only LAF is given since our experiment is similar to LAF [103].

5.6 Discussion

Performance of the proposed methodology depends mainly on the number of MCNs selected for each of the M-CORD and the size of the region selected. Here, region is a square window of size $w \times w$. The number of MCNs detected for each of the M-CORD depends on the size of the window and the dissimilarity threshold δ_{max}. While the size of the window is crucial for obtaining better regional description, δ_{max} controls the number of MCNs selected. The more is the value of δ_{max}, less is the number of MCNs selected. The bigger is the size of the window, better is the representation. Larger windows increase the computational cost of the method. Similarly, if too many MCNs are selected, the methods suffer from the problem of over fitting and it also increases the recognition time.

The values of all the parameters are selected on the basis of several experiments. Win-

dows of three different sizes ($\boldsymbol{w = 10, 16}$ and $\mathbf{25}$) are selected for experiments in COIL-100 dataset. Although, the results obtained are not significantly different, best results are obtained using $\boldsymbol{w = 16}$ and is reported in Table 5.9. It is also seen from Table 5.12 that the number of MCNs increases with the increase in the window size. The other parameter values such as $\boldsymbol{\delta_{max}}$-the dissimilarity parameter between two MCNs, $\boldsymbol{min\_clst\_size}$-the minimum number of pixels needed for a cluster to be valid and $\boldsymbol{r}$-the parameter to check the dissimilarity between two color vectors are selected by varying them over different intervals. The final values are selected by observing the MCNs selected on a number of images. The values of $\boldsymbol{r}$ and $\boldsymbol{min\_clst\_size}$ are varied between 10 and 60 and best results are obtained using $\boldsymbol{r = 30}$ and $\boldsymbol{min\_clst\_size = 20}$. Similarly, the value of $\boldsymbol{\delta_{max}}$ is varied between 20 and 80 and the best results are obtained using $\boldsymbol{\delta_{max} = 40}$.

The same parameter values are used for SOIL-47A dataset except for the window size, to generate the results shown in Table 5.1. Initially, performance of the proposed methods is tested on SOIL-47A using windows of size $\boldsymbol{w = 10}$ for both training and test views. In the SOIL-47A dataset, the size of the frontal(training) view of the object is twice the size of the other(test) views. Thus $\boldsymbol{w = 20}$ is a reasonable size to be selected for the frontal views when the window size for the test views is taken to be 10. This produces better results compared to the case of $\boldsymbol{w = 10}$ for all the views.

The size of an M-CORD increases with the increase in the number of MCNs, resulting in the increase in the memory required to store the M-CORD. Column 3 of Table 5.12 shows the figures for the memory used in bytes against the window size and number of MCNs.

Time taken by a method depends on how efficiently the method is implemented, the optimization criterion used, as well as the machine on which it is implemented. Proposed methods are implemented using C in a 900 MHz Sun-Blade workstation. The implementation of the proposed methods can not be considered as optimal. But some clues can be found regarding the running time for obtaining the M-CORDs and the average comparison time between two M-CORDs.

The time required for the computation of M-CORD is mainly dependent on the size of the window and the size of the image. It also depends on the content of the image. If the image is a complex and colorful texture image then the time taken to find the M-CORD is more compared to a simple image with minimum number of colors and without any texture. The effect of dissimilarity parameter is minimal. In case of M-CORD-Cluster, computation time increases significantly with the increase in the size of the window due to the clustering algorithm. But, the increase in the size of the window does not significantly increase the computation time in the case of M-CORD-Edge.

The time and memory utilization by the proposed methods including the number of MCNs per image are listed in Table 5.12 for two datasets. For SOIL-47A it can be seen that the average number of MCNs detected for the frontal view is almost 2.5 times more compared to the number of MCNs detected for other views with $\boldsymbol{w = 10}$ for both test and training views. This is because images of frontal view are almost twice the size of the other views with more visible object surface, where as for some of the images with the view angles closer to $\mathbf{90}^{\circ}$, the visible object surface is not significant compared to the frontal view. Thus for these images, less number of MCNs are detected. The time required for matching between two M-CORDs is directly proportional to the number of MCNs in each of the M-CORD. This can be seen from Table 5.12. When the number of MCNs is more, time required per match is also more.

In the existing literature, MNS method [61, 82] has close resemblance with the proposed methods. Experimentally, it has been seen that the performance of the proposed methods is better than the MNS method.

Comparison between the two proposed ways of representations : It is to be noted that M-CORD-Edge representation is richer than M-CORD-Cluster because M-CORD-Edge can also be sensitive to the spatial distribution of the colors. Hence, M-CORD-Edge produces better results than M-CORD-Cluster in the case of SOIL-47A and COIL-100. But, rich representation in M-CORD-Edge is obtained at the cost of extra storage space to store the M-CORDs and more comparison time between two M-CORDs. This can be

observed from Table 5.12. In case of ALOI-VIEW dataset, although, the M-CORD-Edge descriptors of the objects are rich, the performance in terms of recognition rate is not good compared to M-CORD-Cluster. The possible reason may be the problem of over representation. Sometimes, the representation richer than needed is not helpful. It can also be seen from Table 5.12 that the construction time and space requirement for M-CORD-Edge is more than M-CORD-Cluster with a smaller window size ($\mathbf{10 \times 10}$) but, the construction time of the descriptor is less for a bigger window size compared to M-CORD-Cluster. Comparatively poor but reasonably good recognition is obtained using M-CORD-Cluster with smaller window sizes ($\mathbf{10 \times 10}$) for all the datasets. Thus to choose between two methods one can opt for M-CORD-Cluster anticipating a reasonably good recognition with a small window size such as $\boldsymbol{w = 10}$. But, to get rich representation of the objects and better results, M-CORD-Edge with bigger window size should be considered.

The main contribution of this chapter is the proposed representation (M-CORD) of an object. Two dissimilarity measures, one is to compare between two M-CORDs and the other one is to compare between two MCNs, have also been proposed. The strength of the proposed methodology is the efficient representation of the colors appearing on the object surface which preserves the local shape information. Proposed methods would perform well when objects are multi-colored and, rich and colorful patterns appear on the object surface.

Chapter 6

Conclusions, Discussion and Scope for Further Works

In this thesis we have approached the problem of object recognition using colors appearing on the object surface. In the process to design a complete object recognition system, solutions to three different problems have been proposed. These three problems are Image Enhancement, Edge Detection and Object Representation. Moreover, two similarity measures are proposed to compare two images. Image enhancement and edge detection are the two problems which have been extensively studied by the researchers in the field of image processing and computer vision. However, in this thesis these two problems have been framed in new paradigms. Object representation is generally a first step in any object recognition or image retrieval task. The performance of the object recognition or image retrieval task highly depends on the kind of features and their representation scheme.

In *Chapter 2* of the thesis, an image enhancement scheme has been proposed. The highlights of the method is the property of preserving the hue component of the color and without encountering gamut problem. Hue is the most important component of a color because change in hue can change the crucial properties of the image. There are some methods which transform the image to other perceptually uniform color spaces such as HSI, HSV, LHS etc., which decorrelate the hue, saturation and intensity components of the

image. Processing is done in these color spaces without modifying the hue component. However, to display the image, the processed image data in these color spaces have to be transformed back to the RGB space and the transformations used to transform from RGB space to these spaces are non linear in nature. Hence, it takes more computational time and there is a chance of enhancing the noise in the original image. The problem occurs when some of the values in the transformed image data (i.e. gray values in RGB components) go out of bounds. The prevailing solutions, which takes care of these two problems need improvements. In *Chapter 2*, we have proposed a scheme by which both the problems are taken care of simultaneously.

In the present study, only the hue component is given importance for preservation. However, affect of the scheme on saturation component is not controlled. Thus sometimes the processed image becomes grayish with the increase in the intensity contrast. This problem demands a solution. The proposed scheme can be used to extend several gray scale image enhancement functions. Any particular image enhancement function is not suitable for all kinds of images for enhancement. When we have thousands of images to be enhanced, tuning the parameters used in the function and the selection of the right kind function is almost an impossible task. This needs automatic selection of functions as well as their parameters. In the context of image enhancement this problem necessitates an elaborate study. Another problem needs attention is the formulation of a quantitative measure that judges the qualitative improvement of the images induced by different enhancement functions. Although, these problems have been studied in the literature, in the context of direct image enhancement schemes [1, 22, 133], they are far away from attaining maturity.

In *Chapter 3* of the thesis, an edge detection method has been proposed. The motivation behind the method is to standardize the edge magnitudes to make the selection of threshold values as easier as possible. Considerable success has been achieved in this regard that is evident from the obtained results using the proposed method. In the present scheme the correlation between the pixels in each individual component is incorporated.

However, the cross correlation between the components has not been exploited. A deep theoretical study is needed to exploit the cross correlation factor, which we believe will produce still better results. In the literature, some measures are existing to compare the performance of different edge detection methods such as the Pratt's Figure of Merit, which depends on the true edge map of the image. These measures are applicable on artificial images only. In the case of natural images some kind of subjective evaluation is possible. A similar study conducted by Heath *et al.* [47], which uses statistical evaluation procedures considering inputs regarding qualitative performance of different methods from human observer, is needed.

In *Chapter 4*, an object representation methodology is presented which exploits the local color structure of the images to represent the image. The color information is kept by storing the mean values of the different parts of a local region and crucial shape features is incorporated by exploiting the local color structure of the image. The scheme is named as "Multi-Colored Region Descriptor (M-CORD)". Each M-CORD is a combination the colors appearing in several "Multi-Colored Neighborhoods(MCN)". Two different methods namely, M-CORD-Edge and M-CORD-Cluster, have been suggested to find the significantly different MCNs. M-CORD-Cluster employs a clustering algorithm to determine MCNs whereas M-CORD-Edge utilizes the edge map of images to determine the multi-colored regions.

In *Chapter 5*, the proposed image representation scheme is used to extract object signatures and experiments are performed using both the methods. Two dissimilarity measures have been proposed to compare two objects. The two dissimilarity measures are (1) for comparing two MCNs to find the significantly different MCNs and (2) for comparing two M-CORDs from two different images. Proposed methodologies are compared in detail with the similar MNS method using the SOIL-47A dataset. It is found that the proposed methods perform outstandingly compared to MNS method. Proposed representation scheme exploits the color structure present in the images effectively. Proposed methodologies have also been compared with some other existing methods on which re-

sults are available on SOIL-47A and COIL-100 datasets. Results using proposed methods on these datasets are found to be better when compared to the other methods considered for comparison. Recognition performance of our methods is found to be significantly better than the other methods for the cases of one, two or four training views and is similar or better for the cases of more than four number of training views. The key behind the success of our method is the use of color features that also carries the shape information. As proposed representation scheme exploits the color structures appearing on the object surface, it would perform better when the objects under consideration possess many colors making a rich and colorful pattern on its surface.

There are several aspects to the problem of object recognition. Some of the aspects of object recognition such as the invariant recognition under different illuminations, recognition of objects when the images having multiple objects, and the deformation of objects due to affine transformations are not explored in the present study. The problem of illumination invariance can be tackled by adopting similar techniques used by Matas *et al.* [82]. Recognition of objects when multiple objects are present in an image can be done by including the spatial information of the detected MCNs. The scale invariant features to some extent can be obtained by considering regions of different sizes in a particular location to find the MCNs in a multi-resolution fashion. Problems of selecting scale and other geometric invariant features due to different affine transformations are more challenging. In the present representation scheme the local color structure has not been exploited to represent the shape features to their fullest potential. This can be extended to a better representation scheme by considering the relative positions of the different parts of the local neighborhoods. These are some of the aspects of object recognition in the context of object representation to be explored in future works.

Selection of optimal views for representation is a problem in 3D object recognition. Generally, knowledge of objects from a single view is not sufficient for representation [88,118] because shape and look of an object vary when it is viewed from different angles. Thus information from different view points are needed to build a model for an object.

An ideal solution is to keep the information from all possible views separately to build the model set. However, in a large dataset, the difficulty in keeping such an enormous amount of data is of two folds. First, more is the number of views considered in the model set, more is the memory required to store information from these views. Second, decision making process for each query object becomes lengthy and computationally intensive when there are more number of views considered in the model set. The general practice is to sample the view space in equal interval and keep the information from the selected views. The problems in this approach are (a) how to determine the number of views to be selected and (b) which views are to be considered. However, it is tenable to say that the images of an object from two different views close to each other are correlated. But, the amount of correlation varies from object to object. Thus the number of views needed to build a model set varies with objects. This particular property could be exploited to select a particular number of views automatically. Moreover, an optimal condition needs to be set to automatically find the required views.

If we have a good representation scheme, the next challenging problem is to design a generalized object classification scheme. Several methodologies in the literature have been proposed using different machine learning tools for image classification such as decision trees and support vector machines (SVM). The problem encountered using an SVM with a local representation scheme is to design a proper kernel function for the SVM because in many local representation schemes the feature vectors can be of different sizes. Thus, it becomes difficult to find a distance between two feature vectors. There are some kernels suggested in the literature in this regard but, there is also disagreement on the mathematical validity of the suggested kernels [13, 72, 143]. It needs investigation to design a suitable kernel function that takes care of the mathematical requirements and heterogeneity present in the feature vectors.

The problem of designing a classification scheme to classify at different levels of specificity is a very challenging problem in object recognition. Human beings can identify objects at different levels of specificity at ease. Human beings can classify a collection

of objects into animals, clothes, vehicles, birds, books etc., and at the same time human beings can classify objects into classes such as living and non-living. The existing class of representation and classification schemes in the literature are far from achieving this level of accuracy for object identification and recognition at different levels of specificity. This is an interesting problem to work upon.

Appendix A

Additional Results from Chapter 3

This appendix is prepared to show some additionl results using the proposed methods, Ruzon *et al.*'s method and Cumani's method. The results can be browsed through an HTML file provided in the CD attached to the thesis. Please click on the link to Appendix given in the index.html file.

Bibliography

[1] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "Transform-based image enhancement algorithms with performance measure," *IEEE Trans. Image Processing*, vol. 10, no. 3, pp. 367 – 382, Mar. 2001.

[2] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004.

[3] A. R. Ahmadyfard and J. Kittler, "Using relaxation technique for region-based object recognition," *Image and Vision Computing*, vol. 20, no. 11, pp. 769–781, Sept. 2002.

[4] W. Alshatti and P. Lambert, "Using eigenvectors of a vector field for deriving a second directional derivative operator for color images," in *Proc. of Int. Conf. on Computer Analysis of Images and Patterns*, pp. 149–156, 1993.

[5] P. Androutsos, A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Aggregation of color and shape features for hybrid query generation in content based visual information retrieval," *Signal Processing*, vol. 85, no. 2, pp. 385–393, Feb. 2005.

[6] R. Basri and D. W. Jacobs, "Recognition using region correspondences," *Int. J. of Computer Vision*, vol. 25, no. 2, pp. 141–162, 1996.

[7] A. Baumberg, "Reliable feature matching across widely separated views," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, vol. 1, pp. 774–781, 2000.

[8] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[9] R. S. Berns, F. W. Billmeyer, and M. Saltzman, *Billmeyer and Saltzman's Principles of Color Technology*, 3rd ed. John Wiley & Sons, Mar. 2000.

[10] H. Bischof, H. Wildenauer, and A. Leonardis, "Illumination insensitive recognition using eigenspaces," *Computer Vision and Image Understanding*, vol. 95, no. 1, pp. 86–104, 2004.

[11] I. M. Bockstein, "Color equalization method and its application to color image processing," *Journal of Optical Society of America*, vol. 3, no. 5, pp. 735–737, 1986.

[12] H. Borotsching, L. Peletta, M. Prantl, and A. Pinz, "Appearance-based active object recognition," *Image and Vision Computing*, vol. 18, no. 9, pp. 715–727, 2000.

[13] S. Boughorbel, J.-P. Tarel, and F. Fleuret, "Non-mercer kernels for SVM object recognition," in *Proc. of the British Machine Vision Conference*, London, England, pp. 137 – 146, 2004. [Online]. Available: http://www-rocq.inria.fr/~boughorb/boughorb-bmvc04.pdf

[14] M. Brown and D. G. Lowe, "Invariant features from interest point groups." in *Proc. of the British Machine Vision Conference*, Cardiff, Wales, pp. 656–665, 2002.

[15] J. Burianek, A. Ahmadyfard, and J. Kittler, "SOIL-47: The Surrey Object Image Library," Centre for Vision, Speech and Signal processing, University of

Surrey. [Online]. Available: http://www.ee.surrey.ac.uk/Research/VSSP/demos/colour/soil47/

[16] V. Buzuloiu, M. Ciuc, R. Rangayyan, and C. Vertan, "Adaptive-neighborhood histogram equalization of color images," *Journal of Electronic Imaging*, vol. 10, no. 2, pp. 445–459, 2001.

[17] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visual shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 4, pp. 373–392, Apr. 1994.

[18] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[19] B. Caputo and G. Dorkó, "How to combine color and shape information for 3D object recognition: kernels do the trick," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., 2002. [Online]. Available: http://lear.inrialpes.fr/pubs/2002/CD02

[20] G. J. Chamberlin, *The CIE International Color System Explained*. Salisbury, England: Tintometer Publication, 1951.

[21] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1055–1064, Sept. 1999.

[22] H. D. Cheng, M. Xue, and X. J. Shi, "Contrast enhancement based on a novel homogeneity measurement," *Pattern Recognition*, vol. 36, no. 11, pp. 2687–2697, Nov. 2003.

[23] A. Cumani, "Edge detection in multispectral images," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 1, pp. 40–51, Jan. 1991.

[24] A. Cumani, "Efficient contour extraction in color images," in *Proc. of the Asian Conf. on Computer Vision*, Hong Kong, vol. 1, pp. 582 – 589, 1998. [Online]. Available: http://www.ien.it/~cumani/accv98.ps.gz

[25] M. Das, E. M. Riseman, and B. A. Draper, "FOCUS: Searching for multi-colored objects in a diverse image," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 756–761, 1997.

[26] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 116–125, Jan. 1986.

[27] P. M. Djuric and J. K. Fwu, "On the detection of edges in vector images," *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1595–1601, Nov. 1997.

[28] G. Dorkó and C. Schmid, "Selection of scale-invariant parts for object class recognition," in *Proc. of Int. Conf. on Computer Vision*, Nice, France, pp. 634–639, 2003. [Online]. Available: http://lear.inrialpes.fr/pubs/2003/DS03

[29] M.-P. Dubuisson and A. K. Jain, "Fusing color and edge information for object matching," in *Proc. of Int. Conf. Image Processing*, vol. 3, pp. 982–986, 1994.

[30] F. Ennesser and G. Medioni, "Finding waldo, or focus of attention using local color information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 805–809, 1995.

[31] J. Fan, W. G. Aref, M. S. Hacid, and A. K. Elmagarmid, "An improved automatic isotropic color edge detection technique," *Pattern Recognition Letters*, vol. 22, no. 13, pp. 1419–1429, Nov. 2001.

[32] J. Fan, K. David, Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1454–1466, Oct. 2001.

[33] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 5, pp. 522–529, May 1995.

[34] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam Library of Object Images," *Int. J. of Computer Vision*, vol. 61, no. 1, pp. 103–112, Jan. 2005. [Online]. Available: http://www.science.uva.nl/~mark/pub/2005/geusebroekIJCV05a.pdf

[35] T. Gevers and A. W. M. Smeulders, "Content-based image retrieval by viewpoint-invariant color indexing," *Image and Vision Computing*, vol. 17, no. 7, pp. 475–488, May 1999.

[36] T. Gevers and A. W. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, Mar. 1999.

[37] T. Gevers and H. Stokman, "Robust histogram construction from color invariants for object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 1, pp. 113–118, Jan. 2004.

[38] A. R. Gillespie, A. B. Khale, and R. E. Walker, "Color enhancement of highly correlated images. I. Decorrelation and HSI Contrast Stretches," *Remote Sensing Of Environment*, vol. 20, no. 3, pp. 209–235, Dec. 1986.

[39] A. R. Gillespie, A. B. Khale, and R. E. Walker, "Color enhancement of highly correlated images. II. Channel Ratio and Chromaticity Transformation Techniques," *Remote Sensing Of Environment*, vol. 22, no. 3, pp. 343–365, Aug. 1987.

[40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley, Apr. 1992.

[41] D. Guillamet and J. Vitria, "A comparison of global versus local color histograms for object recognition," in *Proc. of Int. Conf. on Pattern Recognition*, vol. 2, pp. 422–425, Sept. 2000.

[42] A. Gupta and B. Chanda, "A hue preserving enhancement scheme for a class of colour images," *Patter Recognition Letters*, vol. 17, no. 2, pp. 109–114, 1996.

[43] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 7, pp. 729–736, July 1995.

[44] D. Hall, J. L. Crowley, and V. C. de Verdiere, "View invariant object recognition using coloured receptive field," *Machine Graphics and Vision*, vol. 9, no. 2, pp. 341–352, 2000.

[45] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vision Conf.*, pp. 147–151, 1988.

[46] G. Healey and D. Slater, "Using illumination-invariant color histogram descriptors for recognition," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 355–360, 1994.

[47] M. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for accessing the relative performance of edge-detection algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 12, pp. 1338–1359, Dec. 1997.

[48] G. Heidemann, "Unsupervised image categorization," *Image and Vision Computing*, vol. 23, no. 10, pp. 861–876, Sept. 2005.

[49] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Spatial color indexing and applications," *Int. J. of Computer Vision*, vol. 35, no. 3, pp. 245–268, 1999.

[50] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 9, pp. 850–863, Sept. 1993. [Online]. Available: citeseer.ist.psu.edu/huttenlocher93comparing.html

[51] D. P. Huttenlocher, R. H. Lilien, and C. F. Olson, "View-based recognition using an eigenspace approximation to the Hausdorff measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 9, pp. 951–955, Sept. 1999.

[52] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition*, vol. 29, no. 8, pp. 1233–1244, Aug. 1996.

[53] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 3, pp. 267–278, Mar. 1996.

[54] A. R. Jimenéz, A. K. Jain, R. Ceres, and J. L. Pons, "Automatic fruit recognition: a survey and new results using range/attenuation images," *Pattern Recognition*, vol. 32, no. 10, pp. 1719–1736, Oct. 1999.

[55] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Proc. of European Conf. on Computer Vision*, Prague, Czech Republic, May 2004. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/publications/papers/kadir04.pdf

[56] T. Kadir and M. Brady, "Saliency, scale and image description," *Int. J. of Computer Vision*, vol. 45, no. 2, pp. 83–105, Nov. 2001.

[57] M. S. Kankanhalli, B. M. Mehtre, and R. K. Wu, "Cluster-based color matching for image retrieval," *Pattern Recognition*, vol. 29, no. 4, pp. 701–708, Apr. 1996.

[58] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, Washington, USA, vol. I, pp. 511–517, 2004.

[59] A. Koschan, "A comparative study on color edge detection," in *Proc. of the Asian Conf. on Computer Vision*, Singapore, vol. III, pp. 574–578, 1995.

[60] A. Kostin, J. Kittler, and W. Christmas, "Object recognition by symmetrised graph matching using relaxation labeling with an inhibitory mechanism," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 381–393, Feb. 2005.

[61] D. Koubaroulis, J. Matas, and J. Kittler, "Evaluating colour-based object recognition algorithms using the SOIL-47 database," in *Proc. of the Asian Conf. on Computer Vision*, Jan. 2002.

[62] R. Krishnamoorthi and P. Bhattacharya, "Color edge extraction using orthogonal polynomial based zero crossing scheme," *Information Sciences*, vol. 112, no. 1-4, pp. 51–65, Dec. 1998.

[63] M. Lades, C. Vorbuggen, J. Buhmann, J. Lange, C. v. Malsburg, R. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Comput.*, vol. 42, no. 3, pp. 300–311, Mar. 1993.

[64] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, vol. 2, pp. 409–415, June 2003. [Online]. Available: http://www.vision.ethz.ch/publ/leibe-categorization-cvpr03.pdf

[65] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, Apr. 2000.

[66] A. Leonardis, H. Bischofa, and J. Maver, "Multiple eigenspaces," *Pattern Recognition*, vol. 35, no. 11, pp. 2613–2627, Nov. 2002.

[67] Z.-N. Li, O. R. Zaane, and Z. Tauber, "Illumination invariance and object model in content-based image and video retrieval," *Journal of Visual Communication and Image Representation*, vol. 10, no. 3, pp. 219–244, Sept. 1999.

[68] Z. Lin, J. Wang, and K.-K. Ma, "Using eigencolor normalization for illumination-invariant color object recognition," *Pattern Recognition*, vol. 35, no. 11, pp. 2629–2642, Nov. 2002.

[69] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[70] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[71] L. Lucchese, S. K. Mitra, and J. Mukherjee, "A new algorithm based on saturation and desaturation in the xy-chromaticity diagram for enhancement and re-rendition of color images," in *Proc. of Int. Conf. Image Processing*, Thessaloniki, Greece, pp. 1077–1080, October 7-10 2001.

[72] S. Lyu, "Mercer kernels for object recognition with local features," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 2, pp. 223–229, June 2005. [Online]. Available: http://www.cns.nyu.edu/~lsw/files/cvpr05a.pdf

[73] R. Machuca and K. Phillips, "Applications of vector fields to image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, no. 3, pp. 316–329, May 1983.

[74] D. Macrini, A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker, "View-based 3-D object recognition using shock graphs," in *Proc. of Int. Conf. on Pattern Recognition*, vol. 3, pp. 24– 28, 2002. [Online]. Available: http://www.cim.mcgill.ca/~shape/publications/icpr02.pdf

[75] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proc. of the Indian National Institute Sciences of India*, Calcutta, vol. 2, no. 1, pp. 49–55, Apr. 1936.

[76] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "A generic approach for image classification based on decision tree ensembles and local sub-windows," in *Proc. of the Asian Conf. on Computer Vision*, vol. 2, pp. 860–865, 2004.

[77] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 34–40, June 2005.

[78] D. Marr, *Vision*. W. H. Freeman and Company, 1982.

[79] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. of Int. Conf. on Computer Vision*, vol. 2, pp. 416–423, July 2001.

[80] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Distinguished regions for wide-baseline stereo," Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, Research Report CTU–CMP–2001–33, Nov. 2001.

[81] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference*, Cardiff, London, UK, P. L. Rosin and D. Marshall, Eds., vol. 1, pp. 384–393, Sept. 2002.

[82] J. Matas, D. Koubaroulis, and J. Kittler, "The multimodal neighborhood signature for modeling object color appearance and applications in object recognition and image retrieval," *Computer Vision and Image Understanding*, vol. 88, no. 1-3, pp. 1–23, Oct. 2002.

[83] J. Matas, R. Marik, and J. Kittler, "On representation and matching of multi-coloured objects," in *Proc. of Int. Conf. on Computer Vision*, Cambridge, MA, USA, pp. 726–732, June 1995.

[84] B. W. Mel, "SEEMORE: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition," *Neural Computation*, vol. 9, pp. 777–807, 1997.

[85] F. Mindru, T. Moons, and L. Van Gool, "Recognizing color patterns irrespective of viewpoint and illumination," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, vol. I, pp. 368–373, 1999.

[86] P. A. Mlsna and J. J. Rodriguez, "A multivariate contrast enhancement technique for multispectral images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 33, no. 1, pp. 212–216, Jan. 1995.

[87] P. A. Mlsna, Q. Zhang, and J. J. Rodriguez, "3-D histogram modification of color images," in *Proc. of Int. Conf. Image Processing*, Lausanne, Switzerland, vol. III, pp. 1015–1018, Sept. 16-19 1996.

[88] F. Mokhtarian and S. Abbasi, "Automatic selection of optimal views in multi-view object recognition," in *Proc. of the British Machine Vision Conference*, Cardiff, 2000. [Online]. Available: http://www.bmva.ac.uk/bmvc/2000/papers/p28.pdf

[89] G. Mori, S. Belongie, and J. Malik, "Shape contexts enable efficient retrieval of similar shapes," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, vol. 1, pp. 723–730, Dec. 2001.

[90] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D object from appearance," *Int. J. of Computer Vision*, vol. 14, no. 1, pp. 5–24, Jan. 1995.

[91] K. Nagao, "Recognizing 3D objects using photometric invariant," in *Proc. of Int. Conf. on Computer Vision*, pp. 480–487, 1995.

[92] S. K. Naik and C. A. Murthy, "Hue-preserving color image enhancement without gamut problem," *IEEE Trans. Image Processing*, vol. 12, no. 12, pp. 1591–1598, Dec. 2003.

[93] S. K. Naik and C. A. Murthy, "Hough transform for region extraction in color images," in *Proc. of Indian Conf. on Computer Vision Graphics and Image Processing*, Science City, Kolkata, India, pp. 252–257, 2004.

[94] S. K. Naik and C. A. Murthy, "Distinct multi-colored region descriptors for object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, 2005, (communicated).

[95] S. K. Naik and C. A. Murthy, "Standardization of edge magnitude in color images," *IEEE Trans. Image Processing*, 2005, (Accepted for publication).

[96] S. K. Naik and C. A. Murthy, "Enhancement and edge detection for color object recognition," *Pattern Recognition*, 2006, (communicated).

[97] S. K. Nayar and R. M. Bolle, "Computing reflectance ratios from an image," *Pattern Recognition*, vol. 26, no. 10, pp. 1529–1542, Oct. 1993.

[98] S. K. Nayar and R. M. Bolle, "Reflectance based object recognition," *Int. J. of Computer Vision*, vol. 17, no. 3, pp. 219–240, 1996.

[99] S. K. Nayar, S. A. Nene, and H. Murase, "Real-time 100 object recognition system," in *Int. Conf. on Robotics and Automation*, Minneapolis, Minnesota, vol. 3, pp. 2321–2325, Apr. 1996. [Online]. Available: http://www1.cs.columbia.edu/CAVE/publications/pdfs/Nayar_ICRA96_2.pdf

[100] S. A. Nene, S. K. Nayar, and H. Murase, "Colombia object image library (COIL-100)," Technical Report, CUCS-006-96, Dept. Of Computer Science, Colombia University, Tech. Rep., Feb. 1996. [Online]. Available: http://www1.cs.columbia.edu/CAVE/publications/pdfs/Nene_TR96_2.pdf

[101] R. Nevatia, "A color edge detector and its use in scene segmentation," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 11, pp. 820–826, Nov. 1977.

[102] C. L. Novak and S. A. Shafer, "Color edge detection," in *Proc. of DARPA Image Understanding Workshop*, pp. 35–37, 1987.

[103] S. Obdrzalek and J. Matas, "Object recognition using local affine frames on distinguished regions," in *Proc. of the British Machine Vision Conference*, Cardiff, 2002. [Online]. Available: http://www.bmva.ac.uk/bmvc/2002/papers/134/full_134.pdf

[104] I. K. Park, I. D. Yun, and S. U. Lee, "Color image retrieval using hybrid graph representation." *Image and Vision Computing*, vol. 17, no. 7, pp. 465–474, May 1999.

[105] G. Pass and R. Zabih, "Comparing images using joint histograms," *ACM Journal of Multimedia Systems*, vol. 7, no. 3, pp. 234–240, May 1999. [Online]. Available: http://www.cs.cornell.edu/html/rdz/papers/jms98.pdf

[106] M. Pelillo, K. Siddiqi, and S. Zucker, "Matching hierarchical structures using association graphs," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 11, pp. 1105–1120, Nov. 1999.

[107] I. Pitas and P. Kinikilis, "Multichannel techniques in color image enhancement and modeling," *IEEE Trans. on Image processing*, vol. 5, no. 1, pp. 168–171, Jan. 1996.

[108] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 637–646, June 1998.

[109] M. B. Priestly and M. T. Chao, "Non-parametric function fitting." *Journal of Royal Statistical Society*, vol. ser B3, no. 4, pp. 384–392, 1972.

[110] R. R. Rakesh, P. Chaudhuri, and C. A. Murthy, "Thresholding in edge detection : A statistical approach," *IEEE Trans. Image Processing*, vol. 13, no. 7, pp. 927–936, July 2004.

[111] R. Ramanath, "A framework for object characterization and matching in multi- and hyperspectral imaging systems," Ph.D. dissertation, Dept. of Elec. and Comp. Engg., North Carolina State University, August 13 2003. [Online]. Available: http://www.lib.ncsu.edu/theses/available/etd-08132003-223814/unrestricted/etd.pdf

[112] M. Reiter and J. Matas, "Object-detection with a varying number of eigenspace projections," in *Proc. of Int. Conf. on Pattern Recognition*, Brisbane, Qld., Australia, vol. 1, pp. 759–761, Aug. 1998.

[113] G. Robinson, "Color edge detection," *Optical Engineering*, vol. 16, no. 5, pp. 479–484, Sept./Oct. 1977.

[114] J. J. Rodriguez and C. C. Yang, "High-resolution histogram modification of color images," *Graphical Models and Image Processing*, vol. 57, no. 5, pp. 432–440, September 1995.

[115] D. Roobaert and M. M. Van Hulle, "View-based 3D object recognition with support vector machines," in *Proc. IEEE Workshop on Neural Networks for Signal Processing(NNSP99)*, Madison,Wisconsin,USA, pp. 77–84, Aug. 1999.

[116] D. Roth, M.-H. Yang, and N. Ahuja, "Learning to recognize three-dimensional objects," *Neural Comp.*, vol. 14, no. 5, pp. 1071–1103, May 2002.

[117] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 272–277, 2003.

[118] S. D. Roy and N. Kulkarni, "Active 3-D object recognition using appearance-based aspect graphs," in *Proc. of Indian Conf. on Computer Vision Graphics and Image Processing*, Science City, Kolkata, India, pp. 40–45, Dec. 2004.

[119] M. A. Ruzon and C. Tomasi, "Color edge detection with compass operator," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 160–166, June 1999.

[120] M. A. Ruzon and C. Tomasi, "Edge, junction, and corner detection using color distributions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 11, pp. 1281–1295, Nov. 2001.

[121] E. Saber, A. M. Tekalp, and G. Bozdagi, "Fusion of color and edge information for improved segmentation and edge linking," *Image and Vision Computing*, vol. 15, no. 10, pp. 769–780, Oct. 1997.

[122] J. Scharcanski and A. N. Venetsanopoulos, "Edge-detection of color images using directional operators," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 397–401, Apr. 1997.

[123] B. Schiele and J. L. Crowley, "Recognition without correspondence using multi-dimensional receptive field histograms," *Int. J. of Computer Vision*, vol. 36, no. 1, pp. 31 – 50, Jan. 2000.

[124] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 530–535, May 1997.

[125] T. Sebastian, P. Klein, and B. B. Kimia, "Recognition of shapes by editing shock graphs," in *Proc. of Int. Conf. on Computer Vision*, Vancouver, B.C, pp. 755–762, 2001. [Online]. Available: http://www.lems.brown.edu/vision/researchAreas/ShockMatching/SebastianICCV01.pdf

[126] A. Shiozaki, "Edge extraction using entropy operator," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 1–9, Oct. 1986.

[127] M.-S. Shyu and J.-J. Leou, "A genetic algorithm approach to color image enhancement," *Pattern Recognition*, vol. 31, no. 7, pp. 871–880, Jul. 1998.

[128] K. Siddiqi and B. B. Kimia, "A shock grammar for recognition," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 507–513, 1996.

[129] D. Slater and G. Healey, "Combining color and geometric information for the illumination invariant recognition of 3-D objects," in *Proc. of Int. Conf. on Computer Vision*, pp. 563–568, 1995.

[130] M. A. Stricker, "Color and geometry as cues for indexing,," Department of Computer Science, University of Chicago, Tech. Rep. TR-92-22, 1992. [Online]. Available: http://citeseer.lcs.mit.edu/stricker92color.html

[131] R. N. Strickland, C. S. Kim, and W. F. McDonnel, "Digital color image enhancement based on the saturation component," *Optical Engineering*, vol. 26, no. 7, pp. 609–616, 1987.

[132] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. of Computer Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[133] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain," *IEEE Signal Processing Lett.*, vol. 10, no. 10, pp. 289–292, Oct. 2003.

[134] B. A. Thomas, R. N. Strickland, and J. J. Rodriguez, "Color image enhancement using spatially adaptive saturation feedback," in *Proc. of Int. Conf. Image Processing*, Santa Barbara, CA, vol. 3, pp. 30–33, Oct. 1997.

[135] A. Toet, "Multiscale color image enhancement." *Pattern Recognition Letters*, vol. 13, no. 3, pp. 167–174, Mar. 1992.

[136] P. J. Toivanen, J. Ansamki, J. P. S. Parkkinen, and J. Mielikinen, "Edge detection in multispectral images using the self-organizing map," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 2987–2994, Dec. 2003.

[137] P. E. Trahanias and A. N. Venetsanopoulos, "Color image enhancement through 3-D histogram equalization," in *Proc. of Int. Conf. on Pattern Recognition*, vol. 3, pp. 545–548, 1992.

[138] P. E. Trahanias and A. N. Venetsanopoulos, "Color edge detection using vector order statistics," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 259–264, Feb. 1993.

[139] P. E. Trahanias and A. N. Venetsanopoulos, "Vector order statistics operators as color edge detectors," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, no. 1, pp. 135–143, Feb. 1996.

[140] P. Tu, T. Saxena, and R. Hartley, "Recognizing objects using color-annotated adjacency graphs," in *Shape, Contour and Grouping in Computer Vision*," Springer, ser. Lecture Notes in Computer Science, D. A. Forsyth, J. L. Mundy, V. D. Gesù, and R. Cipolla, Eds., pp. 246–263, 1999, ISBN 3-540-66722-9. [Online]. Available: http://users.rsise.anu.edu.au/~hartley/Papers/sicily/sicily.pdf

[141] M. A. Turk and A. P. Pentaland, "Face recognition using eigenfaces," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[142] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. of Computer Vision*, vol. 59, no. 1, pp. 61–85, Aug. 2004.

[143] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *Proc. of Int. Conf. on Computer Vision*, Nice, France, vol. 1, pp. 257–264, October 13 - 16 2003.

[144] A. R. Weeks, G. E. Hague, and H. R. Myler, "Histogram specification of 24-bit color images in the color difference (C-Y) color space," *Journal of Electronic Imaging*, vol. 4, no. 1, pp. 15–22, Jan. 1995.

[145] C. C. Yang and J. J. Rodriguez, "Efficient luminance and saturation processing techniques for bypassing color coordinate transformations," in *Proc. of Int. Conf. on Systems, Man, and Cybernetics*, Vancouver, British Columbia, vol. 1, pp. 667–672, Oct. 22-25 1995.

[146] C. C. Yang and J. J. Rodriguez, "Saturation clipping in the LHS and YIQ color spaces," in *Proc. of IS&T/SPIE Int. Symposium on Electronic Imaging: Science & Technology – Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, San Jose, CA, Feb. 1996. [Online]. Available: http://www.csis.hku.hk/~yang/spie.ps

[147] Q. Zhang, P. A. Mlsna, and J. J. Rodriguez, "A recursive technique for 3-D histogram enhancement of color images," in *Proc. IEEE Southwest Symp. on Image Analysis and Interpretation*, San Antonio, TX, pp. 218–223, April 8-9 1996.

[148] Y. Zhong and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognition*, vol. 33, no. 4, pp. 671–684, Feb. 2000.

[149] S.-Y. Zhu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Comprehensive analysis of edge detection in color image processing," *Optical Engineering*, vol. 38, no. 4, pp. 612–625, Apr. 1999.