STATISTICAL FEATURE SELECTION AND EXTRACTION FOR VIDEO AND IMAGE SEGMENTATION

By

XIAOMU SONG

Bachelor of Science in Electrical Engineering
Northwestern Polytechnic University
Xi'an, P. R. China
1995

Master of Science in Electrical Engineering
Northwestern Polytechnic University
Xi'an, P. R. China
1998

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2005

STATISTICAL FEATURE SELECTION AND

EXTRACTION FOR VIDEO AND IMAGE

SEGMENTATION

Dissertation Approved:

Guoliang Fan

Dissertation Adviser

Keith Teague

Gary Yen

Douglas Heisterkamp

A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my most sincere thanks to my advisor, Professor Guoliang Fan, for his intelligent guidance, support, and encouragement throughout my Ph.D. study. He has taught me not only the technical knowledge, but also a rigorous attitude towards research and life. I especially thank him for the wonderful research environment he built in our lab, and his advise on pursuing my future academic career. I believe that all the things I learned from him will help me to develop my professional and personal skills in the future. I am also full of gratitude to my advisor committee members: Professor Keith Teague, Professor Gary Yen, and Professor Douglas Heisterkamp (Department of Computer Science) for their constructive suggestions and comments on my research. I benefit a lot from discussions with them. My thanks also go to Professor Rafael Fierro, Professor William D. Warde (Department of Statistics), Professor Mahesh Rao (Department of Geography) and Professor Joseph Havlicek from the University of Oklahoma for their great help, encouragement, and constructive suggestions on my Ph.D. study.

Meanwhile, I would like to thank all former and current members of Visual Communication and Image Processing Laboratory, as well as other cooperators from other departments for their friendship and constructive discussions. Lijie Liu, Ginto Cherian, Jiangming Qian (Department of Geography) and David Monismith (Department of Computer Science) deserve special recognition for their great cooperation in our research.

Most of all, I want to express my deepest thanks to my parents, Luntai Song and Chengzhao Wang. I cannot thank them enough for their everlasting love, support, and understanding, and those are the most precious gift in my life. My Ph.D. work would be impossible without their strongest support behind me. Finally, I am also grateful to my wife, Man Luo, whose love, understanding, patient, consideration, and support make my life full of happiness and joy. I dedicate this dissertation to my parents and my wife.

This work was supported by research grants from the National Science Foundation (NSF), Department of Defense Army Research Laboratory, Oklahoma NASA EPSCoR and Oklahoma State University Environmental Institute.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Motivation

Along with the rapid development of the visual media and digital technologies, it is more and more pressing to provide efficient approaches that can index, store, retrieve, analyze, and transmit visual data, which usually require large mounts of computational and storage sources. Visual data segmentation, including image/video segmentation, is to partition the data into distinct volumes or regions of similar behaviors or properties. For example, video object segmentation separates a sequence of scene into meaningful components, such as moving objects, face, human body, etc. However, due to the natural complexities of colors, intensities, textures, as well as motion properties, visual data segmentation is still a challenging task, especially when there are no or very limited prior knowledge about regions or objects of interest. Generally, visual data segmentation is a pattern recognition problem, where feature selection/extraction and data classifier design are two indispensable steps. Specifically, the feature set is expected to be representative, discriminative, compact, and easy to extract or select. The data classifier should effectively capture the disparity between different regions/objects by exploiting the underlying feature characteristics. Our goal is to study and develop feature selection and extraction approaches for visual data segmentation in three application areas, including video and image segmentation, as well as remote sensing data analysis.

Particularly, five fundamental issues related to feature selection and extraction for visual data segmentation are studied in this research:

1. How to select salient features to support efficient data segmentation via numerical and/or analytical approaches?

2. How to select and/or extract lower dimensional features that are discriminative and capture major information from much higher dimensional features.

3. How to select reliable training samples to convert unsupervised learning to self-supervised learning?

4. How to select proper feature characterizations for supervised and unsupervised learning?

5. How to evaluate and predict the segmentation performance based on training data, and how to adjust the trade-off between different segmentation criteria, e.g., precision and recall?

1.2 Methodology

Feature selection is to select a best subset of the input features set regarding certain criteria. Feature extraction is to create new features based on transformation or combinations of the original input feature set. Before constructing a data classifier, feature extraction is used to generate new discriminative features to facilitate data classification. After constructing a feature set with both extracted and original features, feature selection is applied to select most discriminative features so that the computational load of the following process can be reduced. In this work, the concepts of feature extraction and selection are generalized to any process that generating or selecting features for classification purpose. Moreover, feature selection not only reduces feature dimension, but also feature size.

The methodology of this work is shown in Fig. 1.1. Statistical methods are widely involved when selecting/extracting features and designing a data classifier, which are usually explicitly or implicitly related to a problem of density estimation. When we study visual feature selection and extraction for segmentation, due to different characteristics of visual data, different feature sets and selection/extractiom methods have to be considered. For example, for video segmentation, both spatial

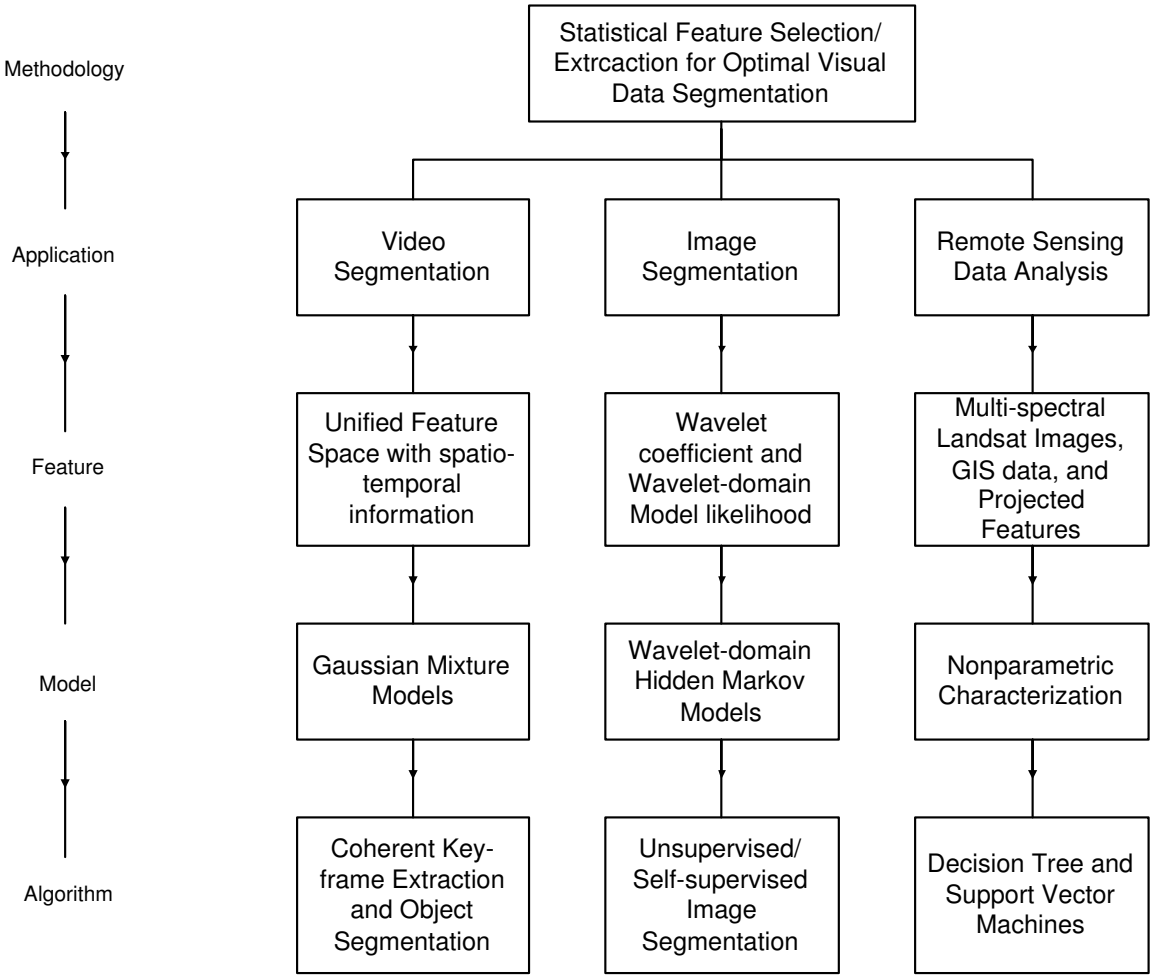


Figure 1.1: Methodology.

and temporal features should be considered because a video sequence records object behaviors in space and time domain. If we study the segmentation of textured image, features that can efficiently characterize spatial texture patterns are the focus. For complex remote sensing image segmentation, single source features might not be enough for to model different cover types, and multisource geospatial features can facilitate such task in most cases.

After determining a representative feature set, feature characterization is another important issue. Parametric and nonparametric methods are two major approaches for feature modeling. Parametric classifier assumes mathematical forms for density or discriminant function. For example, a mixture of Gaussian functions can be used to model spatial and/or temporal features in video data [69, 70], and tree-structured Markov chain model is often used for multiscale image analysis [13, 165, 37]. Nonparametric data classifier does not impose any parametric structure

for the density function. *Parzen* and *k-nearest neighbor* methods are two frequently used nonparametric approaches for density estimation. For instance, nonparametric methods cause more and more attentions for multisource remote sensing data analysis because a single structure parametric model usually cannot efficiently capture the multi-modality of multisource features. Parametric methods have simple forms and are usually computational efficient, but parametric models are not sufficient enough to describe complex densities in most cases. Nonparametric methods can characterize very irregular densities, while they are computationally expensive and need more storage space.

Based on feature modeling, various classification/segmentation methods can be developed. In this research, we study Gaussian Mixture model (GMM)-based maximum likelihood (ML) video segmentation, maximum *a posteriori* (MAP) image segmentation based upon wavelet-domain hidden Markov models (WDHMM), and nonparametric decision tree classifier (DTC) and support vector machines (SVM) for remote sensing image segmentation. It is worth mentioning that feature selection and extraction are not independent to feature modeling and classification. On the contrary, they are integrated into modeling and classification processes. Generally, modeling and classification criteria guide feature selection and extraction because only those that lead to efficient modeling and accurate classification are preferred features. For example, we want to select features that can maximize model likelihood, or maximize divergence between model components, each of which is usually associated with a meaningful object or region, or result in the highest classification accuracy according to test data.

1.3 Applications

1.3.1 Video Segmentation

Content-based video analysis exploits important structures and events based on which efficient and powerful tools can be developed for video access and transmission. Video segmentation is a fundamental step towards content-based video analysis. It often refers to as temporal and object segmentations. Temporal video segmentation

partitions a video sequence into a set of shots, and extracts one or a set of key-frames to represent each shot. Object segmentation partitions video data into meaningful regions or objects for content-based analysis to support various object-oriented video applications. Due to different applications on different semantic levels, key-frame extraction and object segmentation are usually implemented independently and separately based on different feature sets. In order to support more efficient and flexible content-based video analysis, it is helpful to exploit the inherent relationship between key-frame extraction and object segmentation. In addition, the new MPEG-7 standard provides a generic *segment-based* representation model for video data [116], and both key-frame extraction and object segmentations could be grouped into an integrated framework. Therefore, we study how to coherently perform key-frame extraction and object segmentation, where a unified feature space is first constructed to represent video frames and visual objects together in a joint spatio-temporal domain, and key-frame extraction is formulated as a feature selection process for object segmentation. The Gaussian mixture model (GMM) is used to characterize video data in the unified feature space. Issues (1) in Section 1.1 will be studied.

1.3.2 Bayesian Image Segmentation

It is well known that image pixel intensity is not a representative for the segmentation of textured images, and features extracted in transformed domain are more helpful. Wavelet coefficients obtained by wavelet transform are such features. However, the values, or low order statistical information of wavelet coefficients are not robust features for segmentation purpose because they cannot characterize texture behaviors sufficiently. Wavelet-domain hidden Markov models (WDHMM), which capture high order interscale dependencies of wavelet coefficients [37, 137, 60], have shown impressive performances in supervised image segmentation [32, 58] and image retrieval [45]. Nevertheless, existing WDHMMs are not suitable to be directly applied to unsupervised image segmentation because they implement supervised algorithms based on known or pure texture prototypes. In this work, we study how to efficiently implement WDHMMs for unsupervised image segmentation. A new hybrid soft-hard decision approach is suggested to generate discriminative features that contain

high order statistical information of wavelet coefficients for unsupervised clustering, and two new clustering method sare developed to convert the unsupervised problem into a self-supervised one. Two different WDHMMs are used at the unsupervised learning and self-supervised learning steps in order to utilize advantages of different WDHMMs. Issues (2), (3), and (4) in Section 1.1 will be addressed.

1.3.3 Remote Sensing Data Analysis

In remote sensing data analysis, we study two practical problems related to Unite State Department of Agriculture (USDA)'s Conservation Reserve Program (CRP), i.e., CRP mapping and compliance monitoring. CRP is a nationwide program that encourages farmers to plant long-term, resource conserving covers to improve soil, water, and wildlife resources. With recent payments of nearly **$1.8** billion for new enrollments (2003 signup), it is imperative to obtain accurate digital CRP maps for management and evaluation purposes. In this work, CRP mapping is formulated a supervised two-class segmentation problem. Since multispectral Landsat image data cannot provide discriminative features for the classification of some land cover types, multisource geospatial data, including multispectral Landsat imagery and geographic information system (GIS) data are used as the original input features. Because it may not be appropriate to model multisource data by traditional multivariate statistical models [85, 82, 14, 105, 10], we apply nonparametric DTC and SVM to CRP mapping, and study how to increase the system sensitivity (recall rate). When implementing DTC, a entropy-based criterion is used to selection helpful features. When using SVM, principal component anlaysis (PCA) is used to extract new features based on the input feature, and the support vector learning further selects a set of representative samples in a projected high dimensional feature space to construct the data classifier. CRP compliance monitoring checks whether each CRP tract complies with its contract stipulations, and is formulated as an unsupervised segmentation. A newly developed one-class SVM (OCSVM) is used to detect false CRP regions in a CRP tract, and a heuristic method is used to select a feature subset based on the CRP reference data [54]. This method measures the contribution of each feature layer by approximately estimating its effect on the hyperplane. Issues

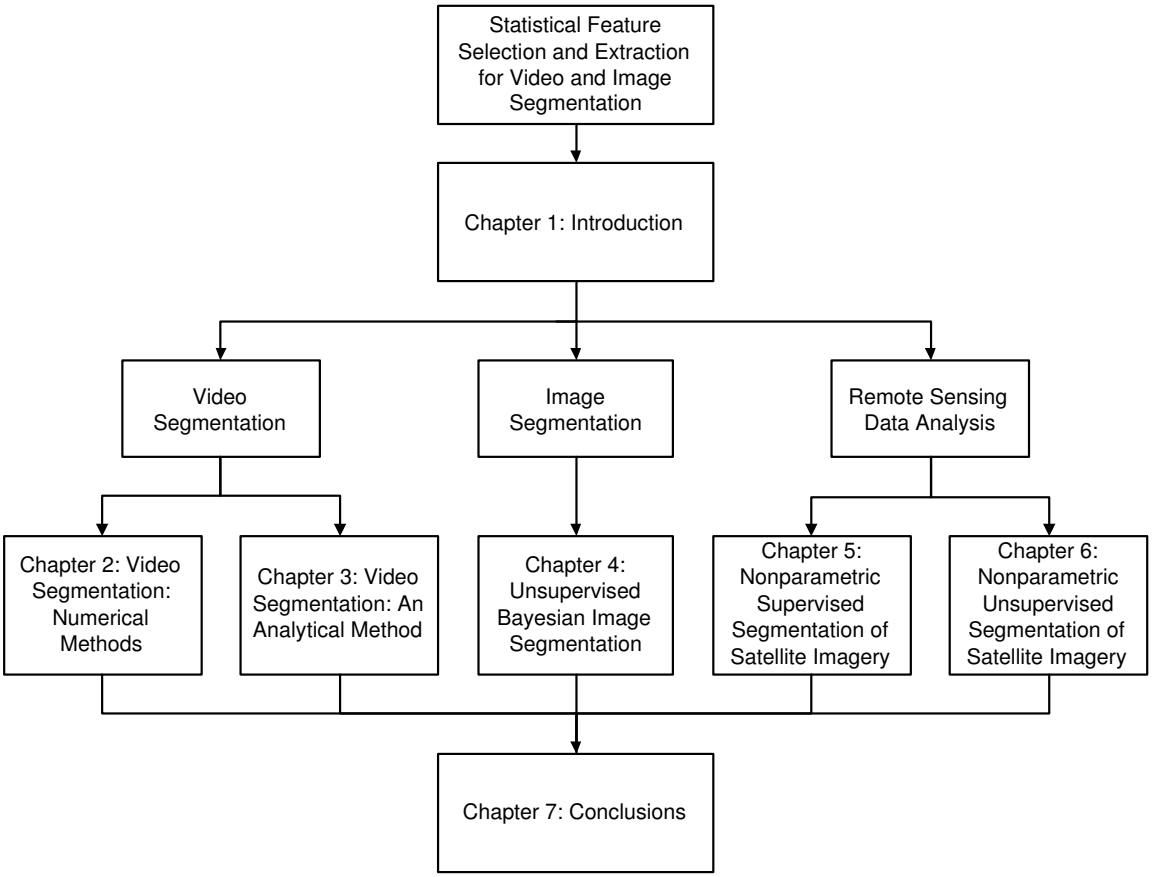


Figure 1.2: Outline of the dissertation.

(2), (3) and (5) in Section 1.1 will be discussed.

1.4 Outline

This dissertation composes seven chapters, and the outline is shown in Fig. 1.2.

In Chapter 2, we study video segmentation, where a coherent framework for video key-frame extraction and object segmentation is proposed. A unified feature space is first constructed to represent video frames and objects simultaneously in the spatial-temporal domain, and key-frame extraction is formulated as a feature selection process that aims to maximize the cluster divergence of distinct video objects by selecting a set of key-frames. Specifically, two divergence-based criteria are applied to achieve joint key-frame extraction and object segmentation with numerical solutions. One criterion recommends the key-frame extraction that leads to the maximum pairwise interclass divergence between objects in the feature space. The other aims at

maximizing the marginal divergence. Simulations with both synthetic and real video data manifest the efficiency and robustness of the proposed methods.

In Chapter 3, we discuss a new analytical approach to jointly formulate key-frame extraction and object segmentation in a statistical mixture model where the concept of frame/pixel saliency is introduced. A modified Expectation Maximization algorithm is developed for model estimation that leads to the most salient key-frames for object segmentation. Based on the coherent segmentation methods, a unified video representation and description framework is also suggested to support content-based video analysis. Simulations on both synthetic and real videos show the effectiveness of the proposed method.

In Chapter 4, a new unsupervised image segmentation method is proposed by exploiting the fitness disparity of local textured behaviors with respect to a global statistical model. A hybrid *soft-hard* decision approach is first developed to generate the fitness disparity, which is measured by the difference of model likelihoods generated from WDHMMs. Additionally, two new clustering approach are suggested to capture the likelihood disparity efficiently so that an initial segmentation map can be obtained. Moreover, a dual-model segmentation framework is suggested in order to fully utilize the capability of different WDHMMs. The simulation results on synthetic mosaics indicate that the proposed unsupervised segmentation algorithm can achieve high segmentation accuracy that is close to the supervised case, and the simulation on real images also show its applicability to real applications.

In Chapter 5, the CRP mapping problem is studied, which is formulated as an uneven two-class supervised segmentation of land covers. CRP mapping is a complex classification problem where both CRP and non-CRP areas are composed of various cover types. Therefore, multisource geospatial data, including Landsat imagery and GIS data, are used to increase the separability of different land cover types in the feature space. DTC and SVM are implemented with different feature selection and extraction approaches. Considering the importance of CRP mapping sensitivity, a new DTC pruning method is proposed to increase the *recall* rate. We also study two post relaxation methods to increase the *recall* rate of SVM. Moreover, a localized and

highly parallel framework is suggested to perform the large scale CRP mapping.

In Chapter 6, we study CRP compliance monitoring by using SVM approaches. CRP compliance monitoring checks each CRP tract regarding its contract stipulations, and is formulated as an unsupervised classification of Landsat imageries given the CRP reference data. Assuming that the majority of a CRP tract is compliant, we want to locate the non-CRP outliers. A one-class SVM (OCSVM) is used to separate minor outliers (non-CRP) from the majority (CRP). ν is an important OCSVM parameter that controls the percentage of outliers and is unknown here. Usually, ν estimation may be complicated or computationally expensive. We propose a novel ν-insensitive approach by incorporating both OCSVM and two-class SVM (TCSVM) sequentially. Specifically, a SVM-based heuristic method is used to select a feature subset for data classification. SVM scores obtained from the OCSVM, which indicate the distance between a data sample and the classification hyperplane in the feature space, is used to select sufficient and reliable training samples for TCSVM. Finally, the CRP tract is reclassified by the trained TCSVM.

Chapter 7 is the conclusions and future research. Based on the results from Chapter 2 to Chapter 6, it can be concluded that statistical feature selection and extraction approaches are effective with various data classification methods for three applications. Both Nonparametric and parametric methods are capable of uncovering the intrinsic data characteristics and structures, proving effective feedback for feature selection and extraction, and supporting accurate and robust visual data segmentation. Future research is also predicted in this chapter.

1.5 Original Contributions

The original contributions of this work are listed as follows:

- **Video segmentation with numerical methods:** A framework for coherent video key-frame extraction and object segmentation is proposed, where video frames and objects are represented in a unified spatio-temporal feature space,

and key-frame extraction is formulated as a feature selection process for object segmentation. Two numerical methods are developed associated with two cluster divergence-based criteria. By building a synergistic interaction between key-frame extraction and object segmentation, the proposed methods can provide not only robust and accurate object segmentation, but also compact and semantically meaningful key-frames to support content-based video analysis.

- **Video segmentation with an analytical method:** The contribution of video key-frame to object modeling and segmentation is quantized and integrated into the estimation of a new generative model derived from GMM, and the key-frame extraction and object modeling/segmentation are performed analytically during the model estimation without any combinatorial search, which could be time consuming if there are many frames.

- **Unsupervised Bayesian image segmentation:** Likelihood Principle is used to theoretically guide the segmentation process based on WDHMMs, where a hybrid *soft-hard* decision approach is proposed to extract discriminative and low dimensional features. Two new clustering methods are developed to obtain initial segmentation maps so that the unsupervised segmentation can be changed to the self-supervised segmentation. Moreover, two different WDHMMs are applied to the segmentation algorithm, where one is for obtaining the largest likelihood disparity with respect to the global model, and the other is used to train each texture type after the clustering.

- **CRP mapping:** In order to increase the separability between different land cover types, multisource geospatial data are used for the selection and extraction of discriminative features. A new DTC pruning method and two SVM post relaxation methods are proposed to increase the classification sensitivity. Additionally, a localized and parallel framework is suggested for the high speed computation with the large data source. Simulation results demonstrate that the suggested approaches can achieve very high recall rates, which might not be achievable by others.

- **CRP compliance monitoring:** OCSVM parameter ν considerably affects the segmentation results. Due to the complexity of the ν estimation, we propose

a novel ν-insensitive approach by incorporating both OCSVM and two-class support vector machine (TCSVM), where OCSVM results are used to select representative land cover samples for TCSVM training. Simulations on real data validate the applicability of the suggested method.

Chapter 2

VIDEO SEGMENTATION: NUMERICAL METHODS

Video segmentation is a fundamental step to support the interpretability and manipulability of visual data for many video applications. According to various needs of video analysis tasks at different semantic levels, such as video parsing or video indexing, video segmentation often refers to as two categories, i.e., temporal video segmentation and object-based segmentation. On one hand, temporal video segmentation usually has two steps. It first partitions a video sequence into a set of shots, each of which is an unbroken sequence of frames captured from one camera perspective. Then each shot can be represented by some key-frames. Temporal segmentation can provide compact video representation for video indexing and browsing. On the other hand, object-based video segmentation extracts objects of interest from a video sequence to support more structured and semantically meaningful representation for many object-oriented video applications, such as object tracking and recognition.

Generally, when there is no prior information about video content, temporal and object segmentation can be formulated as clustering processes in different feature spaces. Specifically, temporal segmentation in this work refers to key-frame extraction within a video shot. Since a frame is usually considered as the basic unit of a video shot, frame-wise features, such as color, texture, and motion information, are first extracted. Then key-frame extraction can be carried out by a clustering process that searches for cluster centers within a video shot, and the frames that are closest to the cluster centers are extracted as key-frames. During this process, similarity measurements [166, 77] or statistical modeling processes [74] are often used based on color histograms, which are invariant to image orientation and insensitive to noise. Moreover, color-based key-frame extraction is usually computationally efficient and

applicable to many online or real-time applications. However, since frame-wise features contain no spatial information about object location, shape, etc., key-frame extraction provides limited semantic meaning.

Compared with temporal segmentation, object segmentation is more semantically meaningful and technically more challenging. According to [117], current video object segmentation methods can be classified into three types: segmentation with spatial priority, segmentation with temporal priority, and joint spatio-temporal segmentation. Recently, more interests are brought to joint spatio-temporal segmentation of video objects [42, 69, 70, 144, 63] due to the nature of human vision that recognizes salient video structures in space and time jointly [67]. The work in [42] uses a modified nonparametric mean shift clustering in the spatio-temporal feature space, and the works in [69, 70] apply the Gaussian mixture model (GMM) to model video objects in a joint spatio-temporal domain, where the Expectation Maximization (EM) algorithm and the minimum description length (MDL) criterion are used for model estimation. Methods based on the graph partition theory are also suggested in [144, 63]. In these approaches, spatio-temporal pixel-wise features are extracted to construct a multi-dimensional feature space for object characterization and segmentation. Video object segmentation is a difficult issue due to the ambiguity of the object definition, as well as the heavy computational load. Feature extraction, selection and characterization play very important roles in current research on object-based video segmentation.

Key-frame extraction and object segmentation are usually implemented independently and separately due to different semantic levels. The work in [61] presents a universal framework where key-frame extraction and object segmentation are implemented independently and then used together to support content-based video analysis, and their outputs can be unified via a high-level description. The new MPEG-7 standard provides a generic *segment-based* representation model for video data [116]. This motivates us to combine key-frame extraction and object segmentations into a unified paradigm, supporting the universal video description scheme proposed in [61]. Recently, a combined key-frame extraction and object segmentation method was proposed in [111], where the extracted key-frames are used to estimate GMM for

model-based object segmentation, and object segmentation results are used to further refine the initially extracted key-frames via the GMM. Compared with [69, 70], this approach significantly reduces the computational load and improves object segmentation results. However, the underlying relationship between key-frame extraction and object segmentation was not explicitly indicated.

In this work, we propose a coherent framework for key-frame extraction and objet segmentation by exploiting an explicit relationship between key-frame extraction and object segmentation. First, a unified feature space is constructed to represent video frames and objects together in the joint spatio-temporal domain. Then video objects are represented by different clusters in the feature space. If a set of probability density functions are used to model the clusters, then model-based cluster divergence measurements can evaluate the separability of the clusters. Therefore, key-frame extraction is formulated as a feature selection problem that aims at maximizing the cluster divergence in the feature space. Specifically, two numerical criteria are suggested to extract key-frames for optimal object segmentation. One is the maximum average interclass Kullback Leibler distance (MAIKLD). The other is maximum marginal divergence (MMD) [156, 157, 158]. MAIKLD considers both temporal and spatial correlations between frames, and requires a greedy combinatorial search of key-frames. In this work, we use Sequential Forward Floating Selection (SFFS) feature selection method [130]. Marginal divergence is defined as the *average distance between each of the marginal class-conditional densities and their mean* in [156]. Instead of trying different combinations of video frames in MAIKLD, MMD tries to maximize the cluster divergence in each frame individually, so that it can be implemented more efficiently than MAIKLD. According to [156], when the mutual information between key-frames is not affected by the class label, the summation of key-frames that have the largest marginal diversity could lead to minimum Bayes classification error. Compared with MAIKLD, MMD might generate less representative key-frames for object segmentation due to the neglect of inter-frame relations. The proposed methods are tested on both synthetic and real video sequences. It is shown that two methods can provide different key-frames sets to support robust and accurate object segmentation.

The rest parts of this chapter are organized as follows. Section 2.1 introduces

some preliminaries about this work. Sections 2.2 discusses the proposed segmentation methods based on MAIKLD and MMD criteria. Section 5.5 shows the experiment setup, the simulations results and discussions. Final conclusions are drawn in Section 6.4.

2.1 Preliminaries

Video key-frame extraction and object segmentation are two major components covered in this paper. Correspondingly, we will first briefly review a color histogram-based key-frame extraction method, a statistical model-based object segmentation method, and a combined key-frame extraction and object segmentation approach. Then we will introduce a unified feature space, which forms the basis of the proposed segmentation framework, to represent video frames and objects simultaneously.

2.1.1 Color-based Key-Frame Extraction

Color information is quite often used for video key-frame extraction. In [166], a clustering-based key-frame extraction is proposed. In this method, a similarity measurement based on the frame-wise 16×8 2-D Hue and Saturation (HS) color histogram in the Hue-Saturation (HS) color space is used to measure the difference between frames $\mathbf{X}_i$ and $\mathbf{X}_j$:

$$Sim(\mathbf{X}_i, \mathbf{X}_j) = \sum_{h=1}^{16} \sum_{s=1}^{8} \min[H_{\mathbf{X}_i}(h, s), H_{\mathbf{X}_j}(h, s)], \tag{2.1}$$

where $H(h, s)$ is the HS color histogram value. A large value of (2.1) means strong similarity between two frames with respect to their HS color histograms. Begin from the first frame, based on the similarity values and a predefined threshold of similarity, a set of clusters are formed. A large threshold could result in more than enough extracted key-frames for video shot representation. After the clustering, the frames that are closest to each cluster center are extracted as the key-frames. Since the color histogram is easy to compute, this method is applicable to real-time systems.

2.1.2 Statistical Model-based Object Segmentation

In [69, 70], the Gaussian mixture model (GMM) is used to model video objects coherently in the joint spatio-temporal domain. If a video shot contains M objects, the probability density of a pixel $\mathbf{x}_i$ is a mixture of M Gaussian components:

$$p(\mathbf{x}_i|\theta) = \sum_{m=1}^{M} \alpha_m p(\mathbf{x}_i|\theta_m), \tag{2.2}$$

where α_m is the weight of the mth Gaussian component $p(\mathbf{x}_i|\theta_m)$ defined by a set of parameters denoted by θ_m. If there are N pixels in the video shot, a maximum likelihood (ML) approach to estimate θ is:

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{N} \log p(\mathbf{x}_i|\theta) \tag{2.3}$$

The iterative Expectation Maximization (EM) algorithm is often used to solve (3.3) [43]. The E step is to compute a so-called Q-function given the current estimation $\hat{\theta}(t)$ and $\mathbf{X} = \{\mathbf{x}_i, i = 1, \cdots, N\}$:

$$Q(\theta, \hat{\theta}_t) = E[\log p(\mathbf{X}, \mathbf{Y}|\theta)|\mathbf{X}, \hat{\theta}_t], \tag{2.4}$$

where $\mathbf{Y} = \{y_i, i = 1, \cdots, N\}$ is the class label of $\mathbf{X}$, and the posterior probability that $\mathbf{x}_i$ belongs to the mth component of GMM is estimated as:

$$w_m^{(i)} = \frac{\hat{\alpha}_m p(x^{(i)}|\hat{\theta}_m(t))}{\sum_{j=1}^{M} \hat{\alpha}_j p(x^{(i)}|\hat{\theta}_j(t))}. \tag{2.5}$$

The M step is to update the parameters by solving:

$$\hat{\theta}(t+1) = \arg\max_{\theta} Q(\theta, \hat{\theta}(t)). \tag{2.6}$$

The EM algorithm can be used together with the minimum description length (MDL) which will determine the order of GMM, i.e., M [136]. After the model estimation, each grouped video object is characterized by a Gaussian density, and video object can be segmented out via the maximum *a posteriori* (MAP) estimation.

2.1.3 Combined Key-frame Extraction and Object Segmentation

In [111], a combined key-frame extraction and object segmentation approach was proposed where the method in Section 2.1.1 is first applied to extract a set of

key-frames to represent an input video shot. In the following, we call these initially extracted key-frames as *key-frame candidates*. Then based on key-frame candidates, the GMM is used to model video objects in the joint spatio-temporal feature space, where the EM algorithm and the MDL criterion are applied to the GMM estimation. After object segmentation, the segmented objects in each key-frame candidate can be characterized by a GMM, and the Kullback Leibler distance (KLD) between each pair of GMM is calculated to estimate the content change between two key-frame candidates, so that the initially extracted key-frame candidates can be refined with a more compact key-frame set. By only involving key-frame candidates to the model estimation which is of a small portion of all video frames, this approach considerably mitigates the computational load compared with the methods in [69, 70], and could provide better segmentation performance due to the more efficient and effective model estimation. Meanwhile, the GMM consisting of both spatial and temporal information can support more salient and representative key-frame extraction after object segmentation.

2.1.4 Unified Feature Space for Video Segmentation

However, the inherent relationship between key-frames and video objects was not explicitly revealed in [111], which will be the focus of this work. Key-frame extraction and object segmentation are usually implemented based on different feature subsets. A unified feature subset is necessary for coherent key-frame extraction and video object segmentation. This feature subset should contain both spatial and temporal information that is capable of characterizing video objects and key-frames simultaneously, and easy to extract. In this work we use a pixel-wise 7-D feature vector suggested in [111], which is an extended version of the one in [69, 70], including (Y, u, v) color features, $x - y$ spatial location, time T, as well as the intensity change over time d_y to provide additional motion information.

For example, as shown in Fig. 2.1, a video shot of N frames contains three objects. Outliers, including noise and insignificant objects that might randomly appear, usually increase the overlapping between the objects/clusters in the feature space. The outliers degrade the accuracy and effectiveness of statistical modeling of major

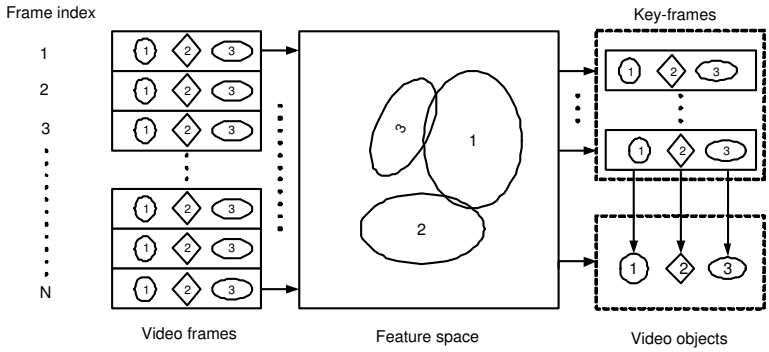


Figure 2.1: Unified feature space.

video objects in the feature space. Usually, the overlapping problem could be mitigated by two different ways. One is to add more discriminative features to construct a higher dimensional feature space where the divergence between objects/clusters could be increased. The other is to extract data samples with less outliers. Since we fix the feature dimension in this work, we attempt to reduce the overlapping phenomenon by extracting a set of key-frames that can effectively support object representation in the feature space. Therefore, key-frame extraction in this work is treated as a *feature selection* process where salient and representative key-frames are extracted so that the overlapping among objects/clusters in the feature space can be minimized. Then the model estimation and object segmentation can be efficiently accomplished based on these key-frames.

We begin from the fundamentals of feature selection in pattern recognition. Given a candidate feature set $\mathbf{X} = \{x^i | i = 1, 2, \cdots, n\}$, where i is the feature index, feature selection aims at selecting a subset $\tilde{\mathbf{X}}$ from $\mathbf{X}$ so that an objective function $F(\tilde{\mathbf{X}})$ related to classification performance can be optimized:

$$\tilde{\mathbf{X}} = \arg\max_{\mathbf{Z} \subseteq \mathbf{X}} F(\mathbf{Z}). \tag{2.7}$$

Generally, the goal of feature selection is to reduce the feature dimension. According to [39], the frames within a video shot represent a spatially and temporally continuous action, and they share the common visual and often semantic-related characteristics. Since a video shot is characterized both spatially and temporally, a set of key-frames could be sufficient to model the object behavior in a video shot. In this work, we apply feature selection methods to extract video key-frames rather than reducing the

feature dimension. Moreover, by extracting a set of representative key-frames that supports salient and condensed object representation in the feature space, we can obtain a compact set of key-frames and accurate object segmentation simultaneously.

2.2 Coherent Key-frame Extraction and Object Segmentation

With the unified feature space, we will consider cluster divergence measurements, which are often used for feature selection. In this work, we study two different criteria. One is to maximize the average pair wise cluster divergence of video objects in the feature space, the other is to maximize the variance of the mean density of the objects. The whole process can be performed as shown in Fig. 3.2. The input

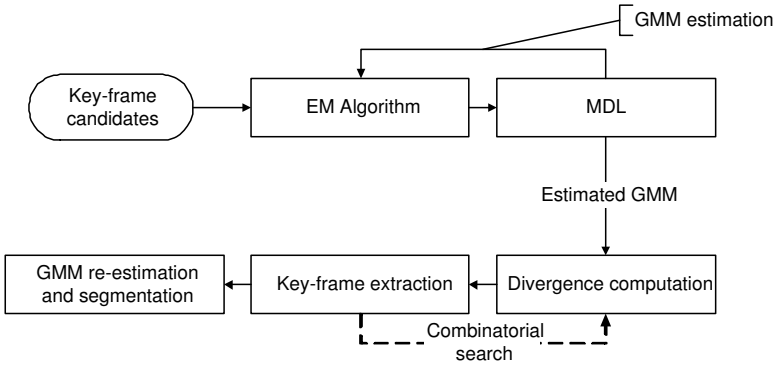


Figure 2.2: The flowchart of the proposed segmentation algorithm.

is a set of key-frame candidates that could be either all frames in a shot, or a set of key-frames initially selected by the method discussed in Section 2.1.1. The GMM is first used to model video objects in the unified feature space, where the EM algorithm associated with the MDL criterion are applied to estimate model parameters. After the GMM estimation, divergence measurements are maximized by searching for an optimal set of key-frames. A combinatorial key-frame search might be used according to the selected criterion where the inter-frame dependencies are considered. Finally, the GMM estimation and object segmentation are performed based on the extracted key-frames. In the next, we will discuss two different criteria for key-frame extraction. The former one considers the dependency between frames, the latter one assumes the independence between frames.

2.2.1 Maximum Average Interclass Kullback Leibler Distance (MAIKLD)

2.2.1.1 Average Interclass Kullback Leibler Distance

Kullback Leibler distance (KLD) measures the distance between two probability density functions [99]. In feature selection, a frequently used criterion is to minimize the KLD between the true density and the density estimated from feature subsets. Nevertheless, this approach aims at minimizing the approximation error rather than extracting the most discriminative feature subsets. Although it is often desired that this criterion can lead to good discrimination among classes as well, this assumption is not always valid [127]. For the purpose of robust classification, divergence-based feature selection criteria are more preferred [127], and the KLD of two densities can be used to measure the cluster divergence between two different clusters in the feature space.

Given two probability density $f_i(x)$ and $f_j(x)$, the KLD between them is defined as:

$$KL(f_i, f_j) = \int f_i(x) \ln \frac{f_i(x)}{f_j(x)} dx, \tag{2.8}$$

KLD is usually not a symmetric distance measurement and is symmetrized by adding $KL(f_i, f_j)$ and $KL(f_j, f_i)$ together:

$$D(f_i, f_j) = \frac{KL(f_i, f_j) + KL(f_j, f_i)}{2}. \tag{2.9}$$

KLD is often used as the divergence measurement of different clusters in the feature space. Ideally, the larger the KLD, the more separability between clusters. If there are M clusters, the average interclass KLD (AIKLD) is defined as:

$$\bar{D} = C \sum_{i=1}^{M} \sum_{j>i}^{M} D(f_i, f_j), \tag{2.10}$$

where $C = \frac{2}{M(M-1)}$. Conventional approaches that reduce the feature dimension based on the maximum AIKLD (MAIKLD) usually have $\bar{D}_0 \leq \bar{D}$, where $\bar{D}_0$ is the AIKLD of clusters in the reduced feature space. As mentioned before, key-frame extraction is formulated as a feature selection process, and we want to extract a set of key-frames where the average pair wise cluster divergence is maximized. Let $\mathbf{X}$ be

the original video shot with N frames and M objects, and be represented as a set of frames $\mathbf{X} = \{\mathbf{x}_i,\ 1 \leq i \leq N\}$ with cardinality $|\mathbf{X}| = N$. Let $\mathbf{Z} = \{x_i^*,\ 1 \leq i \leq N^*\}$ be any subset of $\mathbf{X}$ with cardinality $|\mathbf{Z}| = N^* \leq N$. The objective function is defined as:

$$\tilde{\mathbf{X}} = \arg \max_{\mathbf{Z} \subseteq \mathbf{X}, |\mathbf{Z}| \leq N} \bar{D}_{\mathbf{Z}}, \tag{2.11}$$

where $\tilde{\mathbf{X}}$ is a subset of $\mathbf{X}$ that is optimal in the sense of MAIKLD, and $\bar{D}_{\mathbf{Z}}$ is the AIKLD of M objects within $\mathbf{Z}$ in the 7-D feature space. We have $\bar{D}_{\tilde{\mathbf{X}}} \geq \bar{D}_{\mathbf{X}}$. By extracting frames that contain less aforementioned outliers, it is expected that the cluster overlapping problem can be mitigated.

According to [40], MAIKLD is optimal in the sense of minimum Bayes error. If we use the zero-one classification cost function, then this leads to the maximum *a posteriori* (MAP) estimation. Therefore an optimal solution to (2.11) will result in an optimal subset of key-frames that can minimize the error probability of video object segmentation. Nevertheless, it is usually not easy to find an optimal solution, especially when N is large, and a suboptimal but computationally efficient solution might be preferred in practice.

2.2.1.2 Combinatorial Key-frame Extraction

Feature selection methods have been well studied and some very good reviews can be found in [86, 87]. It is well known that the exhaustive searching method can guarantee the optimality of the feature subset according to the objective function. Nevertheless, the exhaustive searching method is usually computationally expensive and impractical for large feature sets. For example, if a video shot $\mathbf{X}$ has N frames, then the exhaustive search needs to try 2^N possible frame subsets. Various suboptimal approaches are suggested and amongst them a deterministic feature selection method called the *Sequential Forward Floating Selection* (SFFS) method shows impressive performance [130]. When N is not very large, the SFFS method could even provide optimal solutions for feature selection. In this work, we begin with $N' \leq N$ initially key-frame candidates as shown in Fig. 3.2. After the GMM model estimation, key-frame extraction is performed as follows, where the SFFS method is initialized by

using sequential forward selection (SFS):

(1) Start with an empty set $\tilde{\mathbf{X}}$ (no key-frame), and n is the cardinality of $\tilde{\mathbf{X}}$, i.e., $n = |\tilde{\mathbf{X}}|$ and initially $n = 0$;

(2) Based on the MAIKLD criterion, first use SFS to generate a combination that comprises 2 key-frame candidates, and $|\tilde{\mathbf{X}}| = 2$;

(3) Search for one key-frame candidate that maximizes AIKLD when $|\tilde{\mathbf{X}}| = n + 1$, and add it to $\tilde{\mathbf{X}}$, let $n = n + 1$;

(4) If $n > 2$, remove one key-frame candidate from $\tilde{\mathbf{X}}$ and compute AIKLD based on the remained key-frame candidates in $\tilde{\mathbf{X}}$, and go to (5), otherwise go to (3);

(5) Determine if AIKLD increases or not after removing the selected key-frame candidate. If the answer is yes, let $n = n - 1$, and go to (4), otherwise go to (3).

There are a few possible stop criteria for the SFFS method, e.g., the iteration number or the key-frame number. The MAIKLD-based method has several significant advantages: (1) Since the GMM estimation is based on a small number of key-frames, the segmentation is computationally efficient compared with those using all frames [69]. (2) The optimal or near-optimal set of key-frames that maximize AIKLD can be extracted for robust object segmentation. These key-frames could be more representative than those extracted by the method in [111]. (3) The algorithm is very flexible and effective and without significant data-dependent thresholds.

However, there still remains some problems that need further consideration. First, the SFFS method is not efficient enough when N' is very large. Second, the EM algorithm with the MDL criterion for model estimation is time consuming. An alternative approach that makes the algorithm faster is to perform SFFS before the model order estimation, or in other words, the SFFS method is performed based on the estimated GMM using the largest possible number of classes. Then the rest of model estimation can be done based on the extracted key-frames that maximize AIKLD, reducing the computational load tremendously. Nevertheless, this approach

might tend to select some "noisy" frames because within all object subclasses of the largest possible number, some subclasses are from the same object and the MAIKLD criterion could favor those frames with more outliers so that the divergence of these subclasses could be increased. Another alternative approach is to extract key-frames by using the SFS method, which is previously used to initialize SFFS and faster than SFFS. However, it is unable to remove possible redundant key-frame candidates after adding other key-frames. In order to simplify the feature selection process without deteriorate the segmentation performance, we suggest another method that is based on the assumption of frame independence.

2.2.2 Maximum Marginal Diversity

In a recent work [156], a maximum marginal diversity (MMD) criterion based on the *infomax principle* [108] is proposed for efficient feature selection with very simple computation. Under certain constraints, MMD is equivalent to *infomax* that is also optimal in the sense of minimum Bayes error. In this work, we apply MMD for coherent video key-frame extraction and object segmentation. The exhaustive search or the SFFS approach needs to test different combinations of key-frame candidates, while the MMD method only considers the interclass divergence in each key-frame candidate by assuming the frame independence. This considerably reduces the computational load compared with the MAIKLD approach.

The *infomax principle* was originally derived from a viewpoint of neural network where the mutual information (MI) between input and output should be maximized [108]. This principle recommends a system that preserves maximum information about input behavior while reduces the information redundancy to the minimum. In the context of classification, any feature selection method should select certain features that maximize the MI between the features and class labels [156]. When the *infomax principle* is applied to this work, the objective function can be written as:

$$\tilde{\mathbf{X}} = \arg \max_{\mathbf{Z} \subseteq \mathbf{X}, |\mathbf{Z}| \leq N} I(\mathbf{Z}, Y), \tag{2.12}$$

where $\tilde{\mathbf{X}}$, $\mathbf{Z}$ and $\mathbf{X}$ are the same as equation (2.11), and $I(\mathbf{Z}, Y)$ is the MI between

the key-frame subset $\mathbf{Z}$ and class label $Y = \{1, 2, \cdots, M\}$ that is defined as:

$$I(\mathbf{Z}, Y) = \sum_{\mathbf{x}_i \in \mathbf{X}} \sum_{y_j \in Y} p(\mathbf{x}_i, y_j) \ln[\frac{p(\mathbf{x}_i, y_j)}{p(\mathbf{x}_i)p(y_j)}] \tag{2.13}$$

Considering $I(\mathbf{Z}, Y) = H(Y) - H(Y|\mathbf{Z})$, where $H(Y)$ is the entropy of the class label, and $H(Y|\mathbf{Z})$ is the conditional entropy. The *infomax principle* is equivalent to minimize the conditional entropy $H(Y|\mathbf{Z})$. It is shown in [154] that the conditional entropy is a lower bound on the probability of misclassification via Fano's inequality. A relation between the tightest lower bound on the probability of misclassification, i.e., Bayes error, and $H(Y|\mathbf{Z})$ is derived in [156]. This relation indicates that minimum the conditional entropy $H(Y|\mathbf{Z})$ (or the *infomax principle*) is to minimize a lower bound on Bayes error.

From equation (2.13), we have

$$\begin{aligned} I(\mathbf{Z}, Y) &= \sum_{y_j \in Y} p(y_j) KL(p(\mathbf{Z}|y_j), p(\mathbf{Z})) \\ &= E_Y[KL(p(\mathbf{Z}|Y = y_j), p(\mathbf{Z}))] \end{aligned} \tag{2.14}$$

It is derived in [157] that:

$$\begin{aligned} I(\mathbf{Z}, Y) &= E_Y[KL(p(\mathbf{X}|Y = y_j), p(\mathbf{X}))] \\ &= \sum_{i=1}^{N^*} MD(\mathbf{x}_i^*) + \sum_{i=2}^{N^*} I(\mathbf{x}_i^*; \mathbf{x}_{1,i-1}^*|Y) - \sum_{i=2}^{N^*} I(\mathbf{x}_i^*; \mathbf{x}_{1,i-1}^*) \end{aligned} \tag{2.15}$$

where

$$MD(\mathbf{x}_i^*) = E_Y[KL(p(\mathbf{x}_\mathbf{i}^*|Y = y_j), p(\mathbf{x}_\mathbf{i}^*))] \tag{2.16}$$

and $\mathbf{x}_{1,i-1}^* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \cdots \mathbf{x}_{i-1}^*\}$. $MD(\mathbf{x}_i^*)$ is called the marginal diversity (MD) [156], and indicates the variance of the mean density.

MMD only considers the cluster divergence in each frame, and recommends the extraction of key-frame candidates that have the largest MD values. However, only considering the marginal diversity takes a risk of overlooking the joint information between key-frame candidates. The analysis in [157, 158] indicates that the solutions

of MMD and infomax are equal when the mutual information between features is not affected by class labels, i.e.:

$$\sum_{i=2}^{N^*} I(\mathbf{x}_i^*; \mathbf{x}_{1,i-1}^* | Y) = \sum_{i=2}^{N^*} I(\mathbf{x}_i^*; \mathbf{x}_{1,i-1}^*) \tag{2.17}$$

As generalized in [158], this condition is originated from the recent research about image statistics, which suggests that a rough structure of pattern dependencies between some image features follow general statistical laws that are independent of class label. These image features are extracted via various biologically plausible image transforms, such as the wavelet transform. Although this condition is not always strictly held, at least it proves that when the condition of equation (2.17) approximately holds, the MMD approach is near optimal in the sense of minimum Bayes error.

2.2.3 MAIKLD *vs* MMD

Similar to MAIKLD, MMD key-frame extraction is performed after the GMM estimation. During the key-frame extraction process, the MD of video objects in each key-frame candidate is calculated first, then N^* key-frames that have the largest MD are selected as final key-frames. Therefore MMD does not need a combinatorial search in all frames. N^* could be predetermined, or be adaptively determined given a threshold of the MD value. In this simulation, we use the average MD of all key-frame candidates as the threshold. In other words, a key-frame candidate is selected as the key-frame if its MD is greater than the average MD.

The MAIKLD criterion tries to maximize the pair wise inter-class divergence and considers the inter-frame dependencies, while MMD criterion aims at maximizing the variance of the divergence in each individual frame by assuming frame independence. Accordingly, they lead to different key-frame extraction results, although both of them could be lower bounded by the Bayes error. In the context of video object segmentation, MAIKLD could extract more representative key-frames than MMD because maximum divergence variance does not necessarily maximize the pair wise divergence between any clusters, which is expected for object segmentation. Nevertheless, MMD is much faster than MAIKLD because no combinatorial search is necessary.

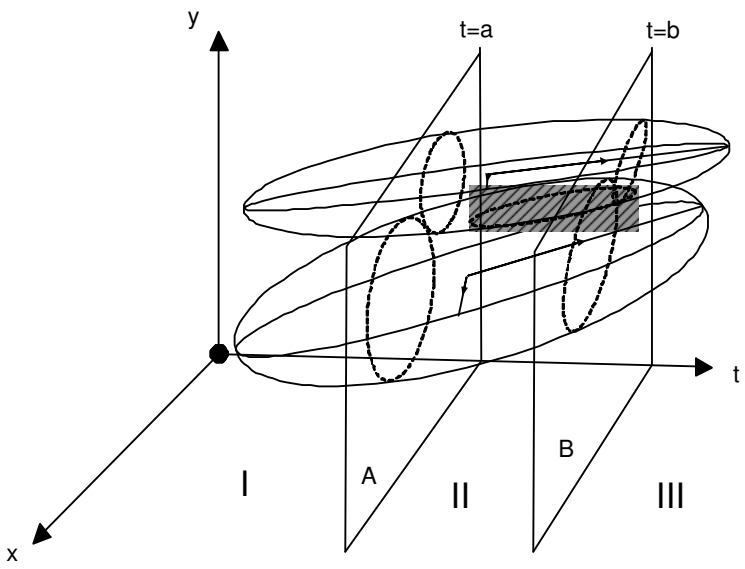


Figure 2.3: Two clusters in the feature space. Axis-t is the time, Axes-x and -y are spatial features. Two slices (frames) at time a and b split the space into three regions, where the clusters in $x - y$ subspace are more separable in regions I and III, and the clusters are overlapped (the shaded area) in region II.

Generally, the difference between MAIKLD and MMD can be illustrated via Fig. 2.3. Axis-t is for time, Axes-x and -y are spatial features characterizing video objects, e.g., spatial location of the objects. There are two clusters corresponding to two video objects in spatio-temporal domain, and the slices (frames) A and B capture the spatial distribution of the two objects at time $t = a$ and $t = b$, splitting the spatio-temporal feature space into three regions. Two clusters are closest to each other with spatial overlapping (the shaded area) in region II. As mentioned before, maximizing AIKLD is equivalent to minimize Bayes error, which is caused by the cluster overlapping in the feature space. Since MAIKLD considers both space and time information of the two objects, this overlapping is mainly characterized by the frames in region II. Consequently, the majority of the extracted key-frames is expected from this region for accurate GMM estimation. On the contrary, MMD does not consider the temporal information between frames, and the frames where two object are well separable in feature plane $x - y$ could be extracted as key-frames, which are mainly located in regions I and III. Therefore, the key-frames extracted by MAIKLD could emphasize more on the inter-relationship between different objects, while those extracted by MMD could mainly highlight the individual object behaviors.

2.3 Performance Evaluation

Numerical criteria are used in this work to evaluate the segmentation performance with respect to all moving objects. For synthetic videos, since the ground truth of object segmentation is available, we calculate segmentation *accuracy*, *precision*, and *recall*. Accuracy means the overall segmentation accuracy regarding all moving objects. Precision is the percentage that the segmented moving objects are true moving objects. Recall shows the percentage that true moving objects can be detected. For the real videos that no ground truth is available, a set of objective measures are used. According to the analysis in [34, 52], these measures include spatial uniformity, temporal stability, and motion uniformity.

2.4 Spatial Uniformity

Spatial uniformity is measured by two methods. One is the texture variance of objects [34]:

$$text\_var(O) = \frac{3 \cdot varY(O) + varU(O) + varV(O)}{5}, \tag{2.18}$$

where $var*(O)$ is the variance of $*$ channel of YUV color space. The other is to measure the spatial color contrast along object boundaries. In this method, we first use morphological dilation and erosion to obtain two video object planes (VOP) with enlarged and diminished objects [1], respectively. After subtracting the VOP with diminished objects from the VOP with enlarged objects, we can have regions E along objects boundaries. Then YUV color histograms are calculated inside and outside objects boundaries within E, and the color contrast is computed as:

$$color\_con(O) = \sum_{B} |HIN_t(E) - HOUT_t(E)|, \tag{2.19}$$

where B indicates all bins of the color histograms, $HIN_t(E)$ and $HOUT_t(E)$ are color histograms inside and outside objects boundaries within E of the tth frame. This approach uses a similar approach as the one suggested in [52], but is much more easy to implement. Obviously, a good segmentation result will lead to a smaller *text_var* and larger *color_con* compared with poor results.

[1] In this work, we only consider moving objects in a video shot.

2.5 Temporal Stability

Temporal stability is tested via three methods. The first two measure the inter-frame difference of the object size and elongation [34]:

$$\begin{aligned} size\_diff &= |area(O_t) - area(O_{t-1})|, \\ elong\_diff &= |\frac{area(O_t)}{(2 \cdot thickness(O_t))^2} - \frac{area(O_{t-1})}{(2 \cdot thickness(O_{t-1}))^2}|, \end{aligned} \tag{2.20}$$

where O_t denotes the objects in the tth frame, and $thickness(O_t)$ counts the number of morphological erosion steps that remove the object O_t. The third method is the temporal color histogram difference [52], where a χ^2 metric is used:

$$\chi^2 = (H_t, H_{ref}) = \frac{1}{N_{H_t} + N_{H_{ref}}} \sum_{i=1}^{B} \frac{[r_1 H_t(i) - r_2 H_{ref}(i)]^2}{H_t(i) + H_{ref}(i)}, \tag{2.21}$$

where $r_1 = \sqrt{\frac{N_{H_{ref}}}{N_{H_t}}}$, $r_2 = \frac{1}{r_1}$, $N_{H_t} = \sum_{i=1}^{B} H_t(i)$, $N_{H_{ref}} = \sum_{i=1}^{B} H_{ref}(i)$, H_t is the YUV color histogram of VOP at time t, H_{ref} is the color histogram of the reference VOP, which is usually set as the VOP of the first frame in the video shot. If the two histograms are identical, $\chi^2 = (H_t, H_{ref}) = 0$, and it tends to be 1 if more differences exist. Consequently, a good segmentation performance should correspond to small $size\_diff$, $elong\_diff$ and χ^2 values.

2.6 Motion Uniformity

Motion uniformity is evaluated via the variance of motion vectors [34], i.e.,

$$motion\_var(O) = varXvec(O) + varYvec(O), \tag{2.22}$$

where $varXvec(O)$ and $varYvec(O)$ are the variances of the motion vectors in x and y direction at a given time. Given the segmentation map of each frame, motion vectors are estimated by the conventional block-matching algorithm [109]. Obviously, a small variance of motion vectors is preferred.

2.7 Simulations and Discussions

In this section, we will study three issues of the proposed methods: (1) the characteristics of extracted key-frames, (2) the necessity and benefit of key-frame

extraction for object segmentation, and (3) the compactness and saliency of the extracted key-frames. In the following, the experimental set-up is introduced first followed by the simulation results and discussions pertaining to the three issues.

2.7.1 Experiment Setup

Three synthetic videos (gray-level), Video-A, Video-B, and Video-C, and three real videos (color), Face, Tachi, and People, are used as shown in Figs. 6.2 and 3.11. When constructing the synthetic videos, we added some additive white Gaussian noise (AWGN). The video frame size is 176×144. Video-A has a rectangular object moving horizontally through two background objects. Video-B has a circular object moves sigmoidally. There are two moving objects in Video-C. One is an elliptic object that is moving diagonally with the size increasing simultaneously, and the other is a rectangular object moving from right to left horizontally. Videos Face and Tachi have some global motion introduced by the camera. Video People has two persons walking toward each other from left and right. In the following, we compare the two suggested methods with those in [69] and [111]. For convenience, we refer to the method in [69] as Method-I with no key-frame extraction. Our previous method in [111] is referred to as Method-II, and two new methods are referred as Method-III (MAIKLD) and Method-IV (MMD), respectively. Simulations are performed on a PC computer with the 3.2GHz Pentium-IV CPU and 1GB memory.

2.7.2 Key-frame Characteristics

We first study the key-frames extracted by Methods-III and IV using Video-A. Each method is controlled to extract 12 key-frames. Specifically, all frames in Video-A are involved as key-frame candidates. As shown in Fig. 2.6, most key-frames extracted by Method-III are those that the objects move close to each other spatially, and key-frames extracted by Method-IV are those that the objects are spatially far away from each other. Since the pixel intensity of the synthetic video is not discriminative enough after adding AWGN. The spatial location of video objects are the major features for object segmentation. Method-III considers both space

and time information of the objects equally. Consequently, in order to well separate the objects, more attention has to be paid when the objects are spatially close to each other. However, Method-IV only consider the cluster separation in each frame by ignoring temporal dependency across frames, resulting in the key-frames where objects are far away from each other spatially. These observations are consistent to our discussion in Section 2.2.3. In addition, we also study the key-frame extracted by Method-II. Since the frame-wise HS color histogram varies slightly across frames in Video-A, Method-II cannot extract salient key-frames to represent significant change of video content, leading to relatively poor object segmentation results, as shown in Figs. 3.13 and 2.8, where Method-I uses all 36 frames, Method-II use 12 key-frames as Methods-III and -IV.

2.7.3 Necessity and Benefit of Key-frame Extraction for Object Segmentation

As we can see from the above simulation, Method-II cannot well segment the moving object based on the key-frames extracted from the color histogram. Methods-I, -III and -IV have similar results while Methods-III and -IV only use 12 key-frames (1/3 of all frames) for object segmentation. We will further manifest that object segmentation using key-frames can have similar or even better performance compared with Method-I if key-frames are appropriately extracted. In order to reduce the computational load, Methods-III and -IV begin with a set of key-frame candidates that are initially extracted via the color histogram. The object segmentation results of the four methods on Videos-B and -C are shown in Figs. 2.10 and 6.4. It can be seen that Methods-II, -III, and -IV outperform Method-I. Method-II still produces good performance here because the spatial overlapping of video objects causes the significant change of the color histogram, leading to a set of salient key-frames. All the observations indicate that key-frame extraction is necessary and beneficial for accurate statistical video modeling and object segmentation. Among three key-frame-based methods, Methods-III and -IV can extract more compact and salient key-frames that support effective object segmentation than Methods-II. In most cases, Method-III outperforms Method-IV due to the consideration of dependency across frame.

2.7.4 Results of Real Video Sequences

We now study Methods-II, -III, and -IV on the three real videos. The number of initial key-frame candidates and finally extracted key-frames are listed in Tab. 3.3. We can see that Methods-III and -IV further reduce the redundancy key-frames for object segmentation. In order to further compare the three methods in terms of the effectiveness of key-frame extraction for object segmentation, we fix the number of extracted key-frames to be the same for the three methods. Specifically, some objective criteria introduced in Appendix are used to evaluate the video segmentation performance. The numerical segmentation results on the three videos are illustrated from Fig. 3.15 to Fig. 2.14, and the mean and variance of each measurement are listed from Tab. 3.4 to Tab. 2.4.

There are two major observations as follows. (1) Methods-III and -IV usually produce more representative and salient key-frame sets for object segmentation. This is supported by both subject and object evaluations. As shown in Fig. 3.17 to Fig. 2.17, the moving objects are effectively segmented from the background with better accuracy. From Fig. 3.15 to Fig. 2.14, as well as Tab. 3.4 to Tab. 2.4, we see that the key-frames extracted by Methods-III and -IV can also lead to numerically improved object segmentation results in terms of temporal stability (smaller $elong\_diff$, $size\_diff$, χ^2), motion uniformity (smaller $motion\_var$), and spatial uniformity (smaller $text\_var$ and larger $color\_con$). (2) It is also interesting to notice that if original data samples are clearly separable in the feature space where the temporal information contributes little to key-frame extraction, e.g., video People, Methods-III and -IV would produce the similar key-frames as well as segmentation results, as shown in Fig. 2.14, Fig. 2.17, and Tab. 2.4. These observations are consistent with our initial motivations and analysis of Methods -III and -IV.

2.8 Summary

This chapter presents a coherent framework for key-frame extraction and object-based segmentation within a video shot. We first define a unified spatio-temporal feature space where video frames and visual objects are represented jointly.

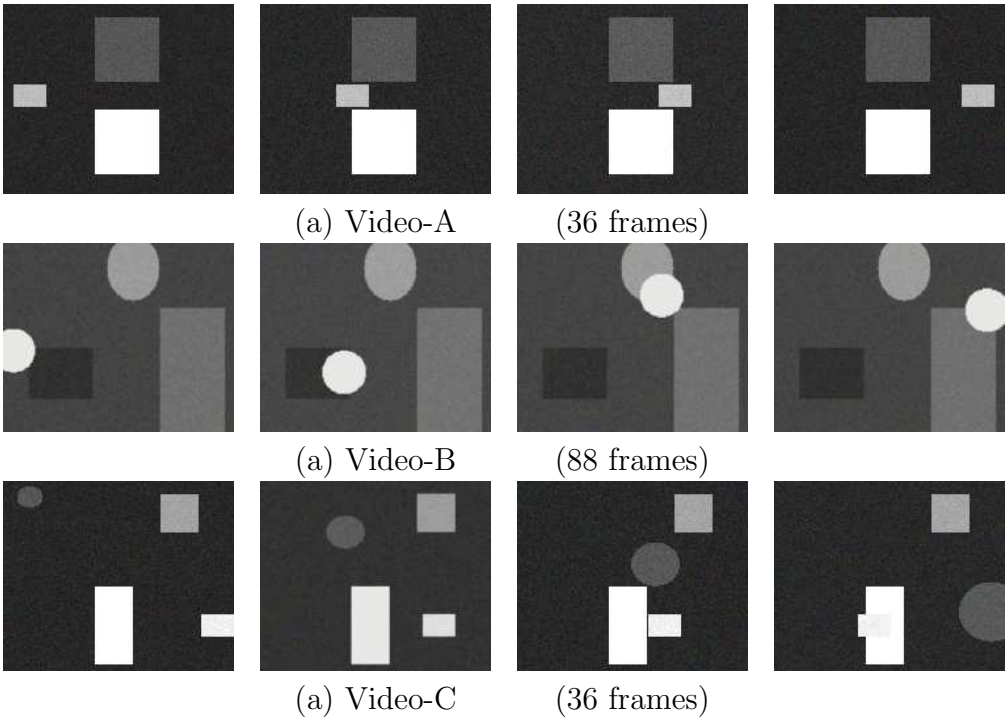


Figure 2.4: A selection of frames in synthetic videos.

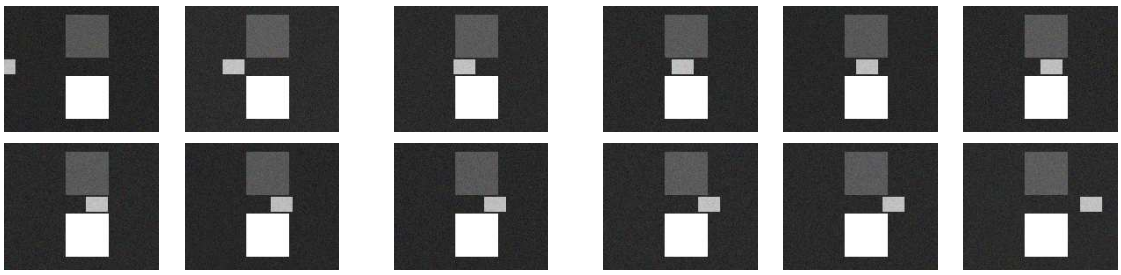
Then key-frame extraction is formulated as a feature selection process that aims at maximizing the cluster divergence in the unified feature space. Specifically, two divergence-based criteria, i.e., MAIKLD and MMD criteria, are used to implement key-frame extraction. In the context of object segmentation, the proposed framework explicitly reveals the inherent relationship between key-frames and objects in a video shot. Compared with the previous methods with and without key-frame extraction, the proposed approaches can provide more robust and accurate object segmentation results, as well as more compact temporal representations of video shots using key-frames. This work also provide a more integrated segmentation scheme to support content-based video analysis.

(a) Face (150 frames)

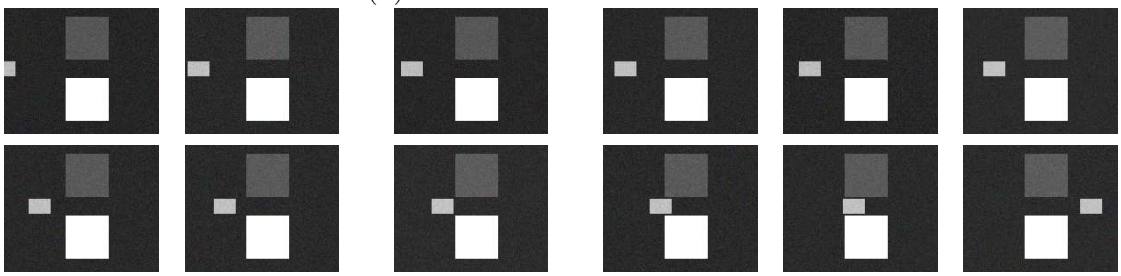
(a) Taichi (358 frames)

(a) People (215 frames)

Figure 2.5: A selection of frames in real videos.

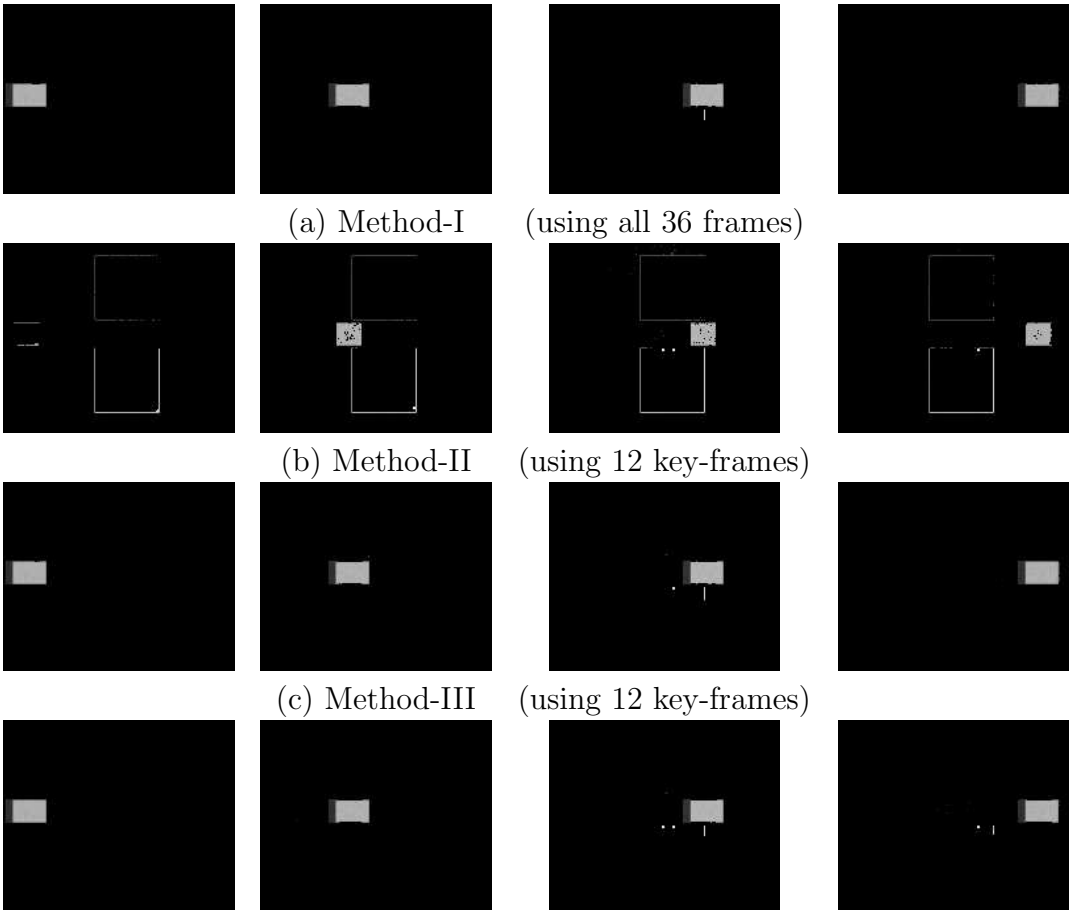


(a) Method-III.



(b) Method-IV.

Figure 2.6: Extracted key-frames (12 key-frames) of Video-A using Methods-III and -IV.



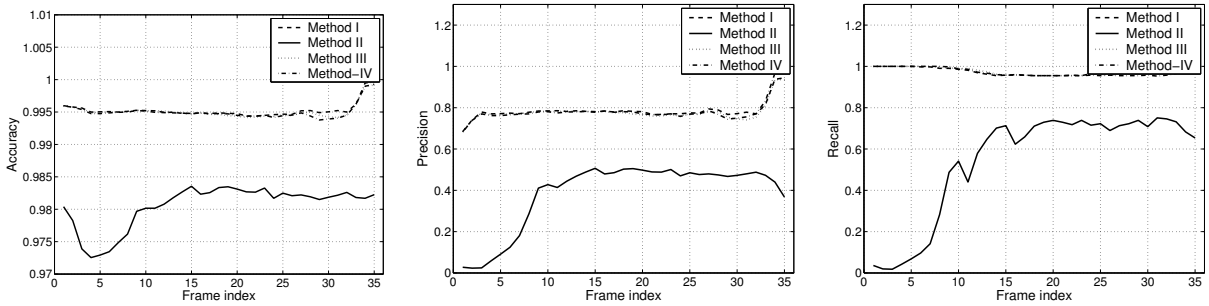
(a) Method-I (using all 36 frames)

(b) Method-II (using 12 key-frames)

(c) Method-III (using 12 key-frames)

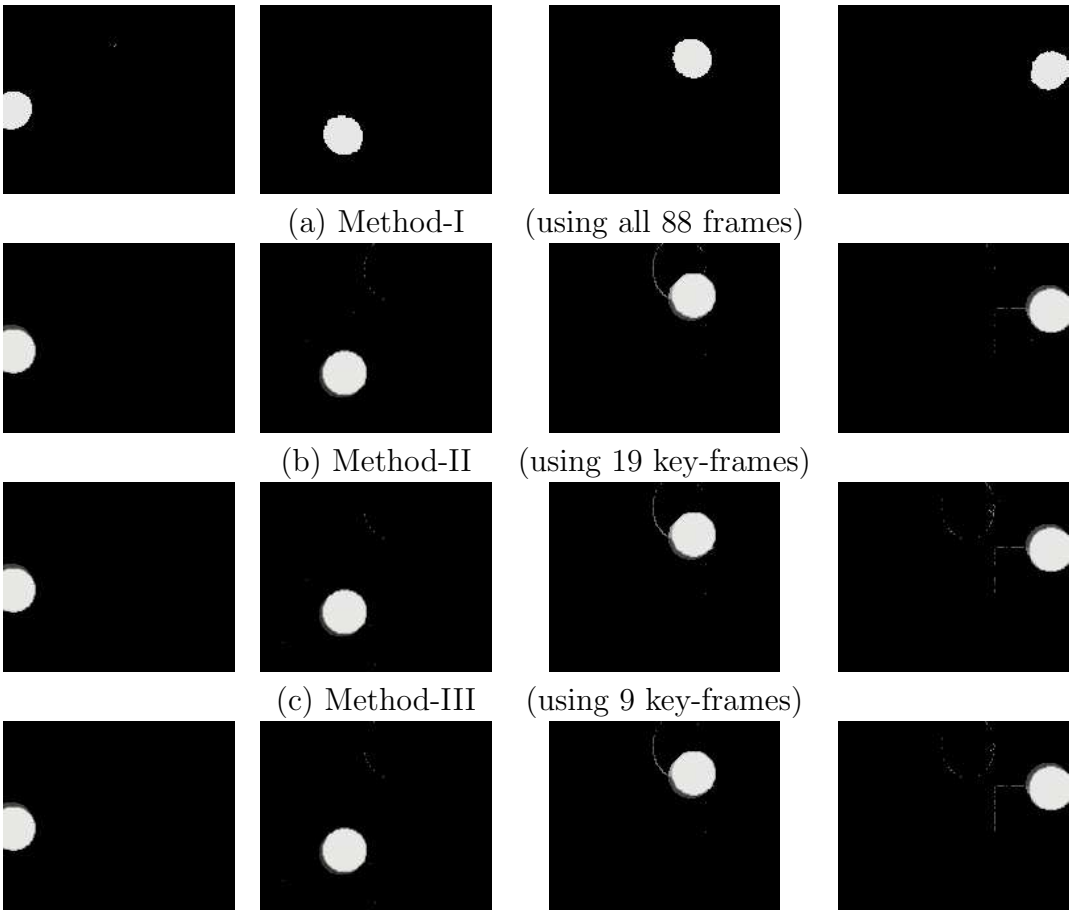
(d) Method-IV (using 12 key-frames)

Figure 2.7: Segmented moving object of Video-A using different methods.



(a) Accuracy (b) Precision (c) Recall

Figure 2.8: Numerical results of Video-A. Dashed, solid, dotted, and dash-dot lines indicate the results of Method-I, -II, -III, and -IV, respectively.



(a) Method-I (using all 88 frames)

(b) Method-II (using 19 key-frames)

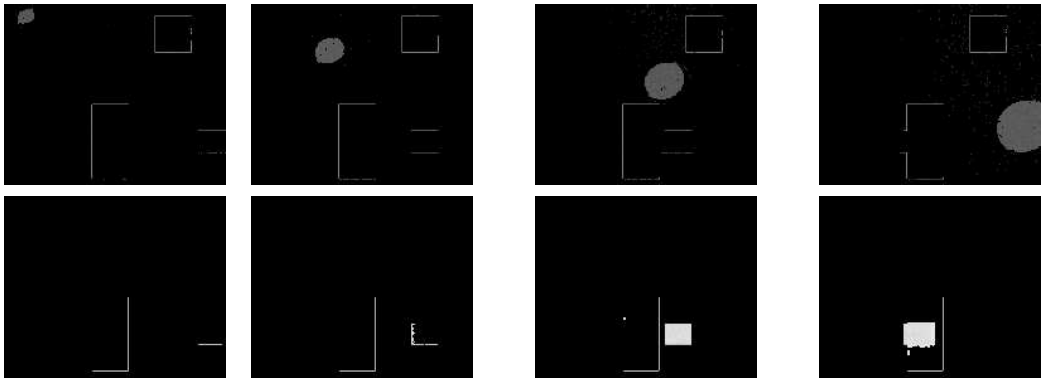
(c) Method-III (using 9 key-frames)

(d) Method-IV (using 9 key-frames)

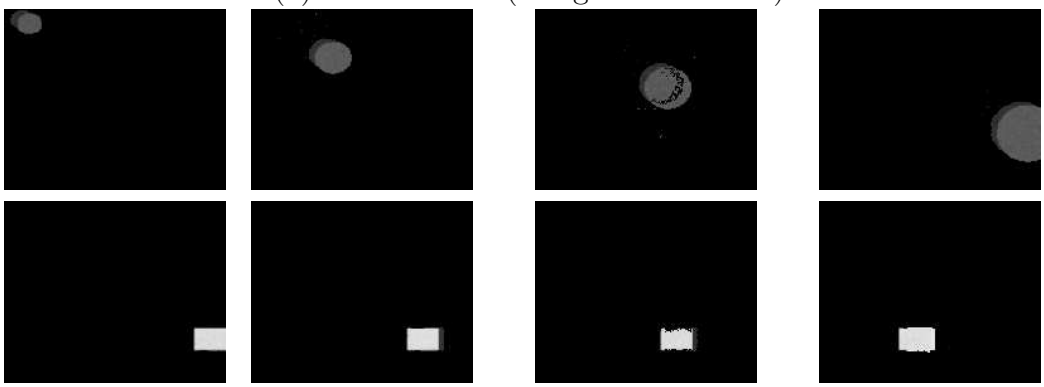
Figure 2.9: Segmented moving object of Video-B.

Table 2.1: Computational loads. NF: The number of used frames; CT: Computational time (seconds); N/A: not available.

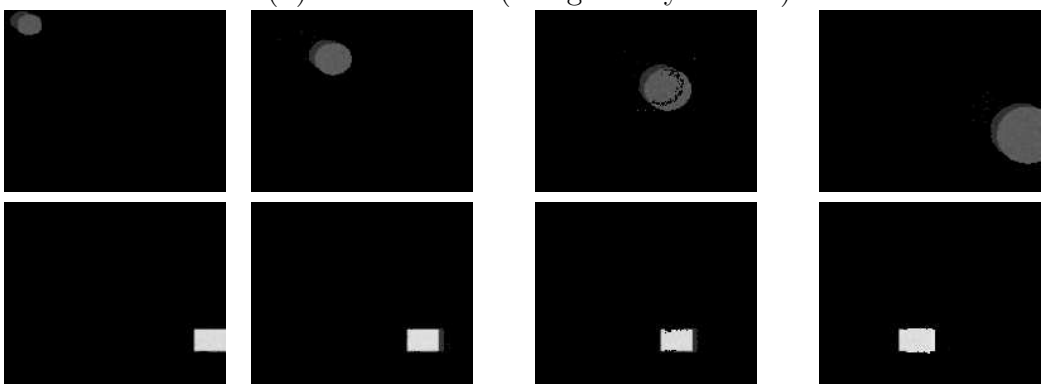
Video sequences	Method-I		Method-II		Method-III		Method-IV	
	NF	CT	NF	CT	NF	CT	NF	CT
Video-B	88	851	19	215	9	278	9	217
Video-C	36	272	17	175	8	223	7	173
Face	150	N/A	16	201	8	261	9	213
Taichi	358	N/A	16	278	8	356	13	290
People	215	N/A	13	190	6	213	9	197



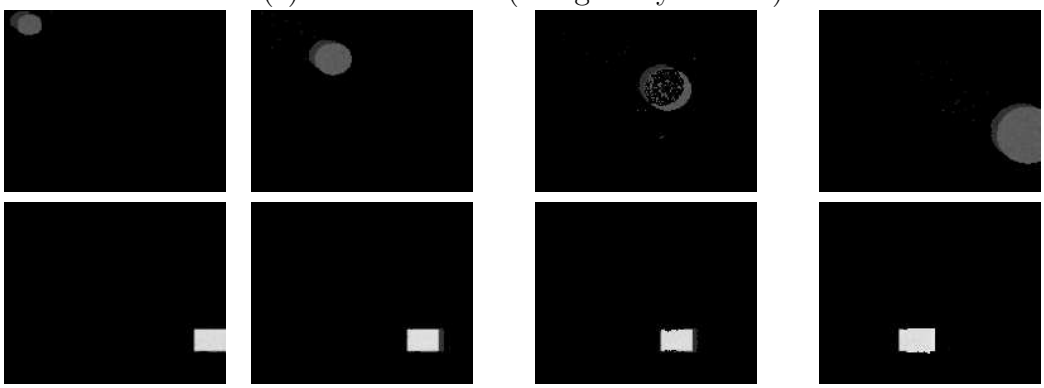
(a) Method-I (using all 36 frames)



(b) Method-II (using 17 key-frames)

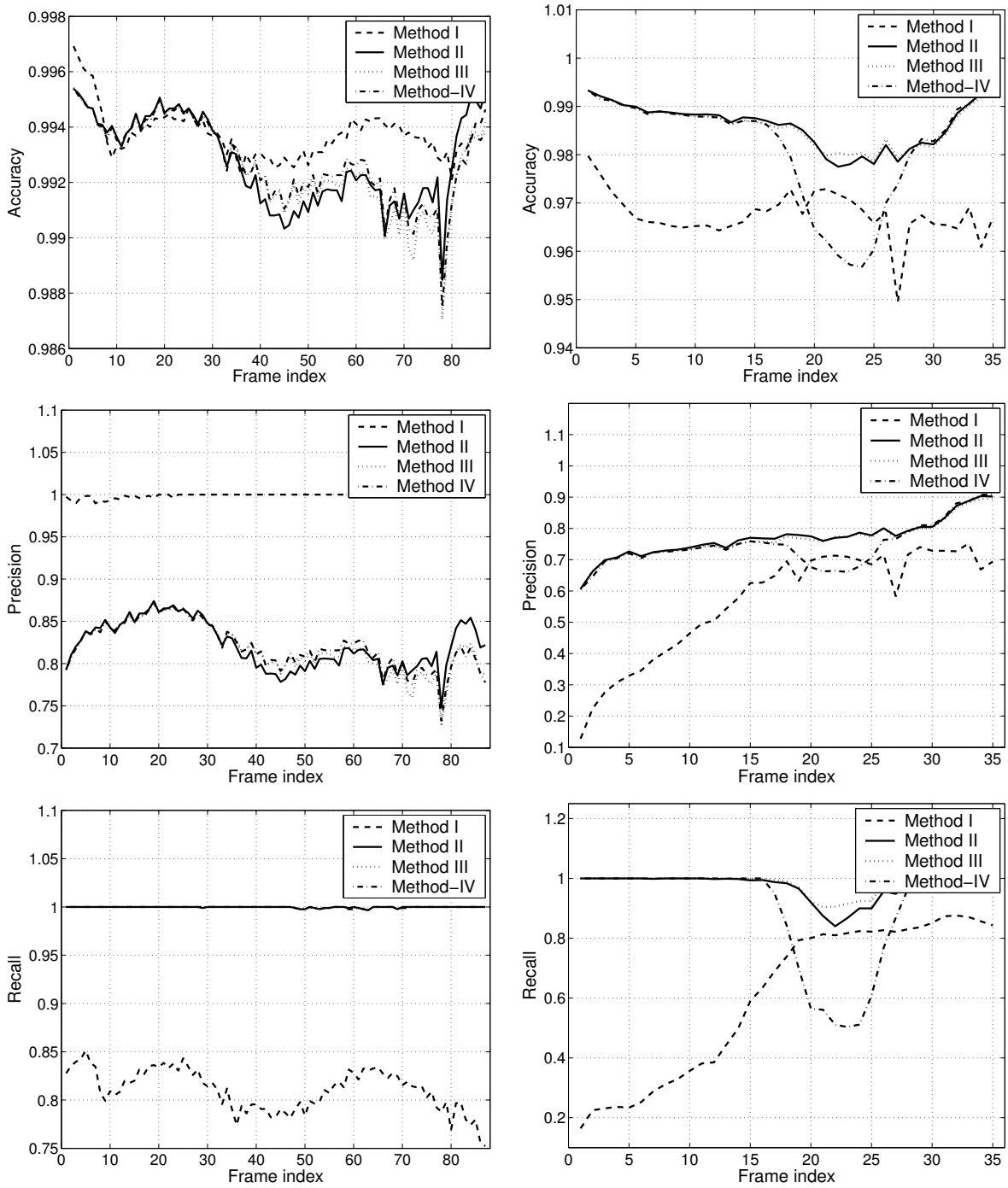


(c) Method-III (using 8 key-frames)



(c) Method-IV (using 7 key-frames)

Figure 2.10: Segmented moving objects of Video-C.



(a) Video-B (b) Video-C

Figure 2.11: Numerical results. Dashed, solid, dotted, and dash-dot lines indicate the results of Method-I, -II, -III, and -IV, respectively.

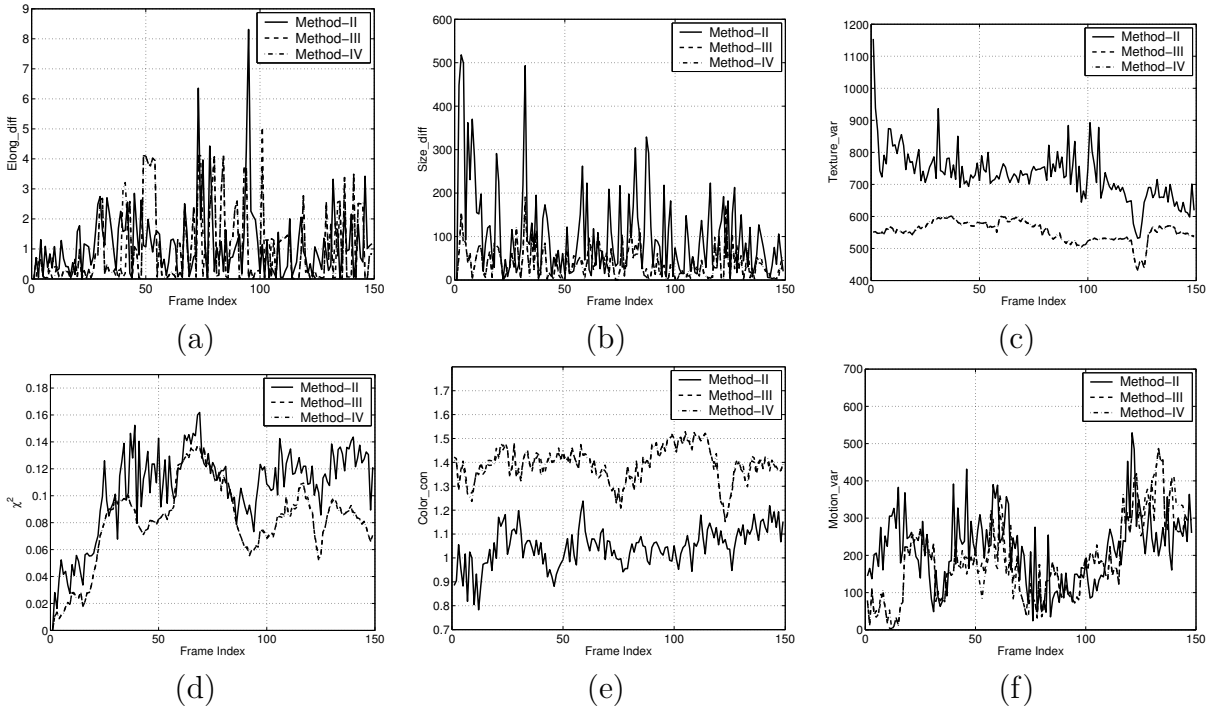


Figure 2.12: Objective evaluation of video Face.

Table 2.2: Numerical performance of video Face.

Measurements	Method-II		Method-III		Method-IV	
	Mean	Var	Mean	Var	Mean	Var
$Elong\_diff$	1.16	1.34	1.0	1.6	1.04	1.67
$Size\_diff$	103.21	1.07e4	39.69	1.28e3	40.06	1.35e3
$Texture\_var$	729.57	6.36e3	552.84	1.04e3	553.29	1.03e3
χ^2	0.11	9.95e-4	0.08	9.31e-4	0.08	9.27e-4
$Color\_con$	1.05	0.007	1.39	0.005	1.39	0.005
$Motion\_var$	214.01	9.12e3	188.13	1.13e4	188.2	1.19e4

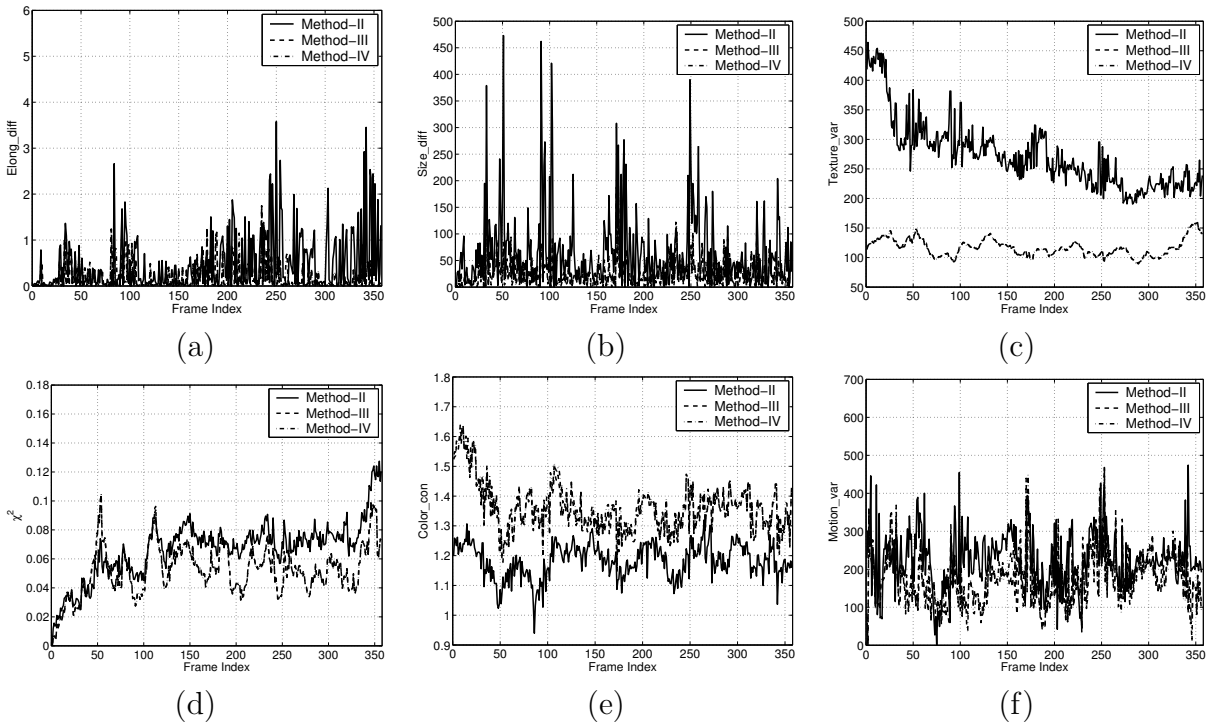


Figure 2.13: Objective evaluation of video Taichi.

Table 2.3: Numerical performance of video Taichi.

Measurements	Method-II		Method-III		Method-IV	
	Mean	Var	Mean	Var	Mean	Var
$Elong\_diff$	0.47	0.39	0.17	0.08	0.19	0.09
$Size\_diff$	63.89	5.25e3	24.91	703.44	25.37	720.81
$Texture\_var$	272.98	3.36e3	116.64	210.41	116.55	209.48
χ^2	0.07	3.93e-4	0.05	2.93e-4	0.05	2.91e-4
$Color\_con$	1.18	0.003	1.36	0.006	1.37	0.007
$Motion\_var$	223.41	5.45e3	171.87	4.66e3	171.83	5.07e3

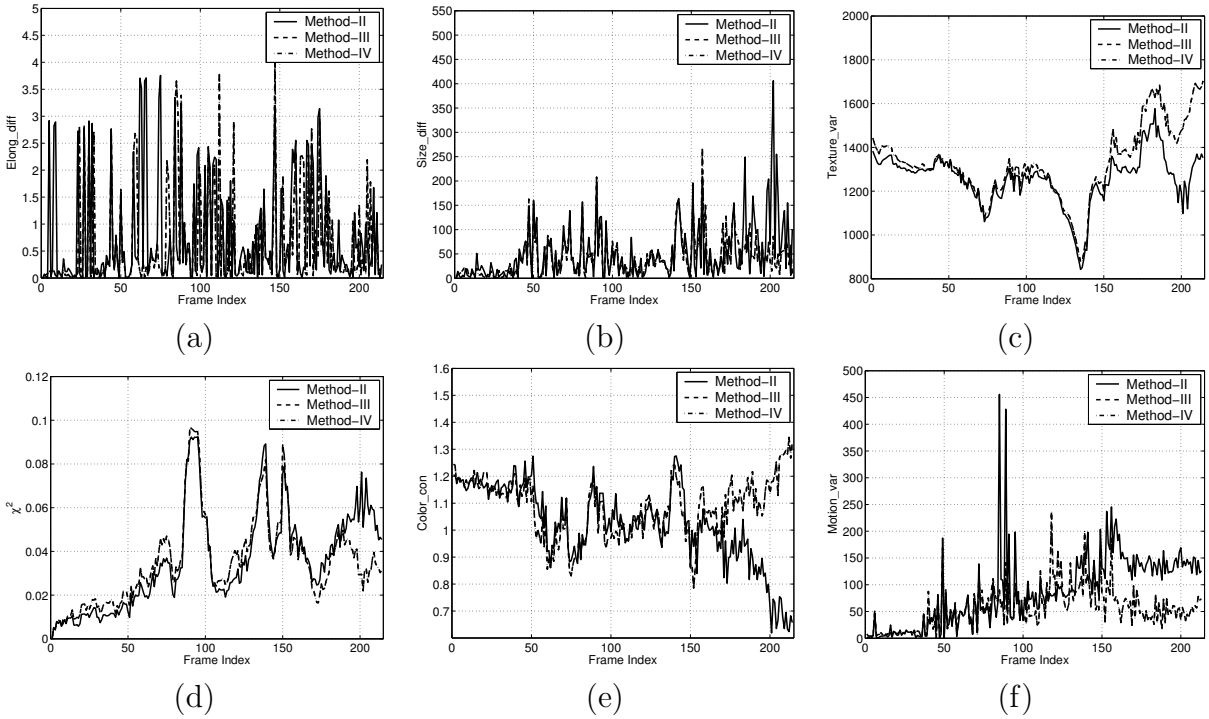


Figure 2.14: Objective evaluation of video People.

Table 2.4: Numerical performance of video People.

Measurements	Method-II		Method-III		Method-IV	
	Mean	Var	Mean	Var	Mean	Var
$Elong\_diff$	0.76	1.01	0.66	0.88	0.66	0.88
$Size\_diff$	51.65	3.41e3	43.22	1.78e3	43.22	1.78e3
$Texture\_var$	1.26e3	1.39e4	1.32e3	2.72e4	1.32e3	2.72e4
χ^2	0.036	5.04e-4	0.035	4.16e-4	0.035	4.16e-4
$Color\_con$	1.02	0.02	1.08	0.01	1.08	0.01
$Motion\_var$	87.52	4.67e3	55.51	1.63e3	55.51	1.63e3

(a) Method-I.

(b) Method-II.

(c) Method-III.

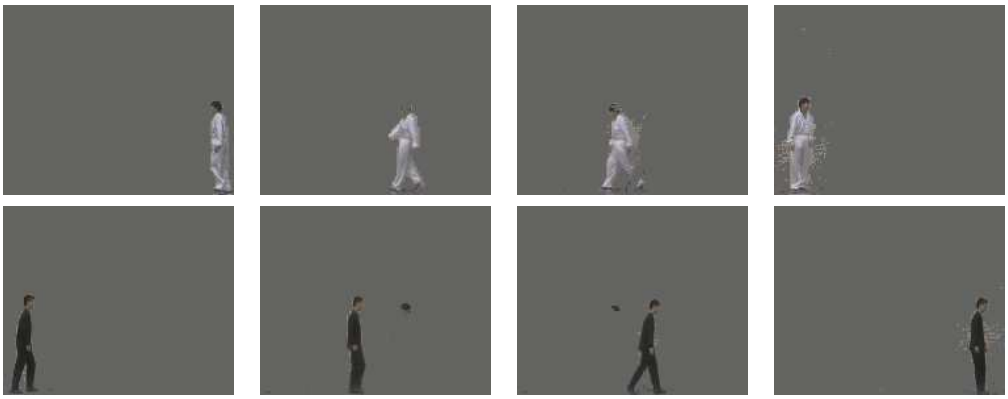
Figure 2.15: Segmentation results of Video-Face using the same number of key-frames (8 key-frames).

(a) Method-I.

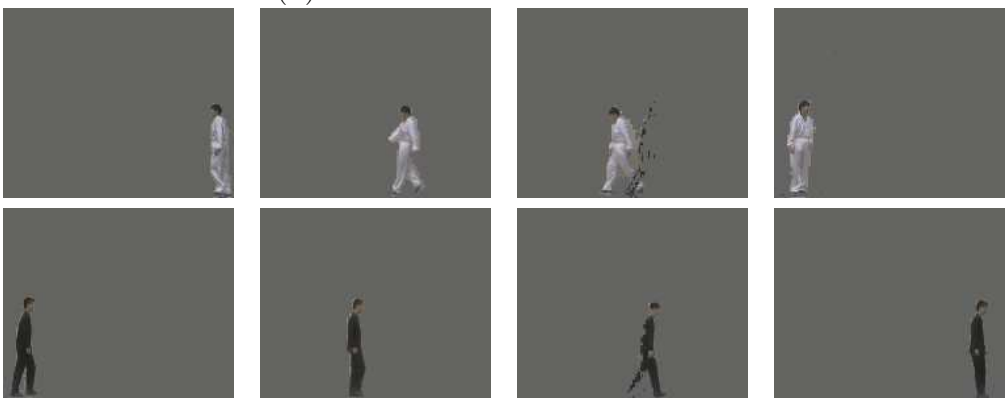
(b) Method-II.

(c) Method-III.

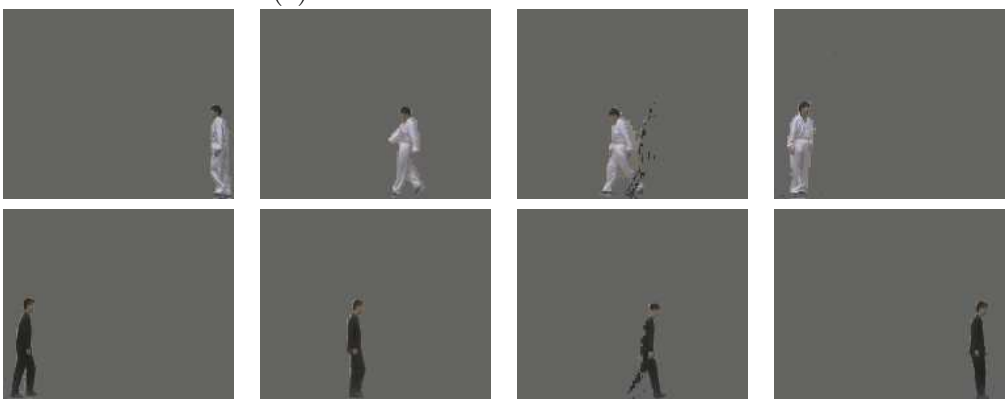
Figure 2.16: Segmentation results video Taichi using the same number of key-frames (8 key-frames).



(b) Method-II.



(c) Method-III.



(c) Method-IV.

Figure 2.17: Segmentation results of video People using the same number of key-frames (6 key-frames).

Chapter 3

VIDEO SEGMENTATION: AN ANALYTICAL METHOD

In Chapter 2, we discuss a novel framework for coherent video key-frame extraction and object segmentation, where two numerical methods are proposed associated with two cluster divergence criteria. Within the numerical methods, the key-frame extraction is performed based on estimated models, where any inaccuracy in model estimation could lead to improper key-frames that could affect the following model re-estimation. In this work, we suggest an analytical method where key-frame extraction is integrated in model estimation. This approach is inspired by a recent work of simultaneous feature selection and model estimation [104, 103], where the contribution of feature subsets is parameterized and estimable during the model estimation. Since key-frame extraction reduce the sample size rather than feature dimension, a different formulation to [104, 103] is derived by formulating the key-frame contribution to model estimation, called frame saliency, as part of model parameters. After model estimation, the frames with the highest saliency are extracted as key-frames.

It has been shown that extracted key-frames could contain certain semantic meaning if motion and/or object information are involved [159, 94, 124]. However, key-frame selection is subjective. Due to various purpose and corresponding criteria, different key-frames would be selected. For example, those of minimum motion are identified as key-frames in [159, 47], while in [122], frames of intensive motion are selected. By exploiting the inherent relationship between key-frames and video objects, we have shown that the numerical methods can provide semantically meaningful key-frames showing spatial interaction between video objects in [148]. In this work, we will show that the estimated frame saliency is associated with object behaviors, resulting in semantically meaningful key-frames, too. Moreover, different

saliency values are associated with different object behaviors, giving us the flexibility to select different key-frames for video browsing. Generally, key-frames either capture the scene of interest or summarize the content of entire video [77]. Particularly, the capability of locating video segments of interest based on semantically meaningful key-frames can facilitate content-based video retrieval and browsing. Based on the both analytical and numerical approaches, a general analysis framework for video representation and description is also suggested to support various description schemes of MPEG-7, where video temporal and spatial analysis are unified from low to high semantic level. Simulations are performed using both synthetic and real video data, and subjectively and objectively evaluated.

The rest of this chapter is organized as follows. In Section 3.1, we introduce a work of simultaneous feature selection and model learning, which inspired this part of work, other preliminary information about GMM-based video object modeling, combined key-frame extraction and object segmentation method, unified feature space, and cluster divergence-based criteria and approaches can be found in Chapter 2. In Section 2.1.4, we introduce the concept of frame/pixel saliency, and derive the analytical approach for coherent key-frame extraction and object segmentation. Section 5.5 shows the simulations and discussions. Final conclusions are made in Section 6.4.

3.1 Simultaneous Feature Selection and Model Learning

An integrated feature selection and GMM estimation method is proposed for unsupervised object segmentation [104, 103], where an important term, i.e., feature saliency, is introduced to describe the contribution of a feature to model estimation. Given data samples represented by a set of d dimensional feature vector, where each feature may have different contribution to model estimation, if redundant features can be removed, the accuracy of model estimation could be improved whilst the computational load will be reduced. In [104, 103], feature saliency is measured by the probability of relevance. A feature is irrelevant if its distribution is independent to class labels, or in other words, its distribution is another probability density rather than the GMM. Within this approach, a component-wise EM (CEM) algorithm is

suggested for model and feature saliency estimation [22, 62], and the minimum message length (MML) criteria is used for model order estimation.

Inspired by the conception of feature saliency, we develop a new analytical method for coherent video key-frame extraction and object segmentation by introducing a measurement of frame relevance to the GMM estimation, where we are not going to reduce the feature dimension d but the sample size.

3.2 Proposed Analytical Method

It has been shown in [151, 150] that the proposed numerical methods can provide more accurate object segmentation results, as well as more salient and compact key-frames compared with the segmentation methods that use all frames within a shot [69] or initial key-frames extracted merely via color histogram. The numerical methods perform key-frame extraction after GMM estimation using all key-frame candidates, and GMM estimation is performed again after final key-frames are extracted. During this process, outliers in key-frame candidates could lead to inaccurate GMM estimation, affecting the following key-frame extraction. In addition, the objective function of MAIKLD and MMD do not have close-form solutions, thus numerical approaches have to be used in [151, 150] to obtain suboptimal or near-optimal solutions via different search methods, which increase the computational load.

In this work, we develop an analytical approach to integrate key-frame extraction as a part of the model estimation. The proposed method is originally inspired by the work in [104, 103], which integrate feature selection and GMM estimation into one process as mentioned in Section 3.1. Since our objective is to reduce key-frame amount rather than to remove redundant features (reduce feature dimension), we have different formulations and solutions to [104, 103] as introduced in the following.

3.2.1 Video Object Modeling

Given a video shot contains N objects, the probability density function (PDF) of video pixel $\mathbf{x}_l$ is formulated as a GMM of N components, i.e., $\mathbf{\Theta} = \{\theta_n, \alpha_n | n =$

$1, ..., N\}$, as:

$$p(\mathbf{x}_l|\mathbf{\Theta}) = \sum_{n=1}^{N} \alpha_n p(\mathbf{x}_l|\theta_n), \tag{3.1}$$

where α_n is the weight of the nth Gaussian characterized by $\theta_n = \{\mu_n, \Sigma_n\}$. If there are L pixels, i.e., $\{\mathbf{x}_l|l = 1, ..., L\}$, $\mathbf{\Theta}$ can be estimated via maximum the model likelihood:

$$\mathbf{\Theta}_{ML} = \arg\max_{\mathbf{\Theta}} \sum_{l=1}^{L} \log p(\mathbf{x}_l|\mathbf{\Theta}), \tag{3.2}$$

The label of each video pixel is represented by a binary vector $\mathbf{y}_l = [y_l^{(1)}, \cdots, y_l^{(N)}]$. If $\mathbf{x}_l$ is from the mth component of the GMM, then $y_l^{(m)} = 1$, and $y_l^{(n)} = 0,\ n \neq m$. The complete log-likelihood is:

$$\log p(\mathbf{X}, Y|\mathbf{\Theta}) = \sum_{l=1}^{L} \sum_{n=1}^{N} y_l^{(n)} \log[\alpha_n p(\mathbf{x}_l|\theta_n)] \tag{3.3}$$

Expectation Maximization (EM) algorithm is often used as a solution to maximum likelihood (ML) estimation of GMM parameters. The E step is to compute a so-called Q-function given the current estimation $\hat{\mathbf{\Theta}}(t)$ and Y:

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}(t)) = E[\log p(\mathbf{X}, Y|\mathbf{\Theta})|\mathbf{X}, \hat{\mathbf{\Theta}}(t)], \tag{3.4}$$

and posterior probability of $y_l^{(m)} = 1$ is estimated as:

$$w_{l,m} = \frac{\hat{\alpha}_m p(\mathbf{x}_l|\hat{\theta}_m(t))}{\sum_{n=1}^{N} \hat{\alpha}_n p(\mathbf{x}_i|\hat{\theta}_n(t))}. \tag{3.5}$$

The M step is to update the parameters by solving:

$$\hat{\mathbf{\Theta}}(t+1) = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}(t)) \tag{3.6}$$

in the case of ML estimation. After the model estimation, grouped feature vectors are characterized by a Gaussian density, and class label of each feature vector can be estimated via the maximum *a posteriori* (MAP) estimation using (3.5).

3.2.2 Frame/Pixel Saliency

Based on the GMM modeling of video objects in the joint spatial-temporal domain, we develop the concept of frame/pixel saliency in this section. As introduced

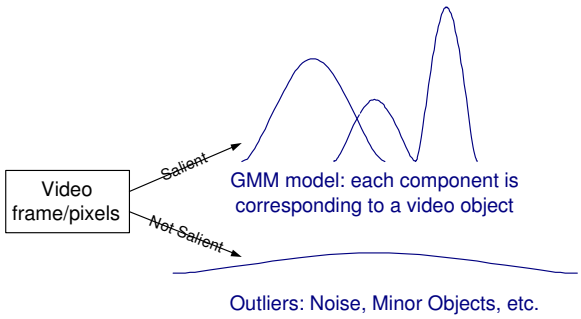


Figure 3.1: Video feature and modeling

in Section 2.1.4, representing video frames and objects in a unified feature space is the first step towards coherent key-frame extraction and object segmentation. Fig. 3.1 (a) illustrates an example. An input shot has N frames with three major objects denoted as objects 1, 2, and 3. The objective of the analytical method is to extract a set of frames that are highly relevant to these three objects, so that accurate object modeling can be achieved.

Given a video shot with N objects, M frames and K pixels in each frame, we define the saliency of the jth frame as: $\phi_j \in \{0,1\}, j = 1, \cdots, M$, where $\phi_j = 1$ means the jth frame is relevant to the GMM for the object segmentation, and $\phi_j = 0$ means this frame is relevant to a class-independent density rather than the GMM as shown in Fig. 3.1. This class-independent model is suggested to characterize aforementioned outliers, and useless data samples such as some background pixels. Similarly, we also define *pixel saliency* as $\phi_i \in \{0,1\}, i = 1, \cdots, MK$, and let $\Phi = (\phi_1, \cdots, \phi_{MK})$ be a binary parameter set for all pixels. Then frame saliency can be obtained by considering all pixels' saliency within this frame. Therefore, given $\Gamma = \{\Theta, \theta_\eta\}$ consisting of GMM Θ and class-independent model θ_η, for pixel $\mathbf{x}_i$, we have the conditional density function as:

$$p(\mathbf{x}_i|\Phi, \mathbf{\Gamma}) = [\sum_{n=1}^{N} \alpha_n p(\mathbf{x}_i|\theta_n)]^{\phi_i} q(\mathbf{x}_i|\theta_\eta)^{1-\phi_i}, \tag{3.7}$$

where $q(\mathbf{x}_i|\theta_\eta)$ is the class-independent density, which is set as a Gaussian density of very large variance in this work, i.e., $\theta_\eta = \{\mu_\eta, \Sigma_\eta\}$. If we let $P_i = P(\phi_i = 1)$, then:

$$\begin{aligned} p(\mathbf{x}_i, \Phi|\mathbf{\Gamma}) &= p(\mathbf{x}_i|\Phi, \mathbf{\Gamma}) * P(\Phi) \\ &= p(\mathbf{x}_i|\Phi, \mathbf{\Gamma}) * P_i^{\phi_i}(1 - P_i)^{1-\phi_i} \end{aligned}$$

$$= [P_i \sum_{n=1}^{N} \alpha_n p(\mathbf{x}_i|\theta_n)]^{\phi_j} [(1-P_i) q(\mathbf{x}_i|\theta_\eta)]^{1-\phi_i}. \quad (3.8)$$

After have all P_is in frame j, frame saliency P_j is determined by averaging all pixel saliency of frame j. Since frame saliency indicated the frame relevancy to the GMM that characterizes the major video objects, the frames with highest saliency values will be finally selected as key-frames for object segmentation.

The GMM and class-independent model are proposed to characterize different parts of the video shot. The GMM is expected to represent the major objects in the feature space, while the class-independent model will capture outliers and insignificant background information. Therefore, this modeling process is in fact a video foreground/background modeling issue that has been widely studied for video object extraction [106, 152, 160], and is extended to coherent key-frame extraction and object segmentation in this work. Since each Gaussian component of the GMM could be associated with a video object, the entry values of Σ_n are less than those of Σ_η, which captures more "noisy" behavior. Therefore, when initializing the EM algorithm, we set larger variance/covariance values for the class-independent density.

3.2.3 A Modified EM Algorithm

In this section we derive an EM algorithm to simultaneously estimate pixel saliency and GMM parameters. Given a pixel $\mathbf{x}_i$ and its class label $y_i = n$, which indicates that it belongs to the Gaussian component θ_n in Θ, its complete-data likelihood is:

$$p(\mathbf{x}_i, y_i = n, \phi_i) = [\alpha_n P_i p(\mathbf{x}_i|\theta_n)]^{\phi_i} [(1-P_i) q(\mathbf{x}_i|\theta_\eta)]^{1-\phi_i}. \quad (3.9)$$

The Q-function is obtained by calculating the expectation of the logarithm of the complete-data likelihood:

$$\begin{aligned} & E[p(\mathbf{X}, \mathbf{Y}, \mathbf{\Phi}|\Gamma)] \\ &= \sum_{n,i,\Phi} p(y_i = n, \Phi|\mathbf{x}_i)[\phi_i \log \alpha_n + \phi_i \log p(\mathbf{x}_i|\theta_n) \\ &\quad + (1-\phi_i) \log(1-P_i) + (1-\phi_i) \log q(\mathbf{x}_i|\theta_\eta)] \\ &= \sum_{n,i} [p(y_i = n, \phi_i = 1|\mathbf{x}_i)(\log \alpha_n + \log P_i + \log p(\mathbf{x}_i|\theta_n)) \end{aligned}$$

$$+p(y_i = n, \phi_i = 0|\mathbf{x}_i)(\log(1 - P_i) + \log q(\mathbf{x}_i|\theta_\eta))]. \tag{3.10}$$

Let $w_{i,n} = p(y_i = n|\mathbf{x}_i)$, $u_{i,n} = p(y_i = n, \phi_i = 1|\mathbf{x}_i)$, and $v_{i,n} = p(y_i = n, \phi_i = 0|\mathbf{x}_i)$, then we have:

$$\begin{aligned} & E[p(\mathbf{X}, \mathbf{Y}, \mathbf{\Phi})] \\ = & \sum_{n,i} u_{i,n} \log \alpha_n + \sum_{n,i} (u_{i,n} \log P_i + v_{i,n} \log(1 - P_i)) \\ & + \sum_{n,i} u_{i,n} \log p(\mathbf{x}_i|\theta_n) + \sum_{n,i} v_{i,n} \log q(\mathbf{x}_i|\theta_\eta). \end{aligned} \tag{3.11}$$

The maximization of the expectation is to maximize the four parts in equation (3.11) separately. Finally, the EM algorithm can be derived as:

E Step:

$$\begin{aligned} a_{i,n} &= p(\phi_i = 1, \mathbf{x}_i|y_i = n) = P_i p(\mathbf{x}_i|\theta_n) \\ b_{i,n} &= p(\phi_i = 0, \mathbf{x}_i|y_i = n) = (1 - P_i) q(\mathbf{x}_i|\theta_\eta) \\ c_{i,n} &= p(\mathbf{x}_i|y_i = n) = a_{i,n} + b_{i,n} \\ w_{i,n} &= p(y_i = n|\mathbf{x}_i) = \frac{\alpha_n c_{i,n}}{\sum_{m=1}^{N} \alpha_m c_{i,m}} \\ u_{i,n} &= p(y_i = n, \phi_i = 1|\mathbf{x}_i) = \frac{a_{i,n}}{c_{i,n}} w_{i,n} \\ v_{i,n} &= p(y_i = n, \phi_i = 0|\mathbf{x}_i) = w_{i,n} - u_{i,n}. \end{aligned} \tag{3.12}$$

and

M Step:

$$\begin{aligned} \alpha_n &= \frac{\sum_i u_{i,n}}{\sum_i P_i} \\ \mu_n &= \frac{\sum_i \mathbf{x}_i u_{i,n}}{\sum_i u_{i,n}} \\ \Sigma_n &= \frac{\sum_i u_{i,n} (\mathbf{x}_i - \mu_n(\theta_n))(\mathbf{x}_i - \mu_n(\theta_n))^T}{\sum_i u_{i,n}} \\ \mu_\eta &= \frac{\sum_i (\sum_n v_{i,n}) \mathbf{x}_i}{\sum_{i,n} v_{i,n}} \\ \Sigma_\eta &= \frac{\sum_i (\sum_n v_{i,n})(\mathbf{x}_i - \mu(\eta))(\mathbf{x}_i - \mu(\eta))^T}{\sum_{i,n} v_{i,n}} \\ P_i &= \sum_n u_{i,n}, \end{aligned} \tag{3.13}$$

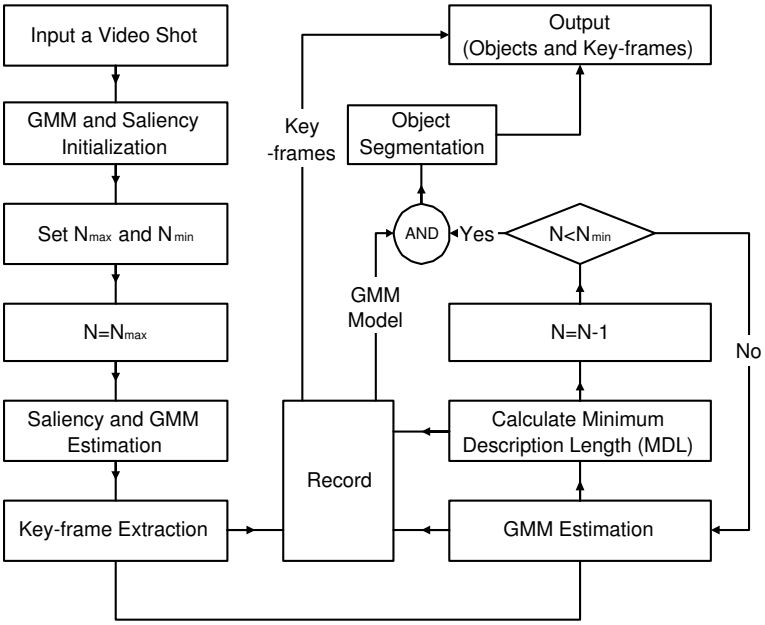


Figure 3.2: The flowchart of the algorithm.

3.2.4 Algorithm Implementation

Given a video shot with M frames, instead of beginning with all frames in this shot, we apply the method in [166, 111] to extract $M^{'} \leq M$ initial redundant key-frame candidates, where a similarity measurement based on the frame-wise 2-D Hue and Saturation (HS) color histogram is used. Pixel saliency and GMM parameters are estimated via the derived EM algorithm, and the MDL criterion is used to estimate model order N. Originated from [136], we have:

$$\hat{\mathbf{\Theta}} = \arg\min_{\Theta} L(\mathbf{\Theta}, N)\{\frac{1}{2}\{N[1+d+\frac{d(d+1)}{2}]-1\}\log(MKd) - \log p(\mathbf{x}|\mathbf{\Theta})\}, \quad (3.14)$$

where d is the feature dimension. Given the largest N value, after the convergence of the EM algorithm, key-frames can be extracted based on their frame saliency. Then the whole process is repeated with $N-1$ based on the extracted key-frames, which considerably mitigate the computational load. During the simulation, we found that most key-frames can be determined with the largest N value, and there is few more frames that could be removed from the key-frame set during the EM iteration when model order is reduced to $N-1$. Therefore, in order to further reduce the computational load, we do not apply the above EM algorithm to all candidate N values.

Instead, after extracting key-frames using the largest N value, we apply the conventional EM algorithms to the following GMM estimation and object segmentation. The flowchart of the whole algorithm is shown in Fig. 3.2.

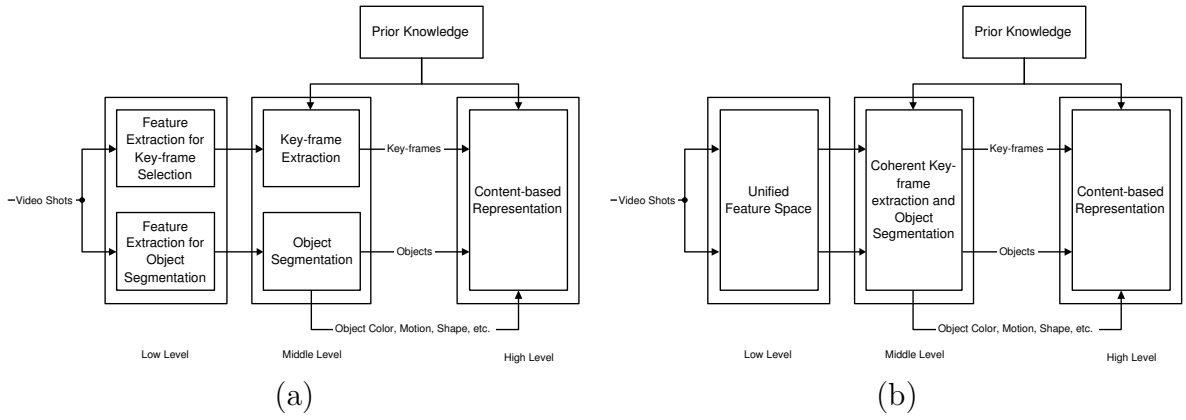


Figure 3.3: Framework.

Based on the coherent segmentation methods, a general analysis framework for video representation and description is suggested as shown in Fig. 5.4 (b), where video key-frame and object analysis are unified from low to high semantic level. As a comparison, Fig. 5.4 (a) shows the conventional video analysis framework where key-frame extraction and object segmentation are implemented separately with different feature sets. Within the unified framework, the unified feature space shown in Fig. 2.1 is first constructed to represent video shot and object on low level of video analysis. Key-frame extraction is performed as a feature selection process for object segmentation on the middle level, and key-frames and objects are finally applied to high level analysis. The unified framework has several advantages: (1). On the low level, feature extraction and representation are efficient. (2). On the middle level, coherent segmentation methods are computationally efficient, providing accurate object segmentation results, and compact, representative, as well as semantically meaningful key-frames. (3). The advantages in (1) and (2) facilitate video representation and description schemes of MPEG-4 and MPEG-7, respectively. The simulation is performed on both synthetic and real videos, we expect the proposed analytical method can achieve similar or better performance compared with the numerical methods.

3.3 Semantically Meaningful Key-frames

3.3.1 Key-frame and Semantic Meaning

In the combined key-frame extraction and object segmentation method [111], extracted key-frames contain very limited semantic meaning. There are two reasons. The first is that frame-wise color histogram provide little information about video object location, motion, and interaction, which are absolutely necessary components for high level video description. The second is that the key-frame refinement is based on the frame-wise comparison of GMMs, where no more content information can be added besides the objects similarity if the relationship between individual objects are not involved. As mentioned in the introduction, key-frame extraction could has semantically meaningful results if motion and/or object information are exploited. It means that we can know certain information about video objects without inspecting the extracted key-frames. Many quality works have been developed to extract key-frames with different semantic meaning, and we only list part of them in Table 3.1. It can be seen that if object-based features are involved, extracted key-frames would contain more information about video content.

Table 3.1: Key-frame characteristics

Key-frame Definitions	Feature Representation	Semantic Meaning
The first frame of a shot [142]	None	Limited
Significant change of ceratin feature [166, 77, 163]	Frame wise color, intensity, texture, etc.	Limited
Significant change of frame wise motion [159, 47, 124, 122]	Motion vector, frame wise difference, etc.	Frame wise motion intensity, irregularity, smoothness or stillness
Appearance of certain object [90, 138, 95, 102]	Object color, texture, shape, motion	Appearance of objects, such as face, skin, etc.
Appearance of certain object behavior [47, 94]	Object position, motion, etc.	Object motion, interaction, etc.

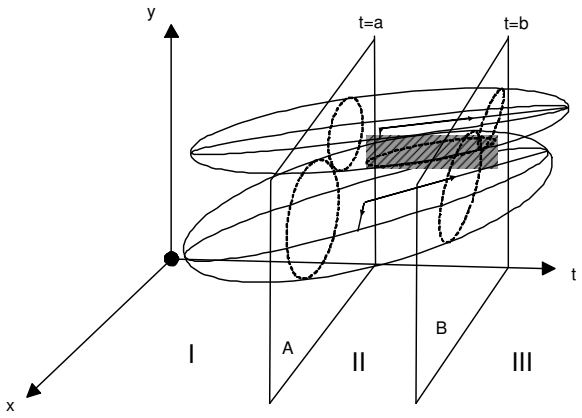


Figure 3.4: Object distribution and interaction in the feature space

3.3.2 Key-frames Extracted by Divergence-based Criteria

Both MAIKLD and MMD criteria measure object divergence based on the characterization of spatial-temporal behavior of video objects, and extract key-frames that contain high level semantic meaning. According to our assumption of video shot in Section **??**, the appearance of major object does not change with in a shot. Consequently, there is no significant change of their color, and extracted key-frames are associated with spatial location of video objects and motion. It has been shown in [148] that MAIKLD extracts key-frames within which major objects are spatially close to each other, while key-frames extracted via MMD are those where objects are spatially far away from each other. In this work, we further exploit the possible motion information contained in key-frames extracted via MAIKLD and MMD. If there are two or more moving objects with nearly constant relative position, the difference between their motion patterns would play more important role than other factors. Therefore, we expect that both MAIKLD and MMD criteria extract key-frames where objects have significant distinction in their motions.

3.3.3 Key-frames Extracted by the Analytical Method

In the analytical method, object modeling is based on the frames with the highest saliency values. On the other hand, estimated frame saliencies not only show the relevance to the GMM, but also imply certain object behaviors, giving us the

flexibility to select different key-frames for video browsing and retrieval:

- Spatially close video objects cause or increase the cluster overlapping in the feature space. If mapping the clusters to the $x - y$ plane, the overlapping regions are most probably located within Gaussian tails, and the probability that they are characterized by the large variance class-independent probability density is high. Therefore, frames with spatially close objects would have low saliency values.

- Intensive/irregular object motion will increase the cluster volume of moving object, causing or increasing cluster overlapping in the feature space. If the relative object distances do not change, intensive/irregular motions could lead to low saliency values.

- Contrary to the above two items, if the objects motion are smooth or spatially far away from each other, high saliency values are expected.

We also illustrate these items in Fig. 3.4, where time (t)and spatial coordinate (x and y) are used to represent the object. As we can see, both spatial closing or intensive/irregular motion will increase the volume of object clusters in the feature space, introducing or increasing the overlapping between clusters as shown in the shaded area of Region II in Fig. 3.4. Simulations on both synthetic and real videos will demonstrate above analysis. Besides these expectations, there exists more potential semantic meaning that can be exploited within the key-frames extracted by the coherent methods. It is worth to mentioning that if we use motion vector rather than pixel-wise frame difference as motion feature, the extracted key-frames can provide more specific information about object behavior. However, this will increase the computation expense.

Generally, key-frames either capture the scene of interest or summarize the content of entire video [77]. Especially, efficiently locating video segments of interest is more interesting and challenging. For example, it is relatively easy to find video sequences of baseball game. However, it is not so effortless to locate clips showing moments of batting. Key-frames implying the interaction between bat and ball will

be very helpful in this case, and the numerical method with the MAIKLD criterion can provide such key-frames. For content-based retrieval and indexing, MPEG-7 proposes approaches to describe and represent visual information by a set of standardized descriptors, which are obtained by video content analysis. After splitting a video scene into shots, video content organization is an important step that helps users group video shots of similar content to increase the efficiency of retrieval and indexing [76, 161]. It is often implemented by content-based matching and classification of shots based on their key-frame similarities regarding visual contents, which are typically represented by color, motion, texture, etc. It is highly expected that representative feature sets about visual content are involved to key-frame extraction so that extracted key-frames are associated with salient/important points of video content. Usually, frame-wise features, such as color histogram, cannot achieve this goal satisfactorily. The coherent key-frame extraction and object segmentation methods represent frames and object in the same spatial-temporal feature space, which characterizes object behaviors joint spatially and temporally. Therefore, extracted key-frames are related to some salient points of video content described by object behavior, facilitating the shot grouping process.

3.4 Simulations and Discussions

3.4.1 Experiment Setup

Simulations are performed on both gray-level synthetic and real video sequences based on a computer with 3.2GHz CPU and 1GB memory. We deliberately add some Additive White Gaussian Noise (AWGN) to the synthetic videos. In the simulation, we compare the proposed analytical approach with the previous methods. Specifically, we denote the method in [111] as Method-I, and two numerical methods as Method-II (MAIKLD) and Method-III (MMD), the proposed analytical method as Method-IV, respectively. The frame size of all the video sequence is 176×144.

3.4.2 Performance Evaluation

Both subjective and objective evaluations are applied to evaluate the segmentation performance regarding moving objects. For synthetic videos, we compute segmentation *accuracy*, *precision*, and *recall* based on the ground truth data. Accuracy is the overall pixel-wise segmentation accuracy regarding all moving objects. Precision shows the pixel percentage that the segmented moving objects are true moving objects. Recall is the pixel percentage that true moving objects can be detected. For the real videos without ground truth, we use a set of objective measures derived from those in [34, 52]. These measures include spatial uniformity, temporal stability, and motion uniformity. The YUV color variance of objects (*text_var*) [34] and the spatial color contrast along object boundaries (*color_con*) [52, 148] are used to measure spatial uniformity. A good segmentation result has a smaller *text_var* and larger *color_con* compared with poor results. Temporal stability is measured by the inter-frame difference of object size and elongation ($size\_diff$ and $elong\_diff$) [34], as well as a χ^2 metric that shows the temporal color histogram difference [52]. A good segmentation performance should have small $size\_diff$, $elong\_diff$ and χ^2 values. The summation of motion vector variance in x and y directions is applied to evaluate motion uniformity [34]. Usually, a small motion variance is related to a smooth motion. More details of these measurements can be found in [34, 52, 148].

3.4.3 Study on Key-frames

Before studying object segmentation, we first demonstrate previous analysis in Section 3.3. An example of object motions using MAIKLD and MMD are illustrated in Fig. 3.5. Fig. 3.5 (a) shows a synthetic video with two moving objects, and Fig. 3.5 (b) is the motion trajectory of the objects. As we can see, there is more motion pattern differences between two objects in the latter part of the video. The motion pattern difference between two objects are increased in the latter part, resulting in more distinction between object models. Therefore, both MAIKLD and MMD criteria extract key-frames majorally located in the latter part of Video-A as shown in Fig 3.6.

Several examples of the analytical method are shown in Fig. 3.7. The columns

from left to right refer to a frame in a synthetic video, object motion trajectory, and average frame saliency, respectively. The first row show a synthetic video with a moving ball, which has two different motion patterns. Compared with the motion in the middle $\frac{1}{3}$ part of the video, the motion in the first and last $\frac{1}{3}$ parts are more intensive and irregular. We calculate the average frame saliency of these three parts, and find that a relatively low average frame saliency implies intensive or irregular object motion, and vice versa. The second row of Fig. 3.7 shows an example of object interaction, where a rectangular object is moving horizontally through two background objects. When the moving object is close to either of two static objects, average frame saliency is low. The last row of Fig. 3.7 illustrates another example of object motion, which is more intensive and irregular in the latter half part of the video than the former part. After calculating the average frame saliency of two different parts, we get the same conclusion as the example in the first row of Fig. 3.7.

This issue is also studied on two real videos as shown in Fig. 3.8. The first row of Fig. 3.8 shows a vehicle running away from the camera. Due to the camera perspective, the vehicle seems to slow down when it is leaving. Therefore, the average vehicle speed in the first half of the video is faster than the latter half. The second row of Fig. 3.8 illustrates two people is walking close to each other with uniform speeds. The average spatial distance between two people in the first half of the video is less than that of the second half. The average frame saliency of these two real videos are shown in Fig. 3.9. As we can see, fast object motion or small spatial object separability is related to small saliency.

3.4.4 Synthetic Videos

Simulations on object segmentation are first performed on the synthetic videos as shown in Figs. 6.2 In Fig. 6.2, Video-B shows a circular object moving sigmoidally. There are two moving objects in Video-C, one is an elliptic object that is moving diagonally from the top-left to the bottom-right, increasing size simultaneously. The other is a rectangular object that is moving leftward. All methods begin with a set of key-frame candidates that are initially extracted via the color histogram [166]. Fig. 6.4 and Table 3.2 show the numerical results of object segmentation, and Table

3.3 shows the computational load, including original video frame number, and the number of key-frames that are finally extracted for the model estimation and object segmentation, and computation time. In order to show the significant decreasing of computation time, we also show the computational load of the method in [69].

For Video-B, all four methods have similar performance regarding accuracy, precision, and recall rate of moving object segmentation, and Method-IV (analytical method) slightly outperform other three methods. Comparing their computational load, Method-IV uses the least number of key-frames for object segmentation, and the least computation time. For Video-C, Method-IV has the highest accuracy and precision rate, but the lowest recall rate. This means that it under-detect the moving object. It also use the least number of key-frames and the least computation time compared with the others. The segmentation results are shown in Figs. 3.13 and 3.14.

Generally, even though less number of key-frames are used, by exploiting the inherent relationship between key-frames and objects, Methods-II, -III, and -IV can provide better segmentation results than Method-I. Moreover, simulation results also show that the analytical method can have similar or better performance using less key-frames compared to numerical methods. This implies that the analytical method can provide compact and representative key-frames sets for object segmentation. This also validates the introduction of frame/pixel saliency and the proposed model.

3.4.5 Real Videos

We also compare Methods-I, -II, -III, and -IV using two real video sequences as shown in Fig. 3.11. In order to compare the four methods in terms of the effectiveness of key-frame extraction for object segmentation, we fix the number of extracted key-frames to be the same for all four methods. The aforementioned objective criteria are used to evaluate the video segmentation performance. The numerical results on the two videos are illustrated in Figs. 3.15 and 3.16, and the mean and variance of each measurement are listed from Table 3.4 to Table 3.5. Final segmentation results of the moving objects are illustrated in Figs. 3.17 and 3.18. As we can see, Methods-II, -III, and -IV outperform Method-I in terms of temporal stability (smaller

$elong\_diff$, $size\_diff$, χ^2), motion uniformity (smaller $motion\_var$), and spatial uniformity (smaller $text\_var$ and larger $color\_con$). Compared with Methods-II and -III, Method-IV provides similar or even better performance. For example, in video Face, Method-IV can correctly separate the bow tie from the moving face, which cannot be done by all three other methods, leading to the significant improvement (much smaller) with respect to $text\_var$. In video Taichi, three coherent methods have similar performance with respect to the objective evaluations, and Method-IV slightly outperforms Methods-II and -III in all evaluation items except for $Elong_{diff}$. These results further indicate the affectivity of the proposed analytical method.

3.4.6 More Discussions

During the simulation, we found some issues that need further study.

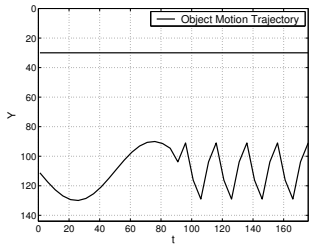
- **Motion features**: In the spatial-temporal feature space, motion information is very important to represent object behaviors. However, pixel-wise frame difference is not enough for accurate motion description because it cannot directly show the motion intensity and direction. In addition, when using pixel-wise frame difference, the newly revealed and concealed background regions around the moving object boundaries within the adjacent pairs of frames would be easily misclassified during the object segmentation. If we replace it by motion vector, we can obtain better object representation, leading to better object segmentation results, and richer semantic meaning within key-frames. The problem is that the motion vector estimation is not so computationally efficient as the calculation of frame difference. We are studying how to extract representative motion feature with acceptable computational load.

- **Intelligent initialization**: How to properly initialize the class-independent probability density is another interesting issue. Frame/pixel saliency is a probability measure and we hope that $P_i \geq 0.5$ would be a proper implication for a salient frame or pixel. However, In the simulation, with different initialization of the class-independent probability density, the same frame has different salient value. In this situation, relative comparison of frame saliency is more

reasonable. Consequently, we have to extract key-frames that have the largest salience values rather than using $P_i \geq 0.5$. Although the extracted key-frames are most salient ones, this introduce an implicit threshold that is not expected according to the modeling. We are trying to develop an adaptive way to solve this problem.

- **Piecewise video analysis**: In order to reduce the computational load when we are dealing with a long video shot, e.g., video Taichi, the frame-wise color-histogram is used to initially extract a set of key-frame candidates, and the coherent key-frame extraction and object segmentation are performed based on these key-frame candidates. Since this initialization could ignore some frames that could be significant for object segmentation, the finally extracted key-frames might not be so representative as those extracted based on the whole shot. Nevertheless, the processing of the whole long shot needs tremendous computational source and time, and is not realistic for real applications. Moreover, GMM may not be efficient enough to characterize video objects within a long shot. Object occlusion, nonlinear and irregular motion patterns, and outliers could affect the integrity and accuracy of object characterization in the spatial-temporal feature space, resulting in inaccurate or fragmented segmentation results. The work in [70] propose a piecewise approach to approximate complex object behaviors. We could also use a piecewise approach, where a video shot is first splitted into much smaller segments, and the coherent key-frame extraction and object segmentation could be implemented in each segments parallelly. This would lead to more representative key-frames and robust object segmentation with high computational efficiency. To split video and combine segmentation results with certain criteria is another interesting research for this purpose.

(a) Video-A



(b) Object trajectory of Video-A

Figure 3.5: Synthetic Video-A

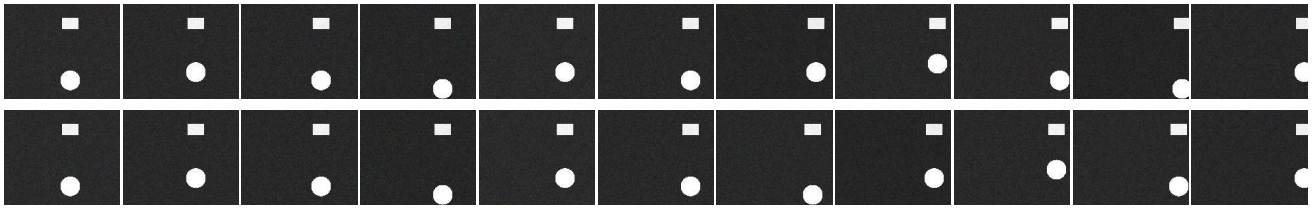
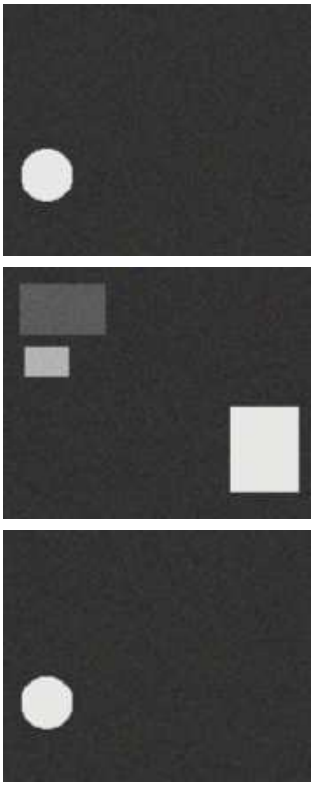
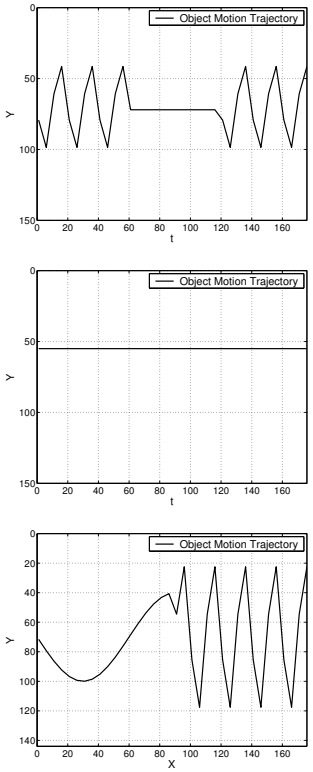


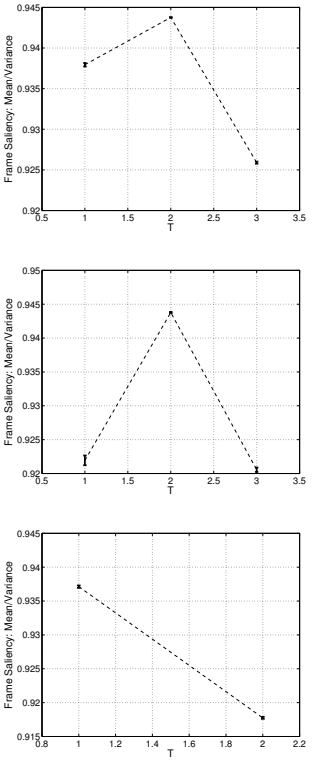
Figure 3.6: Extracted key-frames (11 key-frames) of Video-A using Methods-II (MAIKLD, the first row) and -III (MMD, the second row).



(a) Videos

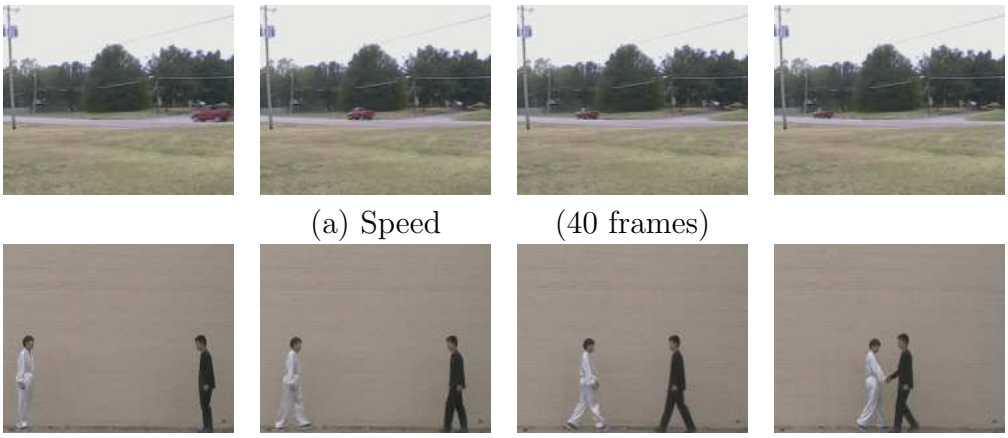


(b) Object trajectory



(c) Mean and variance of frame saliency

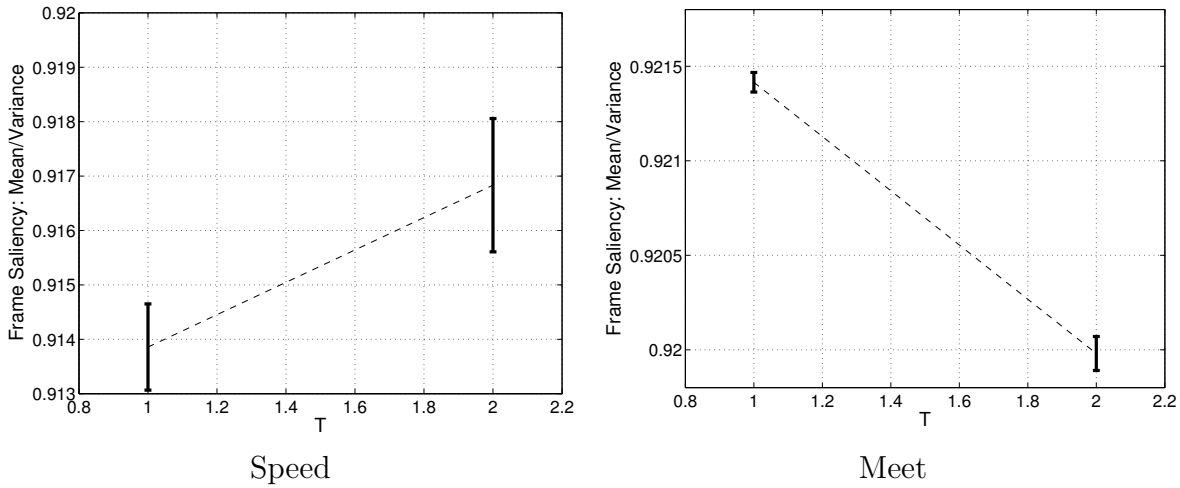
Figure 3.7: Frame Saliency and Object Behavior: synthetic videos



(a) Speed (40 frames)

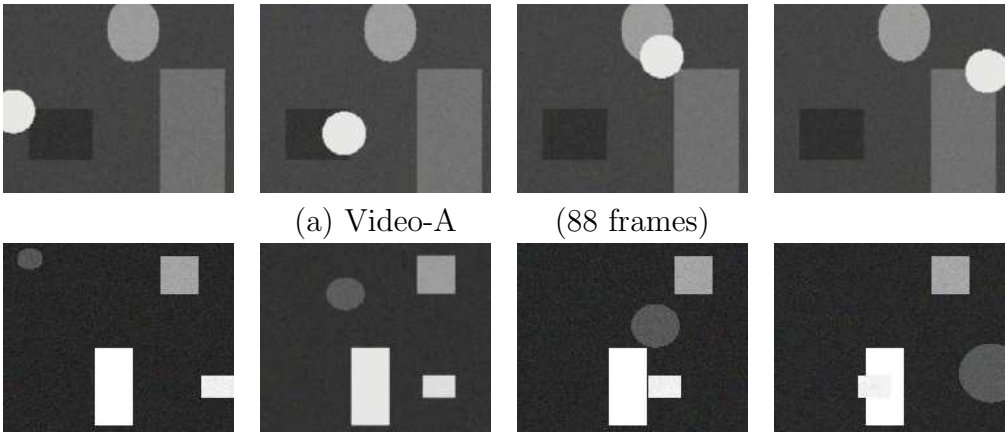
(b) Meet (40 frames)

Figure 3.8: A selection of frames in real videos.



Speed Meet

Figure 3.9: Frame Saliency and Object Behavior: real videos



(a) Video-A (88 frames)

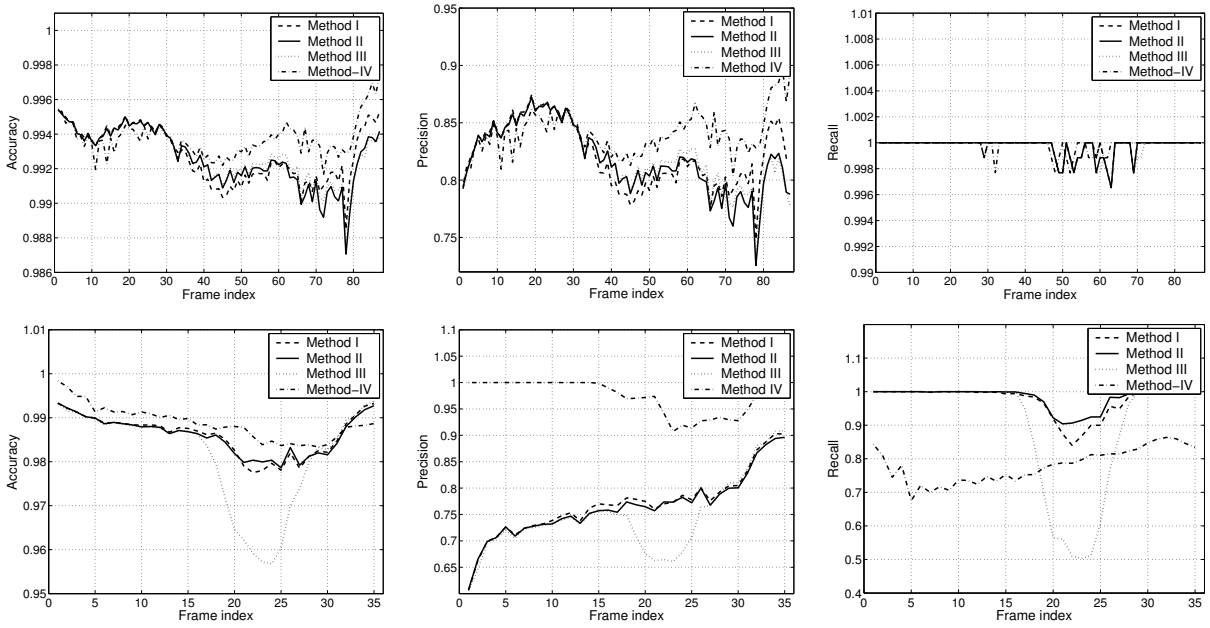
(a) Video-B (36 frames)

Figure 3.10: A selection of frames in synthetic videos.

(a) Face (150 frames)

(a) Taichi (356 frames)

Figure 3.11: A selection of frames in real videos.



(a) Accuracy (b) Precision (c) Recall

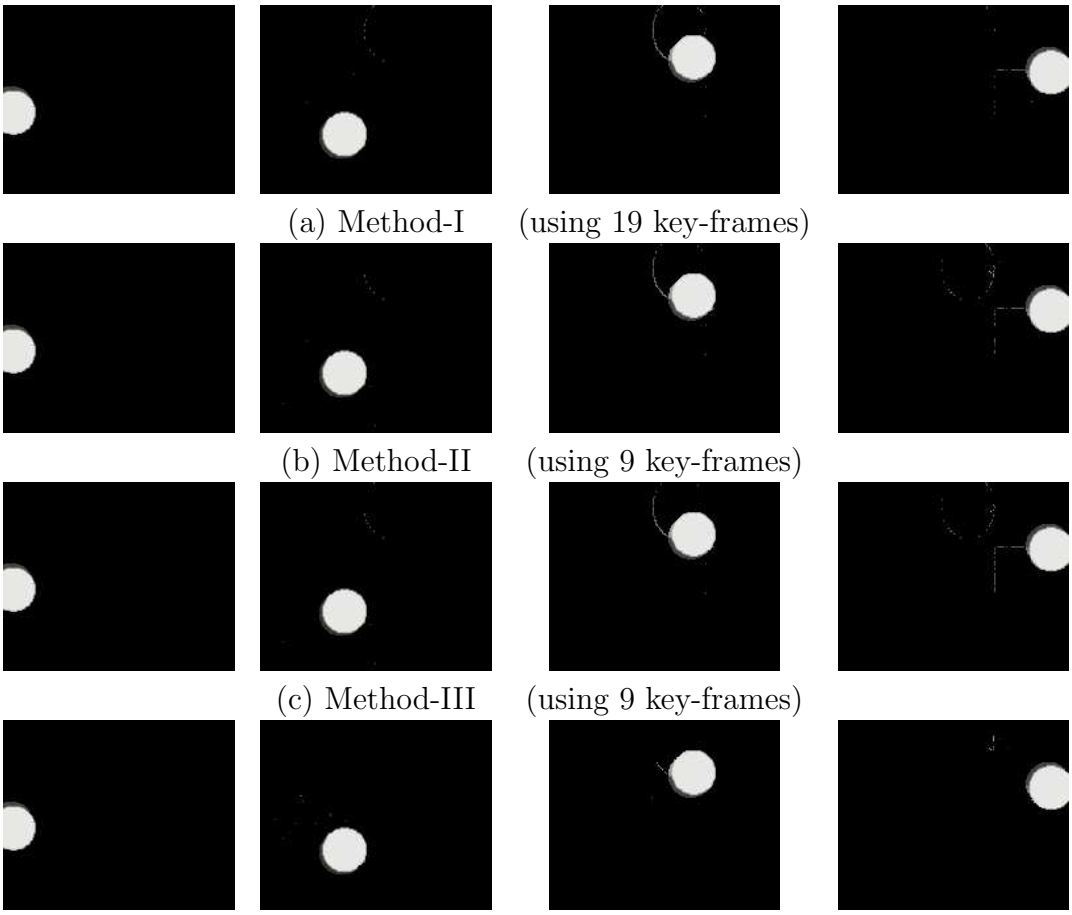
Figure 3.12: Numerical results of Videos-B and -C. Dashed, solid, dotted, and dash-dot lines indicate the results of Method-I, -II, -III, and -IV, respectively.

Table 3.2: The performance of video segmentation (%).

Video sequences		Method-I		Method-II		Method-III		Method-IV	
		Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv
Video-B	Accuracy	99.28	0.16	99.26	0.16	99.28	0.15	99.37	0.11
	Precision	82.16	2.8	81.83	2.9	82.08	2.7	84.05	1.9
	Recall	99.97	0.07	99.97	0.07	99.97	0.07	99.98	0.05
Video-C	Accuracy	98.61	0.47	98.61	0.43	98.16	1.14	98.88	0.39
	Precision	76.92	6.15	76.46	5.98	74.52	7.3	97.29	3.12
	Recall	97.45	4.5	98.29	3.17	89.55	17.74	78.02	5.18

Table 3.3: Computational load. NF: The number of used key-frames. CT: Computation time (s)

Video sequences	Greenspan		Method-I		Method-II		Method-III		Method-IV	
	NF	CT	NF	CT	NF	CT	NF	CT	NF	CT
Video-A (88 frames)	88	851	19	215	9	278	9	217	4	184
Video-B (36 frames)	36	272	17	175	8	223	7	173	5	172



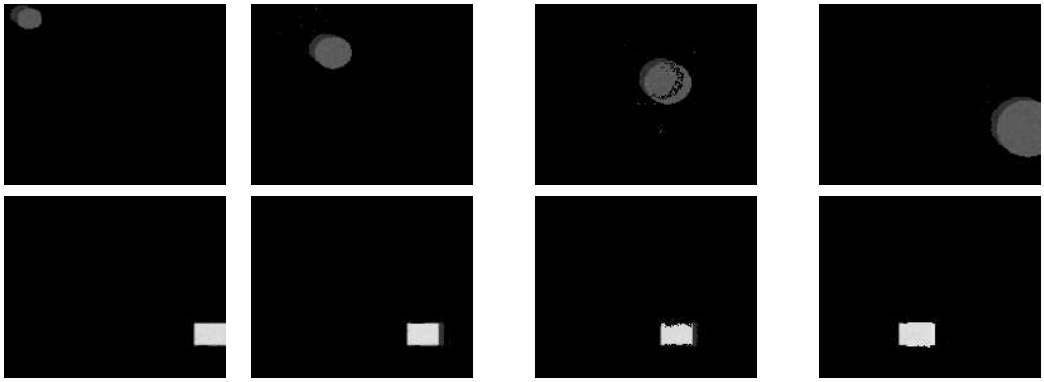
(a) Method-I (using 19 key-frames)

(b) Method-II (using 9 key-frames)

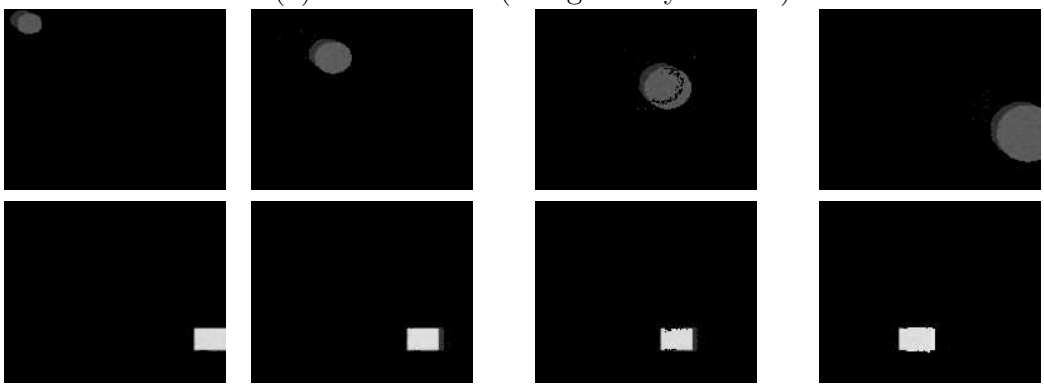
(c) Method-III (using 9 key-frames)

(d) Method-IV (using 4 key-frames)

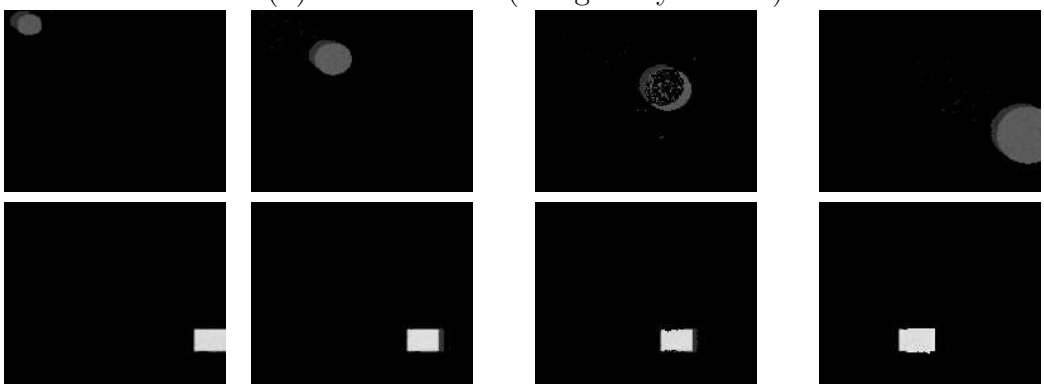
Figure 3.13: Segmented moving object of Video-A.



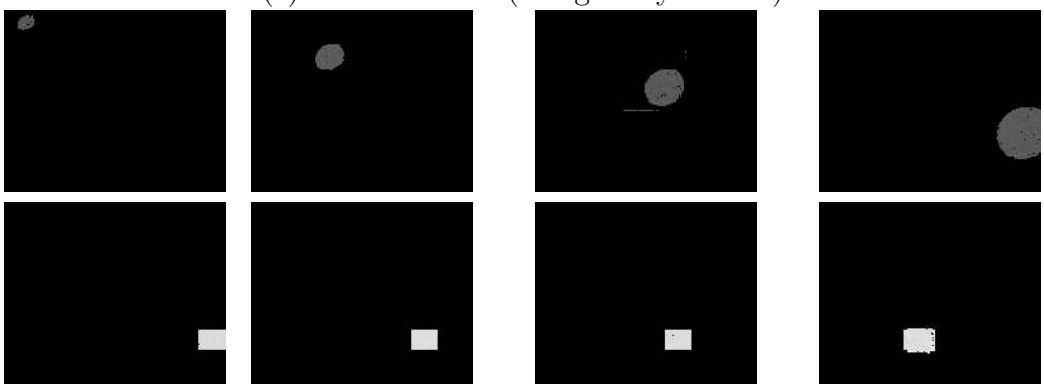
(a) Method-I (using 17 key-frames)



(b) Method-II (using 8 key-frames)



(c) Method-III (using 7 key-frames)



(c) Method-IV (using 5 key-frames)

Figure 3.14: Segmented moving objects of Video-B.

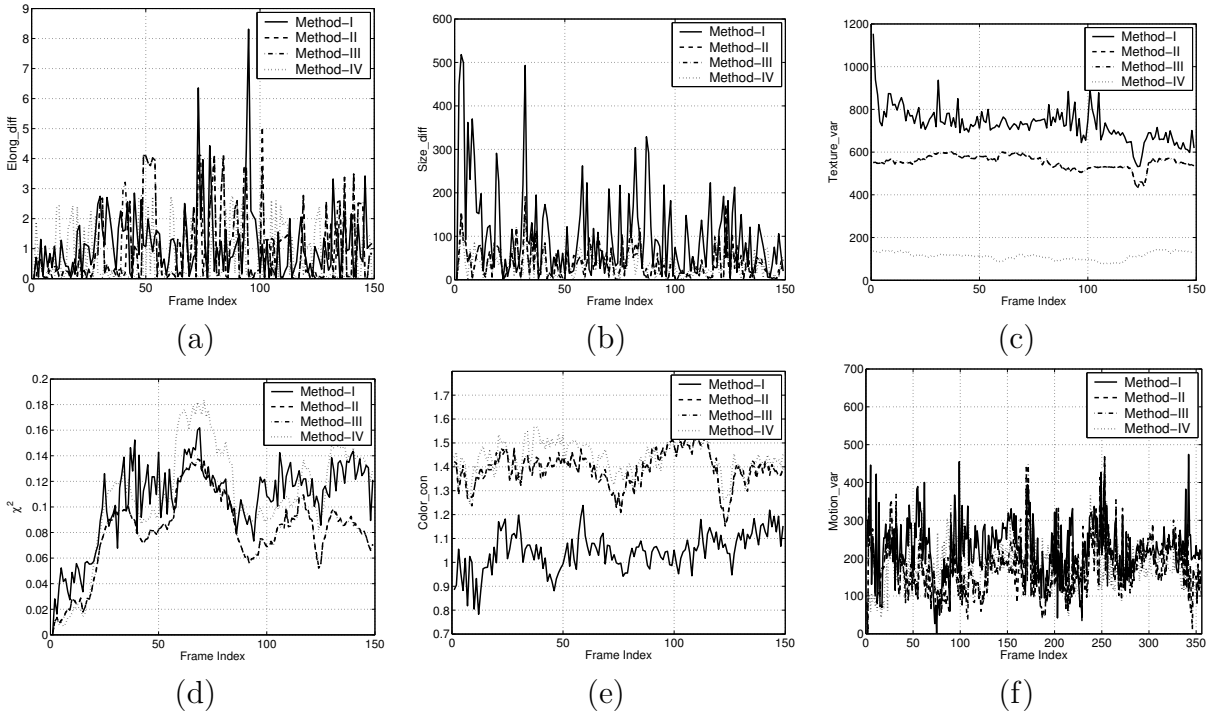


Figure 3.15: Objective evaluation of video Face.

Table 3.4: Numerical performance of video Face.

Measurements	Method-I		Method-II		Method-III		Method-IV	
	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv
$Elong\_diff$	1.16	1.16	1.0	1.27	1.04	1.29	0.73	0.94
$Size\_diff$	103.21	103.37	39.69	35.73	40.06	36.69	35.26	29.39
$Texture\_var$	729.57	79.76	552.84	32.26	553.29	32.16	113.27	17.79
χ^2	0.11	0.03	0.08	0.03	0.08	0.03	0.1	0.04
$Color\_con$	1.05	0.08	1.39	0.07	1.39	0.07	1.44	0.07
$Motion\_var$	214.01	95.51	188.13	106.38	188.2	109.36	158.06	58.44

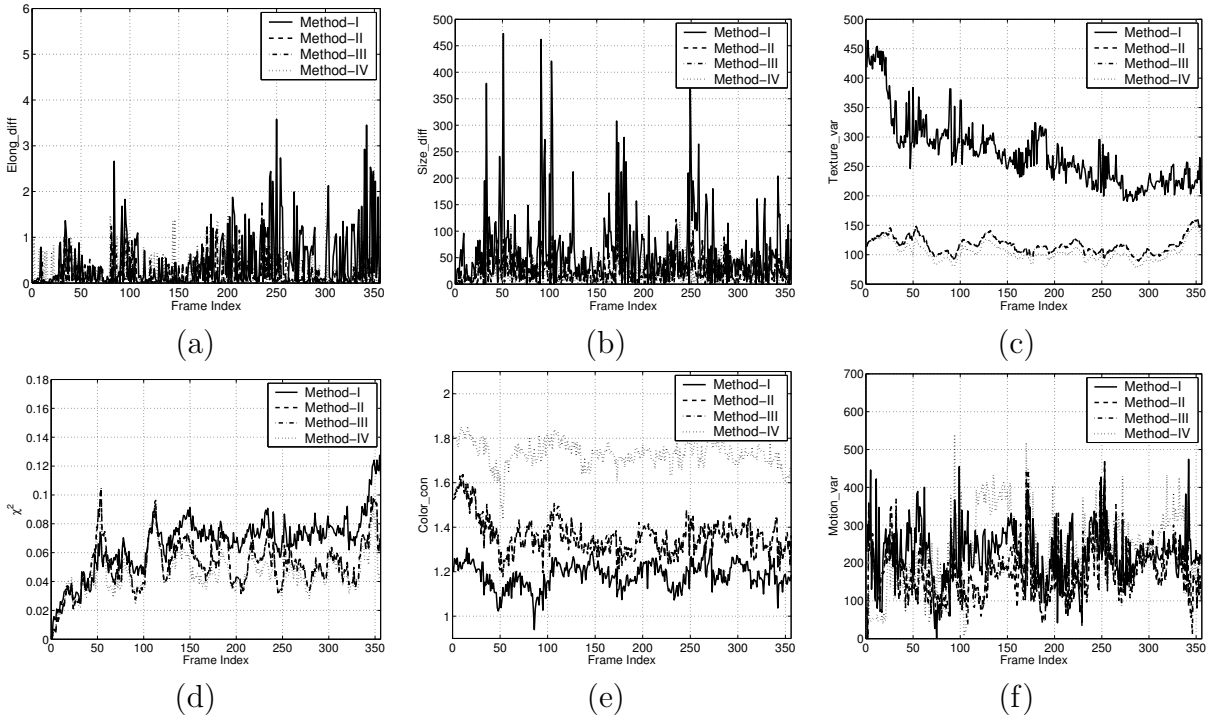
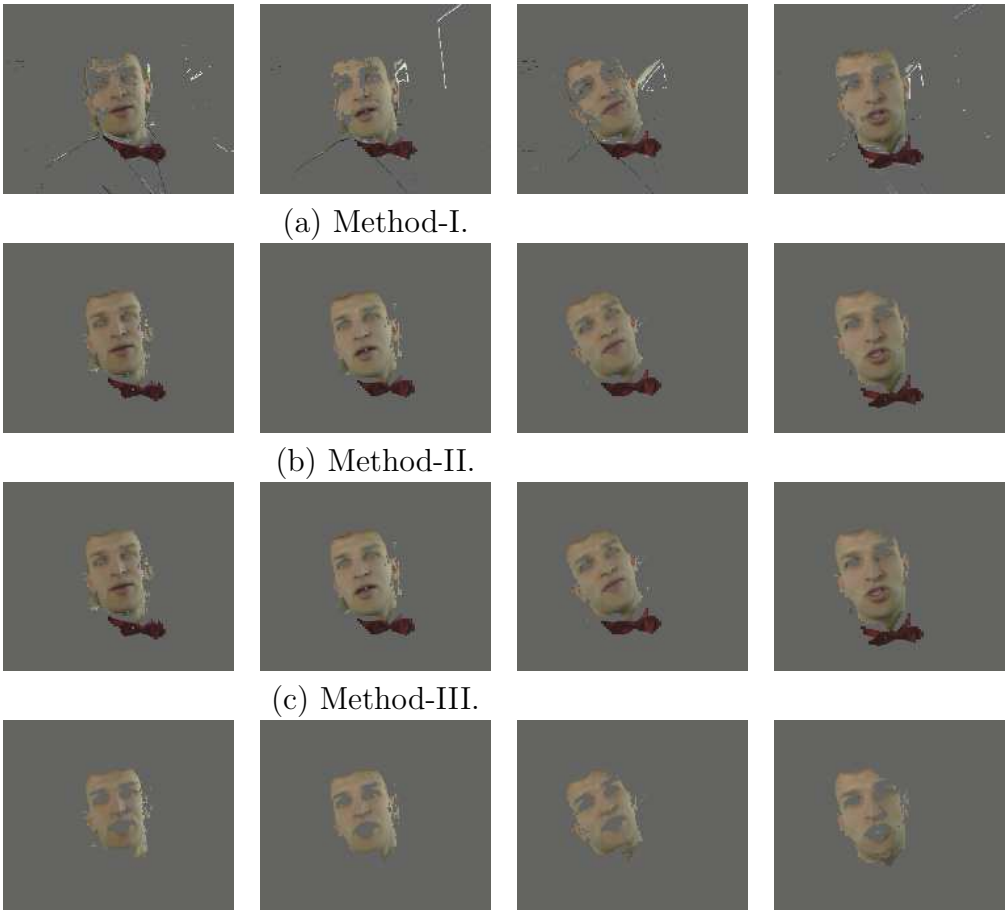


Figure 3.16: Objective evaluation of video Taichi.

Table 3.5: Numerical performance of video Taichi.

Measurements	Method-I		Method-II		Method-III		Method-IV	
	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv
$Elong\_diff$	0.47	0.63	0.17	0.28	0.19	0.29	0.23	0.36
$Size\_diff$	63.89	72.43	24.91	26.52	25.37	26.85	18.66	20.45
$Texture\_var$	272.98	57.98	116.64	14.51	116.55	14.47	104.64	15.26
χ^2	0.07	0.02	0.05	0.02	0.05	0.02	0.04	0.01
$Color\_con$	1.18	0.06	1.36	0.08	1.37	0.08	1.73	0.06
$Motion\_var$	223.41	73.81	171.87	68.29	171.83	71.23	177.25	58.53



(a) Method-I.

(b) Method-II.

(c) Method-III.

(c) Method-III.

Figure 3.17: Segmentation results of Video-Face using the same number of key-frames (8 key-frames).

(a) Method-I.

(b) Method-II.

(c) Method-III.

(c) Method-IV.

Figure 3.18: Segmentation results video Taichi using the same number of key-frames (8 key-frames).

3.5 Summary

This chapter presents an analytical approach for coherent key-frame extraction and object segmentation. Specifically, a video shot is characterized by a statistical mixture model that is a weighted combination of a GMM and a class-independent density model. The estimable weight, which is called frame/pixel saliency, shows the contribution of a frame/pixel to GMM estimation. Simulation shows that the proposed algorithm can extract compact and representative key-frames with semantic meaning for effective object segmentation. Moreover, based on the proposed coherent segmentation methods, a unified video representation/description framework is suggest to support content-based video analysis by providing accurate object segmentation, and semantically meaningful key-frames. The major contribution of this

method is the joint formulation of key-frame extraction and object segmentation that could lead to an optimal solution via the EM algorithm.

Chapter 4

UNSUPERVISED BAYESIAN IMAGE SEGMENTATION

Segmentation of a textured image is to partition the image into distinct regions of homogeneous behavior. In this chapter, we study the unsupervised parametric image segmentation that is widely involved in a variety of computer vision tasks, such as medical diagnosis, remote sensing, and automatic surveillance/detection systems, etc. In spite of its paramount applications, unsupervised image segmentation has been a challenging topic because of the complexity of natural texture behaviors in real images. There is no general approach that works well everywhere.

Usually a textured image is composed of deterministic or random structure patterns, and thus a structural pattern that consists of a group of pixels is of more significance than a single pixel for texture modeling. Consequently, statistical modeling based segmentation methods have drawn substantial attention because of their efficiency in capturing local texture behavior and their prominent methodological advantages [5]. Specifically, texture features are assumed to be generated by an underlying 2-D stochastic model, and there exist closed forms or approximate approaches for estimating model parameters used for segmentation. For example, Markov random field (MRF) models have attracted considerable interest in characterizing the pixel intensity distribution [66, 44, 11, 133], as well as the prior knowledge of labeling processes when Bayesian inference is involved. Under the Bayesian framework, an image can be segmented by computing a maximum *a posteriori* (MAP) estimation of the pixel labels under zero-one loss. However, the solutions to MAP are usually computationally expensive due to the non-causal neighborhood structure and consequent iterative approximations [66, 44, 11]. In addition, MRF models are not capable enough of describing large scale texture behaviors [13, 165], and the usage of more sophisticated models could deter the efficiency of model estimation. Therefore finding a

tradeoff between computational efficiency and statistical modeling accuracy remains essential for developing practical Bayesian image segmentation methods.

In order to develop high-performance segmentation approaches with moderate computational loads, a number of methods have been proposed to model/estimate class-conditional density of texture features and labeling process of texture classes under the Bayesian framework. Instead of modeling texture intensity via a class-conditional density in the spatial domain directly, the interest has been steadily growing nowadays on modeling texture features in the wavelet domain, where the spatial locality of texture information can be preserved and the image structure is recomposed in a way that facilitates the statistical representation of images [114]. Especially, the Haar wavelet transform is suitable for texture segmentation due to its best spatial locality. Usually there are two types of methods proposed to model the statistics of wavelet coefficients. One is the independent model that only considers the marginal distribution of wavelet coefficients, such as generalized Gaussian function [114], which have been successfully used in image compression [7], denoising [121] and retrieval [46], or mixture models [31, 128]. The other is the models that not only consider the marginal distribution, but also capture high-order dependencies of wavelet coefficients, such as interscale dependencies [37, 60, 45], intrascale dependencies [56, 119], or both of them [36, 18, 165, 126, 25, 110, 57]. These models show significant advantages in statistical image modeling and have been applied to image denoising [37, 110, 56, 119, 25], segmentation [32, 60, 126], retrieval [45], compression [18], texture analysis and synthesize [165, 146, 60]. Particularly, Wavelet-domain Hidden Markov Models (WDHMMs), such as hidden Markov tree (HMT) [37, 137], are newly developed statistical models that impose a tree-structured Markov chain across scales to capture interscale dependencies of wavelet coefficients. HMT was improved by capturing dependencies across both wavelet subbands and scales in HMT-3S [60], and/or using redundant wavelet transforms [45, 137]. In our work, WDHMMs are applied to multiscale unsupervised segmentation because of their two salient advantages: 1) They are efficient statistical models which have closed forms to calculate model likelihoods via a fine-to-coarse recursion. 2) The embedded tree structure supports the multiscale texture characterization that is involved in multiscale Bayesian segmentation [32, 28]. As mentioned before, the MAP estimation of labeling process

based on the MRF model could be approximated by suboptimal solutions. However, these solutions could be unstable regarding initial conditions and usually runs risks of converging to local optima [13]. In order to reduce the possibility of converging to local optima, and to characterize the large scale textured behaviors efficiently, multiscale statistical modeling was suggested for image segmentation [13, 97, 165]. One specific model is the multiscale random field (MSRF) model, which was proposed to characterize the labeling process by capturing interscale dependencies of class labels across scales of a pyramid structure, and an efficient sequential MAP (SMAP) estimator was developed as an approximation to MAP [13]. Compared with MAP, SMAP is computational efficient and tends to minimize the spatial size of the classification error due to its scale-dependent cost function, leading to more desirable segmentation results. HMT was well integrated with SMAP for efficient supervised segmentation in a recent work, where an interesting approach HMTseg was proposed [32]. The SMAP was further studied by a joint multicontext and multiscale approach (JMCMS) for exploiting robust contextual information via multiple context models for Bayesian inference [58]. Moreover, HMT-3S was incorporated into JMCMS to further improve the segmentation performance [60]. It was shown that both statistical texture characterization and contextual modeling of multiscale class labels are important in the context-based supervised Bayesian segmentation [59, 60, 55].

The significant advantages of WDHMMs and MSRF previously mentioned in supervised image segmentation have motivated us to extend them to the unsupervised case [149], which was applied to the sea floor sonar image segmentation [49]. In supervised segmentation methods using WDHMMs, given a textured image, the estimation of its WDHMM parameters is a model estimation process. Then it is worth pointing out that these segmentation methods are realized by mapping the image into a set of model estimations of WDHMM to explore the disparities of different textures, where each estimation corresponds to a specific texture in the image. According to the *Likelihood Principle* [50], the disparities amongst different textures can be shown as the likelihood disparities in each model estimation. Nevertheless, this mapping process is not so straightforward in unsupervised segmentation because there is no information *a priori* about model estimations. In such case, the model estimation is

usually approximated by an iterative *hard* or *soft*-decision approach [164]. In *hard*-decision, an image is divided into non-overlapping homogeneous textured windows. Each window provides a model estimation, and a raw segmentation map can be obtained by clustering of the model estimations [115, 125]. In *soft*-decision [164, 101], the segmentation is performed by repeating an estimation/segmentation iteration after initializing model parameters, such as Expectation Maximization (EM) algorithm. Both two approaches need accurate model estimations for all textures to achieve a good segmentation performance. However, some problems arise when WDHMMs are applied to unsupervised segmentation. In *hard*-decision, despite the fact that the accurate texture characterization could be achieved via WDHMMs, the following clustering could be cumbersome because a WDHMM is a high-dimensional Gaussian mixture model with many model parameters [32], resulting in expensive computation and running a risk of the "curse of dimensionality" when the number of data samples does not significantly outnumber the feature dimension [83]. Existing WDHMMs are not suitable to be directly applied to *soft*-decision either because they implement supervised algorithms based on known or pure texture prototypes. In a recent work [141], a mean shift clustering in a 7-dimensional feature space was used to obtain an initial segmentation map, and the WDHMM was applied to estimate model parameters for each texture. This approach is basically a *hard*-decision approach where WDHMM was used after the raw segmentation is obtained by the clustering.

In this work, a hybrid *soft-hard* decision approach, which consists of both *soft*- and *hard*-decision approaches, is proposed where WDHMMs are efficiently integrated into the entire process of unsupervised image segmentation. This approach is also applicable to other statistical modeling frameworks where a closed form of likelihood computation is available. Within this hybrid approach, the *soft*-decision step maps an input image to a WDHMM estimated by the EM algorithm, where the model likelihood can be computed for all image blocks in a multiscale representation. Then the *hard*-decision step generates a raw segmentation by a clustering of the likelihood values, each of which is associated with an image block in the original image and shows the goodness of fit between this area and the estimated WDHMM. Therefore, the problem of unsupervised segmentation is converted to a self-supervised segmentation process. Both the *soft*- and *hard*-decision steps are significant to the final

segmentation performance. The *soft*-decision step implicitly determines feature separability/cluster divergence in a WDHMM, which could be measured by likelihood disparity. Numerical experiments are developed to search for a proper WDHMM where the separability of different clusters are maximized. The *soft*-decision step essentially determines whether the clusters of likelihood values can be well separated by clustering in the *hard*-decision step. In the *hard*-decision step, the K-mean and EM clustering are first briefly discussed, and two new clustering approaches are suggested. One is the context-based multiscale clustering (CMSC) involving local context and multiscale information, and the other is multiple model clustering (MMC) where the likelihood values regarding multiple WDHMMs are used to construct a multidimensional feature space for clustering. Simulations show that higher clustering accuracies could be obtained by CMSC and/or MMC compared with the K-mean and EM approaches. The numerical analysis of the cluster separability in the *soft*-decision step also inspire us to propose a dual-model unsupervised segmentation framework where two WDHMMs are utilized in different parts of the segmentation. Particularly, simulation results on a set of synthetic mosaics show that the unsupervised segmentation performance comes close to the supervised case.

The rest of the paper is organized as follows: Section 4.1 reviews the supervised Bayesian image segmentation approaches using SMAP and WDHMMs. In Section 4.2 we address the model specification and identification by investigating the cluster divergence measured by the likelihood disparity of WDHMMs. Two new clustering methods are developed based on the K-mean and EM clustering in Section 4.4. In Section 4.5 we propose a dual-model unsupervised segmentation framework, and the simulations on both synthetic mosaics and real images are shown in Section 5.5. Finally conclusions are presented in Section 4.7.

4.1 Supervised Bayesian Image Segmentation

4.1.1 Multiscale Random Field Model

Under the Bayesian segmentation framework where both image features and prior knowledge are incorporated, maximum *a posteriori* (MAP) estimation is usually

involved to estimate the class label of image pixels:

$$\hat{x} = \arg\max_{x} E[C_{MAP}(X, x|Y = y] \tag{4.1}$$

where Y is an image observation with an unseen class label field X [1], and $C_{MAP}(X, x)$ is the zero-one cost function. It is well known that the MAP estimation, which aims at maximizing the probability that all pixels are correctly classified, is excessively conservative and computationally expensive. In the statistical modeling based image segmentation, an ideal modeling scheme should characterize both large and small scale behaviors in textured images effectively and efficiently [13, 32]. However, the modeling of large homogeneous texture behavior with MRF models needs complicated neighborhood systems and estimation algorithms. In order to efficiently model the large scale texture behavior, a multiscale random field (MSRF) model was suggested [13, 28]. The MSRF model is composed of a set of random fields with different resolutions. Assume $Y^{(j)}$ is an image observation at scale j with its unseen label field $X^{(j)}$, the principal assumption of the MSRF model is that the distribution of class label $X^{(j)}$ is conditional independent on others given $X^{(j+1)}$ at the coarser scale. This assumption forms a Markov chain across scales from coarse to fine that captures interscale dependencies of class labels, resulting in a causal contextual structure that simplifies the model estimation. This one-order Markov chain is formulated as:

$$P(X^{(j)}|X^{(j+1)}, X^{(j+2)}, \cdots) = P(X^{(j)}|X^{(j+1)}). \tag{4.2}$$

Based on the MSRF model, a sequential MAP (SMAP) estimator was developed with an alternative weighted cost function $C_{SMAP}(X, x)$ [13]. Compared with the MAP estimation, the SMAP method is computational efficient and can minimize the spatial size of errors, leading to more desirable segmentation results. The SMAP estimator can be reformulated as [13]:

$$\begin{aligned} \hat{x}^{(j)} = \quad & \arg\max_{x^{(j)}} \{\log p_{y^{(j)}|x^{(j)}}(y|x^{(j)}) \\ & + \log p_{x^{(j)}|x^{(j+1)}}(x^{(j)}|\hat{x}^{(j+1)})\}, \end{aligned} \tag{4.3}$$

The two terms in (4.3) are related to the texture representation and the modeling of the contextual information from the next coarser scale. It was shown that both

[1] Upper case letters denote random variables, whereas low case letters denote their realizations

image features and multiscale contextual modeling are important in the multiscale Bayesian image segmentation [59, 60, 55].

4.1.2 Wavelet-domain Hidden Markov Models

Wavelet-domain hidden Markov models (WDHMMs) capture high-order dependencies of wavelet coefficients. As the first WDHMM, hidden Markov tree (HMT) model is a multidimensional Gaussian mixture model that applies tree-structured Markov chains across scales to capture interscale dependencies of wavelet coefficients [37, 137]. It is parameterized by:

$$\theta_{HMT} = \{p_J^B(m), \epsilon_{j,j-1}^B(m,n), \sigma_{B,j,m}^2 | B \in \mathcal{B}; j = 1, ..., J; m, n = 0, 1\}. \tag{4.4}$$

In (4.4), $p_J^B(m)$ is the probability of state m at the coarsest scale J in subband $B \in \mathcal{B}$, $\mathcal{B} = \{LH, HL, HH\}$ is the set of three subbands with different orientations, $\epsilon_{j,j-1}^B(m,n)$ is the state transition probability of the Markov chain from scale j to scale $j-1$ in subband B, and $\sigma_{B,j,m}^2$ is the variance of the wavelet coefficients at scale j in subband B given state m. Given wavelet coefficients $\mathbf{w}$ of a $N \times N$ image, the model parameters can be estimated by the EM algorithm that maximizes the model likelihood $f(\mathbf{w}|\theta_{HMT})$:

$$\hat{\theta}_{HMT} = \arg_{\theta_{HMT}} \max f(\mathbf{w}|\theta_{HMT}), \tag{4.5}$$

$$f(\mathbf{w}|\theta_{HMT}) = \sum_{B \in \mathcal{B}} \sum_{k,i=0}^{N_J - 1} \log \left(\sum_{m=0}^{1} f(\mathcal{T}_{J,k,i}^B | \theta_{HMT}, m) \right),$$

where $N_J = N/2^J$, $\mathcal{T}_{j,k,i}^B$ denotes the wavelet subtree rooted at the wavelet coefficient (k,i) at scale j, i.e., $w_{j,k,i}^B$, and $w_{j,k,i}^B$ with its state variable form the root node of the HMT subtree $\mathcal{T}_{j,k,i}^B$. The model likelihood of the subtree $\mathcal{T}_{j,k,i}^B$ with respect to θ_{HMT} is calculated in a recursive fine-to-coarse fashion as follows [37]:

$$f(\mathcal{T}_{j,k,i}^B | \theta_{HMT}, m) = p_j^B(m) g(w_{j,k,i}^B | 0, \sigma_{B,j,m}^2) \left(\prod_{s=2k}^{2k+1} \prod_{t=2i}^{2i+1} \sum_{n=0}^{1} \left(\epsilon_{j,j-1}^B(m,n) f(\mathcal{T}_{j-1,s,t}^B | \theta_{HMT}, n) \right) \right), \tag{4.6}$$

and in the finest scale, i.e., $j = 1$, we have

$$f(\mathcal{T}_{1,k,i}^B | \theta_{HMT}, n) = p_1^B(n) g(w_{1,k,i}^B | 0, \sigma_{B,1,n}^2), \tag{4.7}$$

where $g(w^B_{j,k,i}|0,\sigma^2_{B,j,m})$ is the Gaussian density. Equation (4.5) implies the subband independency of wavelet coefficients. Usually the regular spatial structures in natural images may result in significant statistical dependencies across wavelet subbands B [18]. An improved HMT model, called HMT-3S, was developed to capture these dependencies by grouping three subbands into one quad-tree structure, and the number of states in each node is changed from two to eight, while the marginal distribution of wavelet coefficients is still a two states Gaussian mixture [60]. The likelihood computation of HMT-3S has a similar recursive fine-to-coarse fashion as HMT. The improvement of HMT-3S on texture characterization is demonstrated by the performance of texture analysis and synthesis [60]. Another notable improvement of HMT model was suggested as a vector WDHMM using the redundant steerable wavelet transform [45], where multivariate Gaussian density is incorporated. The statistical dependencies across wavelet subbands are captured by the variance-covariance matrix of the multivariate Gaussian density, and has shown impressive capability in rotation invariant texture retrieval [45]. In our work, only the WDHMMs based on the orthogonal DWT, i.e., HMT and HMT-3S, are studied for unsupervised segmentation. If the redundant wavelet transform is used, such as the vector WDHMM suggested in [45], it is expected that a comparable or better segmentation performance could be achieved, especially in the aspect of the rotation invariance.

4.1.3 SMAP, HMTseg, and JMCMS

As shown in (4.3), the SMAP estimation indicates that the estimation of pixel classes is determined by image features and multiscale context information. Based on the framework of SMAP, a context-based Bayesian segmentation algorithm, HMTseg, was developed in [32]. In HMTseg, the HMT model is used to characterize textures in the wavelet domain and an efficient context model is applied for the interscale context fusion. Specifically, the contextual information is modeled as a context vector $v^{(j)}$ and a contextual prior $p_{x^{(j)}|v^{(j)}}(c|u)$ is involved in SMAP as the second part of (4.3). Assume there are N different textures and the SMAP estimation can be obtained by

$$\hat{x}^{(j)} = \arg\max_{x^{(j)}} p_{x^{(j)}|v^{(j)},y^{(j)}}(x^{(j)}|\hat{v}^{(j)}, y^{(j)}), \tag{4.8}$$

where

$$p_{x^{(j)}|v^{(j)},y^{(j)}}(x^{(j)}|\hat{v}^{(j)},y^{(j)}) = \frac{p_{x^{(j)}}(x^{(j)})p_{v^{(j)}|x^{(j)}}(\hat{v}^{(j)}|x^{(j)})f(y^{(j)}|x^{(j)})}{\sum_{c=1}^{N} p_{x^{(j)}}(c)p_{v^{(j)}|x^{(j)}}(\hat{v}^{(j)}|x^{(j)}=c)f(y^{(j)}|x^{(j)}=c)},$$

$p_{x^{(j)}}(c)$ is the probability mass function of class c at scale j, and $f(y^{(j)}|x^{(j)}=c)$ is the HMT likelihood function of image block $y^{(j)}$ with respect to class c.

The simulation results in [13, 32] show that the segmentation results in homogeneous regions are usually better than those around boundaries because the selected context model favors the formation of large uniformly classified areas with less consideration on texture boundaries. In order to improve the segmentation performance around boundaries, a joint multi-context and multiscale approach (JMCMS) was proposed to capture robust contextual information with multiple context models [58]. JMCMS is a multi-objective optimization problem associated with multi-context models that favor either forming homogeneous classified regions or having high sensitivity to boundaries. Since the optimal solution is too hard to obtain, the sub-optimal solution can be acquired by converting the multi-objective optimization to multiple single objective optimizations [38]. That means the SMAP estimation in JMCMS is performed based on the multiple context models individually and sequentially, and the decision is only made at the final step. JMCMS was further combined with HMT-3S where even better segmentation results can be obtained [60]. In view of significant advantages of WDHMMs in characterizing texture behaviors, as well as the capability of MSRF to model the labeling process, we want to extend them to unsupervised image segmentation which is more practical in real applications.

4.2 A Hybrid Soft-hard Decision Approach

In order to implement efficient unsupervised texture segmentation using WDHMMs, we propose a new hybrid *soft-hard* approach based on the *Likelihood Principle* [50]. As we discussed before, the supervised segmentation of an image I of N different textures can be explained as capturing the disparity of model likelihoods at a set of model estimations of N textures: $\theta_k,\ k = 1, 2, ..., N$. We define that a mapping of image I into θ_k is a model fitting process between I and θ_k, and the goodness of fit is quantified by the model likelihood $f(I|\theta)$. If there exists N different textures in I,

then after mapping I into θ_k, the obtained model likelihood preserves the disparities of N textures that can be captured by θ_k.

An insightful understanding of this mapping can be explicated by the *Likelihood Principle* that is stated as [6]: *All of the relevant information about the parameter(s) provided by the sample data is completely captured by the likelihood function alone*. The *Likelihood Principle* asserts that the evidential meaning of any data with respect to a hypothetical model is contained completely in the likelihood function determined by the data [123]. Thus given image I comprising N different textures $x_1, x_2, \ldots, x_N$, if there is only one hypothesis of model θ, which could be either associated with I by treating I as a *mixed* texture, or associated with any other texture or image, different model likelihoods can be shown due to the different goodness of fit between θ and local texture behaviors, and the model likelihood disparities preserves the disparities of the N textures captured by θ. For instance, if image blocks y_i and y_k in I are from two textures x_i and x_k, respectively, then their model likelihoods $f(y_i|\theta)$ and $f(y_k|\theta)$ should be distinct from each other. This can be represented by a cluster divergence measurement $div(.)$ between two different textures as:

$$\begin{aligned} \text{if } \ div(y_i, y_k|\theta) < \beta, \quad \text{then} \quad & f(y_i|\theta) \approx f(y_k|\theta) \text{ and } x_i = x_k, \\ \text{else} \quad & f(y_i|\theta) <> f(y_k|\theta) \text{ and } x_i \neq x_k, \end{aligned} \tag{4.9}$$

where β is a relatively small constant.

Previous discussion implies that different textures could be roughly segmented out by capturing the likelihood disparities. Hence given image I, after mapping it into a certain model θ, a raw segmentation could be obtained by a clustering on likelihood values of image blocks. In this process, mapping I into θ is a *soft*-decision approach, where θ is estimated by the EM algorithm, and image I is considered as one *mixed* class. After calculating the model likelihood of all image blocks, the clustering on likelihood values is a *hard*-decision approach that provides a blockwise raw segmentation of image I. In *hard*-decision, the estimation of β in (4.9) is circumvented by clustering. The combination of the *hard*- and *soft*-decision steps generates a hybrid *soft-hard* decision, which eventually converts the unsupervised segmentation into self-supervised segmentation.

The suggested hybrid *soft-hard* decision possesses two major advantages. First, the segmentation problem is simplified via capturing the likelihood disparity of different textures because we do not have to find a model that best describes the image, which is usually the goal of the traditional segmentation approaches and computational demanding. Second, both large scale and small scale texture behaviors are well captured by likelihood values at different scales. This will facilitate the clustering process by changing the complicated texture modeling problem to a simpler low-dimensional feature modeling problem, namely, the modeling of likelihood values.

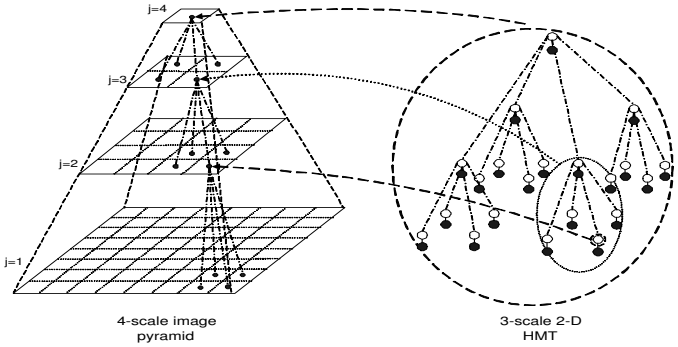


Figure 4.1: Multiscale image representation and 2-D HMT model, where the white node represents the discrete hidden state variable and the black node denotes a continuous coefficient variable [37]. The interscale dependencies are captured by tree-structured Markov chains connecting hidden state variables across scales.

When WDHMMs are applied to the hybrid approach, model θ of any single or *mixed* texture can be estimated by the EM algorithm with a tying operation across wavelet subtrees [37]. Because there is a closed form to calculate the WDHMMs model likelihood at any scale, the likelihood value can be easily obtained at the coarsest scale of WDHMMs, where each node covers the largest spatial area with robust likelihood computation as shown in Fig. 4.1. In Fig. 4.1, the left part is a 4-scale image pyramid and scale $j = 1$ is the pixel level image. The right part of Fig. 4.1 is the corresponding 3-scale 2-D HMT of the Haar wavelet subtree in one subband $B \in \{LH, HL, HH\}$, where the white node represents the discrete hidden state variable and the black node denotes a continuous wavelet coefficient variable [37]. The interscale dependencies are captured by tree-structured Markov chains connecting hidden state variables across scales. A subtree rooted at a node (white or black) at the coarsest scale of the 3-scale

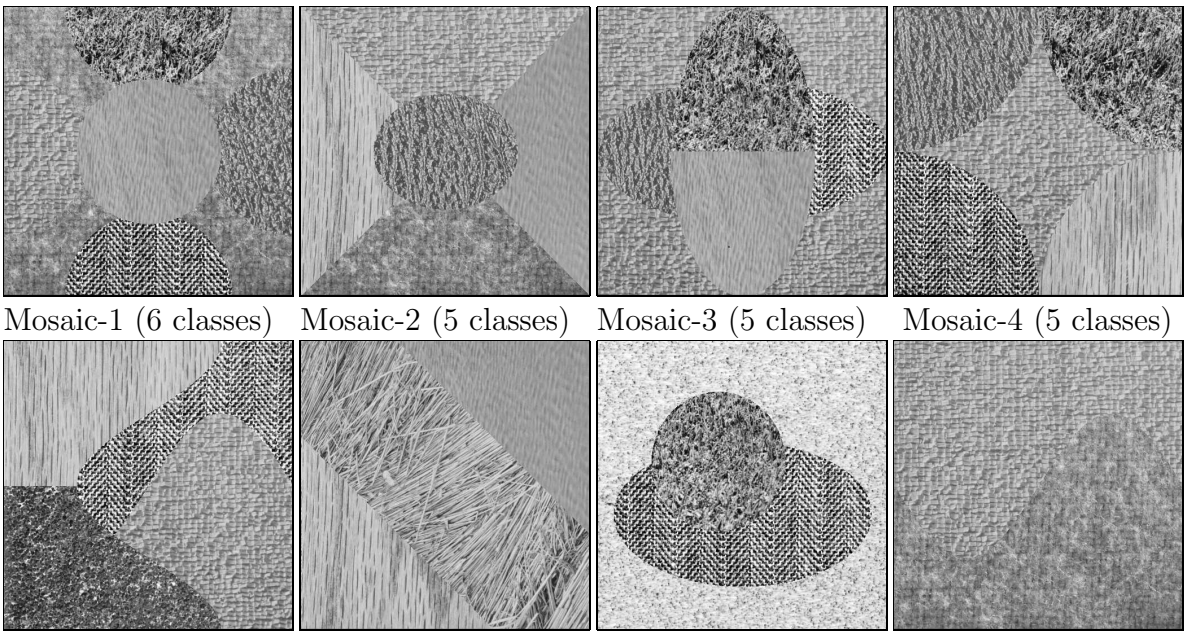


Figure 4.2: Eight synthetic mosaics for the study of cluster divergence and segmentation

2-D HMT is associated with an image block at the coarsest scale of the 4-scale image pyramid. This block covers 8×8 pixels in the image. In the following two sections, we will discuss in detail what roles the *soft-* and *hard* -decisions play in unsupervised segmentation.

4.3 Soft-decision Step: Cluster Divergence

In the hybrid *soft-hard* decision approach, the cluster divergence of different textures, which is measured by the likelihood disparity, is determined during the *soft*-decision step. For studying cluster divergence, it is necessary to select proper divergence measurements, and to explore a proper model θ to which image I is mapped so that different textures in image I are as separable as possible before clustering. This is equivalent to find a $\hat{\theta}$:

$$\hat{\theta} = \arg\max_{\theta} DIV(\theta, I), \tag{4.10}$$

where $DIV(\theta, I)$ measures the cluster divergence of multiple textures in image I regarding θ, and it is not the same as the pairwise divergence measurement $div(.)$ in (4.9). We address this problem by studying two key issues: *model specification*, or

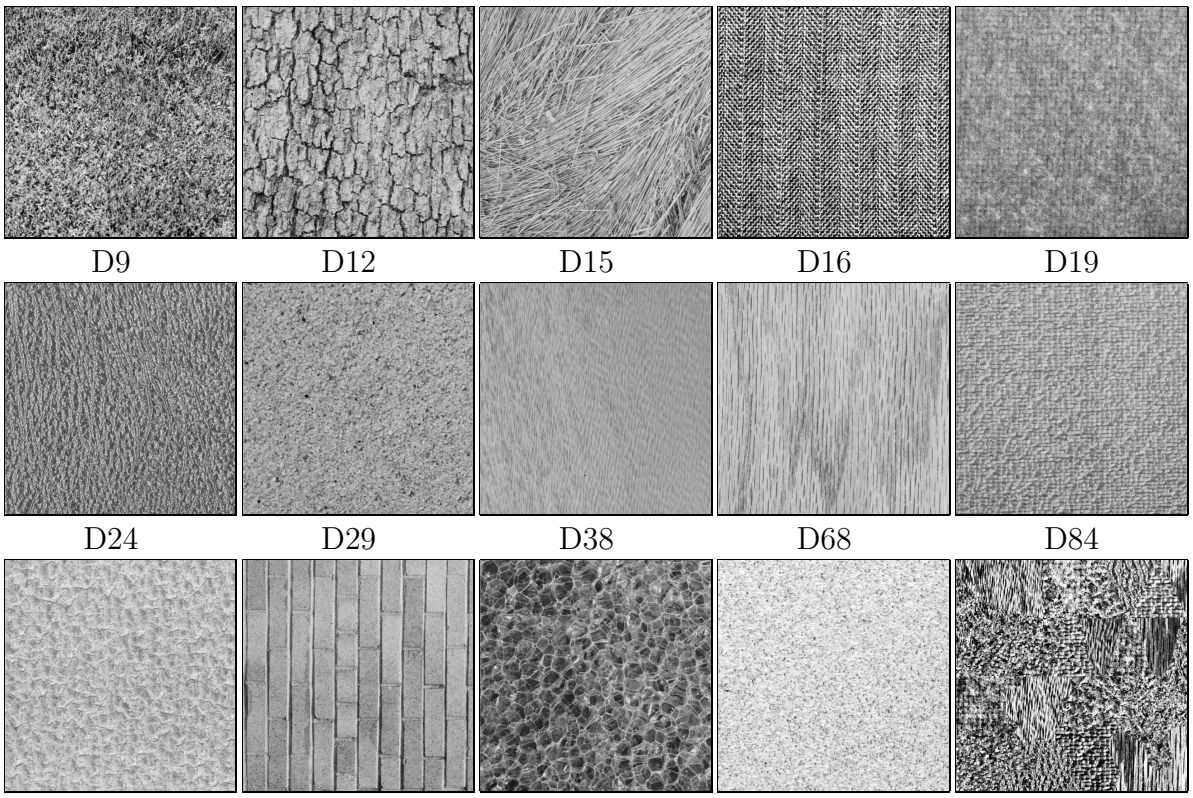


Figure 4.3: 13 Brodatz and 2 other textures from USC database [3].

equivalently, how to estimate θ, and *model identification*, i.e., which model should be chosen, e.g., HMT or HMT-3S?

4.3.1 Experimental Setup

In this work, numerical experiments are implemented to study the two issues based on eight synthetic mosaics as shown in Fig. 6.2, and 15 textures in Fig. 4.3. All images are of size 512×512, and the synthetic mosaics are composed of 2 to 6 different textures. In the experiment, after a 4-scale Haar DWT, image I (a mosaic in Fig. 6.2) is mapped into 16 different model specifications of HMT, as well as 16 specifications of HMT-3S. These model specifications are estimated from image I itself and 15 textures in Fig. 4.3, and we use notation θ_{name} to represent them (HMT or HMT-3S): θ_I[2], θ_{D9}, θ_{D12}, θ_{D15}, θ_{D16}, θ_{D19}, θ_{D24}, θ_{D29}, θ_{D38}, θ_{D68}, θ_{D84}, θ_{D92}, θ_{D94},

[2] In the following sections, θ_I is used to represent the WDHMM model specification of the image to be segmented.

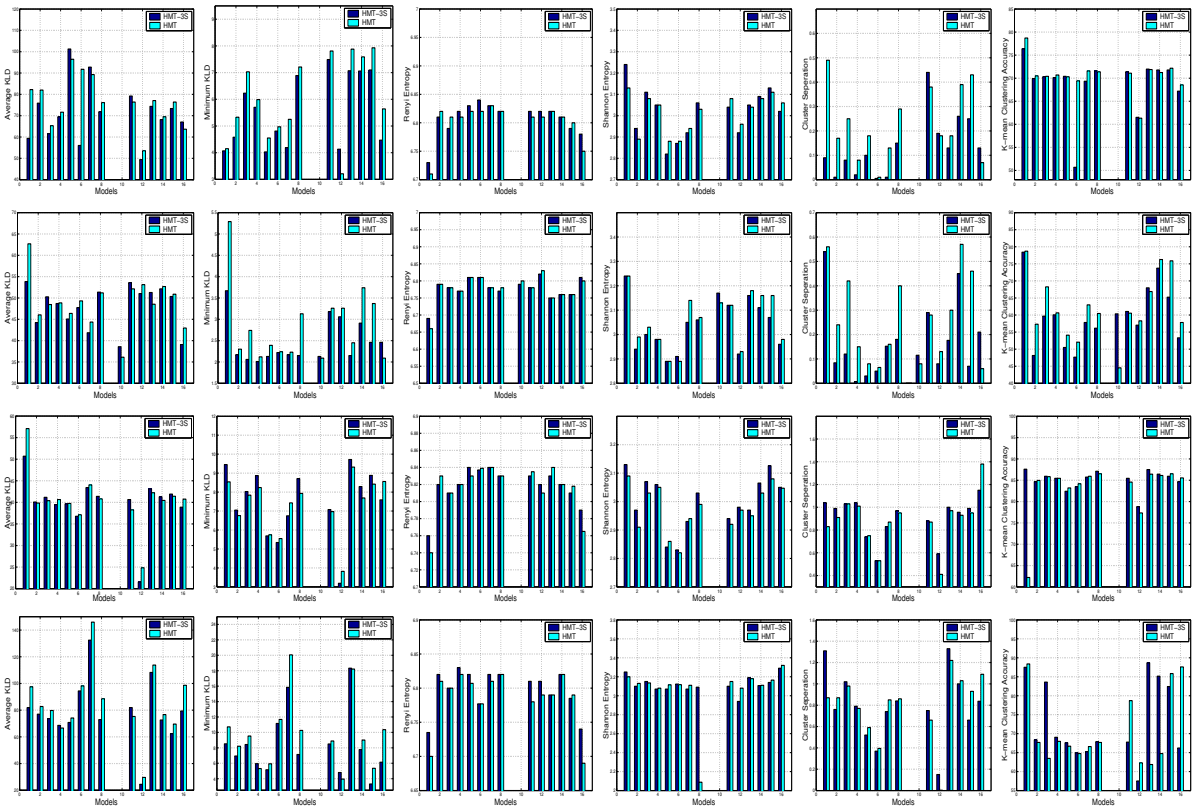
Table 4.1: Numerical measurements of cluster divergence

Blind Evaluation		
Criteria	Definition	Explanations
Shannon Entropy	$I_{Shannon} = -\sum_{i=1}^{N} p_i \ln(p_i)$	Shannon entropy is used to evaluate the histogram of the WDHMM likelihood values. If most likelihood values concentrate to a small region, which means that different classes in image I are not separable after being mapped to certain θ, the entropy value should be small. On the other hand, if the likelihood values are uniformly distributed or concentrate to several ranges which are associated with different textures in image I, the entropy value should be large.
Renyi Quadratic Entropy	$I_{Renyi} = \frac{1}{1-\alpha} \ln(\sum_{i=1}^{N} p_i^{\alpha})$, when $\alpha = 2$, it is called Renyi Quadratic entropy: $I_{Renyi} = -\ln(\sum_{i=1}^{N} p_i^2)$	Renyi entropy is a more general entropy measure developed by Renyi [135]. It relates to Shannon entropy as: $\lim_{\alpha \to 1} I_{Renyi} = I_{Shannon}$. When $0 < \alpha < 1$, $I_{Renyi} > I_{Shannon}$; when $\alpha > 1$, $I_{Renyi} < I_{Shannon}$ [8]. It was suggested as an easier nonparametric estimator for entropy in [129]. Renyi entropy and its derivation were used as cluster divergence/similarity measurements [68, 75]. In this work, Renyi entropy is directly computed on the WDHMMs likelihood values, which is not the same as Shannon entropy. After mapping an image to a model θ, if likelihood values are centralized around several centers with small variance, the Renyi entropy value is smaller than those nearly uniformly distributed (large variance).
Supervised Evaluation		
Kullback-Leibler distance	$D(f,g) = \int f(x) \ln \frac{f(x)}{g(x)} dx$. Symmetrized KLD: $\tilde{D}(f,g) = D(f,g) + D(g,f)$. If both f and g are Gaussian densities: $f \sim N(\mu_f, \sigma_f^2), g \sim N(\mu_g, \sigma_g^2)$, then $\tilde{D}(f,g) = \frac{\sigma_f^2}{\sigma_g^2} + \frac{\sigma_g^2}{\sigma_f^2} + (\mu_f - \mu_g)^2 (\frac{1}{\sigma_f^2} + \frac{1}{\sigma_g^2})$	The Kullback-Leibler distance (KLD) is a widely used divergence measurement between two probability density functions $f(x)$ and $g(x)$ [98]. The KLD is usually not a symmetric measurement and can be symmetrized by adding $D(f,g)$ and $D(g,f)$ together. In this work, the average inter-class KLD (AKLD) and minimum inter-class KLD (MKLD) are used, where the former is the average of KLD of all class pairs, and the latter shows minimum divergence between two clusters. The Gaussian mixture is used to approximately model the HMT likelihood value, where each component is related to a texture in the image. Ideally the larger the KLD, more distinction between clusters.
Cluster Separation	$CS = \frac{r_{min}}{\sigma_{max}}$, where $r_{min} = \frac{1}{2} \min_{i \neq j} \lvert\mu_i - \mu_j\rvert$, $\sigma_{max} = \max_i \sigma_i$, μ_i and σ_i are the mean and standard deviation of the i^{th} Gaussian density, and μ_j and σ_j are the parameters of the j^{th} density.	Cluster Separation (CS) is a specific and efficient divergence measurement if each cluster are Gaussian distributed [91]. According to its definition, CS explicitly considers not only the inter-cluster (between clusters) variability by the mean difference $\lvert\mu_i - \mu_j\rvert$, but also intra-cluster (within a cluster) variability by variance σ_i. Apparently a large CS value is relate to more cluster divergence.

θ_{D112}, θ_{Sand}, and $\theta_{Mixture}$. Based on these mosaics and textures, we want to find a model where the cluster divergence of image I could be maximized.

4.3.2 Numerical Criteria

To quantify the cluster divergence, two types of numerical criteria are involved: one is the blind evaluation criteria without ground truth information of the label field, including Shannon entropy of the likelihood histogram and Renyi entropy of the likelihood values. The other is the supervised criteria that can only be obtained based on the ground truth of the label field, including average inter-class KLD (AKLD), minimum inter-class KLD, and Cluster Separation (CS) [91]. The definition and description of these criteria are listed in Table. 4.1. In the experiment, the numerical criteria are studied associated with the K-mean clustering accuracy at the coarsest



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

Figure 4.4: Cluster divergence in different model specifications and spaces, where model indices from 1 to 16 are corresponding to: mosaic image (Mosaics-1 to -4 from top to bottom), D9, D12, D15, D16, D19, D24, D29, D38, D68, D84, D92, D94, D112, Sand, and Mix. (a) Average KLD. (b) Minimum KLD. (c) Renyi entropy. (d) Shannon entropy. (e) Cluster separation. (f) K-mean clustering accuracy.

scale of WDHMMs.

Amongst these numerical criteria, the blind evaluation is more practical because ground truth information is not available in real segmentation cases. According to the analysis in Table. 4.1, a large Shannon entropy indicates more cluster divergence. However, the largest value appears when the likelihood values are uniform distributed, which is not preferred in this case. In line with the analysis in Section 4.2, the desirable distribution is that the likelihood values are concentrated to several cluster centers with small variances, where each cluster center is expected to be associated with a texture in image I. Clearly, only Shannon entropy is not enough to provide reliable evaluation to cluster divergence. Therefore, we use Renyi quadratic

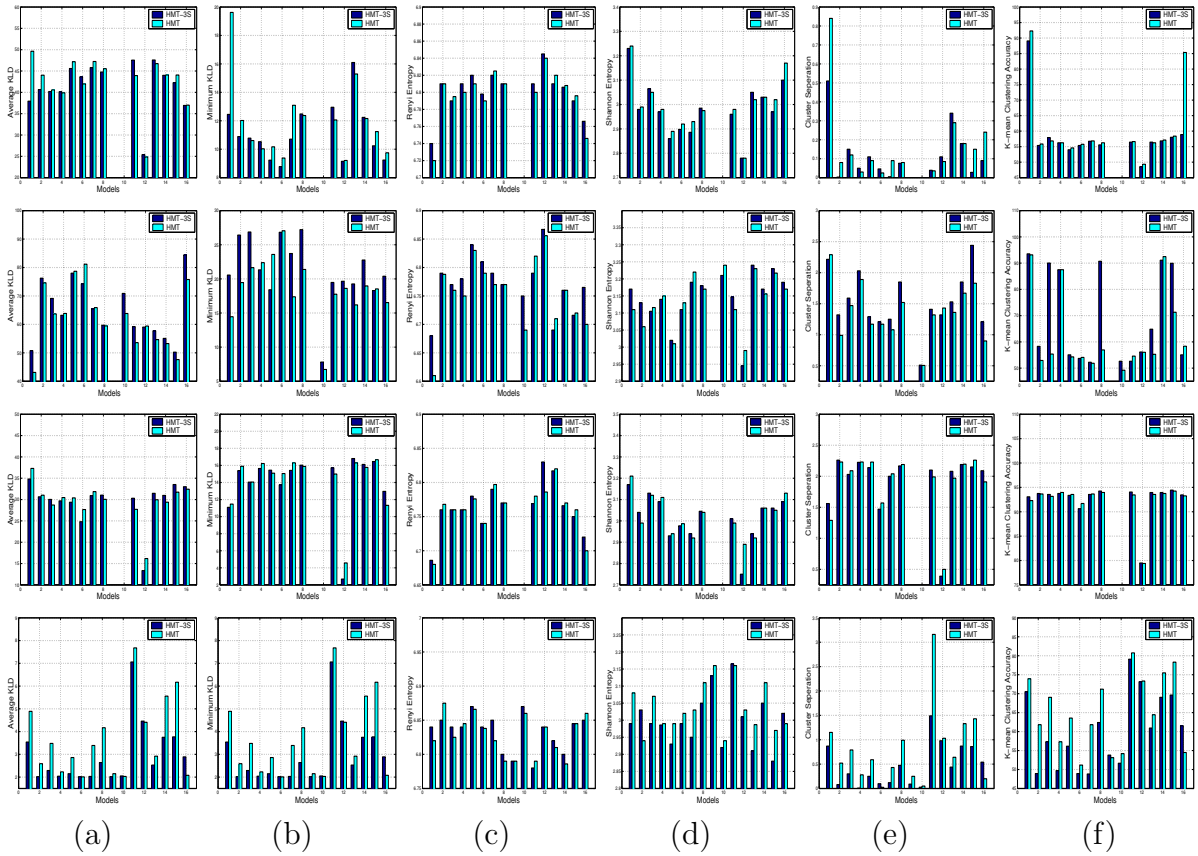


Figure 4.5: Cluster divergence in different model specifications and spaces, where model indices from 1 to 16 are corresponding to: mosaic image (Mosaics 5 to 8 from top to bottom), D9, D12, D15, D16, D19, D24, D29, D38, D68, D84, D92, D94, D112, Sand, and Mix. (a) Average KLD. (b) Minimum KLD. (c) Renyi entropy. (d) Shannon entropy. (e) Cluster separation. (f) K-mean clustering accuracy.

entropy to explore additional information about intra-cluster variation of the likelihood values. When Renyi quadratic entropy is computed directly on the WDHMM likelihood values, less variation of the likelihood values leads to smaller entropy. It is worth pointing out that only Renyi quadratic entropy still cannot give a trustworthy evaluation, either. If all the likelihood values are concentrated to one center with small variance, which means the clusters are not separable, the Renyi quadratic entropy might be also smaller. Generally, a preferred model specification should be the one with a relatively larger Shannon entropy of the likelihood histogram and a relatively smaller Renyi entropy of the likelihood values.

Besides the blind evaluation, supervised evaluation is also helpful to select proper models in the experimental study. Table 4.1 lists the details of KLD and CS

measurements. Here we would like to mention that it was justified that maximum AKLD is equivalent to minimize the Bayes error (probability of misclassification) [40], and consequently the MAP estimation. Therefore, if the Gaussian assumption of the cluster distribution holds true, the model specification with the largest AKLD provides nearly the best cluster separability in the sense of minimizing the Bayes error.

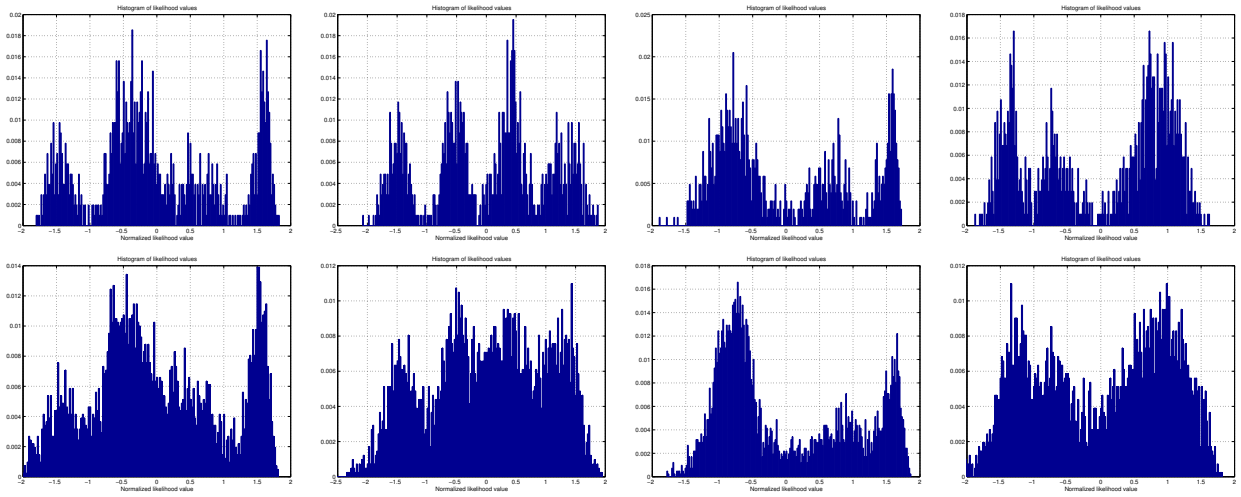
4.3.3 Model Specification and Identification

In the experiment, each of the eight mosaics is mapped into 16 model specifications of HMT and HMT-3S as mentioned in Section 4.3.1. Both blind and supervised cluster divergence measurements are calculated with respect to these models, and Figs. 4.4 and 4.5 show all numerical results. Some model specifications are ignored because they can barely fit the test image with extremely small likelihood values. Experiments reveal that typically the mapping to θ_I leads to a large Shannon entropy, a small Renyi entropy, and large KLD and CS values, which means the model of the image to be segmented itself could be a choice for model specification. Although some mappings to some other θ could also result in better cluster separability according to the divergence measurements and K-mean clustering accuracy, other model specifications could not be always available in real applications, and mapping image I to its own model θ_I is more practical. Furthermore, experimental results implicitly shows that the goal of traditional unsupervised segmentation approaches, i.e., achieving a good fitness between a model and data, is not always desirable in the *soft*-decision step of the hybrid approach. The new goal is to find θ to maximize the separability of different textures.

The cluster divergence in terms of both HMT and HMT-3S are also studied in the experiment. Interestingly when image I is mapped to θ_I, quite often, larger or similar cluster divergence can be obtained from HMT than that from HMT-3S. Although HMT-3S characterizes texture behavior more completely by considering dependencies across both wavelet subbands and scales, HMT provides more cluster distinctions for different textures in an image. This is not unexpected. Recall that HMT and HMT-3S are high dimensional Gaussian mixture, and their parameters

are estimated by maximizing the model likelihoods via the EM algorithm. Since a ML estimator is equivalent to a least square error estimator under the Gaussian assumption of feature distribution, the ML estimation of a more complete characterization (HMT-3S) reduces the fitness disparities between different textures in image I, resulting in less cluster divergence, i.e., likelihood disparities.

4.3.4 Summarization



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

Figure 4.6: First row: histogram of likelihood values at the coarsest scale of HMT model. Second row: histogram of likelihood values at the second coarsest scale of HMT model. (a) Mosaic-2 (5 classes). (b) Mosaic-5 (4 classes). (c) Mosaic-6 (3 classes). (d) Mosaic-7 (3 classes)

To summarize, when segmenting image I, we suggest to map it into a HMT model θ_I that is estimated from I itself via the EM algorithm. Four examples about this mapping are illustrated in Fig. 4.6 by their resultant likelihood histograms obtained at the two coarsest scales of HMT, where Mosaic-2, Mosaic-5, Mosaic-6, and Mosaic-7 in Fig. 6.2 are used. In Fig. 4.6, the first row shows the likelihood histograms obtained from 32×32 likelihood values at the coarsest scale of HMT, where each likelihood value is associated with a 16×16 image block. The second row is the likelihood histograms of 64×64 likelihood values at the second coarsest scale of HMT, and each likelihood value corresponds to an 8×8 image block. It is obvious that more cluster separability could be achieved at the coarsest scale of HMT where the computation of the likelihood values is more robust due to the larger spatial coverage

of each node than those at finer scales. As we can see from the first row of Fig. 4.6, 4 clusters in Mosaic-5, and 3 clusters in Mosaic-6 and Mosaic-7 are well separated, and a simple clustering is capable enough to obtain a raw segmentation map. 2 clusters in Mosaic-2 are moderately overlapped, requiring more powerful clustering approaches to separate them.

4.4 Hard-decision Step: Clustering

The cluster divergence of different textures in an image is determined as the likelihood disparity in the *soft*-decision step. Accordingly, the goal of the *hard*-decision step is to yield a raw segmentation map of the image by capturing the likelihood disparity. This is realized by the clustering on HMT likelihood values. The raw segmentation map provides training samples to estimate the WDHMMs for each texture in the image so that the unsupervised segmentation can be converted into a self-supervised one. Therefore the clustering result is essential to the segmentation performance. In this section, we first review two frequently used clustering methods: K-mean and the EM algorithm, then two new efficient clustering approaches are suggested. The first is a context-based multiscale clustering (CMSC) method involving local context and multiscale information, and the second is a multiple model clustering (MMC) approach where the cluster separability can be increased by introducing multiple models to construct a multi-dimensional feature (likelihood values) space.

4.4.1 K-mean Clustering

In our previous work [149], K-mean clustering was used to identify training samples for each class at the coarsest scale of WDHMMs. Assume there are N different textures in an image, the goal of K-mean clustering is to minimize the square error J_e:

$$J_e = \min_{N} \sum_{k=1}^{N} \sum_{z_l^{(J)} \in \Gamma_k} \|z_l^{(J)} - m_{k,J}\|^2, \tag{4.11}$$

where $z_l^{(J)} = f(y_l^J|\theta)$ is the likelihood value of node l corresponding to class k at the coarsest scale J of a WDHMM, and $m_{k,J}$ is the likelihood mean of class k at scale J.

K-mean clustering is efficient when it is performed at the coarsest scale of WDHMM with a small number of nodes. However, K-mean is a *hard* clustering approach, and only works well if the clusters are hyperspherically distributed and well separated in the feature space. In addition, if a node at the coarsest scale of a WDHMM is related to an area consisting of different textures, it could be misclassified because K-mean algorithm considers the distance between each likelihood value and cluster centers without taking into account the weight of each cluster. In such cases, soft clustering is more desirable.

4.4.2 EM Algorithm

The EM algorithm is a widely used soft clustering approach for finite mixture distribution. When the HMT likelihood values are modeled as a finite Gaussian mixture of N components, the EM algorithm aims at finding:

$$\hat{\mathbf{\Theta}}_j = \arg\max_{\mathbf{\Theta}_j} p(\mathbf{z}^{(j)}|\mathbf{\Theta}_j), \tag{4.12}$$

$$p(\mathbf{z}^{(j)}|\mathbf{\Theta}_j) = \sum_{k=1}^{N} \alpha_{k,j} g(\mathbf{z}^{(j)}|\mathbf{\Theta}_{k,j}),$$

where $\mathbf{z}^{(j)}$ is all likelihood values at scale j, $\alpha_{k,j}$ is the weight of the k^{th} class at scale j, $\mathbf{\Theta}_{k,j}$ is the parameters defining the k^{th} component $g(\mathbf{z}^{(j)}|\mathbf{\Theta}_{k,j})$ of the Gaussian mixture, and $\mathbf{\Theta}_j = \{\alpha_{k,j}, \mathbf{\Theta}_{k,j}, k = 1, ..., N\}$ is the set of parameters defining the whole mixture. The soft clustering process is composed of an assignment (Expectation) and an update (Maximization) step:

⟨1⟩ Assignment Step: the probability that likelihood $z_l^{(j)}$ is of class k can be calculated as follows:

$$\begin{aligned} p_k(z_l^{(j)}) &= \frac{\alpha_{k,j} g(z_l^{(j)}|\mathbf{\Theta}_{k,j})}{\sum_i \alpha_{i,j} g(z_l^{(j)}|\mathbf{\Theta}_{i,j})}, \\ g(z_l^{(j)}|\mathbf{\Theta}_{k,j}) &= \frac{1}{\sqrt{2\pi}\sigma_{k,j}} \exp[\frac{-(z_l^{(j)} - m_{k,j})^2}{2\sigma_{k,j}^2}]. \end{aligned} \tag{4.13}$$

⟨2⟩ Update Step: the model parameters are estimated as:

$$\alpha_{k,j} = \frac{\sum_l p_k(z_l^{(j)})}{\sum_i \sum_l p_i(z_l^{(j)})}, \quad i = 1, ..., N,$$

$$
\begin{aligned}
m_{k,j} &= \frac{\sum_l p_k(z_l^{(j)}) z_l^{(j)}}{\sum_l p_k(z_l^{(j)})}, \\
\sigma_{k,j}^2 &= \frac{\sum_l p_k(z_l^{(j)})(z_l^{(j)} - m_{k,j})^2}{\sum_l p_k(z_l^{(j)})},
\end{aligned} \tag{4.14}
$$

where $z_l^{(j)}$ is the likelihood value of node l at scale j as defined in (4.6), $m_{k,j}$ is the likelihood mean of class k at scale j, and $\sigma_{k,j}^2$ is the likelihood variance of class k. In each iteration, the class probability is estimated based on a set of model parameters by (4.13), and the model parameters are updated using (4.14). The EM algorithm can efficiently handle the hyperellipsoidal distributed clusters if they are well separated, and the K-mean clustering could be used as the initialization of the EM algorithm.

4.4.3 Context-based Multiscale Clustering

In the model based soft clustering, the estimation of cluster models considerably depends on two factors. One is the feature representativeness, and the other is the available feature number. Since each node at the coarsest scale of a HMT model is associated with the largest area ($2^J \times 2^J$ pixels) of the original image compared with other nodes at finer scales, the likelihood value of the node is more representative than those at finer scales, and consequently the clustering on them is considered to be reliable in terms of the likelihood computation. Whereas, the limited number of trainable nodes ($N/2^J \times N/2^J$) may lead to inaccurate model estimation for each component of the Gaussian mixture. Although each node at scale $J-1$ is less representative than those at scale J, resulting in less robust feature extraction (likelihood computation), more trainable nodes ($N/2^{J-1} \times N/2^{J-1}$) are available for each class at scale $J-1$. Hence we could improve the clustering accuracy by incorporating nodes at both scale J and $J-1$. Moreover, based on the Bayesian framework, the local context information of the class label in the MSRF framework can be involved to further improve the estimation accuracy of cluster models by encouraging the formation of large homogeneous regions. In this work, a context-based multiscale clustering is proposed by involving local context and clustering results at the two coarsest scales of HMT.

As illustrated in Fig. 4.1, the two coarsest scale of the MSRF model are associated with the two coarsest scales of HMT. In order to simplify the multiscale fusion process, we make two assumptions based on the MSRF framework: 1) given the class label, an observation (likelihood value) is independent to others; 2) a parent block z_l^J at scale J and its four children at scale $J-1$ have the same class label denoted by $x_l^{(J)}$ [3]. If we use $\mathbf{z}_l^{(J)}$ to represent $z_l^{(J)}$ and its four children: $\mathbf{z}_l^{(J)} = \{z_l^{(J)}, z_{l,c}^{(J-1)} | c = 1, 2, 3, 4\}$, a joint conditional density of them is:

$$\begin{aligned} h(\mathbf{z}_l^{(J)}|x_l^{(J)}) &= g(z_l^{(J)}, z_{l,c}^{(J-1)}|x_l^{(J)}) \\ &= g(z_l^{(J)}|x_l^{(J)}) \prod_{c=1}^{4} g(z_{l,c}^{(J-1)}|x_l^{(J)}), \end{aligned} \tag{4.15}$$

where $z_{l,c}^{(J-1)}$ is the c^{th} child of $z_l^{(J)}$ at the finer scale $J-1$, and $g(.)$ is the class conditional density defined as (4.13). The posterior class probability of $\mathbf{z}_l^{(J)}$ can be estimated given its local context $v_l^{(J)}$ at the same scale [58]. In this work, the local 2-order neighborhood system $\mathbb{N}_l$ are adopted to estimate the local context variable $\hat{v}_l^{(J)}$, which is initialized using an soft voting method:

$$\hat{v}_l^{(J)} = \arg \max_{k \in \{1,\ldots,N\}} \sum_{t \in \mathbb{N}_l} p_{x^{(J)}|z^{(J)}}(k|\mathbf{z}_t^{(J)}), \tag{4.16}$$

$\hat{v}_l^{(J)}$ is an approximation to the local non-causal neighborhood system and applied to the context fusion as a causal prior. The contextual prior $p_{x^{(J)}|v^{(J)}}(k|u)$ is the solution that maximize the conditional density function:

$$p_{z^{(J)}|v^{(J)}}(\mathbf{z}^{(J)}|v^{(J)}) = \prod_{l \in S^{(J)}} \sum_{k=1}^{N} h(\mathbf{z}_l^{(J)}|k) p_{x^{(J)}|v^{(J)}}(k|v_l^{(J)} = u), \tag{4.17}$$

where $S^{(J)}$ is the 2-D lattice that contains all individual samples $z_l^{(J)}$ at the coarsest scale. Equation (4.17) can be estimated by the Bayesian rule. The final estimation of $x_l^{(J)}$ is obtained by:

$$\begin{aligned} \hat{x}_l^{(J)} &= \arg \max_{x_l^{(J)}} p_{x^{(J)}|v^{(J)},z^{(J)}}(x_l^{(J)}|\hat{v}_l^{(J)}, \mathbf{z}_l^{(J)}), \\ p_{x^{(J)}|v^{(J)},z^{(J)}}(x_l^{(J)}|\hat{v}_l^{(J)}, \mathbf{z}_l^{(J)}) &= \frac{p_{x^{(J)}}(x_l^{(J)}) p_{v^{(J)}|x^{(J)}}(\hat{v}_l^{(J)}|x_l^{(J)}) h(\mathbf{z}_l^{(J)}|x_l^{(n)})}{\sum_{k=1}^{N} p_{x^{(J)}}(k) p_{v^{(J)}|x^{(J)}}(\hat{v}_l^{(J)}|x_l^{(J)} = k) h(\mathbf{z}_l^{(J)}|x_l^{(J)} = k)} \end{aligned} \tag{4.18}$$

[3] This assumption does not hold true around boundaries or within small textured area, but a majority of the blocks around boundaries will not be involved to the following texture model estimation after the training sample selection [149].

Equation (4.18) has the same form as (4.8) except for two things. First, the cluster density $h(.)$ differs from the HMT density $f(.)$. Second, in the clustering $h(.)$ needs to be updated in each iteration of the context fusion, while $f(.)$ is fixed after it is estimated in the supervised segmentation [32, 58]. The density $h(\mathbf{z}_l^{(J)}, \hat{v}_l^{(J)}|x_l^{(J)}) = p_{v^{(J)}|x^{(J)}}(\hat{v}_l^{(J)}|x_l^{(J)})h(\mathbf{z}_l^{(J)}|x_l^{(J)})$ is used to update $h(\mathbf{z}_l^{(J)}|x_l^{(J)})$ during the context-based fusion because it incorporates the multiscale and neighboring information that eventually amends the cluster models.

Overall, the proposed CMSC consists of two steps. In step 1, soft clustering is performed independently at the two coarsest scales of the MSRF, then (4.15) is computed for a ML classification at the coarsest scale of HMT. In step 2, the local context at the coarsest scale of the MSRF is first initialized using (4.16), and the ML classification results at the coarsest scale is fused with the context prior via an iterative process. In this step, the conditional density function (4.17) is computed in each iteration of context fusion, and is also used as stop criterion when it converges. Special attention should be paid to the usage of the local context because small objects could be removed during the fusion.

4.4.4 Multiple Model Clustering

In practice, it is found that the 1-D likelihood histogram is usually enough for clustering. However, the distinction among some textures may not be effectively manifested by WDHMMs. Two examples are shown in Fig. 4.7. Fig. 4.7 (a) is the 1-D HMT likelihood histogram generated by mapping Mosaic-3 (5 classes) into θ_I. It can be seen that three clusters are well separated, whereas two are overlapped. When the likelihood disparity of different textures is too small to be captured, the above 1-D clustering methods could not work well. In clustering research, one of the most frequently used approaches to increase the cluster separability is to construct a higher-dimensional feature space. In this work, we propose a multiple model clustering method as shown in Fig. 4.8. By constructing a multidimensional feature (likelihood values) space by mapping image I into different HMT model specifications $\theta_i,\ i \in \{D16, D19, \ldots, Mixture\}$, better cluster separability is expected. For instance, after

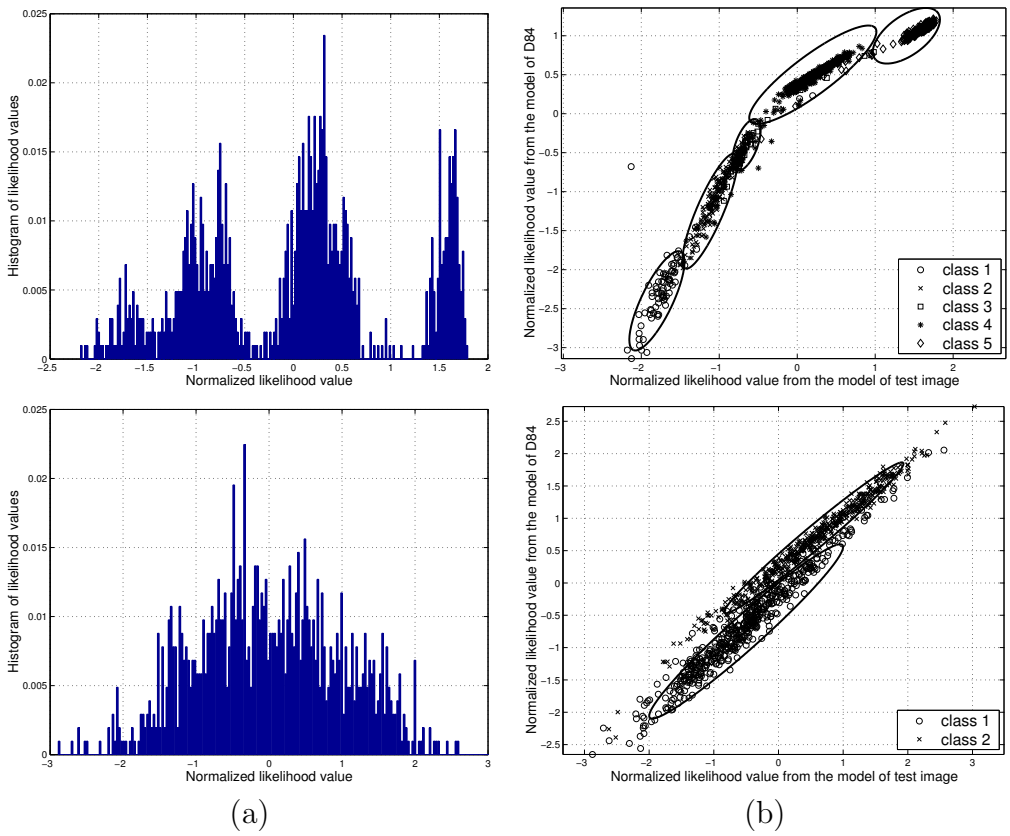


Figure 4.7: First row: Mosaic-3. Second row: Mosaic-8. (a) One dimensional likelihood histogram. (b) Two dimensional likelihood distribution.

adding one dimension of HMT likelihood value that is obtained by mapping Mosaic-3 into θ_{D84}, two overlapped clusters in Fig. 4.7 (a) are more separable as shown in Fig. 4.7 (b). Another example of Mosaic-8 is shown in the second row of Fig. 4.7. It can be seen that two clusters overlaps significantly in the 1-D likelihood histogram. Although K-mean clustering can split two clusters partially, considerable samples are misclassified. After incorporating the likelihood value that is obtained by mapping Mosaic-8 into θ_{D84}, the higher clustering accuracy is achieved by simply using K-mean clustering even though K-mean cannot well handle cigar-shaped clusters [113]. A comparison of MMC accuracy to other mentioned clustering methods is shown in Table 4.6.1 of Section 4.6.1.

How to choose additional model specifications to construct 2-D or higher dimensional likelihood spaces is another interesting issue. Experiments show the models that have large cluster divergence measurements could be helpful to increase the cluster separability in higher dimensional feature (likelihood) spaces. In practice, the

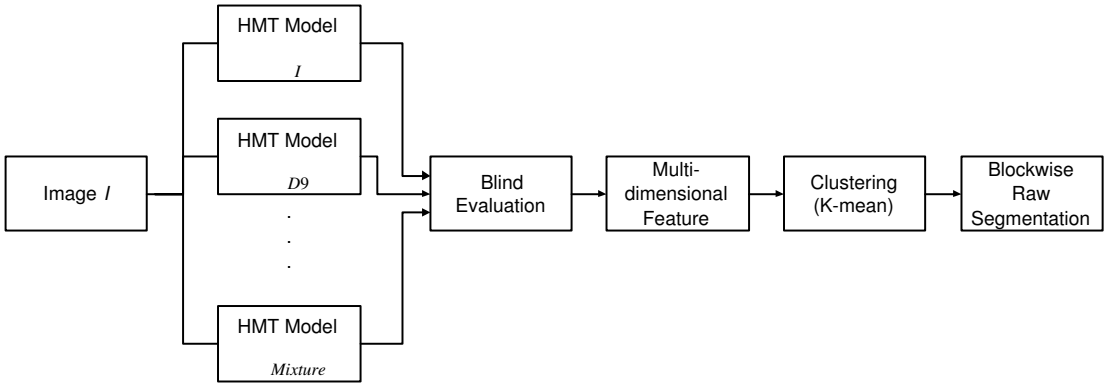


Figure 4.8: Multiple Model Clustering

selection of model specification can be only based on the blind evaluation results of the cluster divergence, namely, Shannon and Renyi entropy. Although MMC is not always necessary if 1-D likelihood histograms show enough cluster divergence, it provides an alternative approach that could further improve the clustering accuracy. Our studies show that the textures in Fig. 6.2 usually show hyperellipsoidal distributed clusters, and some of them are very close to each other in the feature space. If we want to fully explore the cluster separability from multiple model specifications, more complicated clustering methods should be considered.

4.5 Dual-model Segmentation Framework

After the study of the *soft-hard* hybrid decision approach, we now discuss the whole unsupervised segmentation framework. Since HMT model most often provides more cluster divergence, as well as its faster training process than HMT-3S, it is used in the hybrid decision to generate a raw segmentation map. Whereas, since each texture should be modeled more completely for the following self-supervised segmentation based on the raw segmentation map, HMT-3S is taken into account at this time. Therefore, a joint utilization of both HMT and HMT-3S results in a dual-model unsupervised segmentation framework, where the cluster divergence of textures is captured by HMT models, and each texture is remodeled using HMT-3S after the clustering step. The structure of the framework is shown in Fig. 4.9. It can be seen that the the hybrid *soft-hard* approach is well embedded into the dual-model

segmentation framework.

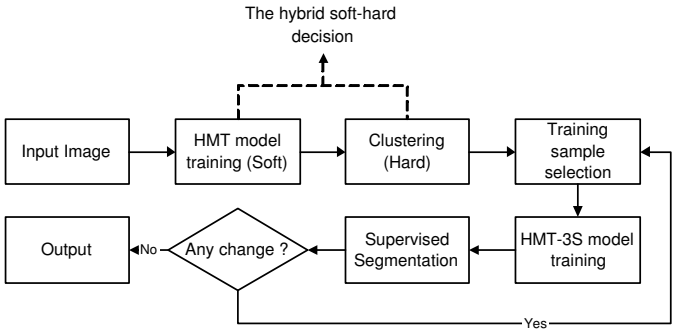


Figure 4.9: The proposed dual-model unsupervised segmentation framework.

In this segmentation framework, selecting reliable training samples (nodes) at the coarsest scale of HMT is another important step. It ensures that the training samples are only selected from the large homogeneous regions because the misclassified samples or those around texture boundaries will cause inaccurate texture model estimation. A sample is selected if a majority of its 2-order neighborhood have the same class label as it. As shown in Fig. 4.9, the iteration process could be repeated several times until no obvious change appears in final segmentation results.

4.6 Simulation Results and Discussion

The synthetic mosaics in Fig. 6.2 and some real images are used in the simulation, where each image is decomposed into 4 scales by the Haar wavelet transform. After building a HMT model for each image, each node at the coarsest scale of HMT corresponds to an image block of 16×16 pixels. Only the coarsest and the second coarsest scale of HMT are involved in the EM algorithm and CMSC. All these clustering methods are computational efficient due to the small number of nodes at the two coarsest scales of HMT (32×32 at the coarsest scale and 64×64 at the second coarsest scale). The overall processing time of a test image is typically less than one minute on a computer of Pentium-4 CPU (2.2GHz).

4.6.1 Synthetic Mosaics

The synthetic mosaics are shown in Fig. 6.2. Three numerical criteria are used to evaluate the segmentation performance [58]: P_a is the percentage of pixels that are correctly classified, showing the *overall segmentation accuracy*, P_b the percentage of boundaries that coincide with the true ones, showing *boundary specificity*, and P_c the percentage of boundaries that can be detected, showing *boundary sensitivity*. The blockwise clustering accuracy at the coarsest scale of HMT is denoted by $\tilde{P}_a$. A comparison of the clustering accuracy of the K-mean algorithm, EM clustering, CMSC and MMC (2 model specifications) methods is shown in Table 4.6.1 based on the synthetic mosaics. The percentage in parentheses is the increasing (+) or decreasing (-) in $\tilde{P}_a$ benchmarked against the K-mean clustering accuracy, and the highlighted bold fonts indicate the highest value in each test set.

Table 4.2: Overall clustering accuracy ($\tilde{P}_a$) at the coarsest scale of HMT (%).

Accuracy(%)	K-mean	Soft Clustering	CMSC	MMC (K-mean)
Mosaic-1	78.71	78.81(+0.10)	**84.38(+5.67)**	80.37(+1.66/Sand)
Mosaic-2	78.71	79.69(+0.98)	**90.33(+11.62)**	79.88(+1.17/D112)
Mosaic-3	64.36	71.58(+7.22)	73.63(+9.27)	**85.74(+23.53/D84)**
Mosaic-4	88.38	87.30(-1.08)	**91.60(+3.22)**	89.65(+1.27/Mix)
Mosaic-5	92.29	92.38(+0.09)	**94.14(+1.85)**	92.09(-0.2/Sand)
Mosaic-6	93.16	90.43(-2.73)	90.63(-2.53)	**94.04(+0.88/D112)**
Mosaic-7	92.29	93.26(+0.97)	**95.02(+2.73)**	93.36(+1.07/Sand)
Mosaic-8	73.92	73.63(-0.29)	**80.08(+6.16)**	77.83(+3.91/D84)

Table 4.3: Segmentation performance comparison (I: K-mean, II: CMSC, III:MMC).

Numerical Results	P_a (Accuracy%)			P_b (Boundary Specificity%)			P_c (Boundary Sensitivity%)		
Mosaics	I	II	III	I	II	III	I	II	III
Mosaic-1	96.50	96.46	**96.71**	39.03	32.25	**42.21**	62.79	57.61	**63.51**
Mosaic-2	98.28	**98.87**	98.72	40.39	**49.38**	45.55	52.64	**58.10**	56.20
Mosaic-3	86.78	89.49	**94.56**	8.06	35.29	**35.33**	43.59	44.81	**67.39**
Mosaic-4	96.59	**98.31**	97.74	21.28	**30.72**	25.77	47.77	54.58	**55.09**
Mosaic-5	99.06	**99.17**	99.08	46.07	**52.27**	49.05	62.26	**63.73**	62.23
Mosaic-6	98.61	94.51	**98.65**	32.28	21.29	**32.69**	37.22	**43.61**	39.01
Mosaic-7	98.72	**98.81**	98.80	39.05	**44.47**	40.16	59.78	60.85	**61.58**
Mosaic-8	98.18	98.35	**98.46**	16.59	8.78	**21.73**	25.02	16.83	**32.14**

As illustrated in Table 4.6.1, the EM clustering yields similar performances or slight improvements compared with K-mean. Since most textures in the test images are hyperellipsoidally distributed in feature (likelihood) spaces, the EM algorithm should learn the cluster model better if clusters are not very close to each other. The EM algorithm is also used to initialize the CMSC approach. The iteration times of context-based fusion in CMSC is adaptively controlled by the convergence of (4.17). CMSC outperforms K-mean on 8 synthetic mosaics except for Mosaic-6. This indicates that the effectiveness of context information as denoted in (4.18). MMC usually

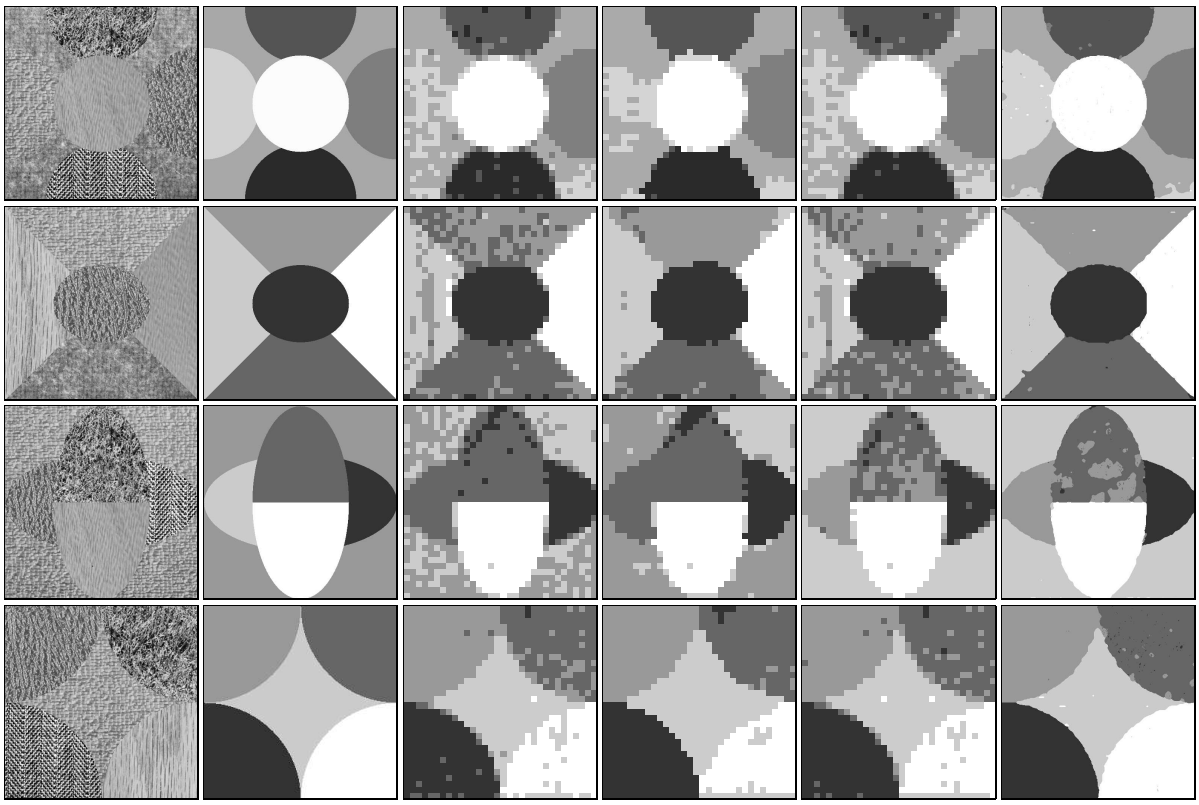


Figure 4.10: Synthetic mosaics and simulation results. (a) Mosaics. (b) Ground truth. (c) K-mean clustering results. (d) CMSC results. (e) MMC results. (f) Final pixel level segmentation results.

can improve clustering accuracy by increasing the dimension of feature (likelihood) spaces. An example of Mosaic-3 is shown in Fig. 4.7 (a). There are 5 textures in this synthetic mosaic, and textures D9 and D24 are highly overlapped in the 1-D HMT likelihood histogram. K-mean and EM algorithm merely classify them as one cluster, resulting in four clusters. CMSC cannot classify these two textures either because the initial EM algorithm cannot identify their disparity. By adding a dimension of likelihood values obtained by mapping Mosaic-3 into θ_{D84}, all five textures can be well segmented. Moreover, we would like to mention that sometimes the 4-scale DWT may not be sufficient to capture large scale texture behaviors, resulting in inadequate model representation. Accordingly, increasing the DWT scale could be necessary when significantly large scale texture behaviors exist.

Table 4.6.1 compares the final pixel level segmentation performances in terms of different clustering methods. It can be seen that sometimes the improvement in $\tilde{P}_a$ cannot increase the final segmentation accuracy. There are three possible reasons:

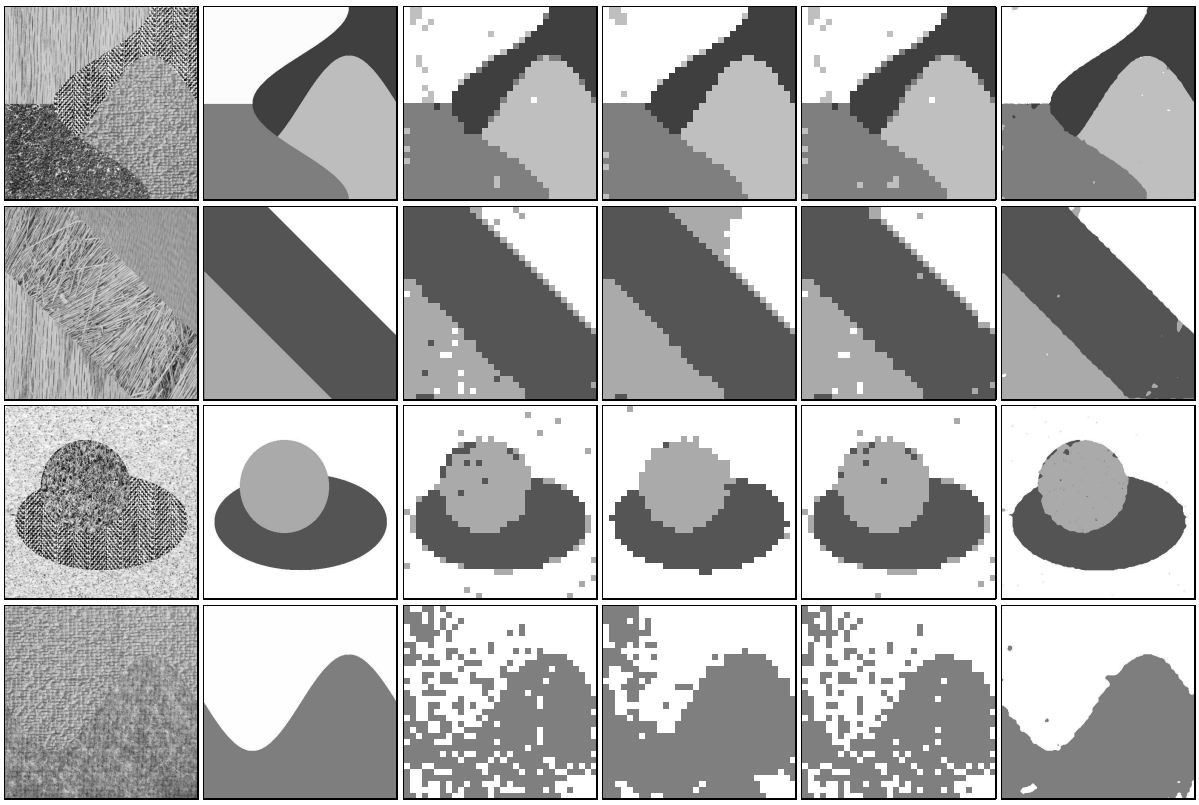


Figure 4.11: Synthetic mosaics and simulation results. (a) Mosaics. (b) Ground truth. (c) K-mean clustering results. (d) CMSC results. (e) MMC results. (f) Pixel level segmentation results.

(1) The total number of nodes at the coarsest scale of HMT is small. A 2% improvement of $\tilde{P}_a$ means about 20 more correctly classified nodes. If the number of classes is more than 4, the average increasing of correctly classified training nodes for each class is less than 5, which might be negligible, contributing little to the following HMT-3S model estimation and the final pixel level segmentation.

(2) If the nodes at the coarsest scale of HMT are not representative enough, they could contribute differently to the HMT-3S model estimation that is related to the final pixel level segmentation accuracy. Accordingly, although more nodes could be correctly classified by CMSC or MMC, the final segmentation accuracy could be improved little or even be worse. This deficiency could be mitigated by increasing the DWT scales, making each node more representative, or by developing new criteria to select training samples, which might be an interesting topic in the future.

(3) CMSC tends to generate large homogeneous regions, where the location of the texture boundary might not be accurate. Training sample selection approach may not remove all the misclassifications located on the boundaries, and the accuracy of pixel-level segmentation will be affected by those remained misclassifications.

The clustering results and the final pixel level segmentation results of 8 synthetic mosaics are shown in Figs. 4.10 and 4.11. The five columns refer to, respectively, the original synthetic mosaics, the ground truth of class label, the K-mean clustering results, the CMSC results, the MMC results, and the pixel level segmentation results. All results in Table 4.6.1 and Figs. 4.10 and 4.11 are obtained with one iteration only. It is shown that a good clustering result is essential to the segmentation performance. Better results could be obtained with more iterations. Because some images do not have much dependency among their neighborhoods at the coarsest scale of the image pyramid, especially the multispectral satellite imagery or Synthetic Aperture Radar (SAR) imagery, the local context fusion may not be helpful. However, if there exist homogeneous textures, the improvement from the local context fusion could be prominent.

4.6.2 Real Images

The proposed segmentation algorithm is also tested on five real images of three types: aerial photo, indoor, and outdoor pictures as shown in Fig. 4.12. The rows refer to the original real images, the clustering results at the coarsest scale of the HMT model using K-mean, CMSC or MMC approach, and the unsupervised segmentation results, respectively. The number of clusters in each image is fixed to 3 classes. Although the texture distribution in these images are not very homogeneous and non-uniform, the proposed method still yields good performance.

Vehicle and ***Peninsula*** are aerial photo. Both K-mean and CMSC performs well on them. We test MMC on ***Vehicle*** to demonstrate its applicability to real images. After mapping ***Vehicle*** into the 16 HMT model specifications mentioned

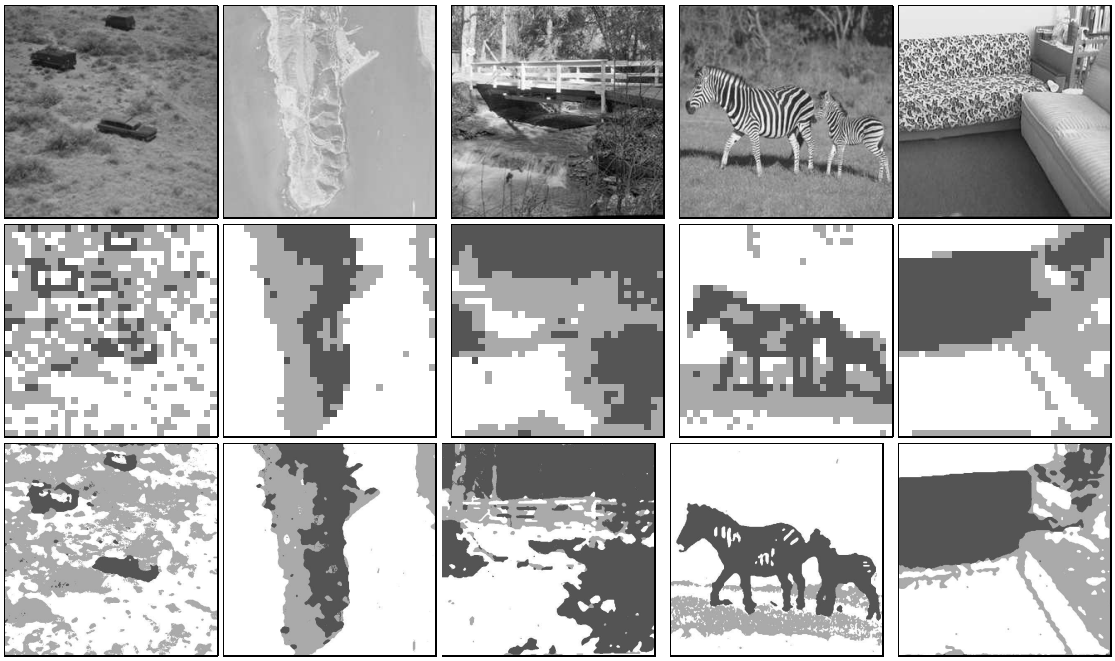


Figure 4.12: Unsupervised segmentation of real images. (a) The clustering (MMC) and segmentation results of ***Vehicle*** image. (b) The clustering (CMSC) and segmentation results of ***Bridge*** image. (c) The clustering (CMSC) and segmentation results of ***Sofa*** image. (d) The clustering (K-mean) and segmentation results of ***Zebra*** image. (e) The clustering (CMSC) and segmentation results of ***Peninsula*** image.

in Section 4.3.1, the likelihood values obtained in θ_I and $\theta_{Mixture}$ are chosen to construct a 2-D feature space. The clustering and final segmentation results indicate the usefulness of MMC. ***Bridge*** and ***Zebra*** are outdoor pictures, and ***Sofa*** is an indoor picture. The man-made structures, natural plants and animals in these pictures can all be well segmented out. In the simulation, we found that all mentioned clustering approaches mentioned can produce semantically correct segmentation maps. Usually for those with large homogeneous regions, such as ***Sofa***, the CMSC could perform better. For the images with small homogeneous regions, such as ***Zebra***, K-mean is preferred to preserve more details.

4.7 Summary

In this chapter we proposed a new unsupervised texture segmentation method based on the WDHMMs. First a hybrid *soft-hard* decision approach is suggested to

obtain an initial blockwise segmentation map at the coarsest scale of a WDHMM. This map is used to identify training samples for the following self-supervised segmentation. In this hybrid approach, the *soft*-decision step determines the cluster divergence measured by the likelihood disparity, and the *hard*-decision step captures the likelihood disparity to generate the raw segmentation map via clustering. Specifically, in the *soft*-decision step, the image to be segmented is mapped into a WDHMM, and the experimental study show that the model generated from the image itself usually provides better cluster separability. In the *hard*-decision step, two new clustering methods are developed and show better performance compared with K-mean and EM algorithm in the simulation. Furthermore, the study of the cluster divergence shows that HMT has comparable or better cluster separability than HMT-3S with respect to a given image. Therefore a dual-model unsupervised segmentation framework is suggested, where the raw segmentation is obtained based on the HMT model, and each unknown texture in the image is modeled by HMT-3S for the self-supervised segmentation. Simulation results show that the proposed segmentation method performs well on complex synthetic mosaics and real images, and the segmentation results of the synthetic mosaics are close to the supervised case.

Given the framework of the unsupervised segmentation algorithm, there are two more interesting issues that might deserve further pursuing. First, Although the orthogonal Haar DWT is used in this work and show very good performance, it is expected that the implementation of redundant wavelets would allow for a better segmentation performance. Second, besides WDHMMs, the proposed hybrid *soft-hard* decision is also applicable to other statistical models if there is a closed form for likelihood computation.

Chapter 5

NONPARAMETRIC SUPERVISED SEGMENTATION OF SATELLITE IMAGERY

The classification of remotely sensed imagery into different areas for Land Use Land Cover (LULC) analysis has been an important topic in past decades. Conventional parametric statistical model-based methods show their efficiency in such problems [48, 93]. In recent years, a variety of works have used multisource geospatial data to facilitate the classification of multispectral imagery [85, 82, 147, 17]. Correspondingly, people found that it may not be appropriate to model multisource data by traditional multivariate statistical models [85, 82, 14, 105, 10]. Therefore, nonparametric methods should be considered. In this work, we study nonparametric machine learning approaches for mapping United States Department of Agriculture (USDA)'s Conservation Reserve Program (CRP) tracts based on satellite imagery, which is a special and complex problem of LULC analysis. CRP is a program that encourages farmers to plant long-term resource conserving covers to improve soil, water and wildlife resources [1]. Very little work has been done for CRP mapping so far, and recent work in [51] requires considerable human interpretation and intervention.

Compared with the traditional LULC applications, CRP mapping has several major characteristics that make it a complicated problem. (1) CRP mapping is a 2-class classification problem (CRP and non-CRP) of complex rural area where each class is a mixture of many different land cover types, resulting in highly overlapped clusters in the spectral spaces of satellite imagery. Therefore, representative feature sets and powerful data classifiers are necessary. (2) Existing CRP reference data provided by Natural Resources Conservation Service (NRCS) is not very accurate or up-to-date, and we need a specific way to select reliable training samples and to

evaluate the mapping performance from the present reference data. Moreover, based on the mapping results, we can also correct some errors in the present reference data. (3) CRP mapping is an uneven classification task where CRP tracts usually consist of less than 10% over all study areas. Accordingly, methods that favor high *recall* rates should be considered. (4) Since CRP mapping is a nationwide program, any CRP mapping tools developed should be computationally efficient with minimum human involvement.

Basically, CRP is a man-made program that possesses strong correlation with geographic information system (GIS) data, such as slope, elevation, and distance-to-waterbody, therefore the importance of multisource GIS data is prominent in the CRP mapping application. For example, if a significant number of training samples are available, the cluster separability of different classes can be increased in a higher dimensional feature space constructed from multisource data. The work in [17] shows the advantages of using neural networks for the classification of complex rural areas. In this work, we study the decision tree classifier (DTC) and support vector machine (SVM) [155] for CRP mapping. The principle of the DTC is to break up a complex classification problem into a union of several simpler classification issues. The SVM constructs a linear classification hyperplane that maximizes the margin between two different training patterns in the original feature space or a high dimensional feature space generated by kernel methods [155, 15, 35, 19]. In our recent work [149], both the DTC and SVM were used to implement CRP mapping based on multisource geospatial data, where the CRP reference data was used as ground truth for performance evaluation. However, there is usually no significant mis-location of CRP tracts in the reference data; therefore, some locality error around CRP boundaries could deteriorate the purity of training sample, and invalidate the performance evaluation. In this work, we use a specific approach to refine the classifier training and to evaluate the performance of CRP mapping.

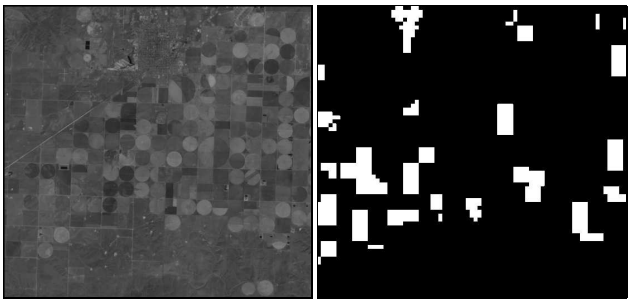
Since the CRP mapping is an uneven 2-class unsupervised classification problem, besides the overall *classification accuracy*, *precision* (user's accuracy) and *recall* (producer's accuracy) are used to evaluate the overall CRP mapping performance. In our previous work [149], we found that pruned DTCs and SVMs favor high *precision*,

leading to low *recall* if the *classification accuracy* cannot be increased significantly. However, failing to detect existing CRP tracts is more undesirable than false CRP tracts when we deal with problems related to CRP mapping, e.g., compliance monitoring; thus, high *recall* outweighes high *precision* in practical CRP mapping. Therefore, we propose a new DTC pruning method to increase *recall*. We also study two relaxation approaches for the SVM to improve *recall* specifically. Moreover, we propose a localized and parallel classification framework to implement CRP mapping for large areas efficiently and effectively.

More details of the above issues will be discussed in the following sections. In Section 5.1, USDA's CRP program and the study area will be introduced. The DTC and the SVM are briefly described in Section 5.2. In Section 5.3, the localized framework is proposed based on the multisource geospatial data. Section 5.4 studies how to improve the sensitivity of the DTC and the SVM for CRP mapping. Section 5.5 shows and discusses simulation results. Conclusions are drawn in Section 6.4.

5.1 The CRP Program and Study Area

This work is originally motivated by the need for mapping USDA's CRP tracts from remotely sensed data. The CRP is a provision of the 1985 Farm Bill that seeks to convert highly erodible lands with active crop production to permanent vegetative cover [23]. It is a voluntary program that uses financial incentives to encourage farmers to enroll in contracts of 10-15 years in duration to remove lands from agricultural production. Enrolled lands must be highly erodible, contribute to a serious water quality problem, or provide substantial environmental benefits if devoted to certain conservation uses. USDA's Farm Service Agency (FSA), in-charge of administering the CRP signups and enrollments, evaluates the fields submitted by the producers based on the Environmental Benefit Index (EBI) score accumulated by each farm applicant (FSA 2003). This process implicitly associates CRP enrollments with multisource GIS information. Depending on the overall applicants, a cutoff EBI is identified, above which farms get selected for long-term retirement with rental benefits. Starting in 1998, with the initial CRP contracts beginning to expire and a nearly $1.6 billion new enrollment in 2003, it is imperative for FSA to evaluate and



(a) (b)

Figure 5.1: The study area in Texas County, Oklahoma (February, 2000): (a) Landsat TM image band 4, which is of size 552×523 pixels, corresponding to an area of 260km^2. (b) CRP reference data.

manage this program based on accurate and detailed digital CRP maps, which are usually not available or need to be updated.

Currently, there is no standardized approach to keeping track of existing CRP tracts. FSA relies on aerial photography to manually delineate CRP tracts on a county level basis. These aerial photographs are at the section level, and provide little information about the CRP from a landscape perspective. Furthermore, when NRCS drafts the CRP reference data, the possible mis-locality and spatial misalignment of CRP tracts deteriorate the reliability and usability of these reference data [51]. Therefore, the goal of this work is to develop an automatic tool for accurate CRP mapping based on the existing CRP reference data provided by NRCS.

The study area of this work is located in Texas County, Oklahoma as shown in Fig. 5.1 (a). This area (552×523 pixels) is about 260km^2, where the accurate CRP mapping framework is being developed and tested. Fig. 5.1 (b) illustrates the imperfect CRP reference data of this area. Texas County is one of the most intensively farmed counties in Oklahoma. Because of the underlying water-rich Ogallala Aquifer, irrigated farming is extensively practiced in the area for corn, sorghum, cotton, and soybeans cultivation. Due to the large scale of agriculture for many years, Texas County also ranks first in the state for CRP enrollments. Therefore, it is a salient region for the study of CRP mapping.

5.2 Machine Learning Approaches

Machine learning is the ability of a machine to recognize patterns that have occurred repeatedly and to improve its performance based on past experiences. It is a typical machine learning problem to acquire general concepts for two different land covers, i.e., CRP and non-CRP, from given training samples. The target function of CRP mapping is defined as:

$$Y = f(\mathbf{X}), Y \in \{0, 1\}, \tag{5.1}$$

where $\mathbf{X}$ is the multisource geospatial data, $f(\cdot)$ is the target concept to be learned, and Y is an indicator where 1 can be defined as CRP and 0 as non-CRP. Both DTC and SVM are inductive inference methods, and their learning goal is to determine a hypothesis of the target concept that best fits the training data. The DTC has shown advantages in real remote sensing (RS) applications for more than ten years [82, 9, 145, 64, 41, 78]; however, considering that the overfitting problem is met by the DTC with poor generalization performance, the SVM is suggested as an alternative to the DTC. Recent research on the SVM in RS applications have shown impressive classification results [73, 72, 71, 80, 118, 16]. In this work, both the DTC and SVM are used for CRP mapping as a semi-supervised classification issue involving multisource geospatial data. Particularly, the generalization performance of the two machine learning approaches is carefully studied to produce accurate CPR maps with high *recall* rates.

5.2.1 Decision Tree Classifier (DTC)

The DTC is a tree-structured classifier built from a training data set, representing rules underlying training data with hierarchical and sequential structures that recursively partition the data. In this work, the *C4.5* DTC is applied to CRP mapping [132]. It is constructed based on the *information gain ratio* criterion, which measures the increase in class purity. Assuming a set of samples S that contains k classes with probability $p_1, ..., p_k$, if S is partitioned into n classes based on a test, the

information gain ratio is defined as:

$$\text{Gain-ratio}(S) \;=\; \frac{\text{Gain}(S)}{-\sum_{i=1}^{n}\frac{|S_i|}{|S|}\log(\frac{|S_i|}{|S|})}, \tag{5.2}$$

where

$$\text{Gain}(S) \;=\; \text{Info}(S) - \sum_{i=1}^{n}\frac{|S_i|}{|S|}\text{Info}(S_i),$$

and

$$\text{Info}(S) \;=\; -\sum_{i=1}^{k} p_i \log(p_i).$$

In equation (5.2), $|S_i|$ is the number of samples in subset i and $|S|$ is the number of samples in the set S. Gain(S) is the gained information of the target function that is obtained from the test with selected features, and Gain-ratio(S) is a normalized *information gain* so that the bias of trivial partition could be avoided [131]. Beginning from the root node, the *C4.5* performs a top-down greedy search through the complete hypothesis space until the stop criterion is met. In this work, the tree stops growing if there are less than five samples in a node.

5.2.2 Support Vector Machine (SVM)

SVMs are newly developed learning methods [155]. Given a set of training samples from two classes: $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\},\ \mathbf{x} \in R^n,\ y \in \{1, -1\}$, the goal of SVM learning is to determine a classification hyperplane induced from the training samples that maximally separates classes, or equivalently, to minimize $\frac{\|\mathbf{w}\|^2}{2}$, subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0,\ y_i \in \{1, -1\},\ \forall i. \tag{5.3}$$

where $\mathbf{w}$ and b are parameters of the hyperplane. If the training data are linearly nonseparable, the hyperplane can be obtained by minimizing:

$$C\sum_{i=1}^{l}\xi_i + \frac{1}{2}\|\mathbf{w}\|^2, \tag{5.4}$$

subject to :

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, \tag{5.5}$$

where $\xi_i \geq 0,\ i = 1, ..., l$ are called *slack variables*, and C indicates the tradeoff between the complexity of classification hyperplane and the ratio of nonseparable data samples.

In SVM learning, kernel methods are often used to map the data vectors in the input space into a higher dimension feature space, then the construction of a linear classification hyperplane in this high dimension feature space is equivalent to a nonlinear decision hyperplane in the input lower dimension space [15, 19]. There are several often used kernel functions, such as the radial basis function (RBF):

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-|\mathbf{x} - \mathbf{x}_i|^2/2\sigma^2), \tag{5.6}$$

where σ is related to the function width. In this work, we use a nonlinear SVM with a RBF kernel, and a SVM software SVM^{light} [88] is used to perform training and classification of geospatial database. SVM parameters, i.e., σ in RBF kernel and regularization factor C, are usually determined by cross validation or experience [30, 139].

5.2.3 Performance Measurements

The generalization performance is one of the most important issues of machine learning approaches because it shows how well the learned hypothesis approximates the true target concept. Regarding the CRP mapping performance, three measurements are used in this work: *classification accuracy* (P_a) is defined as the percentage of pixels that are correctly classified in terms of CRP and non-CRP. *Precision* (P_b) indicates the percentage of detected CRP pixels that are true ones. *Recall* (P_c) is the percentage of true CRP pixels that can be detected. Recent research reveals that any classification system that performs better than a random decision exhibits a tradeoff between *precision* and *recall* if *classification accuracy* is a constant [4]. This implicates that if we cannot further increase *classification accuracy*, we could only improve *precision* by sacrificing *recall*, and vice versa. A further increase of *precision* and *recall* could not happen simultaneously unless *classification accuracy* could be increased, which is difficult and costly. Therefore, searching for a proper tradeoff

is more realistic in the case of CRP mapping. According to [4], a tradeoff between *precision* and *recall* is formulated as:

$$\lambda P_c + (\lambda + P_a - 1)P_b = 2\lambda P_b P_c, \tag{5.7}$$

where λ indicates the probability that a randomly selected sample belongs to the class of interest. Since CRP mapping is an uneven classification problem, where CRP tracts might cover less than 10% of a whole study area, i.e., $\lambda \approx 0.1$ in most cases, it is necessary that the trained classifiers can achieve high P_c for testing data. We will discuss how to increase P_c for DTC and SVM in Section 5.4.

5.3 Classification Framework

5.3.1 Geospatial Database

The geospatial database is composed of the Landsat TM satellite imagery, vegetation indices, texture information, and GIS data, which are all in raster format. Combined with LULC GAP data and CRP reference data, there are a total of 40 layers as shown in Fig. 5.2. During CRP mapping, layer sets A and D are original inputs, and layer sets B and C are automatically generated by the system during run time.

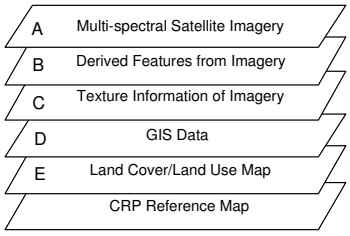


Figure 5.2: Multisource geospatial database.

Layer set A consists of the Landsat TM multispectral images obtained in February and June of year 2000 with a resolution of $30m \times 30m$. Since Band 1 is prone to scattering, we do not use it in the database. Bands 2, 3, 4, 5, and 7 for the two seasons are used in the study, resulting in the first 10 layers of the database from top to bottom. All layers were geometrically and radiometrically corrected before

being applied to the classification. The accuracy of the geometric correction is 0.5 pixels, and the technique outlined in [26] is used to perform radiometric correction.

Layer set B contains vegetation indices that include the *Normalized Vegetation Difference Index* (NDVI), and band ratios TM4/TM3, TM5/TM2, TM5/TM4. The NDVI for Landsat TM is computed as:

$$NDVI = \frac{TM4 - TM3}{TM4 + TM3}, \tag{5.8}$$

where $TM3$ and $TM4$ are spectral values in bands 3 and 4, respectively. The NDVI is calculated from the imagery in each season and the largest one is chosen as the final value. It can be used to discriminate different vegetation cover types. Band ratio TM4/TM3 (Ratio vegetation index) is widely used for vegetation discrimination. Ratio TM5/TM2 is helpful to discriminate different vegetation types [107]. TM5/TM4 (Ratio drought index) can provide more information of plant water content [33], which is useful to discriminate irrigated crops from relatively dry CRP grasses. Totally, there are 7 layers in Layer set B that are composed of NDVI (1 layer), and three band ratios of two seasons (6 layers).

Layer set C consists of 20 layers of texture information, including local mean and local variance of each band in each season. The local mean and local variance are computed on the spectral value within a 3×3 window. The texture layers are followed by GIS data of Layer set D, including *elevation* that ranges from 881 to 986 feet, *slope* that is from 0 to 30, and *distance-to-waterbody* with the extent from 0 to 3230 feet. The LULC GAP data could be used for more robust image analysis with respect to different cover types. The bottom layer is the reference data for training and/or evaluation purposes.

5.3.2 Feature Extraction

The geospatial database is directly applied to the DTC that generates a set of rules that are easy to interpret and understand. On the other hand, the database needs to be pre-processed before implementing SVM. First, it is necessary to normalize each data layer to be zero-mean and unit variance. This normalization can balance

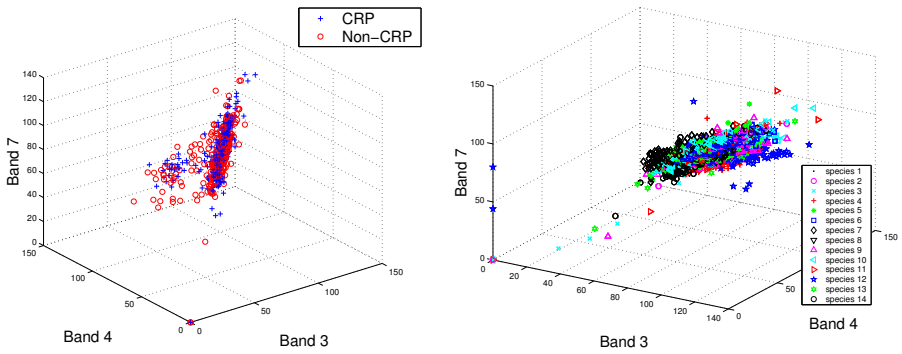


Figure 5.3: (a) 3-D feature spaces of CRP and non-CRP regions. (b). Overlap of CRP species in 3-D feature spaces, where species type 1 is Old World Bluestem, type 2 is Plains Bluestem, type 3 is WW Spar, type 4 is Ganada, type 5 is Plains Bluestem (1986), type 6 is Ganada (1986), type 7 is Old World Bluestem (1987), type 8 is Caucasian (1987), type 9 is Plains Bluestem (1987), type 10 is Plains (1988), type 11 is Plains (1989), type 12 is WW Spar (1989), type 13 is Old World Bluestem (1990), and type 14 is Native Mixture (1990).

the relative importance between different layers. Second, since the Landsat TM spectral channels and the derived features contain highly redundant information, it is necessary to reduce the feature redundance via feature selection or extraction. In this work, we use discriminant analysis feature extraction (DAFE) [65] to extract feature subsets from 5 multispectral image bands, 5 layers of local mean, as well as 5 layers of local variance for each season separately. In each set of 5-layer data, the three layers of extracted features with the largest eigenvalues are preserved. As the result, there are 9 layers for each season and totally 18 layers for two seasons. Including 7 layers of vegetation indices and 3 GIS layers, there are totally 28 layers for SVM-based CRP mapping. Since DAFE only works well when CRP and non-CRP are normally distributed, it might not be able to produce the most discriminative features. It is expected that more effective features could be obtained by using advanced feature extraction methods [65, 100].

5.3.3 Localized Data Classification

When selecting training samples, a straightforward way is to select samples from the whole study area. Nevertheless, it is worth pointing out that there are more

than 30 different CRP species in Texas County, Oklahoma, and there are 14 CRP species in the study area shown in Fig. 5.1. On the other hand, there exist many other cover types in non-CRP regions as well, such as crop, urban, and pasture, etc. For example, the 3-D spectral feature distributions of both CRP and non-CRP in the study area are illustrated in Fig. 5.3 (a). It can be easily observed that the spectral features of CRP and non-CRP highly overlap in bands 3, 4, and 7.

Hence, CRP mapping is actually a multi-class classification problem, where CRP and non-CRP areas are composed of many cover types. Both DTC and SVM approaches can be applied to multi-class problems. Specifically, in the case of CRP mapping, multi-class DTC or SVM approaches would involve the training and classification of each CRP species individually. However, since CRP mapping is a large scale problem, multi-class DTC training could lead to a complex and inefficient tree structure with very large size even after pruning. There are two typical multi-class SVM methods. One is the "one-against-one" [139], where the SVM training and classification are based on each pair of cover types individually. This approach has some problems: (1) Detailed cover types are not available. (2) Certain CRP species are mixed grasses, which also overlap in the feature space as shown in Fig. 5.3 (b). (3) The number of training samples is very large because they should cover all possible ranges of elevation, slope and distance-to-waterbody. This will increase the computational load tremendously. The other multi-class SVM method is the "one-against-all" [12], where a SVM is trained with samples from a CRP species as the positive class, and with samples of all other cover types as the negative class. Still its practical application is also limited by above three problems. More detailed discussions of these methods can be found in [81].

Generally, a multi-class DTC or SVM is not efficient enough to deal with the large scale data sets. In this work, we suggest a localized block-based technique to achieve automatic CRP mapping efficiently. The proposed technique splits the study area into small blocks, and the DTC or SVM training and classification are performed within each block independently. Then the outputs of all blocks are combined to rebuild the whole CRP map. The classification framework is shown in Fig. 5.4.

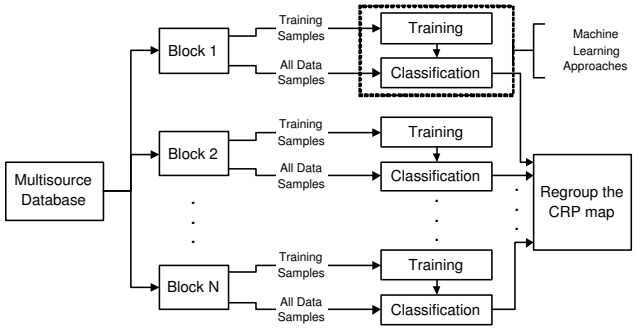


Figure 5.4: Block-based (Localized) Classification framework.

We have four major arguments to support the proposed block-based operation. (1) Less cover types exist in each block and the overlap between CRP and non-CRP areas in the feature space could be reduced. (2) Reliable training samples (above 50% of the true CRP areas) are usually available from the reference data in each block if the block size is sufficiently large. (3) Since CRP mapping is performed in each block individually, training and classification processes are very efficient. Furthermore, the block-based operation leads to a parallel classification structure. For example, when 20% of CRP and non-CRP areas are used for training and the remaining areas are are used for testing, it takes more than half an hour (Pentium IV 2.2GHz CPU, 1GB memory) on the whole study area. If the study area is split into 25 blocks of size around 100×100 pixels, the time for SVM training and classification in each block is about 10 seconds, and 30 seconds for the DTC. If we have a parallel computing architecture, the whole area can be processed efficiently.

The proposed block-based technique assumes independence across all local blocks. In reality, each block is not completely independent to other blocks. We manifest this fact by randomly selecting five blocks. Three different block sizes are studied: 50×50, 100×100, and 150×150 pixels. The SVM is trained from each of five blocks, and it is used to classify all five blocks. Simulation results are listed in Table 5.1, where *NA* denotes no sample is detected as CRP. It is shown that a trained SVM *only* performs well in the block where it is trained. Since CRP mapping is an uneven classification problem, $P_a = 90\%$ does not mean a good performance without high P_b and P_c.

Table 5.1: The study of inter-block dependency via training-classification process at 20% sampling rate.

50	1			2			3			4			5		
	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c
1	**96.95**	**94.47**	**98.81**	74.75	0.0	0.0	71.32	45.07	36.36	44.69	52.0	1.56	80.91	7.52	4.78
2	56.42	66.67	0.59	**97.55**	**84.08**	**94.91**	73.43	NA	0.0	44.89	100.0	0.48	87.61	NA	0.0
3	56.68	51.92	12.02	87.93	3.33	0.56	**98.07**	**95.74**	**97.05**	44.16	23.08	0.36	87.2	0.0	0.0
4	61.22	53.27	91.84	51.02	14.12	72.63	64.01	40.10	71.82	**98.41**	**97.65**	**99.52**	36.34	3.54	15.79
5	56.29	NA	0.0	89.56	NA	0.0	73.43	NA	0.0	44.63	NA	0.0	**95.73**	**79.91**	**87.56**

100	1			2			3			4			5		
	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c
1	**96.96**	**93.79**	**96.38**	79.68	4.00	0.48	77.16	22.11	12.69	90.40	1.72	1.12	90.66	28.87	28.63
2	69.24	37.50	1.66	**98.42**	**95.91**	**95.45**	82.06	25.00	1.85	93.48	3.92	0.45	91.77	9.52	2.95
3	69.10	44.71	6.58	80.63	13.86	1.12	**97.98**	**92.34**	**96.30**	91.61	1.04	0.45	92.16	15.79	4.42
4	69.57	NA	0.0	81.70	NA	0.0	82.65	0.0	0.0	**99.55**	**98.14**	**94.18**	93.42	NA	0.0
5	69.57	NA	0.0	81.71	100.0	0.08	82.71	100.0	0.08	94.10	NA	0.0	**98.93**	**90.45**	**93.68**

150	1			2			3			4			5		
	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c	P_a	P_b	P_c
1	**96.56**	**92.38**	**98.22**	88.89	0.68	5.45	88.22	72.07	12.62	92.68	0.81	0.77	93.99	7.48	1.03
2	65.18	NA	0.0	**99.65**	**82.11**	**91.82**	87.23	NA	0.0	96.20	NA	0.0	94.62	NA	0.0
3	73.16	93.94	24.51	85.06	0.21	2.27	**96.63**	**84.78**	**89.76**	92.92	0.53	0.46	88.85	25.16	54.38
4	65.11	0.0	0.0	98.74	0.0	0.0	87.07	0.0	0.0	**98.47**	**77.46**	**84.36**	94.59	10.0	0.06
5	65.18	NA	0.0	98.75	NA	0.0	87.23	NA	0.0	96.2	NA	0.0	**99.29**	**91.92**	**95.23**

When the block size is smaller, the CRP and non-CRP cover types in one block tend to be purer and more distinct compared with other blocks. Then the locally trained SVM may not be applicable to other areas. Moreover, if block size is larger, due to more complex cover types in each block, the applicability of a locally trained SVM to other areas is not good yet. Therefore, this result validates the assumption of block independence.

CRP training samples indicated by the reference data are usually available for each block if the block size is large enough. In the case of an unknown block without training samples, we still could perform CRP mapping using the aforementioned "one-against-all" method, where CRP training samples can be selected from the known areas and non-CRP training samples could be selected by manual inspection within the unknown block.

5.4 CRP Mapping Implementation

5.4.1 Sample Selection for Training and Evaluation

Considering the error in the existing CRP reference data provided by NRCS, we develop a specific method to select reliable samples for classifier training and

evaluation. The majority of errors in the present CRP reference data are the mis-location of CRP tracts and/or CRP boundaries. If this mis-location is not significant (usually true in most cases), we may still get reliable training and testing samples by sampling away from CRP boundaries. In other words, all data samples are selected from the center areas of CRP tracts. A more reliable way to get data samples is to perform field study of the CRP tracts in question. Based on reliable training samples, the CRP mapping results can even correct some locality errors and spatial misalignment of CRP tracts in the reference data.

5.4.2 CRP Mapping using DTC

In the *C4.5* DTC, the classification hyperplane consists of a set of local splitting operations without guaranteeing global optimality. Moreover, the DTC training process often faces the overfitting problem, i.e., the learned concept is too specified for the training data, which leads to poor generalization performance. Therefore, some pruning methods have been developed to mitigate the overfitting problem. We use a post-pruning approach suggested in *C4.5* [132], which is also called error-based pruning (EBP) [53]. For this approach, we assume there are N training samples covered by a node and E samples are misclassified. If this node is pruned, the error rate is $R = E/N$. For a given confidence level α, the upper bound of the estimated error for the future test can be computed as $R^{'} = R + U_{\alpha}(E, N)$ with the assumption that errors in the training set are binomially distributed, where $U_{\alpha}(E, N)$ is the confidence limit for the binomial distribution. This method conservatively estimates the misclassification rate when pruned trees are applied to the test data.

The EBP method favors higher P_a and P_b, while decreasing P_c, especially when α is small. In this work, based on the same assumption of the EBP method, we develop a *recall*-based pruning (RBP) approach in favor of higher *recall*. When splitting a node, the data samples in this node are divided into two parts: $\{a^+, a^-\}$ and $\{b^+, b^-\}$, where a^+ is called true positive, a^- is false positive, b^+ is false negative, and b^- is true negative. Then the *recall* of this splitting is defined as:

$$P_c = \frac{a^+}{a^+ + b^+}. \tag{5.9}$$

Contrast to EBP, RBP begins from the parent node of each leaf node in the DTC because P_c is only associated with those nodes that are not leaves, and the error that needs to be reduced is b^+. Therefore, each pruning removes a subtree from the constructed tree. If a subtree is pruned, given confidence level α, the upper bound of b^+ is estimated as $B^+ = b^+ + U_\alpha(b^+, b^+ + b^-)$. Since RBP cannot guarantee small a^-, it should be used in conjunction with EBP, i.e., when deciding whether to prune a node or not, we compare both R' and B^+ calculated at the current node with those nodes of a subtree.

5.4.3 CRP Mapping using SVM

Although the overfitting problem of DTC could be mitigated by pruning, the generalization performance still cannot achieve the optimal solution. Moreover, the curse of dimension could arise if training samples do not significantly outnumber the feature dimension. SVM methods avoid these limitations by optimizing a margin-based criterion, resulting in a better generalization than DTC. However, when dealing with the uneven classification problem, SVM usually leads to good P_b but poor P_c. This usually happens in text classification [21, 113, 143], as well as in CRP mapping [149]. Various relaxation approaches have been developed to address this problem [143]. The principle of these methods is to adjust either or both of the position and orientation of the classification hyperplane to achieve a better performance. We study two relaxation methods in this work. One is the SVM based embedded relaxation (SVM-ER) method that assigns uneven costs to the misclassification of positive and negative samples during the SVM training [155, 120], leading to the change of both position and orientation of the hyperplane. The other is an efficient SVM based post-learning relaxation (SVM-PLR) approach suggested in [143], where an adaptive beta-gamma filtering method [162] is used to adjust the position of the hyperplane.

The SVM^{light} outputs indicate both the distance of each sample to the decision hyperplane and the class type with the appropriate sign. After ranking these distances from the positive to the negative, we can build a utility model U by assigning equal or various weights w_1 and w_2 to the true positive and false positive according to the

class label of training data:

$$U = w_1 a^+ - w_2 a^-, \tag{5.10}$$

where a^+ and a^- are defined in Section 5.4.2. In this work, we set $w_1 = w_2 = 1$. Based on the utility model, we search for the distance threshold that has the maximum U, denoted by θ_{opt}, as well as the threshold of the first zero U, called θ_{zero}. Then the final decision threshold is calculated as:

$$\hat{\theta} = \pi\theta_{zero} + (1-\pi)\theta_{opt}, \tag{5.11}$$

$$\pi = \beta + (1-\beta)e^{-N\gamma}, \tag{5.12}$$

where N denotes the number of positive class training samples, and β and γ determine the extent of threshold relaxation from the threshold's optimal value. β and γ can be determined by cross validation or experience [84]. Furthermore, given the training data, we want to study how SVM and RBF kernel parameters affect P_a, P_b, and P_c via the $\xi\alpha-$estimator suggested in [89]. The $\xi\alpha-$estimator is a highly efficient approximation to the time-consuming Leave-one-out (LOO) estimator proposed in [112]. Given training data, the LOO estimator can provide an nearly unbiased estimation of the true generalization performance, and the $\xi\alpha-$estimator provides lower bounds of P_a, P_b, and P_c, which is more conservative than the LOO estimator.

5.5 Simulation Results

In this section, we investigate the CRP mapping performance in the study area as shown in Fig. 5.1. After removing CRP boundary areas, the remaining 60% of the CRP area is considered as reliable CRP sites, where training and testing samples for the data classifier will be selected and used, respectively. Given a sampling rate x, the equivalent sampling rate (ESR) for CRP areas is computed as $0.6 \cdot x$. For example, if $x = 1/3$, the ESR is about 20%, and if $x = 1/6$, the ESR for CRP is about 10%. The selection of non-CRP training samples is done in the same way. CRP mapping is studied based on the block-based operation, where the block size is around 100×100 pixels.

5.5.1 Simulation of DTC

When the *C4.5* DTC is used for CRP mapping, the confidence level α is set as 0.05. We first compare the DTC that is not pruned, the one pruned using EBP, and the one pruned using RBP. Given a sampling rate, we first select CRP and non-CRP samples for DTC training, and we use the remaining samples for testing [1]. The numerical results via cross validation are listed in Table 5.2 with two different ESRs.

Table 5.2: Classification performance of DTC at two different ESRs (I: not pruned, II: pruned using EBP, III: pruned using RBP.

ESR	20%			10%		
	I	II	III	I	II	III
P_a	97.12	97.56	97.37	96.16	96.85	96.52
P_b	75.72	80.29	77.93	71.29	77.76	74.30
P_c	87.17	86.92	87.68	84.83	83.39	84.67

From Table 5.2, we can see that when $\alpha = 0.05$, EBP results in higher P_b, while P_c is decreased. After using RBP in conjunction with EBP, P_c can be increased with the sacrifice of P_b. This is expected because when we try to increase the *recall* rate (P_c), misclassification of non-CRP samples as CRP samples will occur. The tradeoff between P_b and P_c can be predicted via equation (5.7). For example, in the training and testing data, CRP samples consist of about 8% of all samples. Therefore, at ESR 20%, given $\lambda = 0.08$, $P_a = 97.37\%$, and $P_c = 87.68\%$, we predict that $P_b = 80.84\%$ according to (5.7), which is close to the true value, i.e., 77.93%. Moreover, we can further improve the mapping performance by using Bayesian context fusion or morphological operation to remove the isolated misclassified pixels.

We also study the contributions from different combinations of multisource data to the mapping performance. Given 20% ESR, simulation results are shown in Table 5.3, where the numbers in parentheses are the increases compared with the mapping result using the satellite imagery (layer set A in Fig. 5.2) only. It is shown that all multisource data can improve classification performance in terms of P_a, P_b, and P_c. (1) Vegetation indices (Layer set B) provide helpful information to discriminate healthy green vegetation from dead vegetation, bare soil, and urban

[1] Removed CRP regions are not considered for both training and numerical testing.

Table 5.3: Classification performance of DTC at 20% ESR with different data sets (A-D are defined in Fig. 5.2.).

	P_a	P_b	P_c
A	95.14	62.60	77.63
A+B	95.43(+0.29)	64.28(+1.68)	79.54(+1.91)
A+C	96.12(+0.98)	68.36(+5.76)	84.01(+6.38)
A+D	97.05(+1.91)	75.33(+12.73)	86.52(+8.89)
A+B+C	96.28(+1.14)	69.39(+6.79)	84.40(+6.77)
A+B+D	96.78(+1.64)	73.27(+10.67)	85.75(+8.12)
A+C+D	97.24(+2.10)	76.42(+13.82)	88.19(+10.56)

areas, as well as limited disparity information among different green vegetation. The difficult part of CRP mapping is the discrimination of different vegetation types, and layer set B provides only slight improvements. (2) From LULC GAP data we know that more than half of this region is covered by crops, which usually show relatively smooth texture behavior, while CRP areas are unmanaged areas covered by different grass species that tend to show less smooth texture behavior. The texture smoothness/roughness can be efficiently captured by a window-based local mean and variance (Layer set C), which contribute more to classification accuracy than layer set B. (3) The improvement from GIS data (Layer set D) is most significant when there are only three GIS layers. This indicates that GIS data has certain correlations with CRP tracts with respect to elevation, distance-to-waterbody and slope. This observation is consistent with the CRP enrollment policy of FSA, justifying the usefulness of multisource GIS data for CRP mapping.

5.5.2 Simulation of SVM

From the cross validation, it was found that SVM performs well when C is between 10 to 1000, while σ significantly affects *precision* (P_b) and *recall* (P_c). We need to estimate an appropriate σ value that leads to high P_c with acceptable P_b. Therefore, given the training data, the $\xi\alpha-$estimator can be used to select a proper σ by plotting P_a, P_b, and P_c against σ in a certain range, as shown in Fig. 5.5. As we can see, P_a varies slightly. P_b and P_c vary in opposite directions when σ is small, which verifies the existence of a tradeoff between them if P_a remains approximately constant.

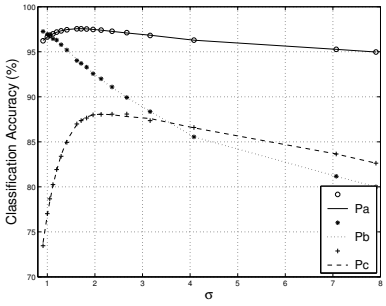


Figure 5.5: SVM classification performance vs. σ at sampling rate 0.2.

Then both P_b and P_c decrease after $\sigma = 2.13$ where $P_b = 92\%$ and $P_c = 88.05\%$. At this point, P_c achieves its highest lower bound. Considering the importance of high P_c, we set $\sigma = 2.13$ in this work. Specifically, we use two relaxation methods introduced in Section 5.4.3 to increase P_c. Simulation results using cross validation are listed in Table 5.4 at two different ESRs.

Table 5.4: Classification performance of SVM at different ESRs (I: No relaxation, II: SVM-ER, III: SVM-PLR)

ESR	20%			10%		
	I	II	III	I	II	III
P_a	99.26	99.26	98.47	98.72	98.72	95.29
P_b	94.58	94.59	83.97	91.79	91.84	62.74
P_c	94.94	94.93	96.89	91.76	91.77	97.61

It is shown above that both P_b and P_c are more than 90% without the relaxation. At 10% ESR, the mapping results of four clips in the study area are illustrated in Fig. 5.6. Fig. 5.6 (a) shows the original CRP tracts in the reference data, and Fig. 5.6 (b) depicts the mapping results using SVM where all data samples in a block are classified. Since the original P_b and P_c are quite high, significant improvement of them could be very difficult. When implementing SVM-ER, we first use the $\xi\alpha -$ estimator to determine a proper relative weight (RW) of CRP and non-CRP samples in the cost function based on the training data, so that P_c could be maximized. We found that RW=0.5 is a preferred value. However, as shown in Table 5.4, SVM-ER can slightly improve P_b and/or P_c. SVM-PLR can increase P_c considerably, but P_b usually suffers. As mentioned before, P_b can also be estimated by equation

(5.7). For instance, at 20% ESR, when $P_a = 98.47\%$ and $P_c = 96.89\%$, we have $P_b = 85.8\%$ near to the true value, i.e., 83.97%.

The contributions from different combinations of multisource data are also studied and listed in Table 5.5 at 20% ESR. The simulation results in Tables 5.3 and 5.5 demonstrate that the *C4.5* and SVM are consistent regarding the feature contribution, where texture information and GIS data are the most important features used to improve CRP mapping accuracy. It is also shown that SVM works better than DTC under the same sampling rate. This demonstrates that SVM has better generalization performance than DTC.

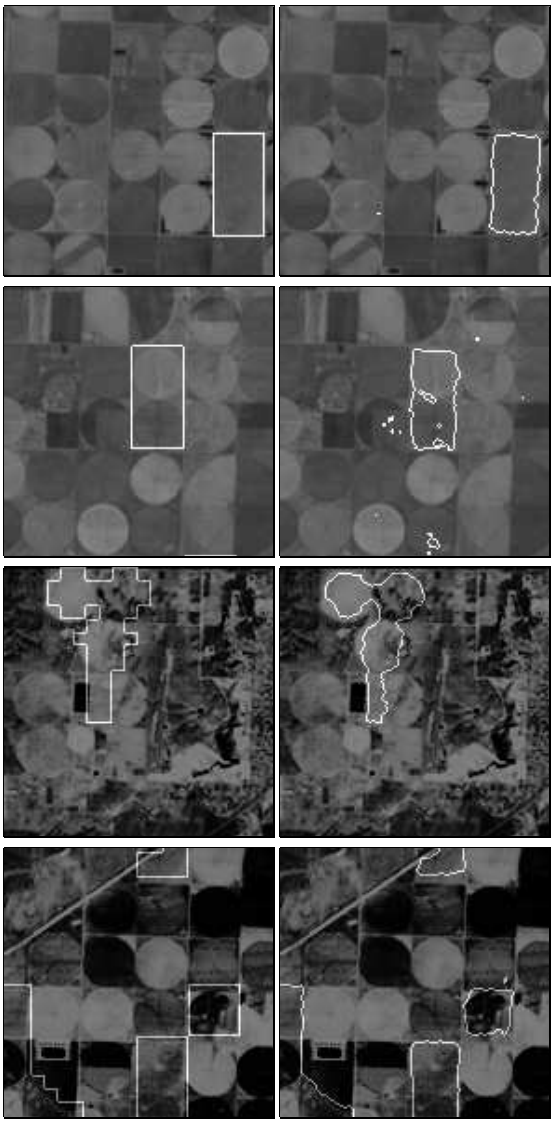
Table 5.5: Classification performance of DTC at 20% ESR with different data sets (A-D are defined in Fig. 5.2.).

	P_a	P_b	P_c
A	96.47	73.91	81.76
A+B	97.64(+1.17)	79.63(+5.72)	89.48(+7.72)
A+C	98.26(+1.79)	84.60(+10.69)	92.09(+10.33)
A+D	98.68(+2.21)	87.57(+13.66)	94.81(+13.05)
A+B+C	98.55(+2.08)	87.61(+13.70)	92.51(+10.75)
A+B+D	98.92(+2.45)	90.09(+16.18)	95.12(+13.36)
A+C+D	99.18(+2.71)	93.41(+19.50)	95.13(+13.37)

We also study the prediction error of SVM via LOO and $\xi\alpha-$estimators. At 10% ESR, different training and testing samples are selected to estimate the mean and standard deviation of the prediction error. Simulation results are shown as error bars in Fig. 5.7. In the study area, there are 19 out of 25 blocks that have significant CRP tracts. The dashed and dotted lines indicate LOO and $\xi\alpha$ estimations, respectively. Since $\xi\alpha-$estimator provides lower bounds of the estimation, the prediction is more conservative but more efficient than LOO estimation. Both estimators can be used to predict the CRP mapping performance. Furthermore, the estimators could also be used to measure the effectiveness of given training samples. If predicted errors are significant, we might want to select more representative training samples or add more training samples.

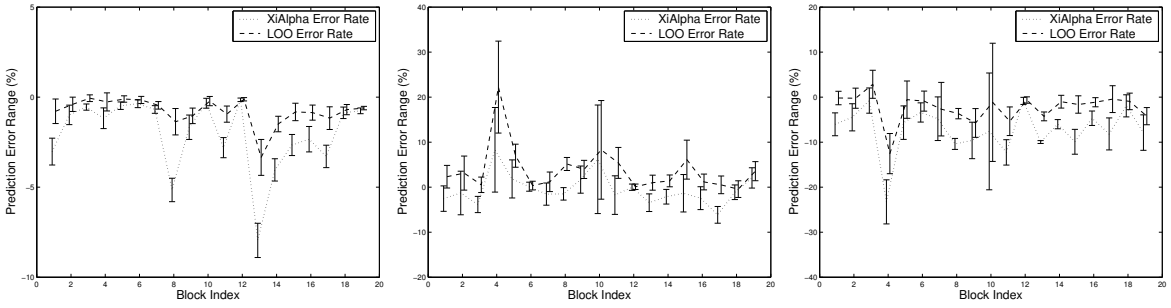
5.6 Summary

We have studied the application of DTC and SVM for automatic CRP mapping, which is a classification problem of complex rural areas. Particularly, a parallel localized classification framework is suggested and validated based on a study area. Considering the importance of classification sensitivity, a new DTC pruning method is proposed to enhance the *recall* rate. Two relaxation methods are also studied for SVM to improve *recall*. Simulation results indicates that SVM-ER cannot improve *recall* significantly, while SVM-PLR can enhance *recall* with acceptable *precision* if we properly choose the relaxation parameters. In addition, the individual contribution of multisource geospatial data is manifested by its improvements on CRP mapping accuracy. Overall, SVM shows a better generalization performance than DTC in this work. Our future research will focus on CRP compliance monitoring based on the proposed CRP mapping approaches.



(a) (b)

Figure 5.6: CRP mapping results (145 × 145 pixels): (a) Original CRP reference data, (b) Mapping results.



(a) (b) (c)

Figure 5.7: Predication errors of LOO and $\xi\alpha -$ estimators. (a) Classification accuracy (P_a). (b) *Precision* (P_b) (c) *Recall* (P_c).

Chapter 6

NONPARAMETRIC UNSUPERVISED SEGMENTATION OF SATELLITE IMAGERY

As discussed in the previous chapter, nonparametric machine learning approach such as support vector machine (SVM), referred to as the two-class SVM (TCSVM) in this chapter [155, 15, 35, 20], has shown superior performance in the classification of remotely sensed data [73, 72, 80, 118, 16]. SVM searches a linear separation plane that maximizes the distance between two patterns in a feature space, and a good generalization performance can be obtained via a tradeoff between the training error and the capacity of a chosen classification function. In addition, the efficient algorithm implementation makes SVM practical in many applications. Recently, a one-class SVM (OCSVM) algorithm was proposed for outlier or novelty detection [140, 153]. OCSVM is an unsupervised approach that separates outliers from the majority. It was shown that OCSVM can produce comparable or superior classification results over traditional unsupervised classification methods for novelty detection in [153]. There is a parameter ν that usually is unknown and significantly affects the OCSVM results. A heuristic method was suggested in [134] that is effective if the majority and outliers are clearly separable.

In this chapter, we will develop a SVM-based method for automatic compliance monitoring of United States Department of Agriculture (USDA)'s Conservation Reserve Program (CRP) based on multispectral Landsat imagery. The CRP is a long-term program that aims to improve soil, water and wildlife resources by encouraging farmers to plant native plant species (mostly grasses) on agricultural land for 10-15 years [2]. In return annual rental payments are made to the farmers by USDA ($1.6 billion in 2002). However, USDA is facing the problem that farmers

are not maintaining CRP tracts according to contract stipulations. Current methods for CRP compliance monitoring involve intensive manual inspection of aerial photographs, which is time-consuming and costly. USDA's Common Land Unit (CLU) data used for general compliance issues is generated from aerial photographs, which are updated every 1-2 years and may not be very timely for CRP compliance monitoring on a large scale [79]. In addition, most existing CRP reference data obtained from USDA's Natural Resource Conservation Service (NRCS) are not very accurate or up-to-date for management purposes. There is a need for an automatic compliance monitoring method that can examine CRP tracts on a large scale more efficiently and promptly with minimum human involvement.

In [29], we have applied both the OCSVM and TCSVM to CRP compliance monitoring that is formulated as an unsupervised classification problem, where more than half of a CRP tract under test is assumed to be compliant, and CRP reference data were used as prior knowledge to locate CRP tracts for testing. The OCSVM is first applied to obtain initial classification results where the majority and outliers can be separated. Then TCSVM training samples are selected with a certain spatial constraint. In the OCSVM, ν is estimated using the method suggested in [134] that estimates optimal ν by computing a distance measure based on many candidate ν values. This may not be efficient when handling large scale remotely sensed data, and it may fail when two clusters are not clearly separable. In this work, we suggest a ν-insensitive [1] approach where a mild deviation from true ν, which is unknown, will not significantly affect the classification performance. ν-insensitivity is achieved by carefully selecting sufficient and reliable TCSVM training samples according to their SVM scores obtained from the OCSVM. Compared with [134], this method reduces the computational load by avoiding ν estimation, and also improves the classification performance. Similar to [29], we use CRP reference data to locate CRP tracts and to evaluate the proposed method. By comparing the classification results with the CRP reference data, the compliance issue can be addressed.

[1] In this work, "*ν-insensitive*" means "*much less sensitive*" to the variation of ν compared with conventional ν-SVM approaches.

6.1 One-class Support Vector Machine (OCSVM)

The OCSVM is an extension of the general TCSVM to the unsupervised classification case [140, 153]. This method aims at providing an approximation function to categorize the majority of data. Basically, the OCSVM tries to find the region in the feature space where the data resides. Two different OCSVM approaches have been proposed. One is Support Vector Data Description method that constructs a spherical boundary to contain as much as possible of data in the feature space while minimizes the volume of the sphere [153]. Those lying outside the sphere are classified as outliers. The other is ν-SVM that computes a hyperplane in the feature space to separate a pre-specified fraction $(1-\nu)$ of data with the maximum distance to the origin (margin) $\frac{\rho}{||\mathbf{w}||}$ [140]. Parameter $\nu \in (0, 1]$ is an upper bound on the fraction of margin errors, and a lower bound on the number of support vectors. The classification hyperplane is constructed by solving:

$$\min_{\mathbf{w}\in F, \xi\in\mathbf{R}^m, \rho\in\mathbf{R}} \frac{1}{2}||\mathbf{w}||^2 - \nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i, \tag{6.1}$$

subject to $y_i(\mathbf{x}_i \cdot \mathbf{w}) \geq \rho - \xi_i,\ i = 1, 2, \ldots, N$, where F indicates the feature space. Both methods are shown to be equivalent when using the RBF kernel in [140, 153]. It is also shown in [153] that both methods operate comparably in practice and perform best when the RBF kernel is used . A connection between OCSVM and TCSVM can be described as follows: if the OCSVM has $\rho > 0$, it is equivalent to a TCSVM with C set a priori to $1/\rho$ [27]. Since ρ shows the threshold to the origin, a large ρ means a better separation, which imply a smaller C in the TCSVM.

6.2 ν-insensitive SVM Classification

6.2.1 Estimating ν for OCSVM

Given a CRP clip of Landsat imagery, we assume the majority (more than half) is compliant. In the OCSVM, we need to set ν . It is ideal to chose the percentage of non-CRP outliers, which is unknown and assumed to be ≤ 0.5. The method proposed in [134] tries out different ν values based on given training data, and the value that

results in the largest separation distance two classes is selected as the optimal one. The separation distance between two clusters is computed as:

$$D_\nu = \frac{1}{N_+} \sum_{f_{\mathbf{w}}(\mathbf{x}) \geq \rho} f_{\mathbf{w}}(\mathbf{x}) - \frac{1}{N_-} \sum_{f_{\mathbf{w}}(\mathbf{x}) < \rho} f_{\mathbf{w}}(\mathbf{x}), \tag{6.2}$$

where N_+ and N_- are the sizes of the majority and outlier classes, respectively, and $f_{\mathbf{w}}(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{w})$. It can be seen that D_ν provides an average estimation of separability between two classes in the feature space, and optimal $\hat{\nu}$ is estimated as:

$$\hat{\nu} = \arg\max_{\nu} D_\nu. \tag{6.3}$$

An accurate estimation of $\hat{\nu}$ requires many tests under different candidate ν values. This is not efficient when we are dealing with very large data sets. In addition, the method suggested in [134] provides accurate estimation only when the majority and outlier are clearly separated in the feature space, which is not always true between CRP and non-CRP regions [149].

6.2.2 Study of Feature Space

The OCSVM classifies a sample $\mathbf{x}$ according its SVM score defined by $s_{\mathbf{w}}(\mathbf{x}) = f_{\mathbf{w}}(\mathbf{x}) - \rho$. This score shows the distance of $\mathbf{x}$ to the constructed hyperplane, and its sign indicates if $\mathbf{x}$ is classified as the majority (positive) or the outlier (negative). A large score magnitude implies that the sample is more likely to be correctly classified. Since ν is the upper bound of the amount of outliers, changing ν actually changes the position and orientation of the classification hyperplane in the feature space. An improper ν would cause some outliers to be mis-classified as the majority class, or vice versa. These samples, which are prone to be misclassified, are usually located around the optimal hyperplane associated with the true ν, i.e., ν^*.

A graphical illustration is shown in Fig. 6.1, where *stars* (outlier) and *squares* (majority) represent two classes that are linearly nonseparable in a 2-D feature space, and the classification hyperplane changes within region C with respect to different ν values. In region C, the hyperplanes I and III are associated with the smallest and largest possible ν values, e.g, ν_{min} and ν_{max}, respectively, and the hyperplane II is associated with true ν^*. The method using equation (6.2) may not be accurate

because there are always some misclassified samples involved in the computation due to the linear non-separability. On the other hand, region A includes outlier samples with large negative SVM scores, and region B contains majority samples with large positive SVM scores. The samples in regions A and B are more probable to be correctly classified when $\nu \in [\nu_{min}, \nu_{max}]$. Thus if we use samples in regions A and B as outlier and majority training samples for TCSVM, the classification results that are insensitive to the variations of ν values could be obtained.

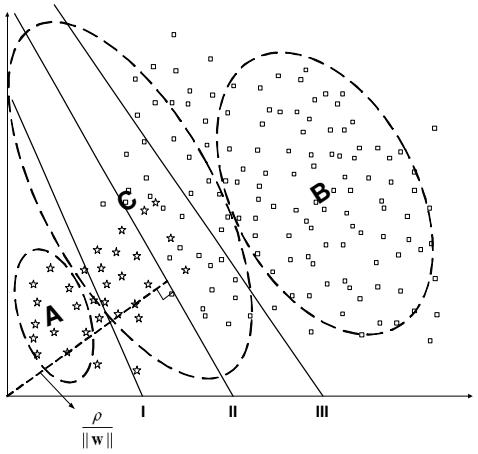
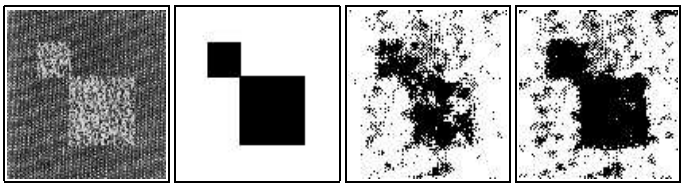


Figure 6.1: SVM hyperplanes with respect to different ν values in the feature space. Hyperplanes I, II, and III are associated with the smallest ν value, the true ν value, and the largest ν value, respectively. The distance from the origin to the decision hyperplane is given by $\frac{\rho}{\|w\|}$ when solving equation (6.1).

6.2.3 Proposed ν-insensitive Approach

In this work, we propose a ν-insensitive method for reliable TCSVM training. Given a test CRP tract $\mathbf{X}$ of N samples, we assume that the majority of $\mathbf{X}$ is compliant, i.e., $\nu^* < 0.5$. After the OCSVM classification, we sort all data samples in the majority and outlier classes according to their SVM score magnitudes, i.e., $|s_{\mathbf{w}}(\mathbf{x})|$, from the largest to the smallest. $\mathbf{X}_M = \{\mathbf{x}_i^{(m)}, i = 1, \cdots, N_+\}$ and $\mathbf{X}_O = \{\mathbf{x}_j^{(o)}, j = 1, \cdots, N_-\}$ denote the sorted majority and outlier data sets, respectively, where $N = N_+ + N_-$, $|s_{\mathbf{w}}(\mathbf{x}_1^{(m)})| \geq |s_{\mathbf{w}}(\mathbf{x}_2^{(m)})| \geq \cdots \geq |s_{\mathbf{w}}(\mathbf{x}_{N_+}^{(m)})|$, and $|s_{\mathbf{w}}(\mathbf{x}_1^{(o)})| \geq |s_{\mathbf{w}}(\mathbf{x}_2^{(o)})| \geq \cdots \geq |s_{\mathbf{w}}(\mathbf{x}_{N_-}^{(o)})|$. We define $\mathbf{X}_M^t$ and $\mathbf{X}_O^t$ as the majority and outlier training sets for TCSVM, which can be constructed as follows:

$$\mathbf{X}_M^t = \{\mathbf{x}_i^{(m)} | i = 1, \cdots, 0.45N\},$$



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

Figure 6.2: Experimental demonstration of the proposed ν-insensitive method based on a synthetic mosaic. (a) Mosaic. (b) Ground data (25% outlier). (c) OCSVM ($\nu = 0.25$, 85.18%). (d) The proposed method ($\nu = 0.5$, 84.32%).

$$\mathbf{X}_O^t = \{\mathbf{x}_j^{(o)} | j = 1, \cdots, (1-\nu)N_-\}. \tag{6.4}$$

On the one hand, since $\nu^* < 0.5$, we might use at least $0.5N$ samples in $\mathbf{X}_M$ with the largest positive SVM scores as majority training samples (e.g., region B in Fig. 6.1). Conservatively, we choose $0.45N$ to avoid selecting samples near the hyperplane. On the other hand, the number of outlier training samples (e.g., region A in Fig. 6.1) is set to be $(1-\nu)N_-$. If we choose small ν, small N_- results. Then most samples in $\mathbf{X}_O$ could be true outliers, and we can use most of them for TCSVM training. On the contrary, if we choose large ν, large N_- results. $\mathbf{X}_O$ may mistakenly contain some majority samples, and we use a small portion of samples in $\mathbf{X}_O$ with the largest negative SVM scores. In practice, $\mathbf{X}_M^t$ and $\mathbf{X}_O^t$ may still have some mis-classified training samples. To further reduce the side-effect of mis-classified data samples, a large margin size is preferred in the TCSVM, which requires small C value in equation (5.4). Furthermore, the OCSVM usually suffers from the problem of having many support vectors and bounded support vectors around the hyperplane, the TCSVM could introduce a more natural decision hyperplane with a relaxed placement of less support vectors, leading to the better generalization performance than the OCSVM alone.

6.2.4 Experimental Demonstration

Here, a synthetic mosaic and its ground data (Fig. 6.2) are used to examine the proposed method. Specifically, the autoregressive features are extracted from

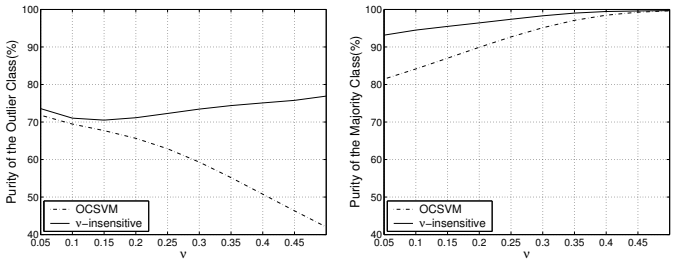


Figure 6.3: Simulation results on the synthetic mosaic. (Left) Purity of outlier training samples vs. ν. (Right) Purity of majority training samples vs. ν.

.

texture pixels within a 7×7 window, resulting in a 25-dimension feature space [96]. The OCSVM is first tested with $\nu \in [0.05, 0.5]$, and RBF kernel is used with $\gamma_1 = 10^{-6}$ determined via cross validation. Small γ_1 indicates a large kernel width that is necessary for this majarity/outlier two-class problem [92]. If we define the *purity* of the training sample as:

$$\text{purity} = \frac{\text{true majority (or outlier) samples}}{\text{detected majority (or outlier) samples}}, \tag{6.5}$$

then based on the OCSVM results, the purity of the outlier and majority classes regarding different ν values are shown in Fig. 6.3 (a) and (b). It is seen that when ν changes from 0.05 to 0.5, the purity of both classes vary considerably.

Our previous work in [29], referred to as Method-I, suggested a simple method to select TCSVM training samples by examining the class homogeneity in a 5×5 window. Although Method-I can improve the purity of training samples, it could be too conservative to select enough training samples. The proposed ν-insensitive method, referred to as Method-II, can select sufficient and reliable training samples with higher purity, as shown in Fig. 6.3 (a) and (b). This leads to ν-insensitive classification results. The highest OCSVM classification accuracy (85.18%) is obtained when $\nu = 0.25$, as shown in Fig. 6.2 (c). When testing Method-II, RBF kernel is also used for TCSVM with $\gamma_2 = 10^{-5}$. Even when $\nu = 0.5$, which deviates from true ν^* significantly, we still obtain the similar accuracy (84.32%) as the OCSVM that requires many attempts, as shown in Fig. 6.2 (c) and (d).

6.3 Experiments and Discussions

6.3.1 Study Area and Experiment Setup

The study area is located in Texas County, Oklahoma, which has the largest CRP enrollments in Oklahoma. Landsat TM multispectral image bands 2, 3, 4, 5, 7 obtained in Spring (February) and Summer (June) of year 2000 are used, based on which an original multi-layer database is constructed. This database also contains the local mean and variance, which are calculated within a 3×3 window of each spectral band, and vegetation indices, which include TM4/TM3, TM5/TM2, TM5/TM4 in each season, and *Normalized Vegetation Difference Index* (NDVI). TM4/TM3 (Ratio vegetation index) and TM5/TM2 are helpful to discriminate different vegetation [107]. TM5/TM4 (Ratio drought index) provides the information of plant water content [33], which is useful to discriminate irrigated crops from relatively dry CRP grasses. The NDVI is calculated from the imagery in each season and the largest one is chosen as the final value. The data in each layer is normalized to zero mean and unit standard deviation. A heuristic method is used to select a feature subset based on the original database and the CRP reference data. This method measures the contribution of an individual feature layer by approximately estimating its effect to the construction of the hyperplane [54]. After the feature selection, we remove the band 2 image and TM5/TM2 in February, resulting 35 feature layers. A software *LIBSVM* [24] is used to implement OCSVM and TCSVM. In the OCSVM, the RBF kernel with $\gamma = 10^{-6}$ is chosen according to cross validation. 10 different ν values are tested, which are from 0.05 to 0.5 at interval 0.05. In the TCSVM, we select $C = 0.5$ and a RBF kernel with $\gamma = 0.01$.

6.3.2 Simulation Results

Simulations are performed on six CRP tracts extracted from Texas County. In each CRP tract, we also deliberately add some non-CRP regions near to CRP boundaries to test the performances of the proposed methods. Method-I needs 10 times (regarding 10 ν values) of OCSVM training and once TCSVM training, while

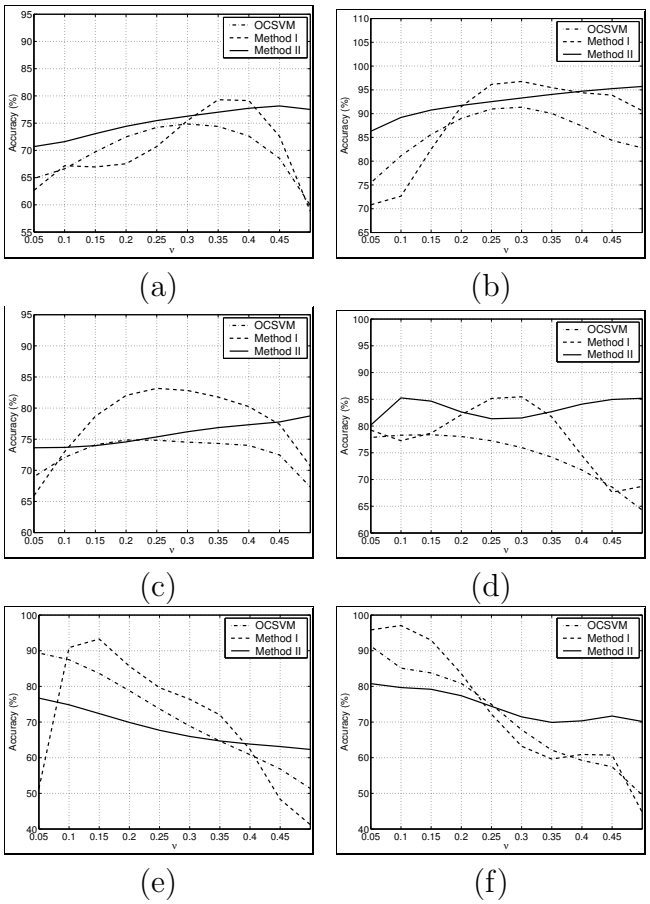


Figure 6.4: The plots of classification accuracy v.s ν for three methods in six tracts: (a) tract 1, (b) tract 2, (c) tract 3, (d) tract 4, (e) tract 5, (f) tract 6.

Method-II trains both OCSVM and TCSVM only once, saving more than 80% computational load. The classification accuracies with respect to different ν values are shown in Fig. 6.4, and the standard deviations (StDev) is computed for each method. Table 6.1 compares the StDev of the classification accuracy for six CRP tracts. As we can see, the performances of both OCSVM and Method-I vary significantly as ν changes, while Method-II is much less sensitive.

Table 6.1: Standard deviations of the classification accuracy.

CRP Tract Index	1	2	3	4	5	6
OCSVM	4.92	5.14	2.64	4.85	13.24	14.03
Method-I	6.88	9.97	5.98	6.84	18.84	18.24
Method-II	2.66	2.96	1.88	1.89	5.16	4.40

We also illustrate the CRP classification results in Fig. 6.5, where five rows

refer to, respectively, 3-band Landsat images, the CRP reference data , the OCSVM classification results, the results of Method-I where $\hat{\nu}$ is estimated from equation (6.2), and the results of Method-II where $\nu = 0.4$. Moreover, the percentage of non-CRP areas according to the CRP reference data (P_{nc}), the percentage of non-CRP areas detected by Method-II (P^*_{nc}), as well as the their differences ($P^*_{nc} - P_{nc}$) are computed for each CRP tract and listed in Table 6.2.

Table 6.2: Non-CRP percentages (%) comparison.

CRP Tract Index	1	2	3	4	5	6
P_{nc}	37.24	30.53	33.74	21.28	9.26	3.7
P^*_{nc}	28.27	27.25	27.98	28.8	34.85	29.47
P^*_{nc}-P_{nc}	-8.97	-3.28	-5.76	+7.52	+25.59	+25.77

In tracts 1, 2, 3, and 4, P^*_{nc} is relatively consistent with or even lower than P_{nc}. Manual inspection further manifests that the CRP areas in tracts 1, 2, 3, 4 have good compliance with respect to the CRP reference data. However, the non-CRP areas in tracts 5 and 6 are found to be significant. This implies that there could be the compliance issue in tracts 5 and 6. As observed from the 3-band Landsat images in Fig. 6.5, there exist some active cultivation areas (darker areas) in those two tracts, which were previously registered as CRP in the reference data. Therefore tracts 5 and 6 need further detailed inspection. Moreover, there are also some man-made buildings in tracts 1, 3, 4, which can be clearly detected by Method I and Method-II as well. Nevertheless, only non-CRP percentage values may not provide sufficient information for compliance monitoring, and additional analysis of the CRP classification maps (the last row of Fig. 6.5) may be necessary.

From the last row of Fig. 6.5, it is interesting to find that Method-II produces better boundary localization around CRP and non-CRP regions than the OCSVM and Method-I. We also found some limitations of our previously proposed Method-I. Largest D_ν is not necessarily related to true ν^*. This fact indicates that CRP and non-CRP are not clearly separated even in the high dimensional feature space mapped via the RBF kernel. For example, in tract 2, D_ν has the largest value when $\hat{\nu} = 0.4$, while the highest OCSVM classification accuracy is obtained when $\nu = 0.25$ which is close to true ν^*.

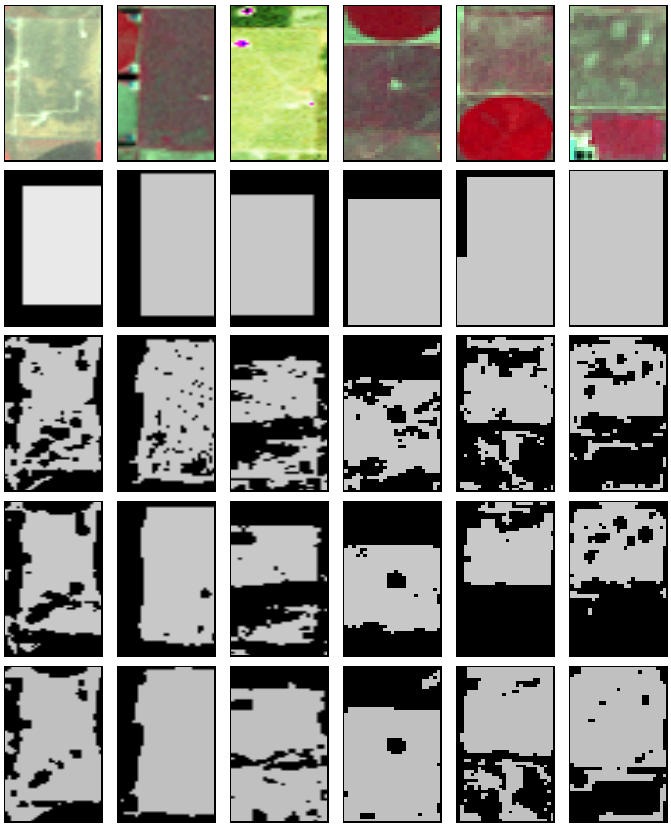


Figure 6.5: Simulation results of the six tracts. The five rows refer to the 3-band Landsat images (June, 2000), CRP reference data (gray: CRP, black: non-CRP), OCSVM results, Method-I results, and Method-II results, respectively.

6.4 Summary

We have developed a ν-insensitive SVM-based method for CRP compliance monitoring. Both OCSVM and TCSVM are used together to accomplish unsupervised CRP classification. Specifically, the proposed method can reduce the side-effect of improper ν setting of OCSVM by selecting TCSVM training samples according to their SVM scores. The percentage of non-CRP/outlier areas could imply whether a given CRP tract is fully compliant, and the classification map can be used to further reveal the detailed information. The proposed method provides a useful guidance for effective and efficient CRP compliance monitoring. One limitation is that we assume the majority of a CRP tract is complaint which is not necessarily true. We are studying the multi-class implementation of one-class SVM where no assumption is made about the dominant class in each CRP tract.

Chapter 7

CONCLUSIONS AND FUTURE WORKS

In this report, we have studied feature selection and extraction approaches for visual data segmentation. In particular, key-frame extraction in video segmentation, WDHMM likelihood computation, decision tree training, and support vector learning are specific approaches of feature selection and/or extraction for segmentation purpose. Both nonparametric and parametric methods are investigated and improved in terms of segmentation performance and computational efficiency. Several new methods are developed that can further inspire our studies towards the real applications. In these applications, we are able to obtain state-of-the-art or promising results as well as efficient algorithms. We conclude this report as follows:

- We propose a novel framework to coherent extract video key-frames and segment objects in a unified spatio-temporal feature space, where key-frame extraction is formulated as a feature selection process. Based on cluster divergence and maximum likelihood -based criteria, two numerical methods and one analytical method are developed to extract key-frames for object segmentation. All methods show impressive performance on both synthetic and real video sequences. The proposed framework explicitly reveals the inherent relationship between key-frames and objects, facilitating content-based video analysis by providing robust and accurate object segmentation results, as well as compact and semantically meaningful key-frame representations of video shots.

- We develop a new approach for unsupervised Bayesian image segmentation. Instead of segmenting an image by estimating a model to fit the image data as much as possible, we suggest to partition the image by exploiting the disparity

of fitness with respect to one global model, which might not be necessary to fit the image data very well. Wavelet-domain hidden Markov models are used here to characterize the image, and WDHMM model likelihood is the key feature for segmentation. A dual-model framework with a new hybrid *soft-hard* decision approach is developed to make WDHMMs applicable to the unsupervised case. Two new clustering approach are suggested to capture the fitness disparity efficiently. The simulations on synthetic mosaics and real images show very good performance of the proposed method.

- We show that features extracted or selected from multispectral Landsat images and GIS data are helpful for complex land cover classification problems using nonparametric decision tree classifier (DTC) and support vector machines (SVM). For mapping USDA's CRP tracts, a new DTC pruning method and two SVM post relaxation methods are studied for increasing the system sensitivity (recall rate) by selecting or extracting proper features. For CRP compliance monitoring problem, we propose a novel method to avoid the estimation of a key OCSVM parameter, i.e., ν, which is usually computationally expensive and complicated, by selecting representative samples in the projected feature space to train a TCSVM. This makes the method practical because ν is usually unknown in many real applications. Simulations indicate the effectiveness and good performance of the suggested approaches.

The perspectives of future work could be highly correlated to the current research. For example, we want to use motion vector rather than pixel-wise frame difference as the motion feature in the unified feature space, consequently, we expect to obtain key-frames with more specific object motion information. Moreover, after building the coherent framework for video segmentation, how to integrate it into a high level video analysis platform? Any progresses of these work could lead to more powerful and efficient tools for content-based video analysis.

BIBLIOGRAPHY

[1] United States Department of Agriculture (USDA), Farm Service Agency, http://www.fsa.usda.gov/dafp/cepd/crp.htm.

[2] *Conservation Reserve Program*. http://www.fsa.usda.gov/dafp/cepd/crp.htm.

[3] *Signal and Image Processing Institute*. Image Database, http://sipi.usc.edu/services/ database/Database.html.

[4] S. A. Alvarez. An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Technical Report BCCS-02-01, Computer Science Department, Boston College*, June 2002.

[5] P. Andrey and P. Tarroux. Unsupervised segmentation of markov random field modeled textured images using selectionist relaxation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 20(3):252–262, March 1998.

[6] G. Antelman. *Elementary Bayesian Statistics*. Edward Elgar, Lyme, 1997.

[7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Processing*, 1(2):205–220, April 1992.

[8] C. Arndt. *Information Measures: Information and its Description in Science and Engineering*. Springer, 2001.

[9] A. S. Belward. A comparison of supervised maximum likelihood and decision tree classification for crop cover estimation from multitemporal Landsat MSS data. *Int. J. Remote Sensing*, 8(2):229–235, 1987.

[10] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy. Conjugate-gradient neural networks in classification of multisource and very-high-dimensional data. *Int. J. Remote Sensing*, 14:2883–2903, 1993.

[11] J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Statist. Soc. B*, 48(3):259–302, 1986.

[12] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: A case study in handwriting digit recognition. In *Proc. Int. Conf. Pattern Recognition.*, pages 77–87, 1994.

[13] C. A. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Image Processing*, 3(2):162–177, March 1994.

[14] G. B. Briem, J. A. Benediktsson, and J. R. Sveinsson. Multiple classifiers applied to multisource remote sensing data. *IEEE Trans. Geoscience and Remote Sensing*, 40(10):2291–2299, Oct. 2002.

[15] B. E. Broser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *in Proc. Fifth Annual Workshop on Computational Learning Theory, ACM*, June 1992.

[16] M. Brown, H. G. Lewis, and S. R. Gunn. Linear spectral mixture models and support vector machines for remote sensing. *IEEE Trans. Geoscience and Remote Sensing*, 38(5):2346–2360, Sept. 2000.

[17] L. Bruzzone, C. Conese, F. Maselli, and F. Roli. Multisource classification of complex rural areas by statistical and neural-network approaches. *Photogrammetric Engineering and Remote Sensing*, (5):523–533, 1997.

[18] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans. Image Processing*, 8(12):1688–1701, Dec. 1999.

[19] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

[20] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, Jun. 1998.

[21] N. Cancedda, N. C. Bianchi, A. Conconi, and C. Gentile. Kernel methods for document filtering. In *in Proc. Eleventh Text Retrieval Conference*, 2003.

[22] G. Celeux, S. Chrtien, F. Forbes, and A. Mkhadri. A component-wise em algorithm for mixtures. *Journal of Computational and Graphical Statistics*, 10:699–712, 2001.

[23] United States Agriculture Census, 1997. Census of Agriculture, USDA, http://www.fsa.usda.gov/dafp/cepd/crp.htm.

[24] Chih-Chung Chang and Chih-Jen Lin. *Libsvm: a library for support vector machines*. 2003. Software available at http: //www.csie.ntu.edu.tw/∼cjlin/libsvm.

[25] S. G. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Trans. Image Processing*, 9(9):1522–1531, Sept. 2000.

[26] P.S. Jr. Chavez. Image-based atmospheric correctionsrevisited and revised. *Photogrammetric Engineering and Remote Sensing*, (9):1025–1036, 1996.

[27] P. H. Chen, C. J. Lin, and B. Schlkopf. A tutorial on nu-support vector machines. *Applied Stochastic Models in Business and Industry*, to appear, 2005.

[28] H. Cheng and C. A. Bouman. Multiscale Bayesian segmentation using a trainable context model. *IEEE Trans. Image Processing*, 10(4):511–525, April 2001.

[29] G. Cherian, X. Song, G. Fan, and M. Rao. Application of support vector machines for automatic compliance monitoring of the conservation reserve program (CRP) tracts. In *Proc. IEEE Int. Geoscience and Remote Sensing Symposium*, Anchorage, Alaska, Sept. 2004.

[30] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley and Sons, 1998.

[31] H. Chipman, E. Kolaczyk, and R. McCulloch. Adaptive Bayesian wavelet shrinkage. *J. Ameri. Stat. Assoc.*, 440(92):1413–1421, Dec. 1997.

[32] H. Choi and R. Baraniuk. Multiscale image segmentation using wavelet-domain hidden markov models. *IEEE Trans. Image Processing*, 10(9):1309–1321, 2001.

[33] E. Chuvieco, D. Riano, I. Aguado, and D. Cocero. Estimation of fuel moisture content from multitemporal analysis of landsat thematic mapper reflectance data: Applications in fire danger assessment. *Int. J. Remote Sensing*, (11):2145–2162, 2002.

[34] P. Lobato Correia and F. Pereira. Objective evaluation of video segmentation quality. *IEEE Trans. Image Processing*, 12(2):186–200, 2003.

[35] C. Cortes and V. N. Vapnik. Support vector network. *Machine Learning*, pages 1–25, 1995.

[36] M. S. Crouse and R. G. Baraniuk. Contextual hidden Markov models for wavelet-domain signal processing. In *Proc. 31th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1997.

[37] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Signal Processing*, 46(4):886–902, April 1998.

[38] P. Dasgupta, P. P. Chakrakbarti, and S. C. DeSarkar. *Multiobjective Heuristic Search*. Vieweg, 1999.

[39] G. Davenport, T. A. Smith, and N. Pincever. Cinematic primitives for multimedia. *IEEE Computer Graphics and Applications*, 11(4):67–74, July 1991.

[40] H. P. Decell and J. A. Quirein. An iterative approach to the feature selection problem. In *Proc. of Purdue Univ. Conf. on Machine Processing of Remotely Sensed Data*, volume 1, pages 3B1–3B12, 1972.

[41] R. Defries, M. Hansen, J. Townshend, and R. Sohlberg. Global land cover classification at 8km spatial resolution: the use of data derived from Landsat imagery in decision tree classifiers. *Int. J. Remote Sensing*, 19(16):3141–3168, 1998.

[42] D. DeMenthon and R. Megret. Spatio-temporal segmentation of video by hierarchical mean shift analysis. *Technical Report: LAMP-TR-090/CAR-TR-978/CS-TR-4388/UMIACS-TR-2002-68*, 2002.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

[44] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9(1):39–55, January 1987.

[45] M. N. Do and M. Vetterli. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden markov models. *IEEE Trans. Multimedia*, 4(4):517–527, Dec. 2002.

[46] M. N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE Trans. Image Processing*, 11(2):146–158, February 2002.

[47] F. Dufaux. Key frame selection to represent a video. In *Proc. of ICME2000*, 2000.

[48] M. M. Dundar and D. Landgrebe. A model-based mixture-supervised classification approach in hyperspectral data analysis. *IEEE Trans. Geoscience and Remote Sensing*, 40(12):2692–2699, Dec. 2002.

[49] Sian Eagles, Dietmar Mller, Michael Hughes, and Peter Hogarth. Automated classification of shallow water seafloor backscatter images. In *Proc. of 3rd Int'l Conf. High Resolution Surveys in Shallow Water*, Sydney, Nov. 2003.

[50] A. W. F. Edwards. *Likelihood*. Cambridge: University Press, 1972.

[51] S. L. Egbert, R. Y. Lee, K. P. Price, and R. Boyce. Mapping conservation reserve program (CRP) grasslands using multi-seasonal Thematic Mapper imagery. *Geocarto International*, 13(4):17–24, Dec. 1998.

[52] C. Eroglu Erdem, B. Sankur, and A. M. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Trans. Image Processing*, 13(7):937–951, 2004.

[53] F. Esposito, D. Malerba, G. Semeraro, and J. Kay. A comparative analysis of methods for pruning decision trees. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (5):476–491, 1997.

[54] T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations and feature selection for multimedia database search. *IEEE Trans. Knowledge and Data Engineering*, (4):911–920, 2003.

[55] G. Fan and X. Song. A study of contextual modeling and texture characterization for multiscale bayesian segmentation. In *Proc. of IEEE Int'l Conf. Image Proc.*, Rochester, NY, Sept. 2002.

[56] G. Fan and X.-G. Xia. Image denoising using local contextual hidden Markov model in the wavelet-domain. *IEEE Signal Processing Letter*, 8(5):125–128, May 2001.

[57] G. Fan and X.-G. Xia. Improved hidden Markov models in the wavelet-domain. *IEEE Trans. Signal Processing*, 49(1):115–120, Jan. 2001.

[58] G. Fan and X.-G. Xia. A joint multi-context and multiscale approach to Bayesian image segmentation. *IEEE Trans. Geoscience and Remote Sensing*, 39(12):2680–2688, Dec. 2001.

[59] G. Fan and X.-G. Xia. On context-based Bayesian image segmentation: Joint multi-context and multiscale approach and wavelet-domain hidden Markov models. In *Proc. 35th Asilomar Conf on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2001.

[60] G. Fan and X.-G. Xia. Wavelet-based texture analysis and synthesis using hidden Markov models. *IEEE Trans. Circuits and Systems, Part I*, 50(1):106–120, Jan. 2003.

[61] A. M. Ferman, A. M. Tekalp, and R. Mehrotra. Effective content representation for video. In *Proc. IEEE Int'l Conference on Image Processing*, Chicago, IL, 1998.

[62] M. A. T. Figueredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Analysis and Machine Learning*, 24(3):381–396, Mar. 2001.

[63] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nystrom method. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 231–238, 2001.

[64] M. Friedl and C. Brodley. Decision tree classification of land cover from remotely sensed data. *Remote Sensing Environment*, 61(3):399–409, 1997.

[65] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press Inc., 1990.

[66] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-6(6):721–741, November 1984.

[67] S. Gepshtein and M. Kubovy. The emergence of visual objects in space-time. In *Proc. of the National Academy of Science*, volume 97, pages 8186–8191, USA, 2000.

[68] E. Gokcay and C. Principe. Information theoretic clustering. *IEEE Trans. Pattern Anal. and Machine Intell.*, 24(2):158–171, Feb. 2002.

[69] H. Greenspan, J. Goldberger, and A. Mayer. A probabilistic framework for spatio-temporal video representation and indexing. In *Proc. European Conf. on Computer Vision*, volume 4, pages 461–475, Berlin, Germany, 2002.

[70] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise GMM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (3):384–396, March 2004.

[71] J. A. Gualtieri and S. Chettri. Support vector machines for classification of hyperspectral data. In *Proc. IEEE Int. Geoscience and Remote Sensing Symposium*, volume 2, pages 813–815, July 2000.

[72] J. A. Gualtieri, S. R. Chettri, R. F. Cromp, and L. F. Johnson. Support vector machine classifiers as applied to aviris data. In *Summaries of the Eighth JPL Airborne Earth Science Workshop: JPL Publication 99-17*, NASA/JPL, Feb. 1999.

[73] J. A. Gualtieri and R. F. Cromp. Support vector machines for hyperspectral remote sensing classification. In *in Proc. 27th SPIE AIPR Workshop*, 1998.

[74] R. Hammoud and R. Mohr. A probabilistic framework of selecting effective key frames for video browsing and indexing. In *International workshop on Real-Time Image Sequence Analysis*, 2000.

[75] A. B. Hamza, Y. He, and H. Krim. An information divergence measure for ISAR image registration. In *Proc. of IEEE Statistical Signal Processing Workshop*, pages 130–133, August 2001.

[76] A. Hanjalic, R. L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Trans. Circuits and System for Video Technologye*, 9(4):580–588, June 1999.

[77] A. Hanjalic and H. J. Zhang. An integrated scheme for automated video abstraction based on unsupervsied cluster-validity analysis. *IEEE Trans. on CSVT*, 9(8):1280–1289, 1999.

[78] M. Hansen, R. Dubayah, and R. Defries. Classification trees: an alternative to traditional land cover classifiers. *Int. J. Remote Sensing*, 17(5):1075–1081, 1996.

[79] J. Heald. USDA establishes a common land unit. In *ArcUser Online*, March-April 2002. http://www.esri.com/news/arcuser/0402/usda.html.

[80] L. Hermes, D. Frieauff, J. Puzicha, and J. M. Buhmann. Support vector machines for land usage classification in Landsat TM imagery. In *Proc. IEEE Int. Geoscience and Remote Sensing Symposium*, volume 1, pages 348–350, July 1999.

[81] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks*, (2):415–425, March 2002.

[82] X. Huang and J. R. Jensen. A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data. *Photogrammetric Engineering & Remote Sensing*, 63(10):1185–1194, October 1997.

[83] H. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Information Theory*, 14(1), 1968.

[84] D. A. Hull and S. Robertson. The trec-8 filtering track final report. In *in Proc. Eighth Text Retrieval Conference*, 2000.

[85] C. F. Hutchinson. Techniques for combining Landsat and ancillary data for digital classification improvement. *Photogrammetric Engineering & Remote Sensing*, 48(1):123–130, Jan. 1982.

[86] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Trans. Pattern Analysis and Machine Interlligence*, 22(1), January 2000.

[87] A. K. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (2):153–158, Feb. 1997.

[88] T. Joachims. Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*, 1999.

[89] T. Joachims. Estimating the generalization performance of an SVM efficiency. In *Proc. Int. Conf. Machine Learning*, 2000.

[90] M. Jones and J. Rehg. Statistical color models with applications to skin detection. *TR98-11, CRL, Compag Computer Corp.*, Dec. 1998.

[91] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 24(7):881–892, July 2002.

[92] S. Keerthi and C. J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation*, (7):1667–1689, 2003.

[93] J. Keuchel, S. Naumann, M. Heiler, and A. Siegmund. Automatic land cover analysis for tenerife by supervised classification using remotely sensed data. *Remote Sensing of Environment*, 86(4):530–541, Aug. 2003.

[94] C. Kim and J. N. Hwang. An itegrated scheme for object-based video abstraction. In *ACM Multimedia 2000*, Los Angeles, CA, 2000.

[95] C. Kim and J. N. Hwang. Object-based video abstraction for video surveillance systems. *IEEE Trans. Circuits and Systems for Video Technology*, 12(12):1128–1138, 2002.

[96] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support vector machines for texture classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11):1542 – 1550, Nov. 2002.

[97] S. Krishnamachari and R. Chellappa. Multiresolution Gauss-Markov random field models for texture segmentation. *IEEE Trans. Image Processing*, 6(2):251–267, Feb. 1997.

[98] S. Kullback. *Information Theory and Statistics*. John Wiley, New York, 1959.

[99] S. Kullback. *Information Theory and Statistics*. Dover, New York, 1968.

[100] B. Kuo and D. A. Landgrebe. Nonparametric weighted feature extraction for classification. *IEEE Trans. Geoscience and Remote Sensing*, (5):1096–1105, 2004.

[101] S. Lakshmanan and H. Derin. Simutaneous parameter estimation and segmentation of gibbs random dields using simulated annealing. *IEEE Trans. Pattern Anal. and Machine Intell.*, 2(8):799–813, August 1989.

[102] L. J. Latecki, D. DeMenthon, and A. Rosenfeld. Extraction of key frames from videos by polygon simplification. In *Proc. IEEE Int'l Symposium on Signal Processing and Its Applications*, pages 643–646, August 2001.

[103] M. Law and A. Zaccarin. Simultaneous feature selection and clustering using mixture models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[104] M. H. Law, A. K. Jain, and M. Figueiredo. Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems 15*, Cambridge, 2003.

[105] T. Lee, J. A. Richards, and P. H. Swain. Probabilistic and evidential approaches for multisource data analysis. *IEEE Trans. Geoscience and Remote Sensing*, pages 283–293, 1987.

[106] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Processing*, 13(11):1459–1472, Nov. 2004.

[107] T. M. Lillesand and R. W. Kiefer. *Remote Sensing and Image Interpretation*. John Wiley and Sons, 2000.

[108] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, 1988.

[109] B. liu and A. Zaccarin. New fast algorithms for the estimation of block motion vectors. *IEEE Trans. Circuits and System for Video Technology*, 3(2):148–157, 1993.

[110] J. Liu and P. Moulin. Image denoising based on scale-space mixture modeling of wavelet coefficients. In *Proc. IEEE Int. Conf. on Image Proc.*, Kobe, Japan, Oct. 1999.

[111] L. Liu and G. Fan. Combined key-frame extraction and object-based video segmentation. *IEEE Trans. Circuits and System for Video Technology*, to appear.

[112] A. Lunts and V. Brailovskiy. Evaluation of attributes obtained in statistical decision rules. *Engineering Cybernetics*, pages 98–109, 1967.

[113] D. J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[114] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. PAMI*, 11:674–693, 1989.

[115] B. S. Manjunath and R. Chellappa. Unsupervised texture segmentation using Markov random field models. *IEEE Trans. Pattern Anal. and Machine Intell.*, 13(5):478–482, May 1991.

[116] Jos M. Martnez. Mpeg-7 overview (ver.8). ISO/IEC JTC1/SC29/WG11 N4980, July 2002.

[117] R. Megret and D. DeMenthon. A survey of spatio-temporal grouping techniques. Technical report, University of Maryland, College Park, March 2002. http://www.umiacs.umd.edu/lamp/pubs/TechReports/.

[118] F. Melgani and L. Bruzzone. Support vector machines for classification of hyperspectral remote-sensing images. In *Proc. IEEE Int. Geoscience and Remote Sensing Symposium*, volume 1, pages 506–508, June 2002.

[119] M. K. Mihcak, I. Kozintsev, and K. Ramchandran. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, Dec. 1999.

[120] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care montoring. In *in Proc. Sixteenth Int. Conf. on Machine Learning*, 1999.

[121] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized-gaussian and complexity priors. *IEEE Trans. Information Theory*, 45(4):909–919, Apr. 1999.

[122] R. Narasimha, A. Savakis, R. M. Rao, and R. De Queiroz. Key-frame extraction using mpeg-7 motion descriptors. In *Proc. of IEEE Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1575–1579, Nov. 2003.

[123] M. Nerlove. *Likelihood Inference in Econometrics.* "http://www.arec.umd.edu/mnerlove/p-mnerlove.htm".

[124] A. G. Nguyen and J. N. Hwang. Scene context dependent key frame selection in streaming. In *Proc. of IEEE International Conference (the 22nd) on Distributed Computing Systems Workshops*, pages 208–213, July 2002.

[125] H. Noda, M. N. Shirazi, and E. Kawaguchi. An MRF model-based method for unsupervised textured image segmentation. In *Proc. of the 13th International Conference on Pattern Recognition*, volume 2, pages 765–769, August 1996.

[126] H. Noda, M. N. Shirazi, and E. Kawaguchi. Textured image segmentation using MRF in wavelet domain. In *Proc. of the 2000 International Conference on Image Processing*, volume 3, pages 572–575, 2000.

[127] J. Novovicova, P. Pudil, and J. Kittler. Divergence based feature selection for multimodal class densities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(2):218–223, 1996.

[128] J. C. Pesquet, H. Krim, D. Leporini, and E. Hamman. Bayesian approach to best basis selection. In *Proc. of IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, volume 5, pages 2634–2673, Atlanta, GA, May 1996.

[129] C. Principe, N. R. Euliano, and W. C. Lefebvre. *Neural and Adaptive Systems: Fundamentals through Simulations.* John Wiley & Sons, New York, 2000.

[130] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, pages 1119–1125, Nov. 1994.

[131] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[132] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[133] A. Rangarajan and R. Chellappa. Markov random field models in image processing. *The handbook of brain theory and neural networks, MIT Press*, pages 564–567, 1995.

[134] G. Ratsch, S. Mika, B. Schölkopf, and K. R. Muller. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (9):1184–1199, 2002.

[135] A. Renyi. Some fundamental questions of information theory. In *Selected Papers of Alfred Renyi*, volume 2, pages 526–552, Akad. Kiado, Budapest, 1976.

[136] J. Rissanen. A universal prior fo rintegers and estimation by minimum description length. *Annals of Statistics*, 11(2):417–431, 1983.

[137] J. K. Romberg, H. Choi, and R. G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden markov models. *IEEE Trans. Image Processing*, 10:1056–1068, July 2001.

[138] H. Rowley, S. Baluja, and T. Kanade. Neyral network based face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[139] B. Scholkopf, J. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Machine*. MIT Press, 1999.

[140] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[141] C. W. Shaffrey, N. G. Kingsbury, and I. H. Jermyn. Unsupervised image segmentation via Markov trees and complex wavelets. In *Proc. of IEEE Int'l Conf. Image Proc.*, Rochester, Sept. 2002.

[142] B. Shahraray and D. C. Gibbon. Automatic generation of pictorial transcripts of video programs. In *Proc. IS&T/SPIE Digital Video Compression: Algorithms and Technologies*, pages 512–519, 1995.

[143] J. G. Shanahan and N. Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *CIKM 2003*, 2003.

[144] J. Shi and J. Malik. Motion segmentation and tracking using Normalized cuts. In *Proc. of Int. Conf. on Computer Vision*, pages 1151–1160, 1998.

[145] M. Simard, S. S. Saatchi, and G. D. Grandi. The use of decision tree and multiscale texture for classification of JERS-1 SAR data over tropical forest. *IEEE Trans. Geoscience and Remote Sensing*, 38(5):2310–2321, Sept. 2000.

[146] E. P. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc. of IEEE Int'l Conf. Image Proc.*, volume 1, pages 62–66, Oct. 1998.

[147] A. H. S. Solberg, A. K. Jain, and T. Taxt. Multisource classification of remotely sensed data: fusion of Landsat TM and SAR images. *IEEE Trans. Geoscience and Remote Sensing*, 32(4):768–778, April 1994.

[148] X. Song and G. Fan. Coherent video key-frame extraction and object segmentation. *Submitted to IEEE Trans. Circuits and System for Video Technologye.*

[149] X. Song and G. Fan. Unsupervised Bayesian image segmentation using wavelet-domain hidden Markov models. In *Proc. of IEEE Int'l Conf. Image Proc.*, Barcelona, Spain, Sept. 2003.

[150] X. Song and G. Fan. Joint key-frame extraction and object-based video segmentation. In *Proc. of IEEE Workshop on Motion and Video Computing (MOTION 2005)*, Jan. 2005.

[151] X. Song and G. Fan. Key-frame extraction for object-based video segmentation. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2005)*, March 2005.

[152] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[153] D. M. J. Tax. *One-class Classification.* Ph.D. dissertation, Technische Universiteit Delft, The Netherlands, 2001.

[154] K. Torkkola and W. Campbell. Mutual information in learning feature transformation. In *Proc. of International Conf. on Machine Learning*, Stanford, USA, 2000.

[155] V. N. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[156] N. Vasconcelos. Feature selection by maximum marginal diversity. In *Neural Information Processing Systems*, Vancouver, Canada, 2002.

[157] N. Vasconcelos. Feature selection by maximum marginal diversity: optimality and implications for visual recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[158] N. Vasconcelos and M. Vasconcelos. Scalable discriminant feature selection for image retrieval and recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Washington DC, 2004.

[159] W. Wolf. Key frame selection by motion analysis. In *Proc. IEEE Int. Conf. Acoust, Speech, and Signal Proc.*, pages 1228–1231, 1996.

[160] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[161] M. M. Yeung and B. Liu. Efficient matching and clustering of video shots. In *Proc. IEEE Int'l Conference on Image Processing*, pages 338–341, Oct. 1995.

[162] C. Zhai, P. Jansen, E. Stoica, N. Grot, and D. A. Evans. Threshold calibration in clarit adaptive filtering. In *in Proc. Seventh Text Retrieval Conference*, pages 149–156, 1999.

[163] H. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4):643–658, 1997.

[164] J. Zhang, J. W. Modestino, and D. A. Langan. Maximun-likelihood parameter estimation for unsupervised stochastic model-based image segmentation. *IEEE Trans. Image Processing*, 3(4):404–420, July 1994.

[165] J. Zhang, D. Wang, and Q. N. Tran. A wavelet-based multiresolution statistical model for texture. *IEEE Trans. Image Processing*, 7(11):1621–1627, Nov. 1998.

[166] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proc. of IEEE Int Conf on Image Processing*, pages 866–870, Chicago, IL, 1998.

VITA

XIAOMU SONG

Candidate for the Degree of

Doctor of Philosophy

Thesis: STATISTICAL FEATURE SELECTION AND EXTRACTION FOR VIDEO AND IMAGE SEGMNETATION

Major Field: Electrical Engineering

Biographical:

Personal Data:

Education: Received Bachelor of Science degree in Electrical Engineering and Master of Science degree in Electrical Engineering from Northwestern Polytechnic University, Xi'an, China in July 1995 and March 1998, respectively. Completed the requirements for the Doctor of Philosophy with a major in Electrical Engineering at Oklahoma State University in July, 2005.

Experience: was an engineer first at the Institute of Remote Sensing Equipments, China, later at the Global Software Group, Motorola; employed by Oklahoma State University, School of Electrical and Computer Engineering as a graduate research assistant, 2001 to present.

Professional Memberships: the Institute of Electrical and Electronic Engineers

Name: Xiaomu Song Date of Degree: July, 2005

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: STATISTICAL FEATURE SELECTION AND EXTRACTION FOR VIDEO AND IMAGE SEGMNETATION

Pages in Study: 151 Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

Scope and Method of Study: The purpose of this study was to develop statistical feature selection and extraction methods for video and image segmentation, which partition a video or image into non-overlap and meaningful objects or regions. It is a fundamental step towards content-based visual information analysis. Visual data segmentation is a difficult task due to the various definitions of meaningful entities, as well as their complex properties and behaviors. Generally, visual data segmentation is a pattern recognition problem, where feature selection/extraction and data classifier design are two key components. Pixel intensity, color, time, texture, spatial location, shape, motion information, etc., are most frequently used features for visual data representation. Since not all of features are representative regarding visual data, and have significant contribution to the data classification, feature selection and/or extraction are necessary to select or generate salient features for data classifier. Statistical machine learning methods play important roles in developing data classifiers. In this report, both parametric and nonparametric machine learning methods are studied under three specific applications: video and image segmentation, as well as remote sensing data analysis.

Findings and Conclusions: For various visual data segmentation tasks, key-frame extraction in video segmentation, WDHMM likelihood computation, decision tree training, and support vector learning are studied for feature selection and/or extraction and segmentation. Simulations on both synthetic and real data show that the proposed methods can provide accurate and robust segmentation results, as well as representative and discriminative features sets. This work can further inspire our studies towards the real applications. In these applications, we are able to obtain state-of-the-art or promising results as well as efficient algorithms.

ADVISER'S APPROVAL: Guoliang Fan