Feature Binding of *MPEG-7* Visual Descriptors

Using Chaotic Series

Hanif Azhar

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

August 2010

Your file Votre référence
ISBN: 978-0-494-71130-9
Our file Notre référence
ISBN: 978-0-494-71130-9

Abstract

Due to advanced segmentation and tracking algorithms, a video can be divided into numerous objects. Segmentation and tracking algorithms output different low-level object features, resulting in a high-dimensional feature vector per object. The challenge is to generate feature vector of objects which can be mapped to human understandable description, such as object labels, e.g., *person*, *car*. *MPEG-7* provides visual descriptors to describe video contents. However, generally the *MPEG-7* visual descriptors are highly redundant, and the feature coefficients in these descriptors need to be pre-processed for domain specific application. Ideal case would be if *MPEG-7* visual descriptor based feature vector, can be processed similar to some functional simulations of human brain activity.

There has been a established link between the analysis of temporal human brain oscillatory signals and chaotic dynamics from the electroencephalography (EEG) of the brain neurons. Neural signals in limited brain activities are found to be behaviorally relevant (previously appeared to be noise) and can be simulated using chaotic series. Chaotic series is referred to as either a finite-difference or an ordinary differential equation, which presents non-random, irregular fluctuations of parameter values over time in a dynamical system. The dynamics in a chaotic series can be *high-* or *low*-dimensional, and the dimensionality can be deduced from the topological dimension of the attractor of the chaotic series. An attractor is manifested by the tendency of a non-linear finite difference equation or an ordinary differential equation, under various but delimited conditions, to go to a reproducible active state, and stay there.

We propose a feature binding method, using chaotic series, to generate a new feature vector, *C-MP7*, to describe video objects. The proposed method considers *MPEG-7* visual descriptor coefficients as dynamical systems. Dynamical systems are excited (similar to neuronal excitation) with either high- or low-dimensional chaotic

series, and then histogram-based clustering is applied on the simulated chaotic series coefficients to generate *C-MP7*. The proposed feature binding offers better feature vector with high-dimensional chaotic series simulation than with low-dimensional chaotic series, over *MPEG-7* visual descriptor based feature vector. Diverse video objects are grouped in four generic classes (e.g., *has_person*, *has_group_of_persons*, *has_vehicle*, and *has_unknown*) to observe how well *C-MP7* describes different video objects compared to *MPEG-7* feature vector. In *C-MP7*, with high dimensional chaotic series simulation, 1) descriptor coefficients are reduced dynamically up to 37.05% compared to 10% in *MPEG-7*, 2) higher variance is achieved than *MPEG-7*, 3) multi-class discriminant analysis of *C-MP7* with Fisher-criteria shows increased binary class separation for clustered video objects than that of *MPEG-7*, and 4) *C-MP7*, specifically provides good clustering of video objects for *has_vehicle* class against other classes.

To test *C-MP7* in an application, we deploy a combination of multiple binary classifiers for video object classification. Related work on video object classification use non-MPEG-7 features. We specifically observe classification of challenging surveillance video objects, e.g., incomplete objects, partial occlusion, background over lapping, scale and resolution variant objects, indoor / outdoor lighting variations. *C-MP7* is used to train different classes of video objects. Object classification accuracy is verified with both low-dimensional and high-dimensional chaotic series based feature binding for *C-MP7*. Testing of diverse video objects with high-dimensional chaotic series simulation shows, 1) classification accuracy significantly improves on average, 83% compared to the 62% with *MPEG-7*, 2) excellent clustering of *vehicle* objects leads to above 99% accuracy for only *vehicles* against all other objects, and 3) with diverse video objects, including objects from poor segmentation, *C-MP7* is more robust as a feature vector in classification than *MPEG-7*. Initial results on sub-

group classification for *male* and *female* video objects in *has_person* class are also presentated as subjective observations.

Earlier, chaos series properties have been used in video processing applications for compression and digital watermarking. To our best knowledge, this work is the first to use chaotic series for video object description and apply it for object classification.

Acknowledgments

I thank my supervisor Dr. Aishy Amer for her continuous support, encouragement and guidance in my PhD work. She has been always very patient, willing to listen and provided useful advice. At every turns on my research challenges she always allowed me to explore the best available options with firm directions. During the writing process all her comments and suggestions immensely helped me to improve my presentation with straight and simple choice of words. This thesis is made possible from her valuable editing. I admire her multitasking hardworking ethics in research pursuits. I also like to appreciate her strong will power and 'can-do' approach with a confident touch of professionalism. I am certain that I will carry her 'to-the-point' working style in my carrier endeavor. Thank you professor!

A special thanks to the members of my thesis committee Dr. Sudhir P. Mudur, Dr. Nawwaf Kharma and Dr. William E. Lynch for their constructive critics which pushed me to overcome initial loopholes in my work, and thus has strengthen the quality of my work. Dr. Mudur raised several flags during his deliberations to set my focus on comprehensive literature review. Dr. Kharma has been very helpful in projecting creative ideas on feature-based classification. His objective perspectives during multiple brain storming meetings made me understand the importance of feature. I will always remember his quote 'feature is the king'. Dr. Lynch, with basic and precise questions, acted as a pivotal guard in my research to clearly identify evaluation criteria in analyzing results. I thank all the respected members for their insightful comments and generous suggestions. They made me realize how important

it is to address hard questions to take the research challenge to the next level on every turn.

Let me say thank you to my colleagues in the 'VidPro' lab who have offered me the best fellowships. On every steps in my work, it was my pleasure to share with them creative visions, to get feedback on first-hand results, to sort out system bottlenecks, and to trigger quality discussions on pretty much anything that matters.

I am truly grateful to my parents, Azhar Ali and Halima Begum, for giving me the very first welcome in this world, for providing the best care to compete for the best, for 'all-out' support and encouragement to pursue my interests, for sacrificing any personal comforts over the priority of my education, for keeping me virtually in their hearts despite the distance, above all, for just being the best dad and mom to walk me to the path of the righteous.

I do also thank my parents-in-law, Shahidullah Khan and Fatema Rayan, for their blessings during these long years, for not questioning my academic intentions, and for staying confident on my potentials.

Last, but not the least, I thank my beautiful wife Jannatul Nayem Rumki for being on my side 'no matter what', for being my mirror when I look for confidence, for being my wings to fly whenever I felt down, for listening candidly to all the chaos and image processing jargons despite her background in business administration, for believing in me, for smiling everyday, for loving me unconditionally, and for delivering my charming prince Yalid Rayan Azhar.

Dedication

To my loving son Yalid,

who has been the most precious gift in my life.

Contents

List of Figures

List of Tables

List of Notation

MPEG-7	Motion Picture Expert Group -7 standard for meta data
XML	eXtensible Markup Language
XM	eXperimental Model (MPEG-7)
CML	Coupled Map Lattice
SVM	Support Vector Machine
kNN	k- Neareast Neighbor
FPS	Frame Per Seconds
KLR	Kernel Logistic Regression
ANN	Artificial Neural Network
MLP	Multi Layer Perceptron
LBP	Local Binary Pattern
MB-LBP	Multi-block Local Binary Pattern
CAVIAR	Context Aware Vision using Image-based Active Recognition [1]
2D	Two Dimensional
3D	Three Dimensional
EEG	ElectroEncephaloGraphy
HSV	Hue Saturation Value color space
HMM	Hidden Markov Model
MDS	Classical Multidimensional Scaling
YUV	A color space in terms of one luma (Y') and two chrominance (UV) components, based on RGB values
ART	Angular Radial Transformation
VSoIP	Video Surveillance over Internet Protocol [2]
CSS	Curvature Scale Space representation [3]
IMB	German abbreviation for Intelligent Multimedia Library
VidAna	Video Analysis module [4]
HMMD	Hue, Max, Min, Diff color space used by the MPEG7 standard, where, Max=max(R,G,B), Min=(R,G,B), and Diff=Max-Min
V1	Visual One
MT	Middle Temporal
PCA	Principal Component Analysis
l-d	Low Dimensional
h-d	High Dimensional
HVS	Human Visual System

SPECT	Single Photon Emission Computed Tomography, a nuclear medicine tomographic imaging technique using gamma rays
PET	Positron Emission Tomography, a nuclear medicine imaging technique to produce a three-dimensional image of functional processes in the body
PETS 2001	Data used at International Workshop on Performance Evaluation of Tracking and Surveillance 2001 [5]
fMRI	functional Magnetic Resonance Imaging
PTZ	Pan Tilting Zoom camera
KIII	Chaos theory based neural network model [6]
RBF	Radial Basis Function
DS	Description Schema
D	Descriptor
RTSP	Real Time Streaming Protocol
GUI	Graphical User Interface
USB	Universal Serial Bus
L	Number of video objects
O_l	List of video objects, where $l \in L$
d_{ijl}	i^{th} *MPEG-7* descriptor of object O_l with j^{th} feature coefficients, $d_{ijl} \in \mathbb{R}$
J_i	Number of feature elements in descriptor i
I	Number of *MPEG-7* visual descriptors per object O_l
YY_l	Feature vector for object O_l
(d_{ml})	Value of feature coefficient m of video object l
N	Iteration number
$f(\cdot)$	Function that generates a chaotic attractor
$H(\cdot)$	Histogram-analysis based feature binding of the re-constructed attractors for all $f(\cdot)$
$\Re$	Real
S_t	State of the system at time t
$f(S_t)$	Function that defines the change of state at time t
S_{t+1}	State at time *t+1*
x_t	State of the system at time t
x_n	State of the discrete system at iteration n
p	*has_person* class label
g	*has_group_of_person* class label
v	*has_vehicle* class label
u	*has_unknown* class label

Chapter 1

Introduction

1.1 Motivation

A content-based video processing system is suppose to interpret the video contents (i.e., *what is happening*?), to raise warning if any anomaly in the description of video contents exist, and also to predict specific description of video contents [7]. The excellence of such a system depends on how intelligently it can interpret huge volume of multidimensional visual data of video contents. The modules in a content-based video processing system are, 1) video analysis, and 2) video interpretation (Fig. 1.1) [7, 8]. *Video analysis* module includes two pre-precessing functions, a) video enhancement (e.g., noise estimation, noise reduction, smoothing, sharpening), and b) temporal segmentation (e.g., shot detection). The pre-processed video signals then go through motion estimation, object segmentation, and tracking. *Video interpretation* module includes description of objects, classification of objects, identification of objects, retrieval of objects, description of underlying concepts and events.

The multimedia description standard *MPEG-7* can be used as an implementation tool in video content description. *MPEG-7* provides standardized visual descriptors

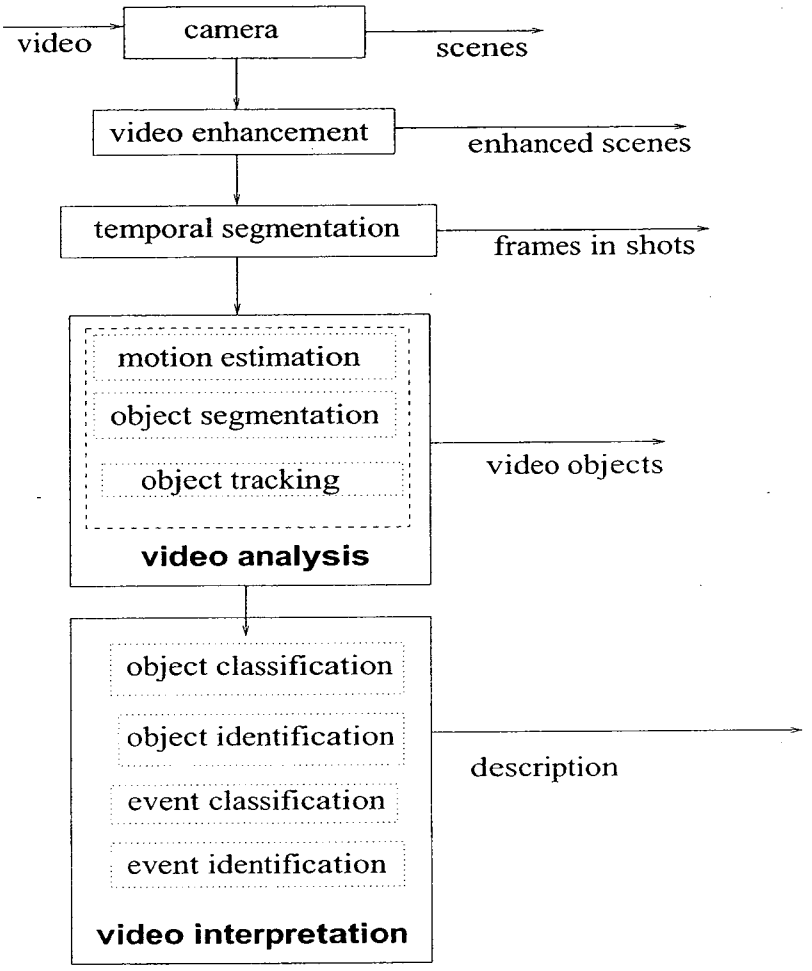


Figure 1.1: Content-based video processing.

which are designed to uniquely identify or retrieve media contents (e.g., objects in video). These visual descriptors are designed on the basis of similar functions in human visual information processing [9]. Examples of such functions are, i) frequency channels in homogeneous texture descriptors with uniform angular but nonuniform radial direction, ii) human characterization of perception in terms of regularity, coarseness, and directionality in texture browsing descriptors, iii) intuitive representation of salient color in dominant color descriptors, iv) Curvature Scale Space, CSS, representation in contour shape descriptor from the observation that, when comparing

shapes, humans tend to decompose shape contours into concave and convex sections. Different visual descriptors expose different features of the media content. Generally, the *MPEG-7* visual descriptors contains redundant feature coefficients, and in natural image retrieval applications, feature reduction techniques can save up to 80% of storage and transmission capacity of these descriptors [10]. For domain specific video processing applications, such as for surveillance video, the redundancy in *MPEG-7* visual descriptors has not been studied. The common approach would be to either refine the *MPEG-7* visual descriptors for domain specific video applications, or to add new visual descriptors for the same. Nonetheless, for human understandable description of video objects, ideal would be, if the *MPEG-7* visual descriptors are processed after some human brain activity.

Brain activities are not fully understood as to-date, yet in some states, e.g., slow-wave sleep stage, the collective oscillatory electroencephalography, i.e., EEG, signals from the neurons in the brain, appears to be chaotic and not random [11, 12]. What previously appear to be *noise* in a dynamical system, eliminated with filters while recording the EEG signals of brain activity, currently appears to be the behaviorally relevant signal. Chaotic properties, thus, can be related to the ability of the brain to generate and process information [11]. As such, neuro-science experiments report that, neuronal excitations in the brain are heavily dependent on chaotic attractor [13]. The chaotic attractor is a non-linear geometrical structure generated by a chaotic series (see Section 2.4).

1.2 Problem Statement

The research in video processing is moving from pixel towards the meaningful content analysis, i.e., interpretation of video contents through human understandable descrip-

tions [8, 14–17]. There are two types of video contents, *primary* and *secondary*. The *primary* contents are video objects (e.g., *person*). The secondary contents in video are concepts (e.g., *indoor*) and events (e.g., *fallen person*). A *video object* is a segmented image-region in a frame which has been tracked over successive frames. Conventional feature-extraction techniques usually output a high-dimensional feature vector to describe objects. This feature vector is vital for creating object description in various object-based applications, e.g., identification, retrieval, classification in specific video domains, e.g., sports, movie, news, surveillance [18]. While, segmentation and tracking modules are yet to be perfect, video object description can also be helpful to interpret secondary contents in video, i.e., concepts and events.

Due to recent advancements in segmentation and tracking [8, 14], a video can be divided into numerous video objects from a scene. The challenge for any video interpretation module is to generate feature vector for primary video contents, i.e., objects, which can be later mapped to human understandable description, e.g., *person*, *car*. When these objects are available from the prior video analysis module, an object classification application in the interpretation module can generate such human understandable description from the feature vector, (Fig. 1.1). The feature vector is critical for discriminative training patterns in video object classification.

Object classification in video is itself a non-trivial challenge. Common problems, which follow through the video analysis module in to object classification, include, a) lack of invariance in scale, lighting, and orientation of objects, b) posture changes for the same object in successive frames, c) existence of parts of background or other video objects inside segmented objects, and d) lost or missing parts of video objects in some frames. Discriminative patterns for the same video object in consecutive frames are, thus, not always available.

Individual *MPEG-7* visual descriptor (e.g., contour shape) coefficients can be

used [19] to describe corresponding feature (e.g., shape) of video objects. *MPEG-7* feature vector of a video object is the combination such feature coefficients from multiple visual descriptors in a multi-dimensional feature space. The *MPEG-7* feature vector can be high-dimensional (e.g., edge histogram descriptor has 80 feature coefficients). During the *MPEG-7* design process, the co-relation among different visual descriptors were not studied. The design goal for each *MPEG-7* visual descriptor was to improve retrieval index in corresponding media databases [9]. The statistical properties, e.g., variance, of *MPEG-7* visual descriptor as a feature vector are not studied. As such, the *MPEG-7* design goal did not include the data quality (e.g., redundancy, discriminancy) of the descriptors as a feature vector.

The problem to solve, in this thesis, is to create a new feature vector, inspired by human brain activity simulation, that,

- is compliant to *MPEG-7* multimedia description standard
- ensures good data quality better than *MPEG-7* visual descriptors
- contains inherent patterns of different video objects to generate human understandable descriptions
- improves discrimination among different class of video objects than *MPEG-7* visual descriptors

The new feature vector is suppose to offer high accuracy for video interpretation applications, such as video object classification in specific domains, such as surveillance video [20–22]. Here the input for video object classification can be the new feature vector, where, the output will be the human understandable class-labels of video objects.

1.3 Proposed Solution

In this thesis, we investigate the scope of chaotic series in grouping feature coefficients of high-dimensional *MPEG-7* feature vector. Our work is motivated by the existence of chaotic series in the neuronal EEG response of human brain activities [11]. In the proposed method (see Fig. 1.2), we perform chaotic feature binding, i.e., group relevant feature coefficients, [23] on the *MPEG-7* feature vector.

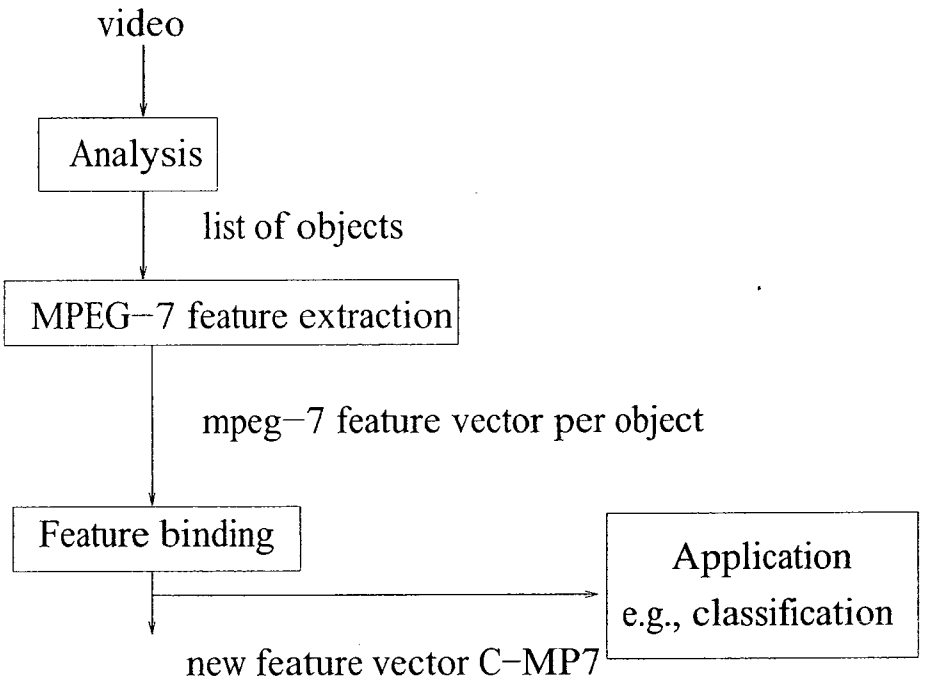


Figure 1.2: Overview of the proposed method.

Our basic idea is to assume feature coefficients from *MPEG-7* visual descriptors as dynamical systems in an abstract multi-dimensional space. We assume such dynamical systems to be similar to biological *neurons*. The multi-dimensional space is a combination of different *MPEG-7* feature spaces. Each feature coefficient is excited, similar to electro-chemical action potential to a neuron, with a chaotic series. The choice of chaotic series is limited to one dimensional finite-difference, and delay differ-

ential equation involving single variable only. Neighborhood interaction of different chaotic series is calculated with Coupled Map Lattice [24]. Statistical properties (e.g., mean) of the chaotic series is then mapped to replace the original *MPEG-7* feature coefficients. A histogram-based clustering of these mapped chaotic feature coefficients is used to locate the new feature vector indexes those form the largest cluster. The *MPEG-7* feature vector coefficients are finally re-mapped to the located feature indexes from the histogram-based clustering. The rest of the *MPEG-7* feature coefficients are discarded in the new feature vector. We name the new *MPEG-7* compliant feature vector as *C-MP7*.

We analyze the goodness of *C-MP7* as a feature vector by observing the data qualities (e.g., variance, discriminancy), multi-class Fisher criteria based binary class separations, and percentage of dynamic feature coefficient reductions compared to those of the original *MPEG-7* feature vector. To test *C-MP7* in an application, we deploy a combination of multiple binary classifiers for video object classification. Related work on video object classification use non-MPEG-7 features. We specifically observe classification of challenging surveillance video objects, e.g., incomplete objects, partial occlusion, background over lapping, scale and resolution variant objects, indoor / outdoor lighting variations. *C-MP7* is used to train different classes of video objects. We apply popular classifiers in supervised learning paradigm to verify the performance of *C-MP7*, for video object classification to generate human understandable description of video objects. Object classification accuracy is verified with both low-dimensional and high-dimensional chaotic series based feature binding for *C-MP7*.

1.4 Contributions

Earlier, chaotic series-based approaches have been studied for pattern classification of EEG signals [25], video compression [26], and digital watermarking [27]. To our best knowledge, this work is the first to use chaotic series to create feature vector for video object description and apply it to object classification [28–32].

The contribution of our work is a new *MPEG-7* compliant feature vector, *C-MP7*. This feature vector is simulated by a chaotic series to describe video objects as available from video analysis, even under poor segmentation and tracking. We,

- propose chaotic feature binding to group a set of *MPEG-7* visual descriptor coefficients inherent in different video object class pattern
- show data quality improves in *C-MP7*, when compared to the statistical properties of the *MPEG-7* feature vector, e.g., higher variance in *C-MP7*
- show feature vector discrimination among different classes of video objects increases with *C-MP7*, when compared to that of the *MPEG-7* feature vector
- show improved classification accuracy of video objects (different scale, resolution, and orientation in assorted indoor and outdoor surveillance shots) with *C-MP7*, when compared to that with *MPEG-7* feature vector

In addition to our proposed core approach for feature binding, this thesis presents analysis on the excellence of *C-MP7* over *MPEG-7* as a feature vector, using either *low-dimensional* or *high-dimensional* chaotic series. We show, cross-validation accuracy with *C-MP7* increases over that with *MPEG-7* feature vector, using both *low-dimensional* and *high-dimensional* chaos. However, cross-validation is not enough to draw conclusion about feature vector for surveillance video (see Section 5.2.2). So,

we further evaluate *C-MP7* with both *high-dimensional* and low-dimensional chaotic series for classification of diverse video objects from public and local databases. With *high-dimensional* chaos, for *C-MP7*, we find,

- in *C-MP7*, descriptor coefficients are reduced dynamically from *MPEG-7*, similar dynamic feature coefficient reduction is attained with *low-dimensional* chaotic series,
- multi-class Fisher-criteria shows increased binary class separation for video objects in different classes, but, class separation decreases with *low-dimensional* chaotic series
- *vehicle* objects are clustered well, which leads to above very high classification accuracy for only *vehicles* against all other objects
- classification accuracy significantly improves when compared to *MPEG-7* for all classes of video objects, but, the accuracy decreases with *low-dimensional* chaotic series
- chaotic series properties, e.g., drifts in attractors due to the existence of transient in *high-dimensional* chaotic series, allow *C-MP7* to include subtle variations in descriptor coefficients for video objects in a class

1.5 Thesis Outline

The rest of this thesis is organized as follow,

- Chapter 2, includes brief introductions on feature space, *MPEG-7*, video content descriptions, and chaos theory.

- Chapter 3, discusses relevant research on the use of chaos theory to simulate brain functions, on the use of *MPEG-7* visual descriptors in video surveillance and on video object classification.
- Chapter 4, provides details on *MPEG-7* visual descriptor selection.
- Chapter 5, contains the proposed method for chaotic feature binding to generate the new feature vector, and statistical analysis of the new feature vector.
- Chapter 6, reports evaluation criteria for *C-MP7* in video object classification compared to *MPEG-7* feature vector.
- Chapter 7, reports the conclusion of our work with a focus on the future work.
- Appendix A, presents sample XML description of MPEG-7 visual descriptors in different video objects.

Chapter 2

Background

This chapter includes brief introductions on description of video contents, *MPEG-7* visual descriptors and chaos theory.

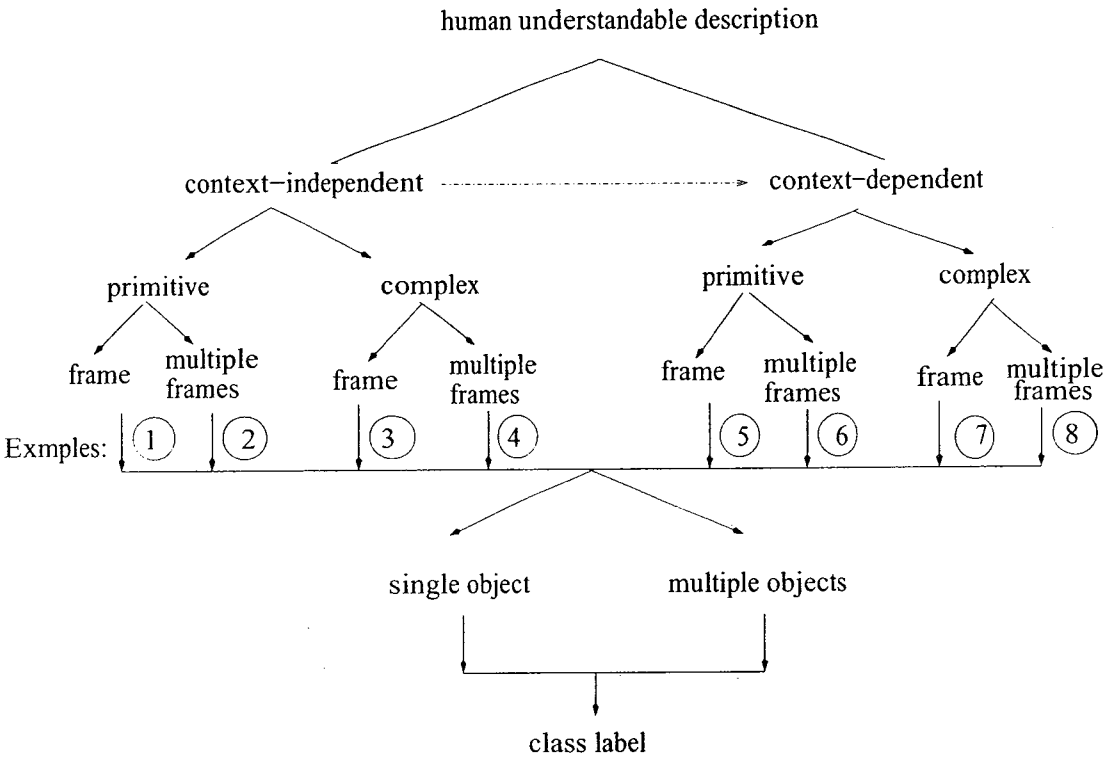
2.1 Introduction

In this thesis, *MPEG-7* visual descriptors and chaos theory are used as integration tools in a content-based video processing system (e.g., surveillance video) to create human understandable descriptions of video objects. The description is generated as object class labels. The proposed method creates a new *MPEG-7* compliant feature vector for video objects, which we feed to the classification framework as input. The output of the classification framework are the class labels for video objects. To better understand the integration tools in this thesis, we include discussion on human understandable description of video contents in Section 2.2, brief information on *MPEG-7* standard in Section 2.3, and fundamental properties of chaos theory in Section 2.4 [11, 12, 33, 34].

2.2 Video Content Description

A video is physically formed by *shots* and conceptually described by *scenes*. A *shot* is a set of successive frames, while, a *scene* is a story unit and consists of sequences of shots. In surveillance, scenes are usually sequences of connected shots. In general, human understandable description of a video can be the meaning of its contents (i.e., objects, concepts, and events). An example of such description for an object is it's class-label. Class-labels describe an object as *car* or *person* from tracked objects, e.g., *object 1*, *object 2*,...*object n* as available in tracking. *Concept* is the extent of idea (abstract or concrete description) that relates a video object to the scene. Concept can be either primitive (e.g., sky) or complex (e.g., harmony, suspicious). Object labels can also be considered as concepts (e.g., human, car, ambulance). *Event* express specific behavior or action of a particular object (or multiple objects) related in a frame or a set of frames [35]. The meaning of a video scene can vary with the presence of relevant context. A *Context* [36, 37] is the accessory information in addition to the low-level features of video contents. Context relates a specific video content to other similar contents or background of the scene.

The hierarchical flow of possible human understandable descriptions of video contents is shown in Fig. 2.1, which shows that the description of concepts and events can be formed from the single or multiple objects. The description of concepts and events can be either *context-independent* or the *context-dependent*. For example, consider similar scenes such as, *a person stopped in the traffic-crossing, a car stopped in the middle of the highway, or a truck stopped in a gas station.* These scenes each have different context-dependent meaning, as the context changes in each scene. In the above example, *stop* is a *context-independent event* which has precisely the same meaning, irrespective of objects (e.g., *'person'*, *'car'*, or *'truck'*). A *context-dependent*



1. person, sky
2. enter, stop
3. fallen person
4. lurking person, occluded vehicle
5. downtown, airport
6. crowed formation, converging persons
7. accident victim, goal, hijack
8. handshake, drunk driving, panic

Figure 2.1: Human understandable description of video contents.

description may also be a composition of simpler context-independent descriptions of contents as shown by the dotted arrow in Fig. 2.1. An example is a possible *robbery* scene, where *a stopped car surrounded by multiple slowing bikes at late-night.* Each context-independent or context-dependent description can be either primitive or complex, also shown in Fig. 2.1. Primitive descriptions describe simple video contents. It can be generated from either a single frame (e.g., the key frame) or from a set of frames. Complex descriptions can be generated from a set of video contents either in a single frame (e.g., the key frame) or in a set of frames. The bottom nodes in Fig. 2.1 show that, there may be single object or multiple objects involved in any video content description. Thus, context-independent descriptions of a concept or an event can be formed from the *individual* low-level feature changes in a single object or in multiple objects contained in a key-frame or in multiple frames. Similarly, context-dependent description of a concept or an event can be formed from *interactive* low-level feature changes in single object or in multiple objects contained in a key-frame or in multiple frames. Here *interactive* refers to the low-level feature change in an object, which has dependency on the context information relative to its own low-level features.

2.3 *MPEG-7* Features For Video Object Description

The *MPEG-7* standard provides a rich set of visual description methods and tools for different viewpoints on multimedia contents. The *MPEG-7* does not standardize the extraction of visual descriptors, nor does it specify the application that makes use of the description of each descriptor. A *MPEG-7 visual descriptor* provides a standard format to describe a media content, e.g., objects in video. Each visual descriptor

has multiple feature coefficients and follows a predefined XML schema to generate description in XML format or use encoding scheme to describe in binary format *.mp7*. The *MPEG-7* visual descriptors cover the basic categories of visual features: color, texture, shape, motion, localization, and face recognition. Each category consists of elementary and compound descriptors. The elementary descriptors are *Color Layout*, *Color Structure*, *Dominant Color*, *Scalable Color*, *Edge Histogram*, *Homogeneous Texture*, *Texture Browsing*, *Region-based Shape*, *Contour-based Shape*, *Camera Motion*, *Parametric Motion* and *Motion Activity* [10]. Other compound descriptors (e.g., *Group-of-Frames/Group-of-Pictures* is based on *Scalable Color*) are defined as descriptor aggregation or localization from the elementary ones. The *MPEG-7* reference software XM [38] can be used to extract these visual descriptors. The details on the algorithms of each visual descriptor is available in [9]. An *MPEG-7* visual descriptor provides a standard format to describe a media content, e.g., video object. Each visual descriptor has multiple feature coefficients and follows a predefined XML schema to generate description in XML format.

The advantages of using MPEG-7 visual descriptors for describing contents in video is based on the knowledge that objects in a scene can have different discriminative low-level features, e.g., dominant color, shape, contour. These features can be used to search, index, classify objects contained in a scene. The visual descriptors provide good clues for locating objects in a visual field [39].

A *video object* is a segmented image-object in a frame which has been tracked from successive frames. Low-level features of a video object can be extracted after segmentation and tracking. Human understandable description of video objects is the feature vector that can describe similar type of video objects of a class in a feature space. An example can be classification label of tracked video objects i.e., image region *object 1*, *object 2*,...*object n* as *vehicle*, *person*.

Description of different classes of video objects in conventional problem-space (i.e., individual feature space) of low-level features is not trivial. One way to visualize video objects is to get a sense of how near or far feature vector data points are from each other. To illustrate individual feature spaces, we group a set of video objects in four different classes, *has_person* (p), *has_group_of_person* (g), *has_vehicle* (v), and *has_unknown* (u) (see Section 5.3.1 for more details on data sets). Multiple descriptor coefficients in a visual descriptor forms a feature vector, and an object can be represented as a data point in the that feature space. Given a set of distances (i.e., dis-similarities) between objects, it is possible to re-create a 2D representation of these objects. We apply classical multidimensional scaling (MDS) function *cmdscale* [40] to visualize video objects in different classes as data points in a 2D feature space. Initially *pdist* [40] function is used to find the pairwise feature distances between objects, which returns a vector containing the Euclidean distances between each pair of objects in an object-feature matrix. Rows of this matrix corresponds to objects, and columns corresponds to features. This vector is a *1-by-(L*(L-1)/2)* row vector, where L is the total number of objects, and this vector corresponds to the *L*(L-1)/2* pairs of objects in the object-feature matrix. Then, *squareform* [40] function is used to reformat the *1-by-(L*(L-1)/2)* row vector into a symmetric, square format, *SF*, so that *SF(aobj, oobj)*, denotes the feature distance between any object, *aobj*, and other object, *oobj*. Finally, *cmsscale* of the *SF* is calculated to obtain 2D embedded feature distances between objects. Figs. 2.2, 2.3, 2.4, and 2.5 show different video objects p,g, v, and u for color layout, contour shape, edge histogram, and region shape *MPEG-7* visual descriptors, respectively. All these figures show video objects among different classes are not clustered well. In color layout descriptor all the classes are over lapped on each other with different video objects except part of p video objects separated away from others (Fig 2.2). In contour shape overlapped video objects are

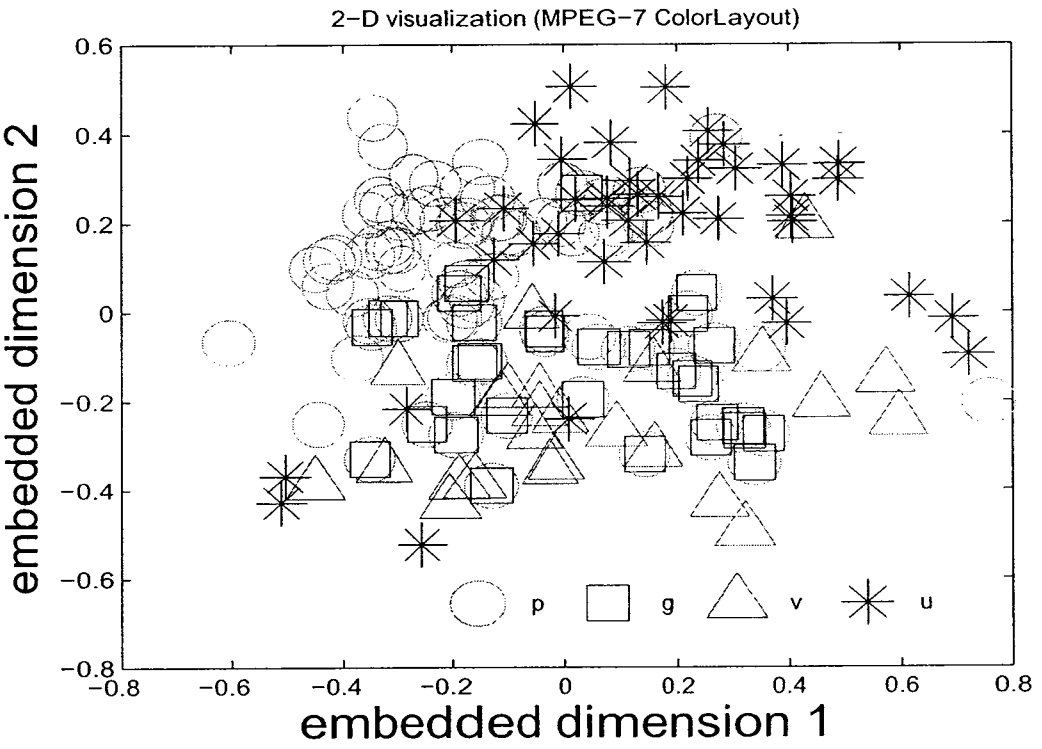


Figure 2.2: 2D representation of color layout feature space, all the classes are over lapped on each other with different video objects. However, parts of p video objects separated away from others.

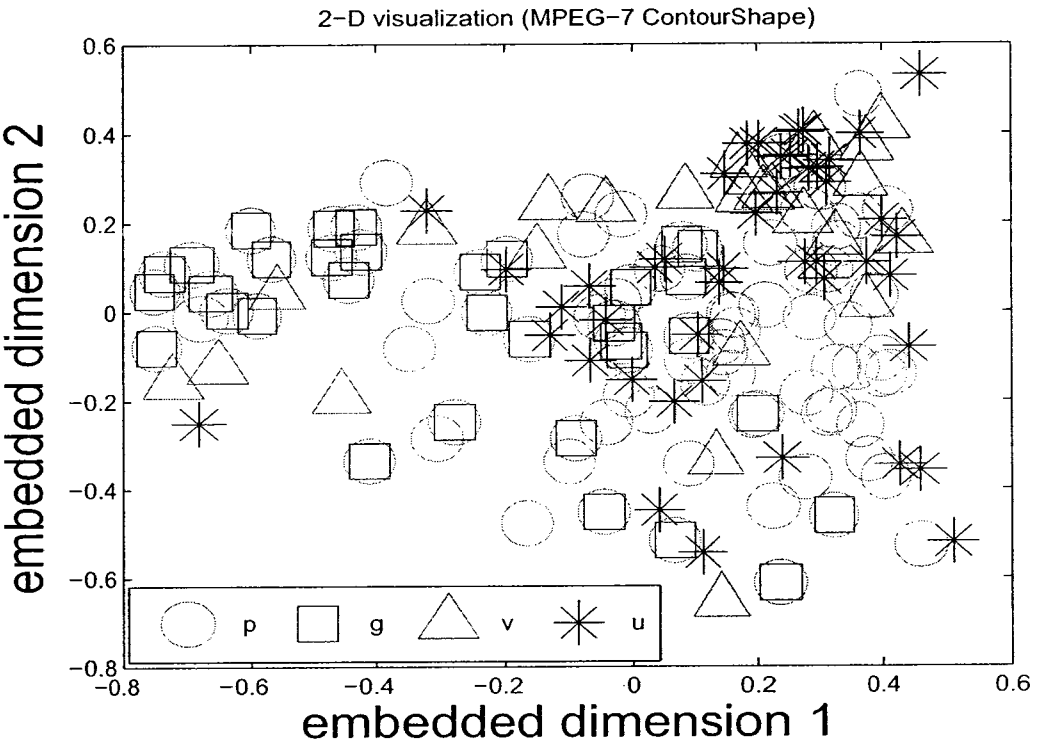


Figure 2.3: 2D representation of contour shape feature space, overlapped video objects with no dominent clustering of video objects.

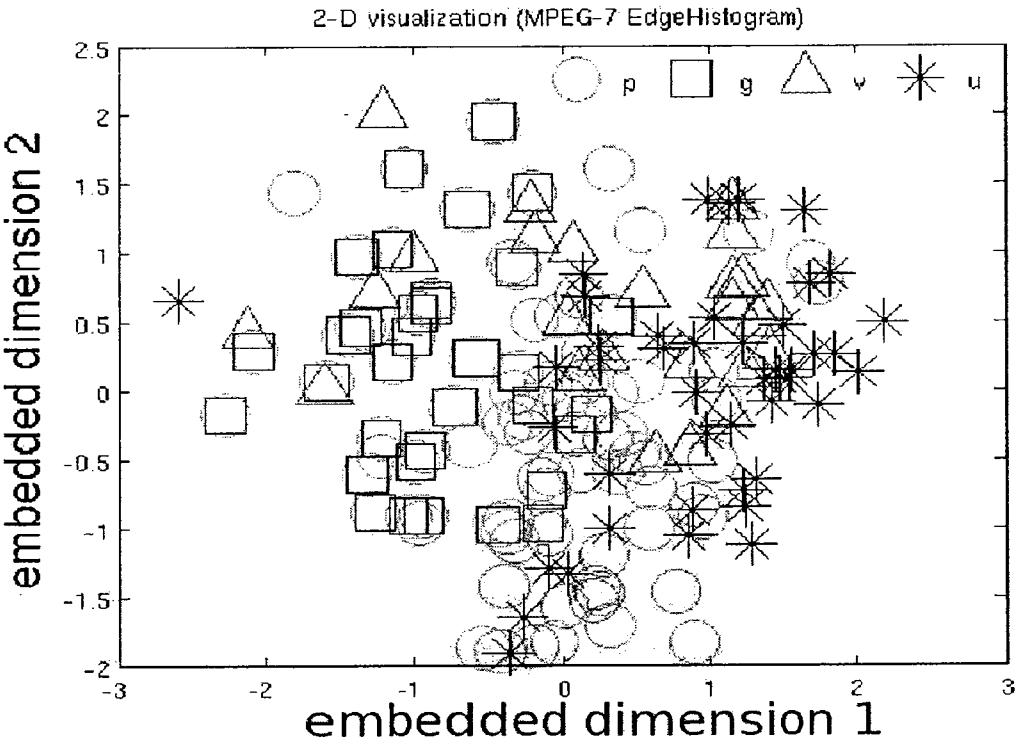


Figure 2.4: 2D representation of edge histogram feature space, overlapping of p with g and v video objects, also u is scattered over p and v.

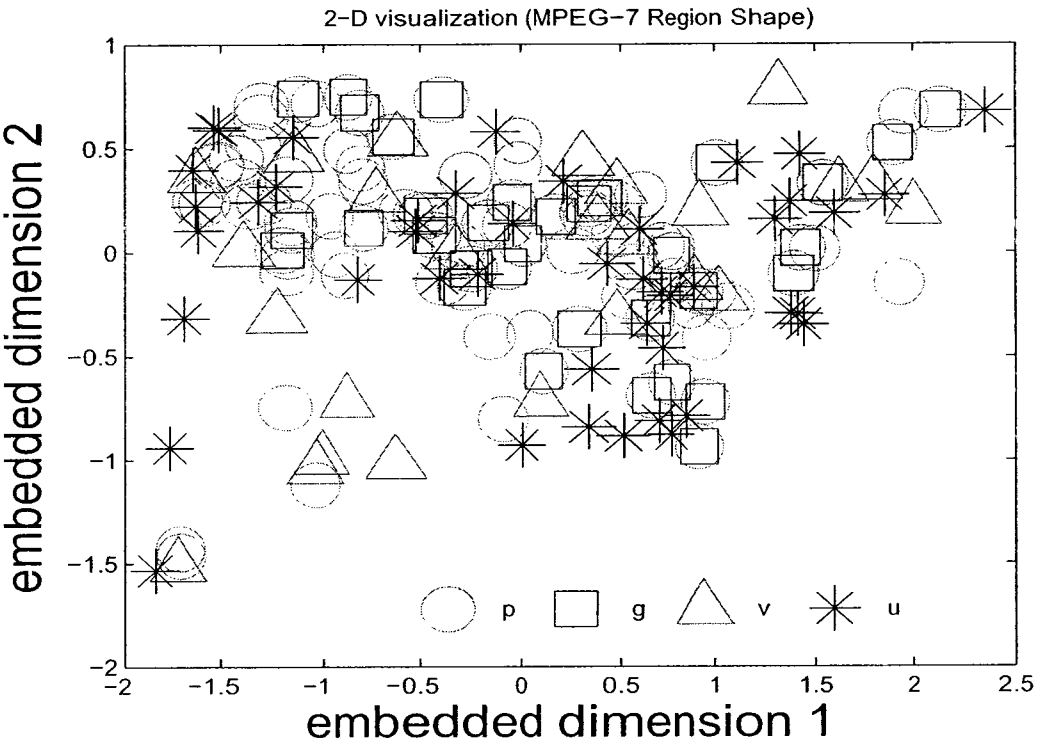


Figure 2.5: 2D representation of region shape feature space, video objects overlapped in different classes.

scattered all over in the feature space, with no dominent clustering (Figs 2.3). Edge histogram descriptor shows overlapping of p with g and v video objects (Figs 2.4), also u is scattered over p and v. In region shape, video objects overlap in different classes, and expose the difficulty to achieve acceptable class separations. Figs. 2.2, 2.3, 2.4, 2.5 demonstrate that using *MPEG-7* visual descriptors directly as feature vector does not ensure generic description of video objects in different classes.

During the *MPEG-7* design process the co-relation among different visual descriptors were not studied. The design goal for each descriptor was to improve retrieval index in corresponding media databases [9]. *MPEG-7* feature coefficients in visual descriptors need to be, either refined for domain specific application, or new visual descriptors need to be added for the same [10]. The low-level features exposed by the existing *MPEG-7* visual descriptors are standard (see Section 3.7) to describe an object. However, using *MPEG-7* visual descriptors, directly as feature vector, usually works for simple object description [41], and does not ensure discriminancy in diverse video objects in different classes. The above question can be verified from Figs. 2.2, 2.3, 2.4, and 2.5.

To improve *MPEG-7* feature vector discrimination for video objects, we take the approach to group a set of feature coefficients inherent to a video object class pattern. Ideal case would be if *MPEG-7* visual descriptor based feature vector, can be processed similar to some functional simulations of human brain activity.

2.4 Chaotic Series

A chaotic series can be generated from non-linear one dimensional finite-difference or ordinary differential equation. Chaos is referred as non-random, irregular fluctuations of parameter values over time in a dynamical system which can be described by de-

terministic equations. Such a deterministic equation can be a simple finite-difference equation,

$$S_{t+1} = f(S_t) \tag{2.1}$$

,which relates values of a system at discrete times. S_t is called the state of the system at time t, $f(S_t)$ is the function that defines the change of state from time t, S_{t+1} is the state at time $t+1$. This equation shows how the state changes in time, i.e., the dynamics of the system.

Many interesting phenomena can arise in the local stability analysis of this finite-difference equation. If $f(S_t)$ is a linear function of R, the finite-difference equation can be

$$S_{t+1} = RS_t \tag{2.2}$$

,A graph of S_{t+1} versus S_t is a straight line, with a slope of R. The solution to (2.2) is a sequence of states, $S_1, S_2, S_3, ...$, that satisfies (2.2) for each value of t. That is the solution satisfies $S_2 = RS_1$, $S_3 = RS_2$, and so on. One way to find a solution to this equation is by the process of iteration. Given the value of S_0 at in the initial time $t = 0$, S_1 value can be calculated. Then the value of S_1 can be substituted in (2.2) to calculate S_2 and so on. The state S_0 is called the initial condition. Following the same pattern we can write, $S_1 = RS_0, S_2 = RS_1 = R^2S_0, S_3 = RS_2 = R^2S_1 = R^3S_0$ and so on. Eventually (2.2) can be written as,

$$S_t = R^tS_0 \tag{2.3}$$

,(2.2) can produce different solutions depending on the value of the parameter R. As shown in Fig. 2.6, the behavior of (2.2) will be exponential decay when $0 < R < 1$, exponential growth when $R > 1$, steady state when $R = 1$, alternating decay when $-1 < R < 0$. alternating growth when $R < -1$ and periodic cycle when $R = -1$.

The linear parameter R in (2.2) can be replaced with a non-linear part $(R - bS_t)$,

$$S_t = (R - bS_t)S_t = RS_t - bS_t^2 \tag{2.4}$$

,where R is the growth rate in the dynamics of the system when S_t is very very small and b is a positive number that controls how the growth rate decreases as S_t increases. In (2.4), two parameter R and b can vary independently. Now replacing $x_t = \frac{bS_t}{R}$, (scaling of (2.4) by $\frac{b}{R}$) in (2.4), and replacing x_t and x_{t+1}, we get the quadratic map equation,

$$x_{t+1} = Rx_t(1 - x_t) \tag{2.5}$$

(2.5), is also known as Logistic map. Fig. 2.7 shows that, choosing different range of value for R, reveals different behavior by this equation. For *R=1.5* solutions to (2.5) shows a steady state system, for *R=2.9* the system approaches to a steady state, and can also alternate. (2.5) can also have periodic cycles, for *R=3.3*, it has periodic cycle of duration *2*, for *R=3.52*, the cycle increases to length *4*. Setting *R=4*, the solution to (2.5) can oscillate in aperiodic manner. This oscillation is irregular, neither exponential growth or decay, nor a steady state. This behavior is called *chaos*, and (2.5) is a *chaotic series.*

A *steady state* is a state of the system that remains fixed, when $x_{t+1} = x_t$. Steady state is associated with the concept of *fixed point.* A fixed point of a function $f(x_t)$ is a value x_t^* that satisfies $x_t^* = f(x_t^*)$. In linear finite-difference equation there can be only one fixed point (in (2.3) at $x_t = 0$ when *R=1*). Non-linear finite-difference equations can have more than one fixed point (in (2.5) when *R=2.9* or *R=3.52*). Fixed points in the Logistic map (2.5) can be found from the roots of the quadratic equation, $x_t = Rx_t(1 - x_t)$ or $x_t(R - Rx_t - 1) = 0$, so the roots are, $x_t = 0$ and $x_t = \frac{R-1}{R}$. In non-linear finite-difference equation fixed points may exists in periodic cycles. Now

we look at some of the basic definitions and properties relevant to a chaotic series.

2.4.1 Stability Of Fixed Point

A fixed point is stable if nearby points approach the fixed point under iteration. A fixed point is unstable if nearby points leave the neighborhood of the fixed point. Let m is the slope of the curve of the non-linear finite-difference equation (2.5) at a fixed point. The fixed point is: unstable when $1 < m$ (nearby points leave the fixed point with exponential growth); stable when $0 < m < 1$ (nearby points monotonically approach); stable when $-1 < m < 0$ (nearby points oscillatory approach), and unstable when $m < -1$ (nearby point oscillatory leave).

A fixed point is locally stable if, given an initial condition sufficiently close to the fixed point, subsequent iterates eventually approach the fixed point. If a fixed point is locally stable, then once the state is very near to the fixed point, it will stay near throughout the future. This is also known as locally asymptotic stability. The term asymptotic dynamics refers to the dynamics as time goes to infinity. Before the state reaches the fixed point, it may show different behavior. Behavior before the asymptotic dynamics is called *transient.*

If the fixed point is approached for all initial conditions then the point is globally stable. For linear finite-difference equations, locally stable fixed points are also globally stable regardless of the initial condition. For non-linear finite-difference equations there can be more than one fixed points. When multiple fixed points are present, none of the fixed points can be globally stable. The set of initial conditions that eventually leads to a fixed point is called the *basin of attraction* of that fixed point. Often the basic of attraction for fixed points in non-linear systems can have very complicated geometry (known as *attractor*).

2.4.2 Phase Space

The *phase space* describes the state of dynamic variable with a set of independent linear vectors. There are several possibilities for defining a phase space. A ten-dimensional phase space can be spanned by $x(t), x(t+\tau), ..., x(t+9\tau)$, where τ means a fixed time increment. Every instantaneous state x_t of a system can be represented [42] by a set $x_1, ..., x_p$, which defines a point in a p-dimensional phase space. The sequence of such points defines a curve 'x_t *versus* x_{t+1}' in the phase space. This curve is called a *trajectory*. As time increases, the trajectories either penetrate the entire phase space or they converge to a lower-dimensional subset, called a *attractor*. An attractor is manifested by the tendency of a non-linear finite difference equation under various but delimited conditions to go to a reproducible active state, and stay there. The trajectory is a mathematical description of the sequence of values taken by a state variable in going from an initial or starting condition to an attractor or through a sequence of attractors. Transition from one attractor to another is called a state change or bifurcation. Attractors can be periodic, quasi periodic or chaotic. With increasing time, all trajectories tend to terminate in fixed points. Stable fixed points are static attractors. A standard example is a pendulum that has come to rest after sometime of oscillation due to frictions.

Chaotic attractors are also known as strange attractors [33]. The manifestation of a strange attractor is its activity which appears to be random, but which is deterministic and reproducible if the input and initial conditions can be replicated.

A set of f first order differential equations $X = F_i(x_i, t)$, $(i = 1 \cdots f)$ is called a dynamical system. If time does not appear explicitly in the function F_i, the system is called autonomous. The non-linearity of F_i is a necessary (but not sufficient) condition for the generation of deterministic chaos. Deterministic chaos means that

the behavior of the system is not predictable over longer periods. Often investigations of dynamical systems have been performed in phase space.

2.4.3 Topological Dimensional

In relation to the geometrical topological dimension of the attractor, one can deduce the dimension of a chaotic series. All strange attractors which have been encountered up to now have a fractal dimension. Grassberger and Procaccia [43] made a proposal to compute the correlation dimension D_2 that computes the geometric dimension of the attractor. Assuming that the attractor represents a 2D manifold in the phase space, it is possible to evaluate the dimension. For every point of the attractor, the number of points lying inside a circle are counted (or in the case of a three-dimensional phase space inside a ball) which have a radius of $r_0, 2r_0, 3r_0$, etc. For the 2D manifold equation $N(r_0) = 8, N(2r_0) = 32, N(3r_0) = 72, \cdots$ Now, if a plot of *log(r)* versus *log N(r)* is performed, a straight line is registered. The slope of this line is *m=2.00.* This is exactly the dimension of the attractor. This result never changes, even if a higher dimension phase space is considered.

It can be shown that if the attractor is not a 2D manifold but a simple curve in the phase space, the evaluation of the dimension leads to a value of $D_2 = 1.0$. The number of points lying inside the circle with radius r is $S(r_0) = 8, S(2r_0) = 16, S(3r_0) = 24$, and so on. By plotting *log S(r)* versus *log (r)*, one finds a straight line with slope *m=1.00.* This is exactly the dimension of the simple curve, which we assume to be the attractor. In fact, Grassberger and Procaccia's [43] algorithm counts the number of points lying inside the circle for every point of the attractor, and averages the results. This method is simple, but time-consuming. Grassberg and Procaccia [43] have shown that it gives reliable results, even though the attractor, and has fractal

dimension.

2.4.4 Limit Cycle

Limit cycle is the closed and recurrent trajectory of a dynamical system in the phase space. All trajectories tend to terminate in this cycle, no other closed cycle lies in its neighborhood. Without external drive, the limit cycle corresponds to a periodic stable position of the non-linear system, whose amplitude and frequency are determined by internal parameters of self-sustained oscillations. Stable limit cycles act as periodic attractors. The standard example is the attractor of a *van der Pol* oscillator [44]. Limit cycles regularly occur with driven oscillators. The trajectory can also move on a two dimensional toroidal surface. In that case, two frequencies are present, oscillations around the torus and along the torus (oscillations with two incommensurable frequencies). The trajectory never closes or covers the whole tours. The trajectory on the tours is then known as quasi periodic

2.4.5 Chaotic Dynamics

One of the major properties of a chaotic series is that it is sensitive to initial condition, x_0. In Fig. 2.8, we see that, with iteration length *t=10*, (2.5) exhibits same series for $x_0 = 0.523423$ and $x_0 = 0.523424$. As t increases, two chaotic series drift apart from each other. Both the series, however, stay on the same chaotic attractor (Fig. 2.9).

Firstly, the system is aperiodic meaning the same state is never repeated. Secondly, the system is bounded meaning that, on successive iterations the state stays in a finite range, and does not approach $\longrightarrow \infty$. As long as the initial condition x_0 is in the range $0 < x_0 < 1$, then all the future iterates will also fall in this range. Thirdly, the system is deterministic meaning that there is a definite rule with no random

terms governing the dynamics of the system. For one-dimensional finite difference equations, deterministic means that for each possible value of x_t, there is only single possible value for $x_{t+1} = f(x_t)$. Finally, the system is sensitive to initial conditions, meaning that two initial points which are initially close, drifts apart as time proceeds.

2.4.6 Delay Differential Equations

Other than the above type of finite-difference equation, chaos can be also found in higher order ordinary differential equations. One dimensional ordinary differential equations of the form,

$$\frac{dx}{dt} = f(x) \tag{2.6}$$

,can have fixed points which grows to $\pm\infty$, but they do not show either periodic cycles or chaos. (2.6) can be approximated by a finite-difference equation. A discrete-time variable $x_t \equiv x(t)$ can be defined for this purpose for $t = 0, \triangle, 2\triangle, ...$, and (2.6) can be re-written as,

$$\frac{dx}{dt} = lim_{\triangle \to 0} \frac{x_{t+1} - x_t}{\triangle} \tag{2.7}$$

,where, approximation $\triangle$ can be very small but finite. (2.7) can be approximated as, $\frac{x_{t+1} - x_t}{\triangle} = f(x_t)$,giving,

$$x_{t+1} = f(x_t \triangle + x_t) \tag{2.8}$$

,If $\triangle$ is small enough (i.e., $\triangle \to 0$), the dynamics of (2.8) will be just like the dynamics of (2.6). If $\triangle = 0$, (2.8) becomes $x_{t+1} = x_t$ which is not a good approximation to (2.6). Analysis of (2.8) reveals that fixed points (x^*) in the finite-difference equation occur at $f(x_t) = 0$, which is the same criterion for fixed points in the original equation (2.6). The stability of a fixed point at x^* depends on $\frac{dx_{t+1}}{dx_t}|_{x^*} = \triangle \frac{df}{dx}|_{x^*} + 1$. Whatever the value of $\frac{df}{dx}|_{x^*}$, by making $\triangle$ small enough $\frac{dx_{t+1}}{dx_t}$ can be made very close to *1*. This means that for $\triangle$ small, (2.8) shows a monotonic approach to or departure

from the fixed point, never the alternating approach or departure for cases where $\frac{dx_{t+1}}{dx_t}|_{x^*} < 0$, as is available in non-linear finite difference equation (2.5) (see Section 2.4.1). A fixed point at x^* is stable when $\frac{df}{dx}|_{x^*} < 0$, and unstable when $\frac{df}{dx}|_{x^*} > 0$, reflecting whether $\frac{dx_{t+1}}{dx_t}|_{x^*}$, is greater than or less than zero. This criterion for the stability of fixed points in the finite-difference approximation (2.8) is the same as in the original differential equation (2.6). Cycles of period *2* can be found in (2.8) for cases where $x_{t+2} = x_t$ and $x_{t+1} \neq x_t$, but,

$$x_{t+2} = \triangle f(x_{t+1}) + x_{t+1} = \triangle f(x_{t+1}) + \triangle f(x_t) + x_t \tag{2.9}$$

,In (2.9), as $\triangle \to 0$ the only points that satisfy $x_{t+2} = x_t$ also satisfy $x_{t+1} = x_t$. So, there are no periodic cycles of length *2* in the dynamics of (2.8), only fixed points exist. The same is true for any cycle of period $n > 1$.

In one-dimensional ordinary differential equations the rate of change of a variable x depends on its present value. However in some cases it is reasonable to assume that the rate of change of a variable depends not only on its value at the present time, but also to its value at some time in past. An example is a variable x, that decays exponentially but is produced at a rate of x's past value at time ℓ [33]. Such dynamics gives a *delay differential* equation [33],

$$\frac{dx}{dt} = \chi(x(t-\ell)) - \alpha x(t), \tag{2.10}$$

,Here function χ is a monotonically decreasing sigmoid function that controls x, and α is a decay constant. If function χ is substituted by a single-hump function, the dynamics generate fixed points, and chaos. (2.10) can then be re-written [33] as,

$$\frac{dx}{dt} = \frac{0.2x(t-20)}{1+x(t-20)^{10}} - 0.1x(t) \tag{2.11}$$

,This equation is known as Mackey-Glass equation. Fig. 2.10 and Fig. 2.11 show the dynamics of (2.11) for iteration length t=100, for two initial conditions apart by 10^{-6}. However, for both the initial condition the chaotic series and attractor shows no drift. For $0 < t < 100$ (Fig. 2.10), two close (apart by 10^{-6}) initial conditions show same series. In Fig. 2.12, for higher iteration (after t=*1000*) the two series (also, corresponding attractors in Fig. 2.13) start to drift apart.

The chaotic dynamics is possible in delay differential equations, but not in one-dimensional differential equations without delays. The reason for this involves the state of a delay differential equation. Whereas, the state at time t_0 of a non-delay, one-dimensional differential equation is a single number $x(t_0)$, the state of a delay-differential equation at time t_0 is given by a function $x(t_0 - s)$ for s in the range of 0 to ℓ. Thus the state of a delay-differential equation is infinite-dimensional.

Ordinary differential equations with either a pair of variable and their first derivatives, or a single variable and its first and second derivatives are called second-order or two dimensional ordinary differential equations. Chaotic dynamics can be possible in higher order or more than one dimensional ordinary differential equations. However, we limit our discussion on chaos theory to equations involving single variable only.

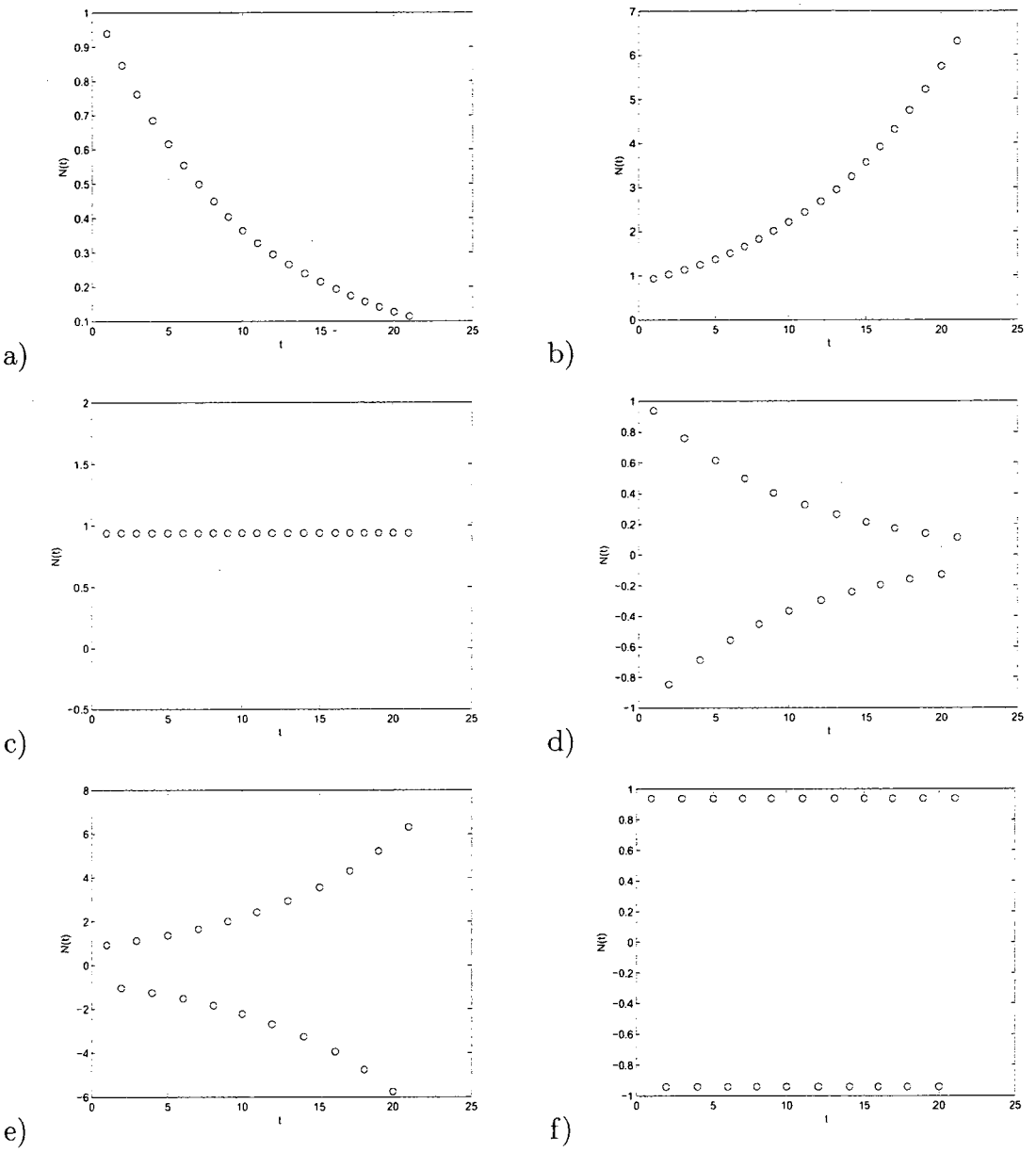


Figure 2.6: Behavior of linear equation for different R, a) decay with R=0.9, b) growth with R=1.1, c) steady state with R=1, d) alternating decay with R=-0.9, e) alternating growth with R=-1.1, and f) periodic cycle of 2 with R=-1.

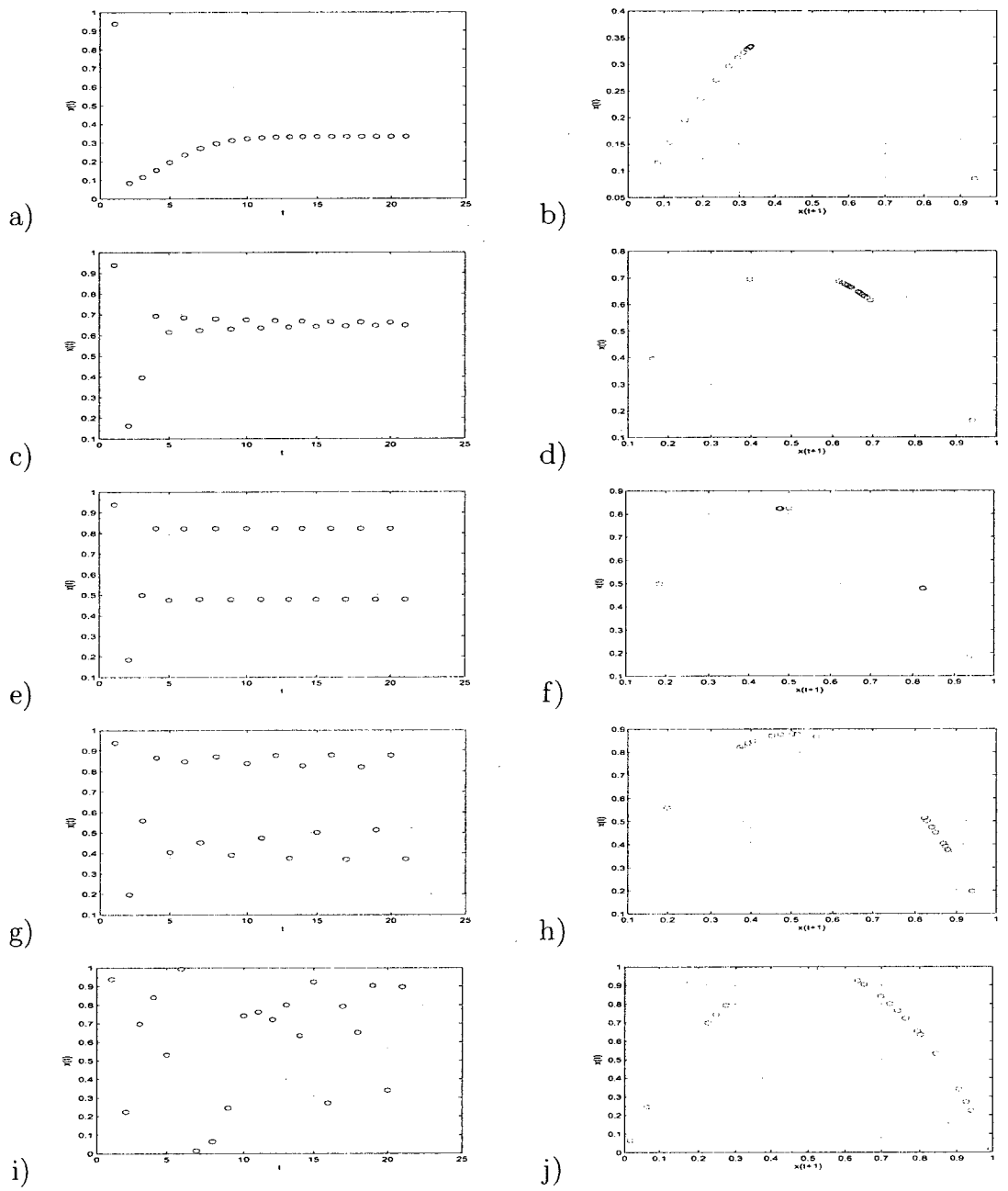


Figure 2.7: Behavior of non-linear equation for different R, a) monotonic steady state with R=1.5, b) state dynamics with R=1.5, c) alternate steady state with R=2.9, d) state dynamics with R=2.9, e) periodic cycles with R=3.3, f) state dynamics with R=3.3, g) periodic cycles with R=3.52, h) state dynamics with R=3.52, i) aperiodic cycles with R=4, and j) state dynamics with R=4.

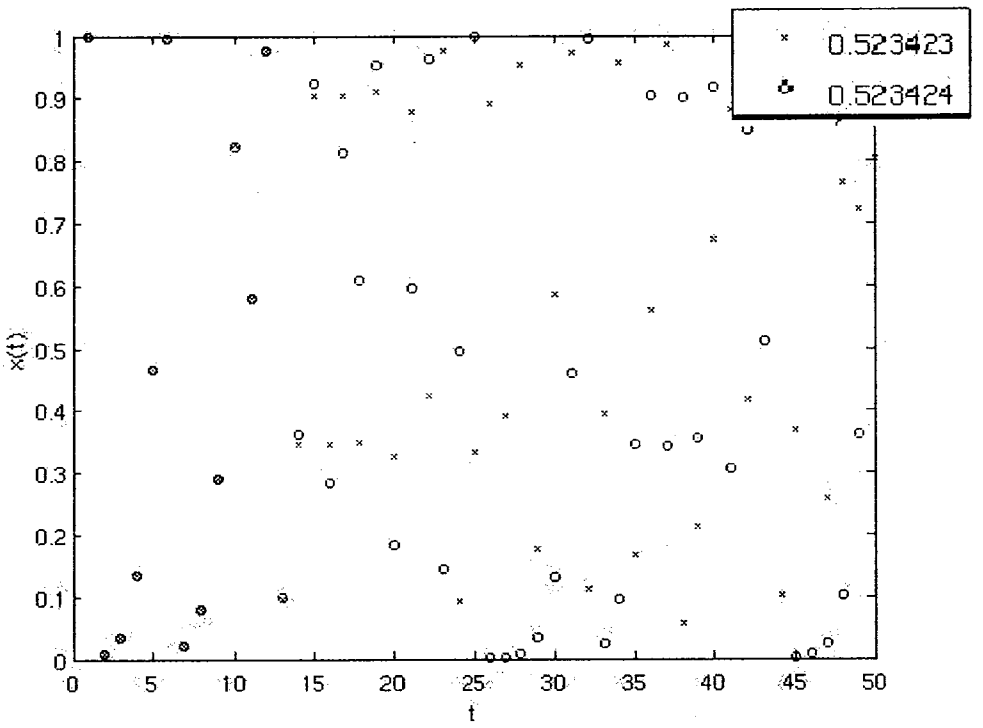


Figure 2.8: Two initial conditions x_0 for two low-dimensional chaotic series, where iteration length *t=50*. Upto iteration length *t=10*, both seeds $x_0 = 0.523423$ and $x_0 = 0.523424$ exhibits same series. As t increases, two chaotic series drift apart from each other.

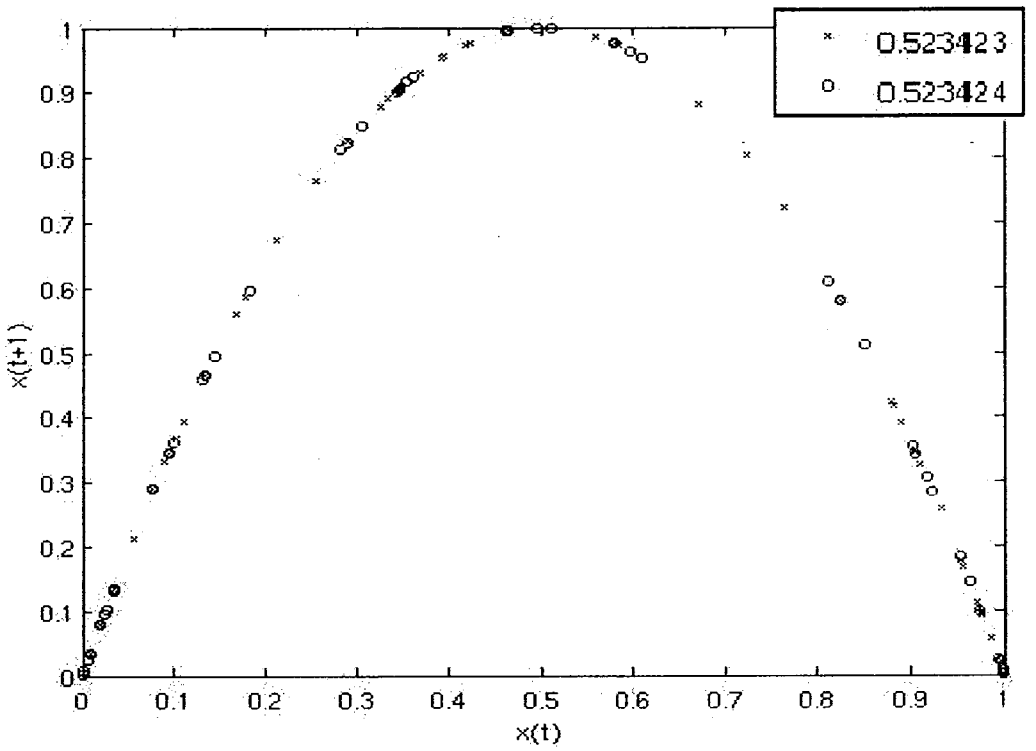


Figure 2.9: Low-dimensional chaotic attractors with seeds $x_0 = 0.523423$ and $x_0 = 0.523424$ where iteration length *t=50*. Both chaotic series, stay on same chaotic attractors.

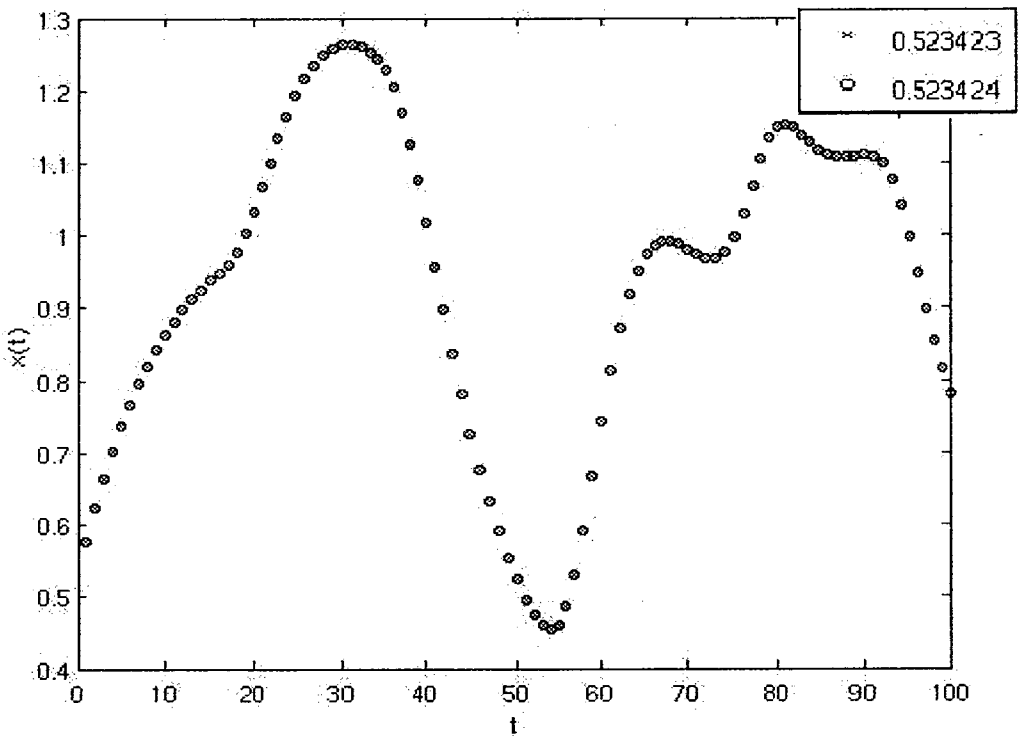


Figure 2.10: Two initial conditions x_0 apart by 10^{-6} for two high-dimensional chaotic series, where iteration length *t=100*. Both chaotic series exactly match, and show no drifts.

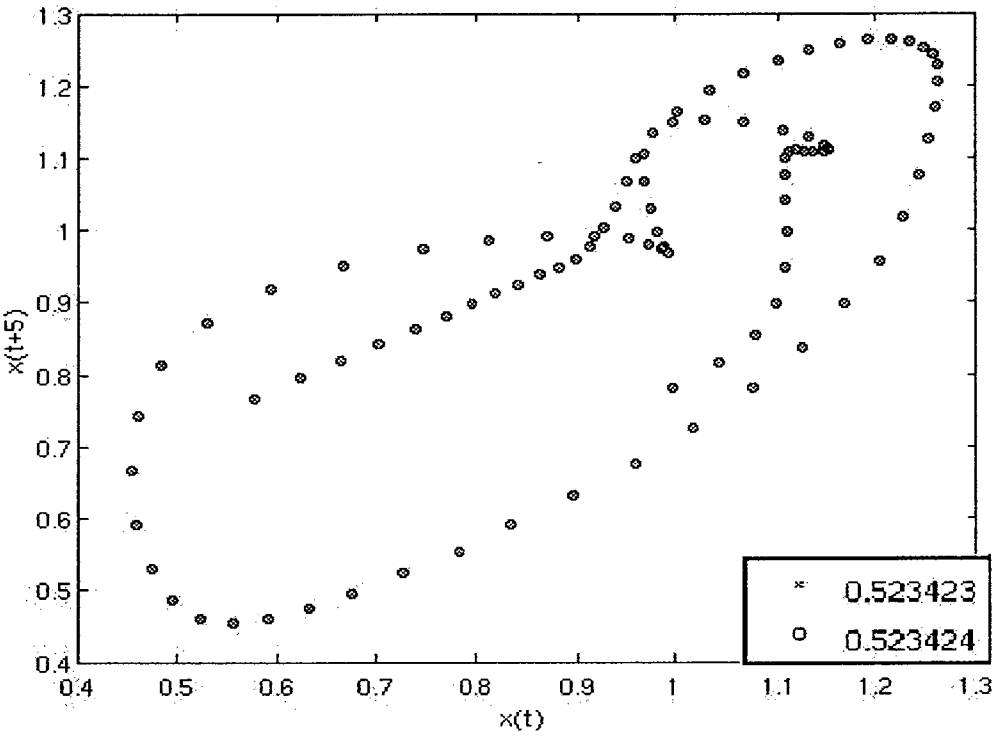


Figure 2.11: High-dimensional chaotic attractors with initial conditions $x_0 = 0.523423$ and $x_0 = 0.523424$ for iteration length *t=100.* Also, both chaotic series exactly match and stay on same attractors. The underlying chaos is hard to detect, and requires larger iteration length t.

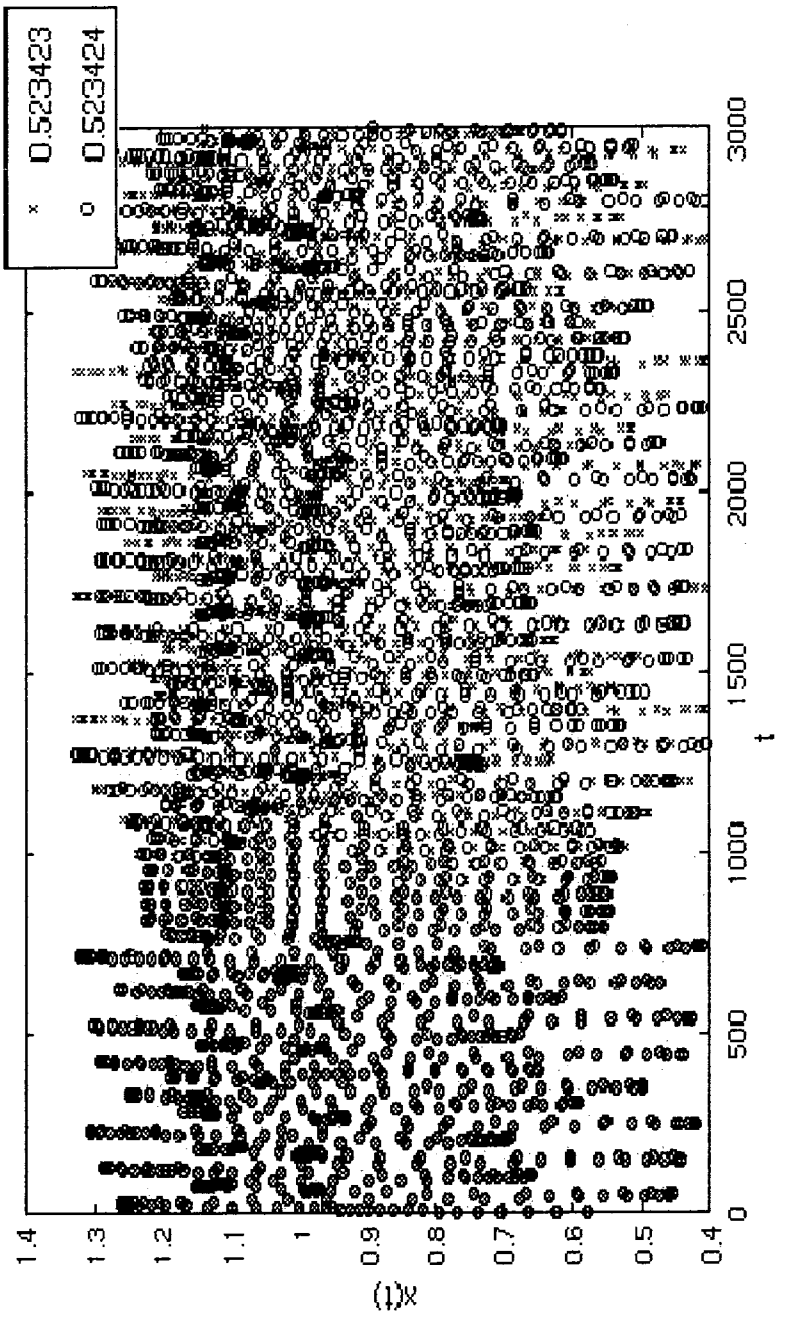


Figure 2.12: Two high-dimensional chaotic series with large iteration length *t=3000*. For initial conditions x_0 apart by 10^{-6}. Significant drift between two corresponding series is observed after *t=1000*

.

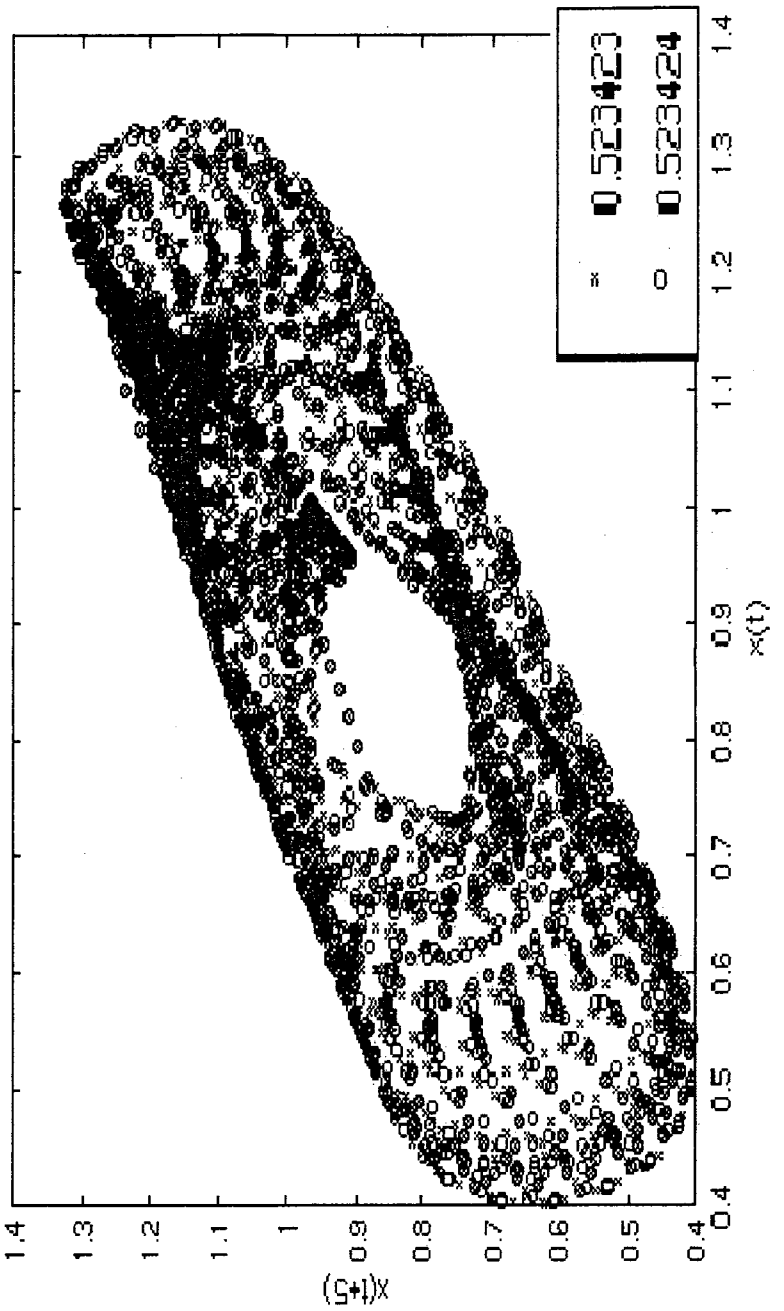


Figure 2.13: High-dimensional chaotic attractors with large iteration length *t=3000*. For two initial conditions x_0 apart by 10^{-6}, the underlying chaos is exposed at high iteration length. The attractors are 'similar' but 'not the same'.

2.5 Summary

This chapter presented a brief background on four major areas related to our work. We summarize the basics on each area as follow,

- *MPEG-7* visual descriptors can be used to generate description of video contents. The information about the contents can be expressed in XML format. However, there exists feature redundancy in each visual descriptors.

- Human understandable description in a video can be the meaning of its contents, i.e., objects, concepts, and events. This description can be generated in the object classification and in the video interpretation module of the video system. Fig. 1.1 shows that an object-based class-labels from object classification can be useful to form concept- or event-based description.

- The basic properties of a chaotic series is presented. The discussion is based on a non-linear finite-difference equation and a delay-differential equation. The formation of chaotic attractors from the stability analysis of fixed points shows the existence of aperiodic states in a chaotic series.

Chapter 3

Related Work

This chapter discusses relevant research on the use of chaos theory to simulate brain functions, on the use of *MPEG-7* visual descriptors in video content description and on video object classification.

3.1 Introduction

Related work is described in different categories. First, we mention relevant neuroscience experiments which report the existence of chaotic attractors in brain function simulations (Section 3.2), and we include major work on pattern classification which use non-linear dynamics of chaos theory (Section 3.3). Later, we present content description work with MPEG-7 features (Section 3.4), non-*MPEG-7* features (Section 3.5), and work on the use of *MPEG-7* visual descriptors to describe contents in surveillance video (Section 3.6). Finally, we present work on video object classification (Section 3.7).

3.2 Chaos In Brain Function Simulation

The human brain is the center of the human nervous system. The brain structure contains millions of interconnected neurons expanded in different hierarchical segments in the brain. Most of the expansion comes from the cerebral cortex, a convoluted layer of neural tissue that covers the surface of the forebrain. The cerebral cortex is the dominant structure in the brain. The cerebral cortex is essentially a 2D sheet of neural tissue, folded in a way that allows a large surface area to fit within the confines of the skull. The electroencephalography, i.e., EEG signals [11] of neurons, in human brain, measures mass changes in neuronal population synaptic activity from the cerebral cortex, and can detect changes over large areas of the brain. The visual features of any real world object get projected in the retina, and then is conveyed to the primary visual cortex, i.e., V1 area in the cerebral cortex. Subsequently, the visual features cause the neurons there to behave as a combination of excitory and inhibitory nodes. V1 is also known as striate cortex, because it can be identified by a large stripe of myelin. Myelin is an electrically-insulating dielectric material that forms a layer, the myelin sheath, surrounding only the axons of many neurons. Axon is the signal-transmitting nerve fiber in a neuron. The myelin sheath is essential for proper functioning of the nervous system. Visually driven regions outside V1 are called extrastriate cortex. There are many extrastriate regions, and these are specialized for different visual tasks, such as visuospatial processing, color discrimination, and motion perception. V1 is highly specialized for processing information about static and moving objects, and is excellent in pattern recognition.

From V1, after being passed from layer to layer through several sets of synaptically connected cells, the information from visual features is sent to several neighboring higher visual areas in the brain. Each of these areas sends its output to several

others. Each of these cortical areas contains three or four synaptic stages. Less is known about how the visual features are processed after the V1, i.e., in other cortical area, visual area 2, or the third area, or the middle temporal (MT) area, to which both the V1 and visual area 2 connect. It is very vaguely known that, in the next two or three areas handle information on color or recognition of complex objects such as faces. After that, for the dozen or so areas which are primarily visual, human still know very almost nothing [45].

However, the brain is a non-linear system par excellence and chaotic dynamics can be used to explain the ability of the brain to generate and process information [11,46]. There has been a established link between the analysis of temporal brain oscillatory signals and chaotic dynamics from the electroencephalography (EEG) signals of neurons [11]. Neuronal excitations in the brain are heavily dependent on chaotic attractors [13]. The chaotic attractor is a non-linear geometrical structure generated by a chaotic series. For some structure and states the brain EEG shows chaotic activity and these states can be an indication to the research in cognitive neuro science. The brain sometimes may as well act noisy tissue. Brain activity, at least, in some states and in some brain parts, appears to be chaotic and not random [12, 46–50].

Babloyantz et al. [51] shows the presence of chaotic attractor (also known as strange attractor) in the EEG behavior during the slow-wave sleep stage in the brain. There work reports analysis of data obtained from a single-variable time series which is independent of any modeling of the brain activity. They show that, from a routine EEG recording, the dynamics of brain activity could be reconstructed. Chaotic attractors could be identified for at least two stages of normal brain activity. In [11], Babloyantz concludes that chaotic property should be related to the ability of the brain to generate and process information.

Freeman and Skarda [13,52] shows that, neurons in cerebral cortex interact synap-

tically by mutual excitation, mutual inhibition, and negative feedback. Typically, the negative feedback connections are locally dense, leading to the formation of local oscillations. The neurons are interconnected by mutually excitory connections over large cortical areas. An appropriate model of cortex is a sheet of distributed coupled oscillators, observation is performed with arrays of surface EEG electrodes. The dynamics of such systems are shaped by tendencies under perturbation to converge to stable states that are identified with attractors of three kinds. An equilibrium attractor is manifested in cortex by a steady state under deep anesthesia; a limit cycle attractor is manifested by regular oscillation, and a strange attractor is manifested by chaos.

Pashaie et al. in [46] proposes a model seeking to emulate the way cortical patches process information and interact with sub cortical areas in visual cortex. The capabilities of such networks in producing sparse codes and computational maps are reported to produce higher level brain functions. in In [46], a macroscopic approach is developed that incorporates salient attributes of the cortex based on combining tools of nonlinear dynamics, information theory, and the known organizational and anatomical features of cortex. Instead of usual simple processing elements such as sigmoidal neurons, Pashaie et al. use nonlinear quadratic maps (i.e., logistic maps) as processing elements. They explored the rich dynamics of quadratic maps that includes fixed-point attractors as well as periodic and chaotic behavior.

3.3 Chaos In Pattern Classification

Chaotic series-based approaches have been studied image compression and communications [26,53–56], image encryption [27,57–61] and image segmentation [62–66]. To our best knowledge, this thesis is the first to use chaotic series to create feature vector

for video object description. However, chaotic series properties is used for pattern classification in EEG, speech and image signal processing [25,67–75]. Relevant major pattern classification work are mentioned below.

[71] proposes a novel classification technique that is used to classify normal and abnormal (epileptic) brain activities through quantitative analysis of electroencephalogram (EEG) recordings. The technique can detecting seizure precursors and correctly classify normal and abnormal EEGs in with a sensitivity of 81.29% and a specificity of 72.86%, on average, across ten patients. k-Nearest Neighbor, kNN is used, where, k= 3 to 13, only odd numbers. However, [71] realizes that kNN is very sensitive to the choice of similarity measure and the EEG characteristics are much more complicated than Euclidean distance can capture. The result is compared in receiver operating characteristics (ROCs) with three similarity measures for EEG time series analysis, Euclidean, T-Statistical and Dynamic Time Wrapping distance, where T-Statistical distance provide best kNN classification performance. Epileptic seizure detection from recorded brain signals also has been reported in [25, 72].

[68] shows that the geometric structures of reconstructed chaotic trajectories contain useful information for nonlinear signal classification and proposes a framework as a feature extraction means for nonlinear signal classification. It uses a set of Poincare surfaces to cut the trajectory that is reconstructed from the nonlinear time series of interest by means of state space reconstruction in order that the structural characteristics in different local regions can be highlighted. A Poincare surface is a hyperplane intersecting with a chaotic trajectory at a given position. As a Poincare surface moves from a position to another, different portions of a chaotic structure are highlighted respectively. [68] also uses shape analysis to characterize the geometric structure of the trajectory. The testing is reported on six classes of real world oceanic signals.

[74] uses wavelet to characterize local features for a chaotic neural network (CNN) by exponentially decaying dilation and chaotically varying translation. Gauss wavelet is used for the self-feedback of the CNN with the dilation parameter acting as the chaos bifurcation parameter. The self-feedback prevents the network from being trapped in the local minima. Experiments on traveling salesman problem (TSP) suggests that the proposed CNN has a higher average success rate for obtaining globally optimal or near-optimal solutions.

[70] proposes a chaotic model of inverse pattern recognition, i.e., to model a clear image object perception which is blurred in certain contexts, when not garbled by with noise by perceiver himself. [70] shows that even without the inclusion of additional noise, perception of an object can be *'blurred'* if the dynamics of the chaotic system are modified. The experimental validity of the model is demonstrated using images of a numeral data set. A by product of the model of [70] is the theoretical possibility of desynchronization of the periodic behavior of the brain (considered as a chaotic system), rendering the possibility of predicting, controlling, and annulling epileptic behavior.

Theoretical and experimental evidence has been reported for the existence of chaotic phenomena in acoustic signals. [67] studies nonlinear regularities in ship-radiated acoustic signals and proposes a chaotic feature for classification. The time series of interest is embedded into a higher dimensional space by using state space reconstruction. Then, the largest Lyapunov exponents computed at different evolution steps are used to form the feature vector. Lyapunov exponent is a property of chaotic series and can be characterized as a measure of a dynamic system's *'sensitive dependence on initial condition'* [33]. The experimental results show that the chaotic feature is not only effective for classification but also can strengthen the spectra based classification in frequency domain.

[69] reconstructed multidimensional attractors of chaotic series is embedded in to phase space to model and analyze nonlinear dynamics in speech signals. Useful features are derived for classification of short time series of speech sounds (i.e., phoneme). [69] uses Lyapunov exponent, which remains intact by the embedding procedure. [69] concludes that even if the speech production system is not chaotic, nonlinear function approximation models can be useful to characterize the speech signals.

[73] uses chaos theory in steganography which is the science of hiding the very presence of communication by embedding secret signals into cover digital signals, similar to digital watermarking. [73] considers that data hiding within a speech signal distorts the chaotic properties of the original speech signal and uses discriminative features such as Lyapunov exponents and a fraction of false neighbors in the speech signal as chaotic features to detect the existence of a embedded secret signals. Fraction of false neighbors is calculated from neighboring points in a chaotic series which fall apart in increasing embedding dimension in phase space, e.g., 2D to 3D.

3.4 *MPEG-7* Features In Video Content Description

The *MPEG-7* visual descriptors have been used in several video content (i.e., objects, concepts and events) description approaches.

[76] proposes a hierarchical *MPEG-7* metadata model to represent video information. The model consists of two separate hierarchies of metadata. The first hierarchy is a directed acyclic graph, captures the relationship between video segments at the semantic level. The second hierarchy is an object composition graph, holds objects

that represent meaningful content appearing in the video.

[77] mentions a video annotation model *VAnnotator*, designed in a way that allows having multiple views over the same video data, enabling users with different requirements to have the most appropriate interface. The format used to store and exchange the information is *MPEG-7*.

[78] presents the IMB, German abbreviation for Intelligent Multimedia Library, that supports annotation and retrieval of content descriptions with a visual annotation and query creation tool. Annotations in IMB are based on one specific domain *ontology*. An ontology is a formal description of a set of concepts within a domain and type of objects and/or concepts based on their domain specific behavior and relations. The retrieval mechanism supports only exact matches based on existing instances of contents, not supporting any ranking or fuzzy matching. [79] uses *MPEG-7* description of video content's visual features for automatic video summarization.

[80] presents a novel approach to human body posture recognition based on the *MPEG-7* contour-based shape descriptor and projection histogram. A combination of the both is used to recognize the main posture and the view of a human based on the binary object mask obtained by the segmentation process.

The system *Caliph and Emir* in [81, 82] allows manual *MPEG-7* compliant annotation of digital images from visualization of objects by set of relation operators in *MPEG-7* description schemes. A similar work [83], manages movie scene content from a growing and evolving set of description that fits into all users and interpretations for generic transcription. The system is built from collaborative effort (through conflict resolution among similar description for same objects, concepts, and events of user's point of view rather than from low-level *MPEG-7* descriptors.

[84] demonstrates Marvel, which is an *MPEG-7* oriented video retrieval system. Marvel can extract up to 200 different semantic concepts such as *summer*, *winter* from

video streams automatically. VideoAL [18] is another project that consists of seven modules: shot segmentation, region segmentation, annotation, feature extraction, model learning, classification, and *MPEG-7* XML rendering. The tool associates semantic labels with each shot or region using a smaller module VideoAnnEx [85]. A concept-learning module builds models for anchor concepts such as *e.g., outdoors, indoors, sky, snow, car, flag* and so forth. The tool automatically assigns a relevance score to the video according to the confidence value of the classification. The Marvel and VideoAL both focuses on the extraction of high level metadata.

In summary, the existing video description works, which have integrated the *MPEG-7* standard, commonly, generate primary content description in a video (i.e., objects), define the structural relationships of the primary contents in the scene, and then use spatio-temporal sequence analysis (independent of *MPEG-7* standard) to label pre-defined secondary video contents (i.e., concepts or events). These works concentrate on the *MPEG-7* structural tools, and lack a functional simulation of human visual system to minimize the semantic gap.

3.5 Non-*MPEG-7* Features In Video Content Description

The review in [8, 14] summarizes non-*MPEG-7* video content (i.e., objects, concepts, and events) description generation approaches, for context-dependent shots. The reviewed approaches mostly use non-*MPEG-7* low-level features as well as human understandable context information (e.g., *restricted zone areas in parking lot*) for content descriptions. The descriptions do not follow any *MPEG-7* visual descriptor or description schema, and are not *MPEG-7* compliant.

What is happening in a video shot can be addressed by the meaning and behavior of a finite set of objects (tracked in the video analysis stage) in a sequence of a small number of consecutive frames [35, 86]. The common practice in description generation broadly corresponds to either the description of specific objects (also, concepts and events) or the description of generic objects (also, concepts and events). The generic description problem can be considered as the classification problem of the time-varying low-level features of objects.

Probabilistic frameworks such as Hidden Markov Models (HMM) [87], Bayesian Network, and Neural Network (NN) are popular methods in spatio-temporal sequence expression to describe video shots [8, 14] in any video shot (including the surveillance video shots). Most of the demonstration are done with specific identification in a shot. HMMs are robust against various temporal segmentation of events. But, the structure and probability distributions in HMM-based approaches are not transparent, and need to be learned using iterative methods [8, 14]. The computation of parameters for a given description structure of a model is expansive. To fit a new problem the structure of the model must be changed, and the learning for variable structure is difficult [88]. Also for complex event descriptions, the network, and the parameter space in the these statistical approaches may become prohibitively large [86].

Other approaches for description of specific events include Curve representation [89], Finite state machine [90, 91], Scene transition graphs(STGs) [92, 93]. In the graphical models mentioned above large amounts of training data are usually required to obtain good models of various actions in the spatio-temporal domain. The training data are usually limited for specific objects, concepts, and events in surveillance shot. Trajectories are also used for scene interpretation [86, 94–97]. It is often not clear over what temporal scale should object behavior be defined without any additional context knowledge of the underlying object behavior. Trajectory-based models also

rely heavily on the assumption of accurate object segmentation and tracking.

Rule-based description approaches [35,98,99] shows their merits for identification for primitive objects, concepts, and events. Even then, rule-based systems may have difficulties in defining precise rules for every possible behavior of a content for a generic description of the content. Usually the rules are defined through numerous feature thresholds. Also, rule-based description approaches depend heavily on object tracking.

[1,100], propose a human visual system-based architecture which uses low-level non-*MPEG-7* features to define the relationships between several processing modules to mimic human-like perception with a fixed camera to generate XML-based description of human behavior in public scenes, such as in streets or in shopping centers. [1,100] addresses the challenges such as data flow regulation, process scheduling and managing situations of high data rates between many processing modules (e.g., image processing modules. tracking modules, behavior modules) which need more resources than a standard surveillance system can provide. They propose two models CAVIAR (with a central controller module) and Psyclone (with distributed controller modules) for fully autonomous configuration of processing and data flow. While CAVIAR has the overhead of governing many layers of data modules to decide *what is happening* in the scene with a one global controller that knows everything about every module, the problem with the distributed controls of Psyclone is that, no single controller knows everything and it is therefore easy to lose sight of what is actually going on globally when trying to mean things locally. Again in addition to these limitations, all the modules in both the approaches concentrate on the structural relationship of features. The semantic relationship is dealt in possible multiple roles of the tracked objects in the context by considering static (CAVIAR) or dynamic (Psyclone) priority in the structural features (e.g., parameter tuning and

module swapping from neural network decisions or pre-defined rules). The semantic gap is addressed at the end-part of their model when it may have been addressed at the initial - part of model. [1,100], with non-*MPEG-7* feature-based architecture, focus on multiple module integration for multiple semantic view of video contents. Multiple views of content do not necessarily expose low-level feature discrimination of contents to minimize the semantic gap between low-level features and human understandable description. However, multiple views of content, provide shared view of contents which can be useful in ontology-based content annotation. Human visual system-based approaches [1, 100], have not yet reported the use of *MPEG-7* visual descriptors to generate video content descriptions.

3.6 *MPEG-7* Features In Video Surveillance

There have been reports on the use of *MPEG-7* standard for automated annotation in sports video [101–104]. Sports applications focus descriptions of domain specific video contents, *players, referees, goal, penalty, home-run.* Surveillance video content descriptions are different from other video domain contents. Examples of such contents are, *person, vehicle, erratic movement* (where an object moves in indirect way from the camera view), *crowed dispersal* (where objects disperse from crowed), *vandalism, robbery.* Few surveillance works [19,105,106] have reported use of *MPEG-7* descriptors to describe limited context-dependent (see Section 2.2) contents. These approaches generally map the pixel-level video features to *MPEG-7* visual descriptors (e.g., shape, color layout, motion activity). The structural relations among these descriptors (along with additional context information) are then statistically mapped to describe domain specific video contents (objects, concepts, events).

The first use of *MPEG-7* visual descriptors in surveillance system is done in [107],

which mainly focuses on automatic segmentation of shots using non-*MPEG-7* temporal and spatial features. The extraction of *MPEG-7* descriptors is done on segmented video objects to represent the objects in a 2D to 3D coordinate transformation. However, [107] proposes non-*MPEG-7* XML schema for content description.

[108] demonstrates quick content identification and retrieval from surveillance shots, and provides *MPEG-7* tags of specific shots based on MEPG-7 description schemes.

Content description initiative [105] performs video plane object extraction, and generates *MPEG-7* complaint shape descriptors, and then use directed acyclic graphs to generate linear ordered description of objects in an *MPEG-4/7* hybrid system.

To describe surveillance video contents, [19] extracts moving objects automatically by mean of video analysis, and then encoded objects in MPEG-4. [19] uses feedback between an object partition and a region partition (of the frame) to track segmented objects in successive frames. The region partition defines homogeneous groups of pixels corresponding to perceptually uniform regions, whereas, the object partition defines video objects which do not usually have invariant physical properties. The feedback also deals with multiple deformed objects, e.g., occlusion, splitting, appearance, and disappearance of objects, and complex motion. For adaptive video delivery to different media networks, the *MPEG-7* descriptors are used as meta data for different video object priorities. Later the *MPEG-7* descriptors are used to create *MPEG-7* description schemes for manual event description (e.g., *who. what*). In the absence of any generic module the *MPEG-7* descriptors only provide description of the segmented object rather than the nature (e.g., *car, person*) of the object.

[106] decomposes a surveillance video into objects of semantic relevance and propose efficient encoding scheme for transmission delivery and visualization (with *MPEG-7* descriptions) for narrow channels and devices. [106] focuses more to content

organization and navigation rather than content description.

[109] uses a layer for *MPEG-7* color descriptors of moving objects in indoor shots, and another layer to define the relationships between the low-level color descriptor ids and unique object ids with descriptions. A graph-based probabilistic module is used to describe the relationships. The XML schema info (e.g., object id, object name, graph of low-level descriptor id references) is available which is not compliant with *MPEG-7* XML schema. The XML schema description in [109] is appropriate to specific object description (e.g., *John Smith*) than a generic object description (e.g., *person*).

3.7 Video Object Classification In Surveillance

Classical approach for content interpretation is to match a pre-compiled template of labeled content descriptions that represent generic behavior that need to be learned by the surveillance processing framework via training sequences [14]. One of the challenge here is the reluctant need to label sparse behavior of large data sets for better classification to interpret complex objects, concepts, and events [110]. This reluctance is due to the unpredictable changes in patterns or behavior of contents in a shot, and also due to the lack of perfect segmentation and tracking algorithms. Examples are: erratic movement (where an object moves in indirect way from the camera view), crowed dispersal (where objects disperse from crowed), vandalism, robbery. Consequently, the major challenge is to generate generic descriptions to reduce the complexity of manually labeling diverse data sets. A generic description generation with effective video object classification can reduce this complexity.

Object classification in surveillance follow two major approaches: *shot-based* and *object-based*. Popular frameworks such as hidden Markov models, Bayesian network,

neural network, k-nearest neighbor (k-NN), support vector machine (SVM), finite state machine, scene transition graphs, can be used as classifier in either of the above approaches.

3.7.1 Shot-based

The *first* approach extracts either, features from objects throughout a shot, or, features from a whole frame throughout a shot. This approach can classify objects [111] or identify a shot with specific concepts [112, 113], such as, *music* or *sports* Features can be both, i) spatial features aggregated over temporal scale, such as, average size, and ii) temporal features, such as, motion.

In [111], objects are classified into 49 concepts as global (e.g., outdoors, indoors) and local (e.g., face, people, car, greenery). The ground truth shots in training are binary ground truth, i.e., either the concept is present or not in a shot. Non-*MPEG-7* features, e.g., color histogram, bounding box height and width, motion vector histogram are extracted from set of frames. SVM classifiers are used with heuristic combination of these features. The reported precision is poor (near 0.3) for surveillance related objects, e.g., *'face'* and *'people'*.

In [112], *MPEG-7* descriptors, i.e., scalable color, color layout, and parametric motion of the whole frame in each shot are used. To group sets of similar shots, [112] employs hierarchical intra-class clustering for specific context specific features. With kNN, they report 70.0% classification accuracy.

In [113], 593 dimensional *MPEG-7* descriptor for each frame, is first reduced and then used to classify shots. The descriptor space for the training shots is partitioned in different small feature spaces to estimate apriori probability. Surveillance shots often contain incomplete video objects in subsequent frames, partitioning the descriptor

space may not be an intuitive approach there. Shot classification accuracy of 93% with Bayesian network, 90% with k-NN, 85% with decision tree, 70% with SVM, and 50% with neural network is achieved. *Basketball*, *education*, *music*, *news*, *soccer*, *swimming*, *tennis*, *table tennis*, and *volleyball* shots are considered.

In [114], repeated motion of objects in a shot is used to classify *people*, *groups of people*, and *vehicle*. Temporal templates of recurrent motion images are computed by aligning and accumulating objects from a non-*MPEG-7* feature-based object detection module. 88.0% accuracy is reported with a rule-based classifier.

3.7.2 Object-based

The *second* approach uses features from objects of selected individual frames throughout a shot to classify video objects. In this approach, prior to classification, a list of video objects related to selected individual frames are available from video analysis, e.g., segmentation and tracking [97]. Features can be both, i) spatial features aggregated over temporal scale, such as, average size, and ii) temporal features, such as, motion. However, unlike the first approach above, the temporal scale is not throughout the shot and thus do not contain any reference or dependency on the shot (i.e., not scene-dependent). The object features are thus scene-independent and object-dependent. The first approach is preferable to train a classifier when good segmentation is available or when objects are simple. The second approach is preferable to train a classifier when the segmentation is poor or when objects are more challenging.

In [97, 115], objects are classified for arbitrary camera viewpoint to distinguish *humans* from *vehicles*. The system employs classical feature-based classification to initialize view-normalization parameters. [115] measures a wide range objects from

non-*MPEG-7* features based on shape, motion and periodicity. k-Nearest Neighbor classifier with $k=10$ is used. The view independent outdoor shots of *pedestrian* with simple shape and motion (without shadows) are considered. Also, only objects whose tracks are stable, i.e., they are not undergoing merge/split, occlusion, or do not lie on the image border, are used for training or testing.

In [41], the *MPEG-7* region shape descriptor is extracted after background subtraction to classify different types of ships with k-Nearest Neighbor, kNN. To classify *people*, [116] uses pairwise constraint, i.e., pair of examples belong to the same class or not, in a margin-based learning framework. In case of similar objects, the mean color histogram from successive frames is considered while selecting pairwise constrains. Adopted version of different classifiers, i.e., kernel logistic regression (KLR) and SVM, with majority voting are used. The classification error rate varies between 6% to 22%. [116] considers objects which do not have any foreground segments containing two or more people, i.e., no occlusion, partial occlusion, overlapped or partly lost objects. Objects maintain different lighting and orientation but similar scale and resolution. In [117], *MPEG-7* visual descriptors are combined for classifiers, e.g., SVM, Nearest Mean, Bayesian, and k-NN. [117] concludes that, either it is required to use multiple features to represent the class, or it is needed to combine different classifiers to fit a distribution to the members of the class in the selected feature space [117]. Both [118,119] aimed for human detection using multi-class classification. However, the relative contribution of features is not considered because it is difficult to measure effectiveness of each feature.

In [120], a hierarchical multi-level neural network is used to classify objects. With 8 dimensional non-*MPEG-7* features they achieve 98% *car*, 9% *cycle* and 11% *pedestrian* accuracy in *parking lot* shots. In [121], *single human*, *human group*, *vehicle*, *bike* are classified. Features invariant to changes caused by environment, scaling, view-

point and lighting are used. The non-*MPEG-7* features include shape (aspect ratio, dispersedness), texture (edge magnitude and connectivity), motion (optical flow vectors). Variance of optical flow vectors distinguishes *walking human*'s non-rigid motion and the *bike*'s rigid motion. 11 dimensional feature for each class is selected to find a linear combination (with weights) of optimal features for that class with AdaBoost. Then they merge the results of four trained classifiers in a multi-class classification module. The *low* dimensional feature vector makes robust use of AdaBoost. The use of Artificial neural network (ANN) reports 95% accuracy. With ANN it is tough to make a desired combination of feature sets in a problem domain with too many training sample. Again, [121] uses limited data sets which do not have any foreground segments containing two or more people (e.g., no occlusion). The data samples maintain different orientation but similar scale, lighting and resolution.

In [122], image regions in video are classified as *person* or *not person*. They develop three generic approaches to discriminate: ridge-based structural models, ridge-normalized gradient histograms, and auto-associative memories (one layer linear neural network). The approaches improve previous computationally expansive appearance-based methods of object recognition (e.g., [36]) or scale and affine transformation sensitive approaches(e.g., [99]). Ridges represent center lines of an oblong structure, and at several scales capture more information about about objects than silhouettes (used in [99]) or skeletons (used in [123]). In total ridge-based features of 10 dimension are used. In auto-associative memory based approach global appearance of the image region of interest is used to handle very low resolution. With KMeans algorithm, ridge based approaches (robust to illumination changes but disruptive to local changes) report precision of 0.9 for 'people' and 0.7 for 'not people'. In classification based on neural network (sensitive to illumination changes) precision is 0.96 for 'people' and 0.9 for 'not people'. All the approaches are mainly sensitive to ridges

(dominantly available in *CAVIAR* database) and not prune to more challenging objects under occlusion or shadows. Also performance with other object classes is not reported.

In [124], a non-*MPEG-7* descriptor, 31 binary (0 or 1) features in sub-block image pixel, is proposed. This descriptor captures the large-scale structures in object appearances by measuring intensity differences between pixel-level sub-blocks in image patches. An AdaBoost algorithm selects a subset of discriminative features from 31 dimensional descriptor, and construct strong two-class classifier. Multiple classes of *cars, vans, trucks, persons, bikes, people groups* are tested, and on average 84% accuracy is reported.

In [125], non-*MPEG-7* low-level features, such as height, width, ratio, are calculated for video objects in each frame, while other features, such as ave height, var speed, are calculated as averages or variances over a group of frames. Only objects with consistent tracking are used in classification. A leave-1-out cross validation is used to divide the training and test data sets. Poor classification accuracy with scene-independent training is reported. In rule-based classifier the accuracy is 56% for 9 classes, 65% for 6 classes, in k-means classifier it is 77% for 6 classes, and in Adaboost classifier it is 75% for 6-classes. [125] concludes that a scene-independent training can not provide sufficient accuracy. Eventually scene-dependent training reports, 85% and 88% accuracy, with Adaboost and a learning-based approach, respectively. On the contrary to the conclusion by [125], our approach is meant to investigate the classification accuracy from scene-independent training.

[126], proposes a 3D and dynamic scene analysis-based algorithm to classify static video objects, e.g., *buildings*, and moving objects e.g., *vehicles* from combat aerial video. 3D spatio object features e.g., object shape, line orientation, color, and texture, 3D static structures of the urban environment of the scene and dynamic video

features, e.g., vehicle motion patterns over time are used to create spatio-temporal feature vectors. These vectors vectors are then used in a probabilistic graphical model for classification. The field of view panoramic mosaics are generated to preserve both 3D and dynamic information. Classification accuracy of 90% is reported to classify *buildings* and *vehicles* from low-resolution airborne surveillance video.

In [127], entity-relation model is used to relate *MPEG-7* visual descriptors with descriptions of primitive events (e.g., *deposit, appear, walk, run, exit*), from hierarchical conceptual units, i.e., ontology. Interactions between moving objects and special regions are represented as spatial relationships, rules and conceptual units. In [128], moving blobs are detected from adaptive Gaussian Mixer Model per background pixel, followed by pixel grouping. *MPEG-7* complaint scene layout for multiple region classes, e.g., *road, parking, buildings*, are created, and is used to generate hypothesis for object class labels. An exhaustive ontology can be created using community collaboration (e.g., [83]) which allows shared views of the same content. We understand that, ontology based shared views of objects do not necessarily expose the underlying low-level feature discrimination to create an effective feature vector for video object classification.

In brief, irrespective of the choice of classifier, most approaches use non-*MPEG-7* features. Also, high classification accuracy with challenging surveillance video objects, e.g., incomplete objects, partial occlusion, background over lapping, scale, and resolution variant objects, indoor - outdoor lighting variations, are not significant in the above approaches.

3.8 Summary

This chapter reviewed relevant work on chaos theory, MPEG-7 and classification. We understand that,

- Chaotic attractors can be related to the ability of the brain to generate and process information. Chaotic series properties are used for pattern classification in EEG, speech and image signal processing
- There has been limited use of MPEG-7 visual descriptors to represent specific video contents
- In the surveillance video systems where MPEG-7 descriptors are used, high classification accuracy with challenging surveillance video objects, e.g., incomplete objects, partial occlusion, background over lapping, is yet to be reported
- Irrespective of the choice of classifier, surveillance video systems use non-*MPEG-7* features
- None of the approaches use chaotic series-based feature vector for video object description

Chapter 4

MPEG-7 Feature Selection

This chapter provides details on different available *MPEG-7* visual descriptors and the rationale behind selected visual descriptors for this work.

4.1 Introduction

The *MPEG-7* visual descriptors cover the basic categories of visual features: color, texture, shape, motion, localization, and face recognition. Each category consists of elementary and compound descriptors. Nine of these descriptors are elementary visual descriptors, others are compound visual descriptors, and are defined as descriptor aggregation or localization from the elementary ones. The compound descriptors are not actually descriptors in the sense that they do not extract low-level features of media contents.

We do preliminary selection of visual descriptors from findings in MPEG-7 literature (e.g., [9]). Then, we apply *MPEG-7* reference software XM [38] on surveillance data sets (see Section 5.3.1) to verify the robustness of each visual descriptors. The XML description of sample video objects for each class is included in Appendix A.

We end up selecting the sets of *MPEG-7* elementary visual descriptors which offer fast extraction, interoperability, are of compact size, and are scale and resolution invariant.

The *MPEG-7* reference software XM [38] can be used to extract *MPEG-7* visual descriptors. The details on the algorithms of each visual descriptor is available in [9]. Each video object can be feed to the XM as an object image for feature extraction. Eidenberger in [10] tested statistical properties of elementary visual descriptor combinations in monochrome texture-based Brodatz data set, Corel color photo data set, and artificial color images with few color gradations. The best descriptor-combination as reported in [10] are, color layout, dominant color, edge histogram, and texture browsing. Considering our application domain, i.e., object classification in surveillance video, we review each elementary visual descriptor definition. We select the set of visual descriptors which offer fast extraction, interoperability, are of compact size, and are scale and resolution invariant. The selected descriptors are color layout, edge histogram, region shape, and contour shape. Sample XML description for these descriptors in different video objects are shown in Appendix B. The discriminancy among different of video objects are expressed as descriptor coefficients in the XML descriptions. However, this discriminancy in *MPEG-7* visual descriptors has to be verified in a video object classification module to qualify these descriptors to be used as feature vector for generic description of video objects of different classes. In the following sections, we briefly explain our justifications on the *MPEG-7* visual descriptors.

4.2 Dominant Color

Color features become useful in fairly high resolution images, in general, they plays important role in tracking rather than object classification [115]. The dominant color is the most common color descriptor. Its feature coefficients consist of number of representative colors (up to eight), their percentages in the region, spatial coherency of the colors, and color variances for each color. It is useful where a small number of colors are enough to characterize the color information in the region of interest. Main target applications are similarity retrieval in image database and browsing based on single or several color values. Our data sets include a wide range of scale and resolution invariant video objects from indoor and outdoor video shots. By observation of the data sets we find that the number of representative colors and corresponding feature coefficients varies in the dominant color XML description of video objects. Feature vector dimension for a class of training or testing video objects are not same. For this reason we drop dominant color descriptor.

4.3 Scalable Color

This descriptor is a restricted color histogram feature of video objects. The color space is fixed to HSV with a uniform quantization of the color space to 256 bins which includes 16 levels in H, 4 levels in S, and 4 levels in V. For compression, the bin information is stored in the *MPEG-7* stream after a Haar transform encoding step. Haar transform is a one-dimensional transform which makes use of the Haar functions [129]. Interoperability between different resolution levels is retained because of the scaling property of the Haar transform. The Haar coefficients maybe extracted from a 128-, 64-, or 32- bin histogram. thus. the scalable color descriptor does not

guarantee interoperability with other similar *MPEG-7* applications, unless the Haar coefficients are extracted in the same precision in all applications. This descriptor is useful for image-to-image matching and retrieval based on color feature. The high-pass coefficients of the Haar transform express the information contained in finer-resolution levels. Scalable color exhibits high redundancy between adjacent histogram bins. The redundancy can be explained by the impurity (slight variation) of colors caused by variable illumination and shadowing effects. It is not prune to variable illumination and shadowing.

4.4 Color Structure

This descriptor expresses local color structure in an object image using a 8x8 structuring element. The algorithm counts the number of times, a particular color is contained within the structuring element as the structuring element scans the image. The HMMD color space is used in this descriptor. Variable number of color bins are possible: 184, 120, 64, and 32. It is sensitive to certain image features to which the color histogram is blind [9]. Color quantization of an image affects its color structure. The descriptor does not guarantee descriptor interoperability for two different extraction or resizing methods. In contrast to most other descriptors, extraction, and resizing of this descriptor is a normative process within the standard, i.e., major steps are specified by the *MPEG-7* standard. Deviation from these steps risks breaking the interoperability of the descriptor. The main functionality of this descriptor is image-to-image matching.

4.5 Color Layout

This descriptor is content independent, because of the luminance coefficients. Spatial distribution of color of visual signals is contained in this descriptor in a very compact form. The captured feature is represented in frequency domain, so users can easily introduce perceptual sensitivity of human vision system for similarity calculation. The descriptor has very small computational costs, and it offers ultra high-speed image sequence-to-sequence matching. The descriptor has no dependency on image or video formats, resolutions, and bit-depths. Also it is feasible to apply the color layout descriptor to mobile terminal applications. It is the first descriptor selection in the *MPEG-7* feature vector in our work.

4.6 Homogeneous Texture

This descriptor provides a quantitative characterization of texture for similarity-based image-to-image matching. This descriptor is computed by first filtering the object image with a bank of orientation- and scale- sensitive kernels, and then computing the mean and standard deviation of the filtered outputs in the frequency domain. The frequency space is partitioned into 30 channels with equal divisions in the angular direction (at 30 intervals) and octave division in the radial direction (five octaves). The image texture energy as well as its deviation in each of the filtered channels is computed. The descriptor is suitable to represent homogeneous textured pattern when viewed from a distance, such as in airborne images. It performs poor on color image retrieval [9]. In a large extent the algorithm is determined by another visual descriptor, edge histogram (see Section 4.8), and for this reason should not be combined with edge histogram. Feature extraction is not fast in surveillance video data

sets.

4.7 Texture Browsing

This descriptor provides qualitative representation (i.e., perceptual characterization) of the texture, in terms of regularity, coarseness, and directionality. It offers a scalable solution to represent homogeneous texture regions in images. It is defined as regularity, direction 1, scale 1, direction 2, scale 2, where regularity may be *not regular, slightly regular, regular, highly regular*, direction in degree may be, *no direction, 0, 30, 60, 90, 120, 150*, and scale may be *no scale, fine, medium, coarse, very coarse.* The descriptor has poor variance which makes it difficult to use in most applications (results may be ambiguous for retrieval application [9]). However, it is suitable for browsing applications. The combination of mostly textual along with numerical descriptor coefficients makes texture browsing descriptors not suitable to be included as feature vector in our work.

4.8 Edge Histogram

This descriptor captures the spatial distribution of edges in the object image. A given image is first subdivided into 4x4 sub images, and edge histograms for each of these sub images are computed. Edges are broadly grouped into five categories: *vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic.* Thus, each local histogram has five bins corresponding to the above five categories. The image partitioned into 16 sub images thus results in 80 bins. Edges play an important role for image perception in human visual system. Edges represent images with similar meaning. This descriptor works good for images with non-uniform edge distribution. It also accounts for edge

orientation. Edge histogram provides effective representation for natural images, sketch images, and clip art images with non-uniform edge distribution [130]. It is the second descriptor selection in the *MPEG-7* feature vector in our work.

4.9 Region Shape

This descriptor defines a set of separable angular radial transformation (ART) basis functions which classify shape along various angular and radial directions. The descriptor has 35 coefficients. It makes use of all pixels constituting the shape within a frame, it can describe any shapes, i.e. not only a simple shape with a single connected region but also a complex shape that consists of holes in the object or several disjoint regions. The descriptor is robust to segmentation noise. It can cope with errors in segmentation where an object is split into disconnected sub regions, provided that the descriptor was computed using all sub regions. It is robust to minor deformation along the boundary of the object. The descriptor size is compact and fixed to 17.5 bytes, extraction is fast. The descriptor also offers scale and rotation invariance. Region shape descriptor performs excellent on any type of media as feature vector [41]. This is the third descriptor selection in the *MPEG-7* feature vector in our work.

4.10 Contour Shape

This descriptor captures characteristic shape features of an object image or region based on its contour. It uses Curvature Scale-Space (CSS) [3] representation, and its feature coefficients represents very well characteristic of the object shape, enabling similarity-based retrieval. It reflects properties of the perception of human visual system and offers good generalization [9]. It is robust to non-rigid motion, to partial

occlusion of the shape, and to perspective transformations. The descriptor is also compact in size. This is the fourth r descriptor selection in the *MPEG-7* feature vector in our work.

4.11 *MPEG-7* Motion Descriptors For Classification?

[114] uses repeated motion of objects to classify *people*, *groups of people*, and *vehicle*. Temporal templates of recurrent motion images are computed by aligning and accumulating the foreground regions from non-*MPEG-7* feature-based object detection module. For training five shots from 6 hours of videos are considered in [114]. Testing is done on two shots (each with 2688 frames) of PETS 2001 [5]. 88.0% accuracy is reported with a rule-based classifier. These works [80, 114] suggests that motion can be a useful feature to distinguish moving objects in video (e.g., non-*MPEG-7* feature posture, recurrent variation in motion direction).

[80] presents a novel approach to human body posture recognition based on the MPEG-7 contour-based shape descriptor and the widely used projection histogram. A combination of the both is used to recognize the main posture and the view of a human based on the binary object mask obtained by the segmentation process. The recognition is treated as a typical pattern recognition task, and is carried out through a hierarchy of classifiers.

On the other hand, [131] does not use any motion feature for classification of *people* and *cars* video objects. Non-*MPEG-7* features of objects are used, e.g., *width*, *height*, *area*, *aspect-ratio*. [131] uses different features for tracking and classification. Normalization is done referring to the scaling of a feature's metric according to its

vertical position, as objects become smaller / larger as they move. [131] states that due to their temporal stability, object classification is usually applied to the descriptors obtained from the tracked objects.

To maintain *MPEG-7* compliance objective for video object description in our work, motion descriptors available in *MPEG-7* needs to be used in our work. There are two sets of motion descriptors in *MPEG-7*: one set (motion activity and camera motion) applies to video segments, and the other (motion trajectory and parametric motion) to moving regions.

The motion activity descriptor captures the intuitive notion of 'intensity of action' or 'pace of action' in video segments, e.g., *fast*, *slow*. In the XM implementation, the descriptor depends on previously computed motion vectors. It does not have its own motion estimation.

The motion trajectory descriptor is calculated based on segmentation. A first- or second- order piecewise approximation of the spatial positions of the representative point along time, for each spatial dimension. Time instants are initialized as successive frame-time instants (one per frame). Then for each time instant, the position (x,y) of the trajectory point are instantiated as coordinates of the center of the region (mask, bounding box), and the interpolating parameters (1st order or 2nd order) are calculated as the local second-order derivatives of the positions. The input format for Motion Trajectory are successive spatio-temporal positions of a point, representative from the object whose trajectory needs to be described. These positions are generated off-line, in a process which is independent from the *MPEG-7* reference software XM. When a segmentation mask is available for defining the object, the input can be obtained by simply calculating for each frame the center of mass of the segmentation mask, and storing these coordinates. Positions can be directly obtained by segmentation of motion field, or by getting the centroid of a bounding box provided as output

of a tracking.

The parametric motion descriptor is calculated from pairs of images. The motion between each image pair is represented. Different motion models (translational 2 parameters, rotation/scaling 4 parameters, affine 6 parameters, perspective 8 parameters, quadratic 12 parameters), for spatial and temporal references can be used. It is used specially when objects are likely to undergo complex motions or deformations that cannot be captured by Motion Trajectory. Parametric motion models have been extensively used within various related image processing and analysis areas, including motion-based segmentation and estimation, global motion estimation, mosaicing (i.e., panoramic view of a video segment, constructed by aligning and blending together frames of a video segment upon each other using common spatial reference) and object tracking. Examples of use are queries such as search for *objects approaching the camera*, or *objects describing a rotational motion*, or *objects translating left.*

In this thesis object-based classification (see Section 3.7) is deployed, for which only moving region based *MPEG-7* motion descriptor are probable candidates for motion feature. The data sets (see Section 5.3.1) contains object image of video objects from fixed camera views (both far field and near field views). By definition [9], MPEG-7 motion descriptors are not suited to exhibit discrimination pose or in postures, among target object classes, e.g., between $person$ and $group\_of\_person$, in consecutive frames. In surveillance video data sets, *MPEG-7* motion descriptors for moving regions lack to exhibit significant discriminancy among pre-defined target class video objects, e.g., between $person$ and $group\_of\_persons$ in a downtown street. [19] uses motion trajectory descriptors to describe the spatial location of video objects in successive frames, and reports very low classification accuracy for the above reason. [19] uses, region locator, contour shape, and dominant color descriptors to improve classification accuracy.

Thus, 1) to maintain *MPEG-7* compliance objective for video object description in our work, and, 2) due to the unavilibility of suitable *MPEG-7* motion descriptors, no *MPEG-7* motion descriptor is included in the target feature vector. Beside, motion descriptors integrate in the temporal domain and would only be comparable with other elementary visual descriptors (Section 4.1) in video object discrimination in this thesis, if the basic color, texture and shape descriptors would be aggregated over time. However, motion estimation for moving image regions is integrated in tracking [132]. Also, the effect of motion is considered in post-classification (see Section 6.2), where class labels for group of video objects from different frames which hold same tracking id are calculated.

4.12 Non-*MPEG-7* Features?

Description generation of video contents (i.e., objects in this thesis) belongs to the interpretation module after segmentation and tracking (see Fig. 1.1) in video processing framework. Video object feature vector for object classification can be independent of the segmentation and tracking feature. To maintain *MPEG-7* compliance in video object description, any non-*MPEG-7* features (i.e., center point x, center point y, compactness, and irregularity) applied in tracking can not be used as feature vector for video object classification. Apart from this reason, 1) any non-*MPEG-7* features which can be useful both in object tracking and in object clasification can be replaced by default *MPEG-7* visual descriptors. For example, *irregularity* in *MPEG-7* texture descriptors and 2) any non-*MPEG-7* features (e.g., center point x provide spatial locations and are not discriminative (see Section 4.11) for our target classes in video object classification.

4.13 Summary

This chapter presented review of eight basic *MPEG-7* visual descriptors to select a set of suitable descriptors for video objects in surveillance systems. We select the set of visual descriptors which offer fast extraction, interoperability, are of compact size, and are scale and resolution invariant. These descriptors are color layout, edge histogram, region shape, and contour shape. Our application focuses on object-based classification where only moving region based *MPEG-7* motion descriptor are probable candidates for motion feature. However, in both training and test data sets, *MPEG-7* motion descriptors for moving regions lack to exhibit significant discriminancy among pre-defined target class video objects, e.g., between *persons* and *group of persons* in a downtown street. To maintain *MPEG-7* compliance, any non-*MPEG-7* features even available from video analysis for video objects are not considered.

Chapter 5

Proposed Feature Binding Method

This chapter contains the proposed method for chaotic feature binding to generate the new feature vector. In Section 6.10.1, we further disscuss important steps of the proposed feature binding which contribute to the robustness of the new feature vector. In Section 5.3, we present the statistical analysis of the new feature vector based on the simulation of the proposed feature binding method.

5.1 Introduction

A *feature space* is an abstract space where each object is represented as a point whose dimension is determined by the number of features. Any feature space is assumed to have a certain geometry (e.g., Euclidean). There is a multi-dimensional (e.g., j) feature space for each *MPEG-7* visual descriptor, where each feature space is j-dimensional. j is not uniform in all descriptors, i.e., it varies in different descriptors. A video object can also be described with a set of *MPEG-7* visual descriptors where each descriptor has its own descriptor coefficients. Then there can be multiple features spaces (e.g.,$j_1 - dimensional, j_2 - dimensional, ..., j_J - dimensional$)

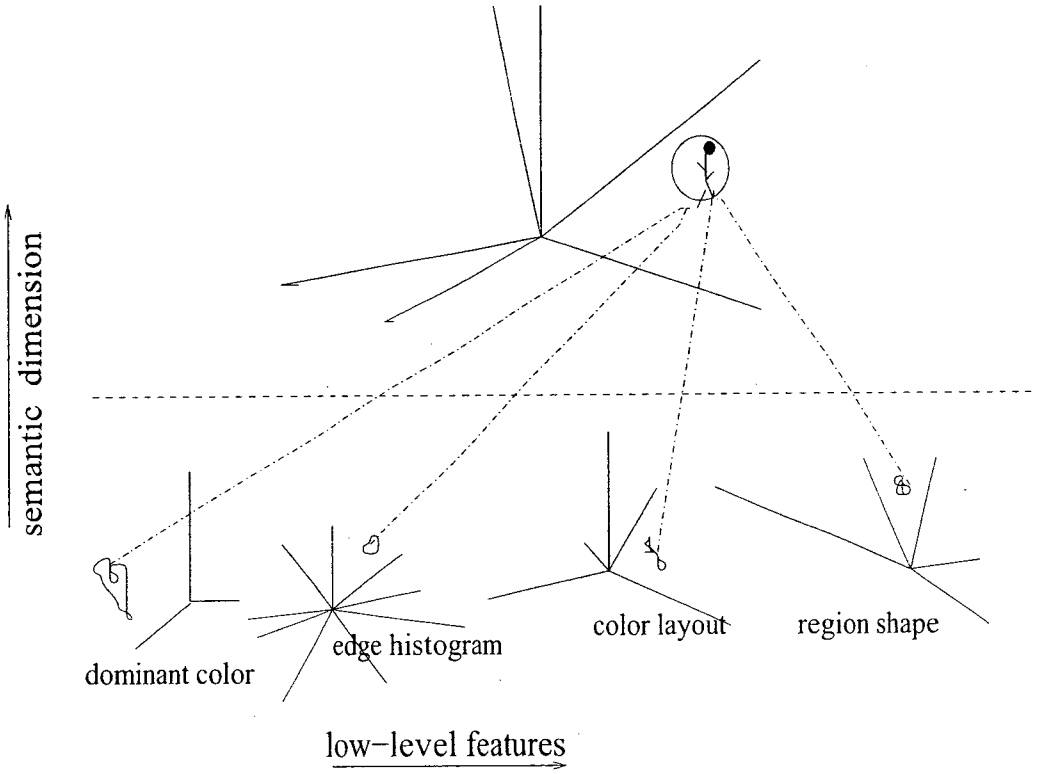


Figure 5.1: A layered multi-dimensional space *binding space*, which can incorporate different feature spaces of different *MPEG-7* visual descriptors.

for total I number of *MPEG-7* visual descriptors. To accommodate multiple feature spaces in different visual descriptors for a video object, we define the concept of a layered multi-dimensional space as *binding space*, which can incorporate different feature spaces from different descriptors. We assume the binding space as the 'problem space' similar to the object recognition 'problem space' of the primary visual cortex. Primary visual cortex is the center of neuronal activity for processing visual features in human brain (see Section 3.2).

Fig. 5.1 shows the diagram of the binding space. Here the y-axis is assumed to have *semantic*, i.e., *meaning*, dimension and represent abstract semantic co-relations of feature coefficients. In real world, an object can have multiple semantic (i.e., meaning)

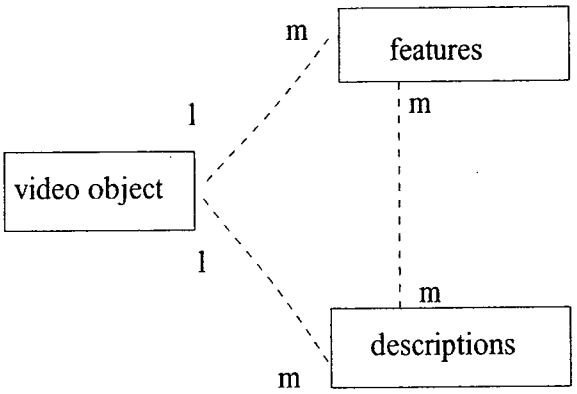


Figure 5.2: Shared relationship among video object, low level features, and human understandable descriptions.

associated with multiple human understandable descriptions, e.g., a video object with a *person driving a car*, can be labeled as both '*has_person*' and '*has_vehicle*). Feature coefficients in a set of descriptors can be common to multiple descriptions for the same object or for multiple objects. As shown in Fig. 5.2, for a video object, different feature coefficients from different descriptors can have shared relationship to define different descriptions. The semantic dimension in Fig. 5.1 is intended to account for any 'undefined' semantic relevance of descriptor coefficients in different visual descriptors.

The x-axis in Fig. 5.1 represents the number of coefficients in different *MPEG-7* descriptors. Multiple (i.e., I) *MPEG-7* feature spaces reside in the lower layer of the binding space. The assumption of the binding space open rooms to process (e.g., chaotic *feature binding*) coefficients of *MPEG-7* visual descriptors.

5.2 Feature Binding

Feature binding [23] is a process to group relevant sets of feature coefficients in a multi-dimensional space (see Fig. 5.1). We apply feature binding on coefficients of different

MPEG-7 descriptors in an abstract multi dimensional space. We use chaotic series to generate a *MPEG-7 compliant* feature vector *C-MP7*. A *MPEG-7 compliant* feature vector is a modified *MPEG-7* feature vector which can generate XML description of media content using the same XML schema as available in the *MPEG-7* standard [9]. *C-MP7* is *MPEG-7 compliant*, because, it contains some of the same *MPEG-7* descriptor coefficients. So *C-MP7* can adopt the same XML schema [9] as offered by each *MPEG-7* visual descriptors to generate XML description files. The difference in *C-MP7* and *MPEG-7* are the null feature elements padded dynamically at different indexes during feature binding. The *MPEG-7* compliance preserve the scopes for *C-MP7* to exchange video object descriptions with other industry standard *MPEG-7* applications. The binding space in Fig. 5.1 groups coefficients from different *MPEG-7* descriptors of different feature spaces into a single feature space. The new feature vector *C-MP7* is suppose to provide generic description of video objects in different classes.

In the framework of the proposed approach (see Fig. 5.3 and Fig. 5.4), the input video goes through a video analysis module [132] that produces a list of segmented and tracked video objects. These video objects are first pre-labeled and grouped in different classes. The feature binding is done from the extracted *MPEG-7* visual descriptors in each video object (Fig. 5.4). The feature binding then, derives feature vector *C-MP7* for video objects of each class.

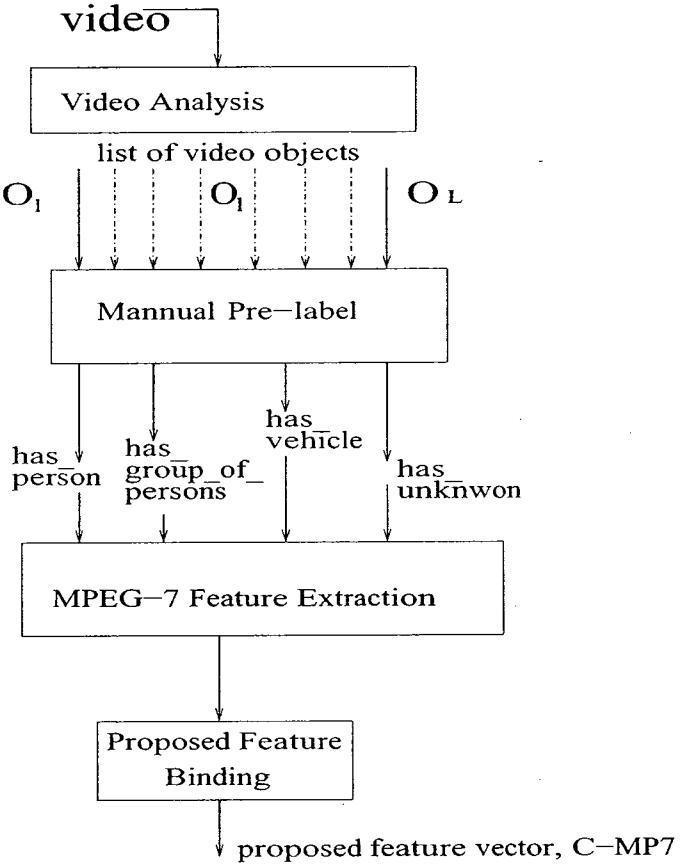


Figure 5.3: Framework of the proposed method.

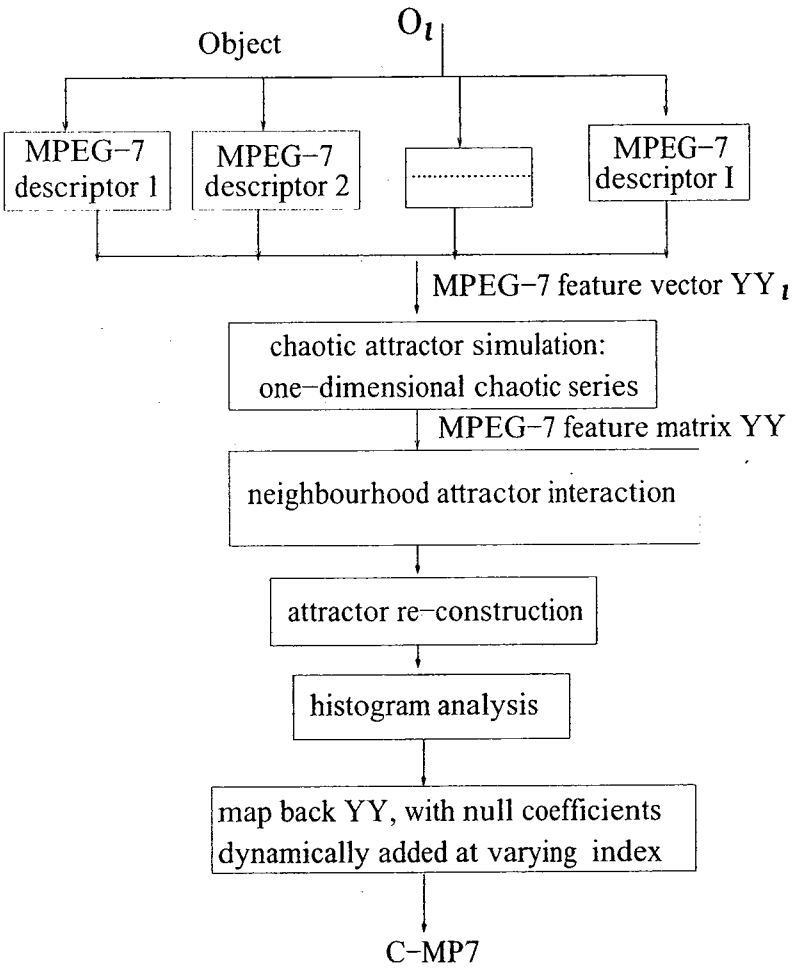


Figure 5.4: Different processing steps in the proposed feature binding method.

5.2.1 Overview

The video analysis module generates a list of video objects O_l, where $l \in L$; L is the total number of video objects. d_{ijl}, where $d_{ijl} \in \mathbb{R}$, is i^{th} *MPEG-7* descriptor of object O_l with j^{th} feature coefficients, where $j \in J_i$ and $i \in I$. The number of feature elements in descriptor i is J_i, and the number of descriptors per object O_l is I. YY_l is the feature vector for object O_l, which contains feature coefficients of all *MPEG-7* visual descriptors, i.e.,

$$YY_l = [d_{11l}d_{12l}\cdots d_{1J_1l}\cdots d_{21l}d_{22l}\cdots d_{2J_2l}\cdots d_{I1l}d_{I2l}\cdots d_{IJ_Il}], \tag{5.1}$$

The dimension of YY_l is 130 (i.e., 80 feature coefficients from edge histogram, 12 feature coefficients from color layout, 3 feature coefficients from contour shape, and 35 feature coefficients from region shape visual descriptors). YY is a 2D array containing feature vectors of all objects. $YY = (d_{ml})$, where $m \in M$, $l \in L$, $M = \sum_{i=1}^{I} J_i$, and d_{ml} is the value of feature coefficient m of video object l. We then, normalize YY using the min-max normalization (to preserve the variances of both rows and columns of the data matrix) as,

$$YY_{nrm} = \frac{YY - d_{ml_{min}}}{d_{ml_{max}} - d_{ml_{min}}} \tag{5.2}$$

After normalization, each feature coefficient is simulated by a chaotic series. The normalized 2D video object array YY_{nrm} then becomes a 3D video object matrix YY_{mx}. by a discrete chaotic series $f(\cdot)$ with a high iteration number, N. Even if N considered to be very high (near ∞), given the inevitable imprecision in characterizing a system's initial state, it is not possible to predict far into the future other than to assert that there is a limiting probability distribution according to which the dynamical system's state will be specified [133]. Considering the computational over-

head, $N = 500$ sufficiently exhibits chaos for different feature coefficients (assumed as dynamical systems) in our simulation (see Section 6.10). We use the discrete form of either Logistic map, a *low-dimensional* chaotic series [33], or Mackey Glass, a *high-dimensional* chaotic series [33], to simulate respective set of chaotic attractors. The discrete form of the Logistic map in (2.5) is,

$$x_{n+1} = r x_n (1 - x_n), \tag{5.3}$$

where $0 \leq n < N$, $N \longrightarrow \infty$, $r = 4$ and $0 \leq x_n \leq 1$ represents an element of the chaotic series at iteration n. The discrete form of the Mackey-Glass equation is,

$$x_{n+1} = x_n + \frac{0.2(x_{n-20})}{1 + (x_{n-20})^{10}} - 0.1 x_n \tag{5.4}$$

where $0 \leqslant n \leqslant N$ and $N \longrightarrow \infty$.

Neighborhood interaction, among the coefficients in YY_{mx}, is then calculated by applying coupled map lattice (CML) [24]. To make the computation with YY_{mx} inexpensive, the CML modified chaotic attractors are reconstructed with finite iteration length. The characteristics (e.g., correlation dimension) of original chaotic attractors are preserved in reconstructed attractors [42]. Statistical mean of the reconstructed attractors are computated then to map back the video object matrix YY_{mx} to a modified video object array YY_{md} $H(\cdot)$ is the histogram-analysis based feature binding of the re-constructed attractors for all $f(\cdot)$ in YY_{md}. $H(\cdot)$ generates a new video object array Y by replacing the *MPEG-7* feature coefficents at varying indexes of YY into YY_{md}. Y contains the new feature vector, *C-MP7*, for each video object.

5.2.2 Feature Excitation

In the human visual system, the retinal receptors do not fire neuronal spikes in the EEG, unless a proper stimuli provides sufficient excitations [45]. We consider each feature coefficient from each extracted descriptor as neuron, and excite the coefficient using a chaotic series. Each coefficient d_{ml} in normalized video object array YY_{nrm} can be used as seed x_0, for the chaos series with iteration length N to form the video object matrix YY_{mx}.

In a chaotic series, the iteration parameter n indicates the time interval (e.g., see Eq. 5.3 and Eq. 5.4) that express the dynamics over that specific time. We consider n as an abstract dimension for feature binding [23]. $f(\cdot)$ generates individual chaotic attractor, i.e., chaotic series $f(x_n)$, n=*1,2,....N* in the phase space of successive series elements, from each d_{ml}. Each d_{ml} can be now substituted by the chaotic series $f(x_n)$. Trajectories of all d_{ml} form different chaotic attractors which have geometrical structure of similar patterns, but, the structures are not exactly the same. The attractors in YY_{mx} can be expressed as $f(x_n(m,l))$.

5.2.3 Attractor Interaction With Coupled Map Lattice (CML)

CML have been investigated for both local as well as global modeling of human information processing tasks [134]. To define the neighborhood interaction within $f(x_n(m,l))$ we apply 2D CML [24]. The 2D CML for feature coefficient $d_{ml} \in YY_{mx}$ is given as,

$$x_{n+1}(m,l) = f(x_n(m,l)) + e(f(x_n(m-1,l)) + f(x_n(m+1,l))$$

$$+f(x_n(m,l+1)) + f(x_n(m,l-1)) - 4f(x_n(m,l)). \tag{5.5}$$

where, $f(x_n(m,l))$ is the chaotic series with initial condition, d_{ml}, n is the neighborhood iteration parameter, and e is a coupling coefficient. The boundary conditions of (5.5) are as in, $f(x_0(m,l)) = f(x_n(m,l))$, $f(x_{n+1}(m,l)) = f(x_1(m,l))$, $f(x_n(0,0)) = f(x_n(m,l))$, $f(x_n(m,l)) = f(x_n(1,1))$.

CML thus couples multiple dynamical systems to take account for coefficients of multiple chaotic series to make the video object-matrix more homogeneous as shown in the partial video object matrix in Figs. 5.5 and 5.6. To apply CML in the proposed feature binding, at least two video objects are required in corresponding class.

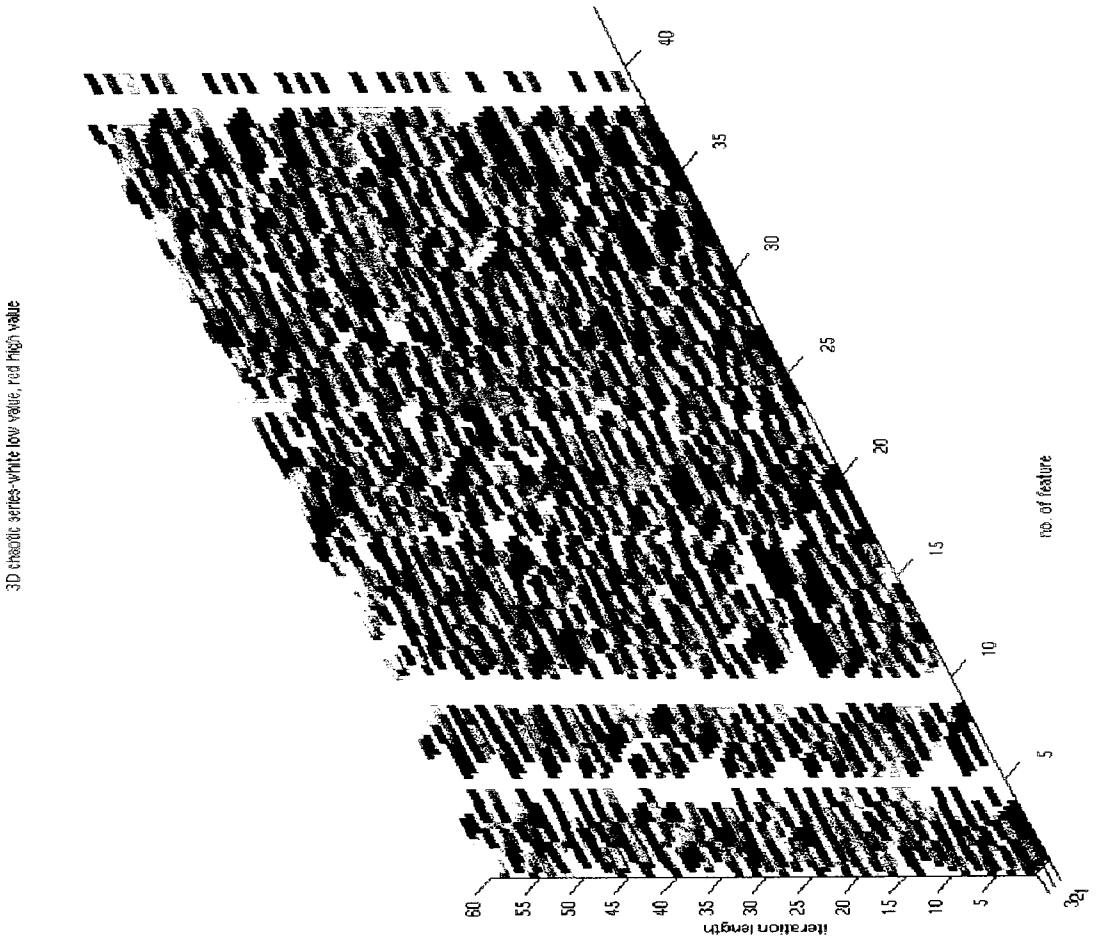


Figure 5.5: Partial view of the 3D video object matrix for *has_person* class. 40 feature coefficients with iteration length $n = 60$ for 3 sample video objects are shown. Here colors represent range of normalized coefficient values, where *red* represents high values of chaotic series coefficients and *white* represents low values of the same

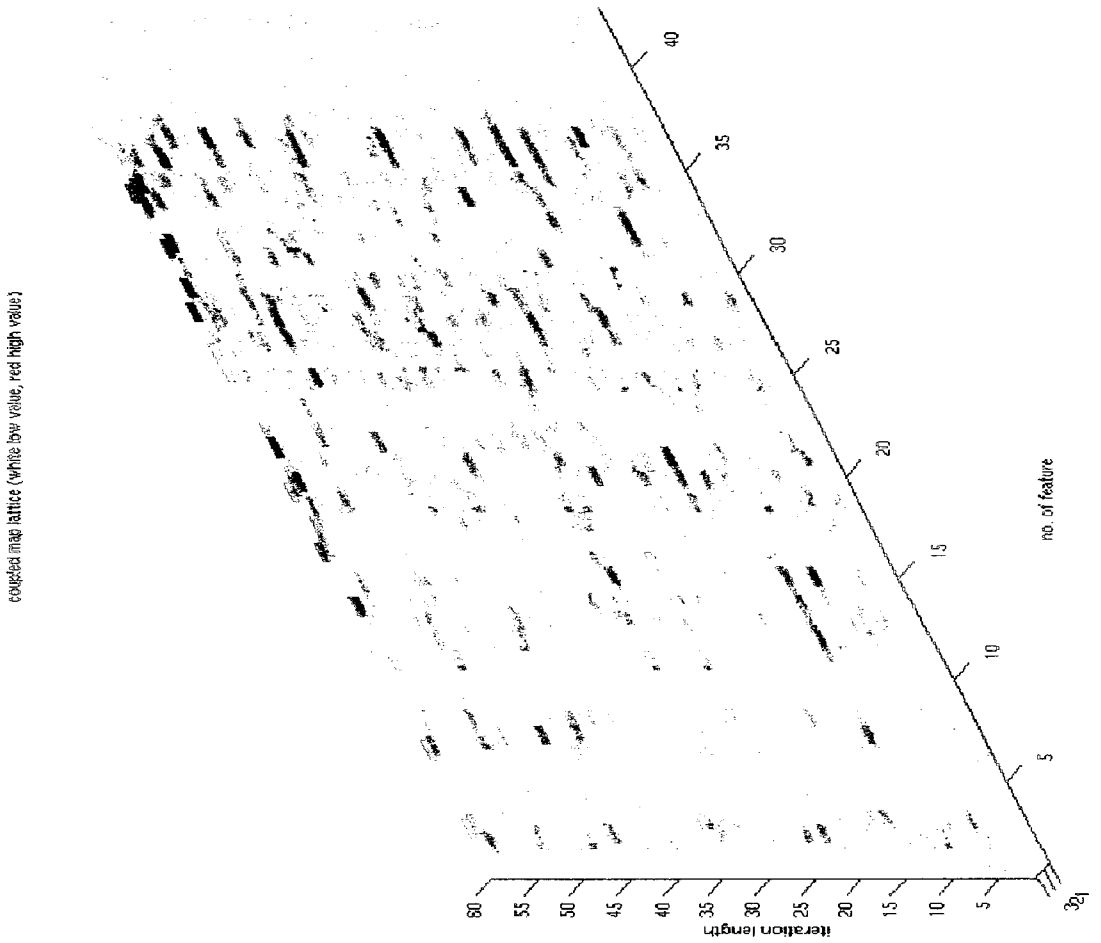


Figure 5.6: After neighborhood interaction is applied using Coupled Map Lattice on the same 3D video object matrix as in Fig. 5.5. Range of chaotic series coefficient values are more homogeneous, where *red* represents high values of chaotic series coefficients and *white* represents low values of the same. The chaotic series coefficients are modified based on the coupling of neighborhood coefficient values in the matrix.

5.2.4 Re-construction Of Attractors

We reconstruct each CML coupled chaotic series $f(x_n(m,l))$ with very small iterative length. [42] shows that the trajectory of a chaotic attractor can be reconstructed with fewer (i.e., $<< N$) chaotic series coefficients maintaining the same pattern (i.e., dimension) of the original attractor. At regular and discrete intervals of iterations the value of some state variables in $f(x_n(m,l))$ can be measured and recorded as, $x_1(p), x_2(p), ..x_n(p), ..$ where $x_n(p) \in \mathbb{R}$. A set of g dimensional vectors, whose components are values at delayed iteration K, can be created [42] as,

$$v_n(p) = x_n(p), x_{n+K}(p), x_{n+2K}(p), \cdots, x_{n+(g-1)K}(p), \tag{5.6}$$

where $v_n(p) \in \mathbb{R}^g$ and $g) <<N$. Thus, for chaotic series coefficients $x_n(p)$ we get vectors $v_n(p)$ in $\mathbb{R}^g$, which is an embedding trajectory space. The dynamics in the one-dimensional coupled chaotic series $f(x_n(m,l))$ is, thus, converted to spatial information in g-dimensional $f(v_g(m,l))$. We re-construct the attractors, as in (5.6), with lower iterative length g. The reconstructed attractors maintain same statistical properties of the original attractors.

5.2.5 Histogram-based Clustering

Intuitively, (see Section 2.4), some visual features cause the neuronal population in the areas of the brain to generate chaotic patterns. We apply histogram-based clustering of chaotic trajectory coefficients of the visual features to syntactically identify indexes of the largest cluster of visual features specific to the pattern of a video object-class. To find scalar property of reconstructed $f(v_g(m,l))$, we calculate the mean ψ for each reconstructed chaotic series,

$$f(v_g(m,l))_{mean} = f(\psi_{ml}) = YY_{md}, \tag{5.7}$$

Now, histogram $H(f(\psi_{ml}))$ identifies the largest cluster of ψ in $f(\psi_{ml})$. As shown in Fig. 5.7, the indexes of the largest cluster can be identified from the center of the corresponding histogram bin. These indexes are then used to map back $f(\psi_{ml})$ to $f(v_n(m,l))$ for each $d_{ml} \in YY$ (feature coefficients before normalization). The indexes outside the largest mean cluster are made null in YY to create a new video object-array Y. We consider the discarded null feature coefficients in Y as *'irrelevant'*. Y constitutes the new feature vector *C-MP7* for L video objects, where Y_l is the feature vector for object O_l.

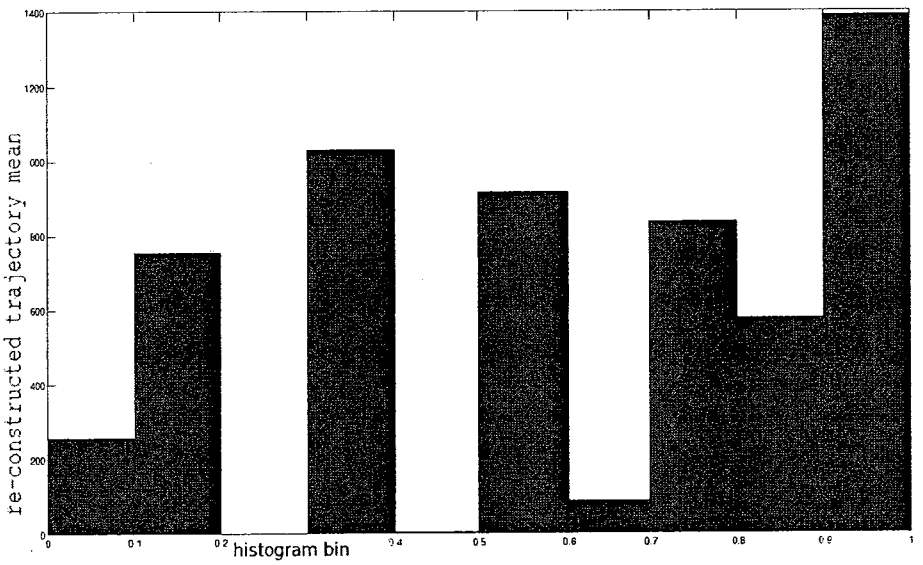


Figure 5.7: Histogram analysis for l^{th} feature coefficients in m^{th} video object in video object array $H(f(\psi_{ml}))$. Here, $m = 45$ and $l = 130$. $H(f(\psi_{ml}))$ identifies the largest cluster of *mean* values ψ in $f(\psi_{ml})$. The *mean* values ψ are calculated as scalar properties in video object matrix with reconstructed chaotic series $f(v_g(m,l))$.

5.3 Statistical Analysis Of The Proposed Method

We evaluated the new feature vector *C-MP7* in terms of feature coefficient reduction, information measures, and multi-class Fisher criteria. The performance of *C-MP7* is compared with that of original *MPEG-7* descriptor-based feature vector. Section 5.3.1 first presents the details on the data sets used for simulation. Section 5.3.2 explains why *C-MP7* is *MPEG-7* compliant, Section 5.3.3 reports the dynamic feature reduction in *C-MP7*, Section 5.3.4 presents the mean and variance of *C-MP7* to illustrate the discriminancy among different video objects, and Section 5.3.5 shows the multi-class Fisher criteria evaluation for *C-MP7*.

5.3.1 Data Sets

For simulation, we select 5500 diverse video objects as ground truth from about 18000 frames of 30 assorted indoor and outdoor surveillance scenes. Scenes are taken from public surveillance video databases, CAVIAR, EPFL, INRS, UROCHESTER, QCIF, and also from a local, AXIS 213 PTZ, camera feed. Video objects are randomly grouped into different data sets, and pre-labeled in four different classes: *has_person* (p), *has_group_of_persons* (g), *has_vehicle* (v), and *has_unknown* (u). In real world video object description, any video object which can be labeled as *has_group_of_persons* also can be labeled as *has_person*, however any video object that can be labeled as *has_person*, does not represent the class *has_group_of_persons*. To keep this distinction as appropriate to real world description of video objects, all the video objects from the *has_group_of_persons* class are made available to the class *has_person*. However, the video objects from *has_person* class are kept independent from the video objects in the class *has_group_of_persons*. To handle spurious image-regions in frames which are mistakenly tracked as video objects, u class is included.

Table 5.1: Data sets from public surveillance video shots, CAVIAR, EPFL, INRS, UROCHESTER, QCIF, and also from a local, AXIS 213 PTZ, camera feed. Video objects are randomly grouped into different data sets, and pre-labeled in four classes: *has_person* (p), *has_group_of_persons* (g), *has_vehicle* (v), and *has_unknown* (u).

Data set	No of objects	*'has_person'*	*'has_groupofperson'*	*'has_vehicle'*	*'has_unknown'*
1	1101	2meet, hall	MeetWalkSplit, vnj	guy11	Sit, survey
2	1192	bishop2entrance, FightChase	MeetSpilt3rdGuy	road2	intelligent, MeetCrowd, RestWiggleOnFloor, LefBag, RestSumpOnFloor, FightRunAway2 MeetWalkSpilt
3	1231	ekrlb, intelligentroom JuliusDeposit,takeobj, MeetWalkSplit, putobj, MeetSpilt3rdGuy, vnj, MeetWalkTogether1	ekrlc, FightChase, MeetWalkTogether1, FightOneManDown, RestFallOnFloor,survey MeetCrowd,LeftBag	road3	putobj, FightChase, vnj, takeobj, hall
4	1039	LeftBag,RestFallOnFloor	MeetWalkTogether2	road3	bishop2entrance, vlab
5	1091	RestWiggleOnFloor, sit, survey	MeetWalkSplit, vnj	guy11	RestFallOnFloor, 2meet, road1, road2
6	1002	FightOneManDown, FightRunAway2	MeetWalkTogether2	road1	guy11, MeetWalkTogether1
7	1537	RestSlumpOnFloor, MeetWalkTogether2	ekrlc, FightChase, FightOneManDown, MeetWalkTogether1, RestFallOnFloor, survey LeftBag, MeetCrowd	road2	ekrlb
11	1434	FightOneManDown, intelligentroom, LeftBag, MeetCrowd, putobj, RestFallOnFloor, RestWiggleOnFloor, takeobj FightRunAway2 MeetWalkTogether1	ekrlc, FightChase, FightOneManDown, LeftBag, MeetCrowd, MeetWalkTogether1, RestFallOnFloor	road1	2meet, FightChase, LeftBag, putobj, RestFallOnFloor, RestWiggleOnFloor, road1, takeobj
12	535	survey, vnj	survey, vnj	road1	road1, survey, vnj
13	698	CarB, PeopleBetter, PeopleBetter3, PeopleBetter4	CarB, PeopleBetter, PeopleBetter3, PeopleBetter4	CarB, PeopleBetter4	CarB, PeopleBetter4
16	193	FightOneManDown, intelligentroom, vnj, CarB, LeftBag, putobj, survey, takeobj	ekrlc, FightChase,takeobj, PeopleBetter3, survey, vnj	CarB, PeopleBetter4, vnj road1, survey,	2meet, FightChase, PeopleBetter4, putobj, road1,

Abrupt changes in lighting, shaking trees, shadows, and occlusions in any scene may influence the segmentation and tracking of video objects [132]. To minimize the influence of segmentation, video objects with good and poor segmentation are both mixed together in a class. Sample video objects from different classes are shown respectively in Tables 5.2, 5.3, 5.4 and 5.5. More video object samples are available on request [135].

Table 5.2: Video objects for *has_person* class, which include poor-segmented video objects, as in 7^{th} row 1^{st} column, 1^{st} row 2^{nd} column, and 2^{nd} row 3^{rd} column.

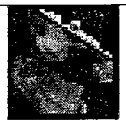
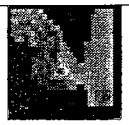
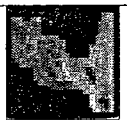
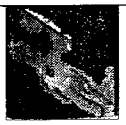
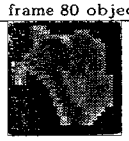
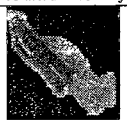
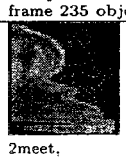
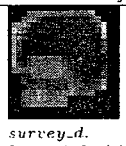
	 LeftBag, frame 18 object 3	 LeftBag, frame 80 object 10027	 LeftBag, frame 86 object 10030	 LeftBag, frame 121 object 29
	 LeftBag, frame 190 object 39	 LeftBag, frame 192 object 10045	 LeftBag, frame 213 object 13	 LeftBag, frame 253 object 30
	 $RestFallOnFloor$, frame 16 object 3	 $RestFallOnFloor$, frame 29 object 3	 $RestFallOnFloor$, frame 68 object 3	 $RestFallOnFloor$, frame 89 object 3
	 $RestFallOnFloor$, frame 235 object 77	 $RestFallOnFloor$, frame 271 object 77	 $RestFallOnFloor$, frame 145 object 1	 $RestFallOnFloor$, frame 401 object 22
	 2meet, frame 598 object 38	 $intelligentroom_vids$, frame 222 object 10004	 July25CarB, frame 44 object 45	 July25CarB, frame 147 object 10266
	 putobj, frame 476 object 10007	 $survey\_d$, frame 63 object 2	 $survey\_d$, frame 72 object 4	 $survey\_d$, frame 193 object 8
	 $survey\_d$. frame 412 object 10060	 $survey\_d$. frame 473 object 63	 takeobj. frame 195 object 6	 vnj. frame 204 object 10005

Table 5.3: Video objects for *has_group_of_person* class which include poor-segmented video objects, as in 1^{st} row 1^{st} column, 5^{th} row 1^{st} column, and 4^{th} row 4^{th} column.

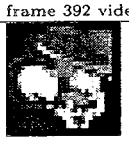
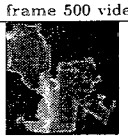
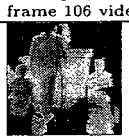
	 *Meet_WalkTogether2*, frame 392 video object 43	 *Meet_WalkTogether2*, frame 392 video object 10077	 *Meet_WalkTogether2*, frame 403 video object 77	 *Meet_WalkTogether2*, frame 460 video object 10087
	 *Meet_WalkTogether2*, frame 500 video object 89	 *Meet_WalkTogether2*. frame 564 video object 89	 ekrlc frame 225 video object 39	 ekrlc frame 239 video object 39
	 ekrlc, frame 290 video object 92	 ekrlc. frame 310 video object 92	 *survey_d*. frame 79 video object 4	 *survey_d*, frame 91 video object 4
	 *survey_d*. frame 98 video object 4	 *survey_d*. frame 106 video object 4	 vnj, frame 176 video object 4	 July25PeopleBetter3, frame 6 video object 10020
	 July25PeopleBetter3. frame 80 video object 10163	 July25PeopleBetter3. frame 84 video object 163	 July25PeopleBetter3, frame 85 video object 163	 July25PeopleBetter3, frame 93 video object 10199
	 LeftBag. frame 16 video object 3	 LeftBag. frame 30 video object 3	 *Rest_FallOnFloor*, frame 92 video object 34	 *Rest_FallOnFloor*, frame 93 video object 34
	 *Rest_FallOnFloor*. frame 95 video object 34	 *Rest_FallOnFloor*. frame 96 video object 34	 *Rest_FallOnFloor*. frame 97 video object 34	 *Rest_FallOnFloor*. frame 98 video object 34

Table 5.4: Video objects for *has_vehicle* class which include poor-segmented video objects, as in 5^{th} row 1^{st} column, 6^{th} row 1^{st} column, 5^{th} row 3^{rd} column, 5^{th} row 4^{th} column, and 7^{th} row 2^{nd} column.

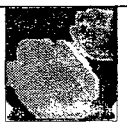
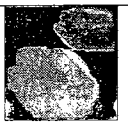
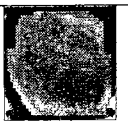
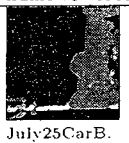
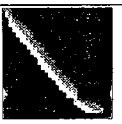
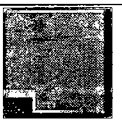
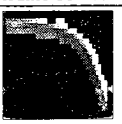
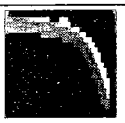
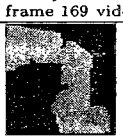
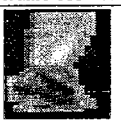
	 *road1_cif*, frame 13 video object 3	 *road1_cif*, frame 22 video object 3	 *road1_cif*, frame 33 video object 3	 *road1_cif*, frame 37 video object 5
	 *road1_cif*, frame 142 video object 4	 *road1_cif*, frame 154 video object 4	 *road1_cif*, frame 154 video object 5	 *road1_cif*, frame 155 video object 10006
	 *road1_cif*, frame 191 video object 5	 *road1_cif*, frame 191 video object 6	 *road1_cif*, frame 191 video object 7	 *road1_cif*, frame 192 video object 10008
	 *road1_cif*, frame 225 video object 5	 *road1_cif*, frame 225 video object 6	 *road1_cif*, frame 225 video object 10	 *road1_cif*, frame 227 video object 10
	 *road1_cif*, frame 261 video object 6	 *road1_cif*, frame 299 video object 5	 July25CarB, frame 44 video object 10070	 July25CarB, frame 45 video object 47
	 July25CarB. frame 76 video object 10114	 July25CarB. frame 82 video object 94	 July25CarB, frame 85 video object 94	 July25CarB. frame 99 video object 134
	 July25CarB. frame 110 video object 157	 July25CarB. frame 120 video object 136	 July25CarB. frame 121 video object 10202	 July25CarB. frame 123 video object 165

Table 5.5: Video objects for *has_unknown* class which include poor-segmented video objects, as in 6^{th} row 1^{st} column, 5^{th} row 2^{nd} column, 7^{th} row 2^{nd} column, and 5^{th} row 4^{th} column.

	 *bishop2enterance_resampled*, frame 14 video object 10001	 *bishop2enterance_resampled*, frame 27 video object 10004	 *bishop2enterance_resampled*, frame 27 video object 10005	 *bishop2enterance_resampled*, frame 29 video object 10006
	 *bishop2enterance_resampled*, frame 34 video object 10012	 *bishop2enterance_resampled*, frame 60 video object 10025	 *bishop2enterance_resampled*, frame 66 video object 23	 *bishop2enterance_resampled*, frame 68 video object 21
	 *bishop2enterance_resampled*, frame 73 video object 27	 *bishop2enterance_resampled*, frame 77 video object 27	 *bishop2enterance_resampled*, frame 96 video object 10039	 *bishop2enterance_resampled*, frame 142 video object 10053
	 *bishop2enterance_resampled*, frame 169 video object 39	 *bishop2enterance_resampled*, frame 172 video object 61	 vlab, frame 397 video object 2	 vlab, frame 405 video object 2
	 2meet, frame 159 video object 7	 2meet, frame 419 video object 23	 2meet, frame 631 video object 50	 *Fight_Chase*, frame 77 video object 10
	 putobj. frame2 video object3	 *road1_cif*, frame2 video object3	 *survey_d*, frame2 video object3	 *survey_d*. frame2 video object3
	 takeobj. frame 249 video object 10007	 takeobj. frame 521 video object 10	 takeobj. frame 249 video object 10007	 vnj. frame 93 video object 10001

5.3.2 *MPEG-7* Compliant

The new feature vector *C-MP7* has the same *MPEG-7* format (except NULL elements added after feature binding) as available in the XML schema (see Section 2.3) for each descriptor. So each descriptor (with reduced feature coefficients) for an video object can be reconstructed back to the original XML description. This compliance allows the *C-MP7*-based video object descriptions to be shared with other potential *MPEG-7* industry applications (e.g. TV Anywhere project [136]).

5.3.3 Dynamic Feature Coefficient Reduction

There are two techniques for dimensionality reduction in a feature vector. One is to select a limited set of features out of the total set [137], e.g., feature binding [23]. The other is to extract a smaller set of features as linear or nonlinear functions of the original set of features using discriminant analysis, e.g., Principal Component Analysis (PCA) [138]. We follow the former technique. The reasons are: 1) it allows us to maintain *MPEG-7* compliance for XML description of objects from the new feature vector, and 2) the video objects are often partially specified, which makes the task of obtaining transformed features using discriminant analysis a non-trivial exercise [113]. Reason 2, is specially true for surveillance applications where segmentation and tracking usually output incomplete video objects.

The dynamic feature coefficient reduction is measured as the total number of null elements (interpreted as irrelevant feature coefficients) at varying indexes of the feature vector *MPEG-7* and *C-MP7*. Separate chaotic feature binding module is used for each class of video objects. The *C-MP7*, with low and high dimensional chaotic series, yield null elements in different indexes of video object-matrix in each class. Fig. 5.8 shows the % of null feature coefficients in video object-matrices in different

classes for different data sets with *MPEG-7* and *C-MP7* (with *low-dimensional* and *high-dimensional* chaotic series). Different data sets are are plotted in x-axis and number of nulls in y-axis, class of video objects are represented by legends. Significant feature coefficient reduction is achieved in Fig. 5.8. *C-MP7* (with *low-dimensional* chaos) has on average 56.36% more null elements than the *MPEG-7* feature vector. In case of *C-MP7* (with *high-dimensional* chaos) this reduction is 37.05% more than the *MPEG-7*.

Although, in Fig. 5.8, *high* dimensional chaotic series simulation offers more feature coefficient reduction in *C-MP7*, the preference of *low-* or *high*-dimensional chaotic series in the proposed feature binding, is yet to be decided. The question is, if *high* dimensional chaotic series simulation can offer effective *C-MP7* without loosing any intel that is already available in *MPEG-7* visual descriptors. We address this question in Section 5.3.4, followed by the discriminant analysis in Section 5.3.5.

The indexes for the null elements vary dynamically depending on the number of the feature coefficients of the *MPEG-7* descriptors (i.e., J_i), number of video objects (i.e., L) per class, and the iteration length N in the chaotic series x_n. The dynamic feature coefficient reduction is illustrated in Fig. 5.9 and Fig. 5.10, where, video object array for *person* class, from data set 2, is displayed for *MPEG-7* and *C-MP7*, respectively.

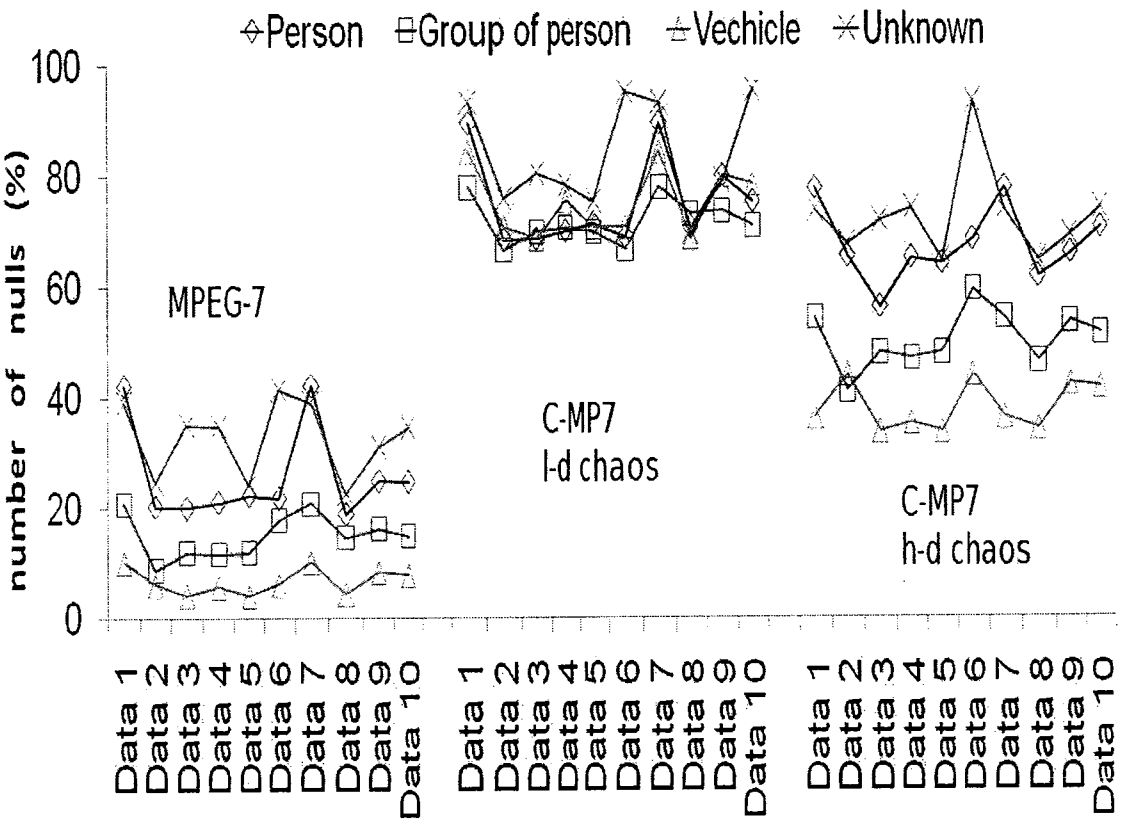


Figure 5.8: Dynamic feature coefficient reduction is measured as the total number of null elements (% of nulls) at varying indexes of the feature vector *MPEG-7* and *C-MP7* in video object array. The *C-MP7*, with low- and high- dimensional chaotic series, yield null elements in different indexes of video object array for corresponding class.

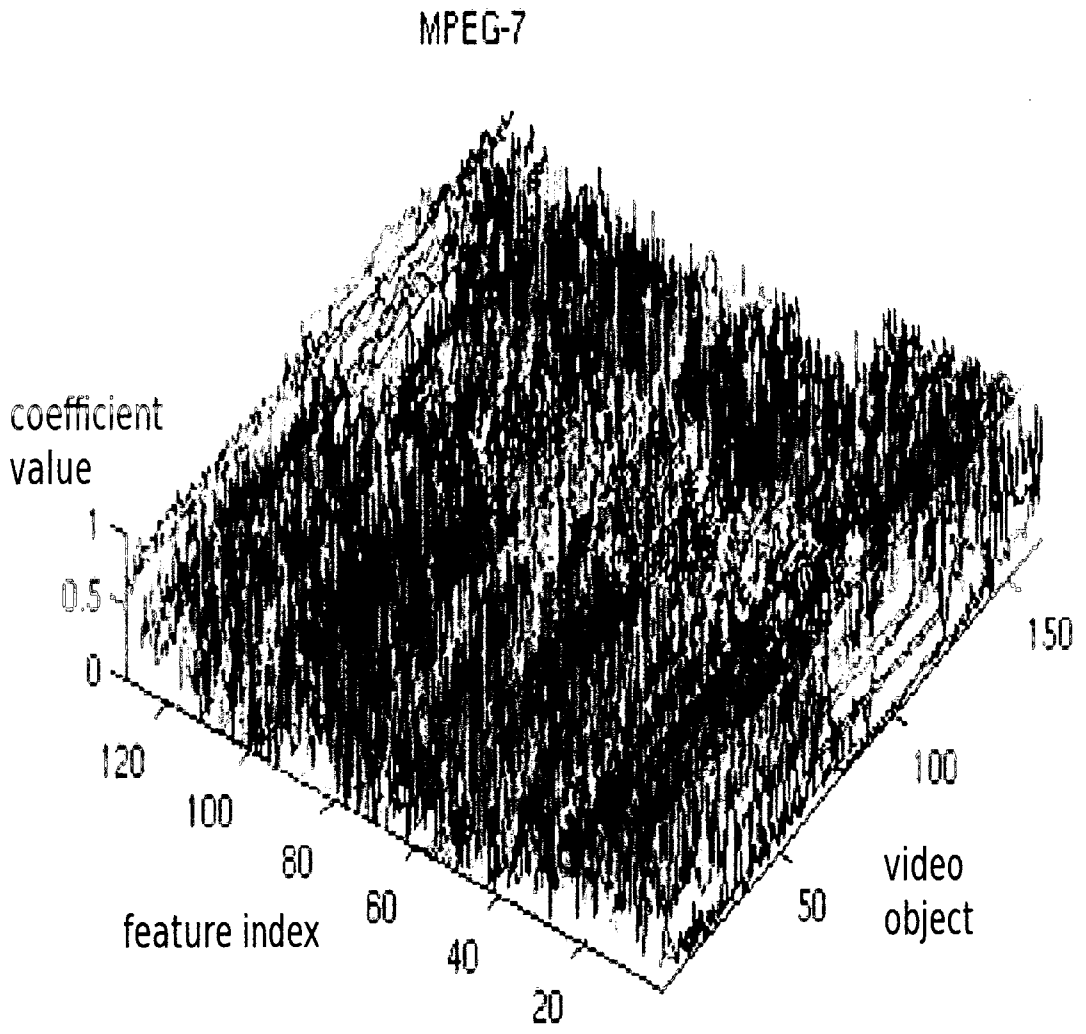


Figure 5.9: Video object array for *person* class, from data set 2, with *MPEG-7*. From Fig. 5.8, less than 10% feature coefficients are null.

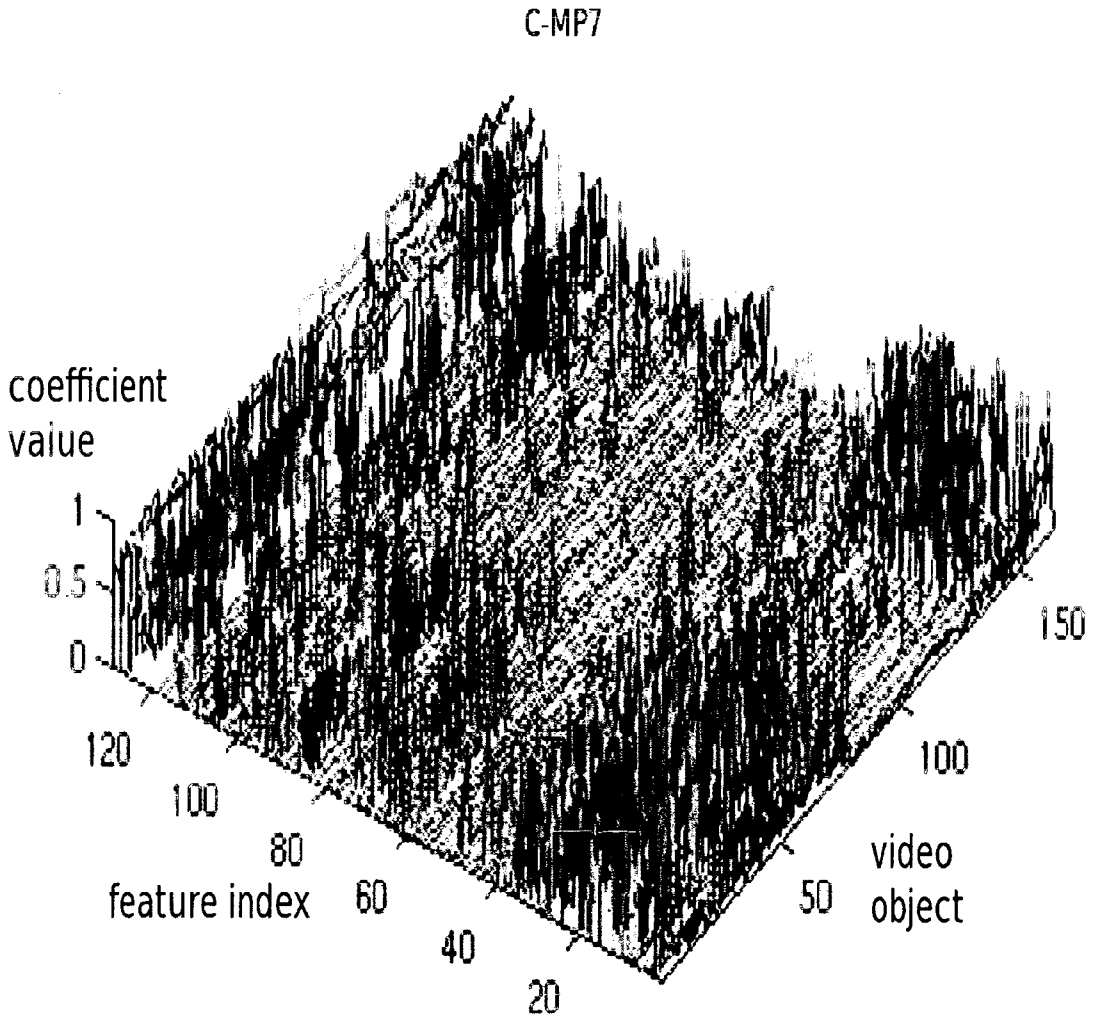


Figure 5.10: Dynamic feature coefficient reduction in video object array for *person* class, from data set 2, with *C-MP7* (low-dimensional chaotic series simulation). From Fig. 5.8, about 70% feature coefficients are null.

5.3.4 Information Measures

A general impression of a data vector can be achieved by calculating statistical properties of (e.g., mean, variance) of the vector [10]. The mean shows the average location of the underlying feature vector. The variance expresses idea on the discriminant. If the variance is near zero, any feature extraction yields same output for any type of given content.

The variance for the *MPEG-7* feature vector, and *C-MP7* are shown in Fig. 5.11 (for high dimensional chaotic series) for all video objects (for data set 16). The variance is higher in *C-MP7* compared to *MPEG-7*. So the discriminancy offered by *C-MP7* is better than *MPEG-7*. Lower variance is observed in *C-MP7* when simulated from *low* dimensional chaotic series (see Fig. 5.11). This lower variance suggests with *low* dimensional chaotic series simulation, the proposed feature binding strips off some useful intel from the original *MPEG-7* visual descriptors. Thus, despite higher % in feature coefficient reduction by *low* dimensional chaotic series simulation (see Fig. 5.8), *high* dimensional chaotic series simulation is preferable in the proposed feature binding. In Section 5.3.5, we further verify this preference between *low*- or *high*-dimensional chaotic series simulation.

The mean for different classes of objects is also shown with *MPEG-7* and *C-MP7* in Figs. 5.12-5.14. It shows that better discriminancy among video object classes is offered from *C-MP7* with high-dimensional chaos simulation than low-dimensional simulation, when compared to that with *MPEG-7*.

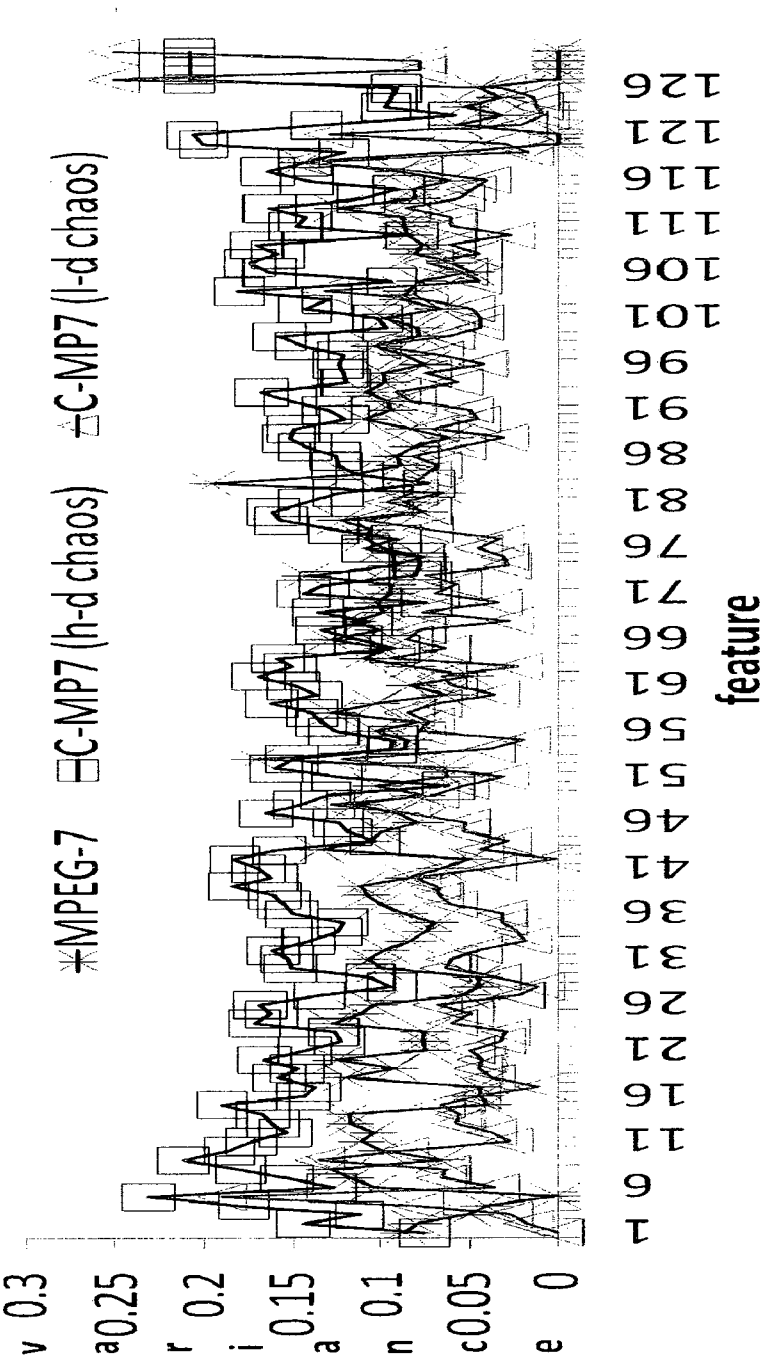


Figure 5.11: Variance for *MPEG-7* and *C-MP7* with low- and high- dimensional chaotic series. All video objects in training, data set 16, is used. The variance is higher in *C-MP7* with high- dimensional chaotic series compared to that of *MPEG-7*. So discriminancy offered by *C-MP7* is better than *MPEG-7*. Lower variance is observed in *C-MP7* with low- dimensional chaotic series, which suggests preference of *C-MP7* with high- dimensional chaotic series as a feature vector.

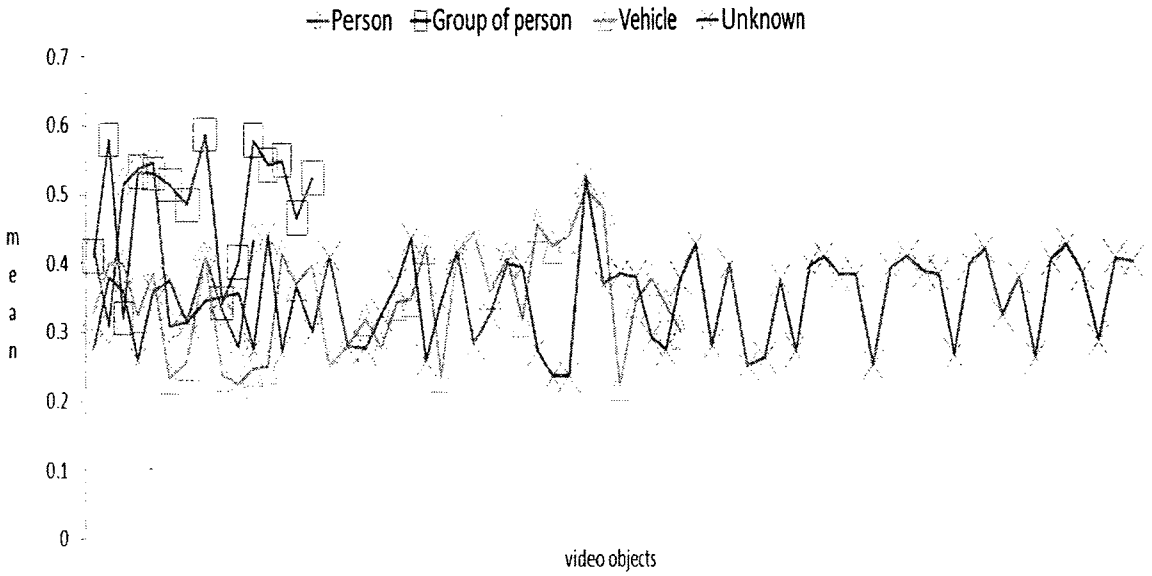


Figure 5.12: Mean of video objects in different classes for *MPEG-7* feature vector, which shows the average location of the underlying feature vector. No clear class separation is evident.

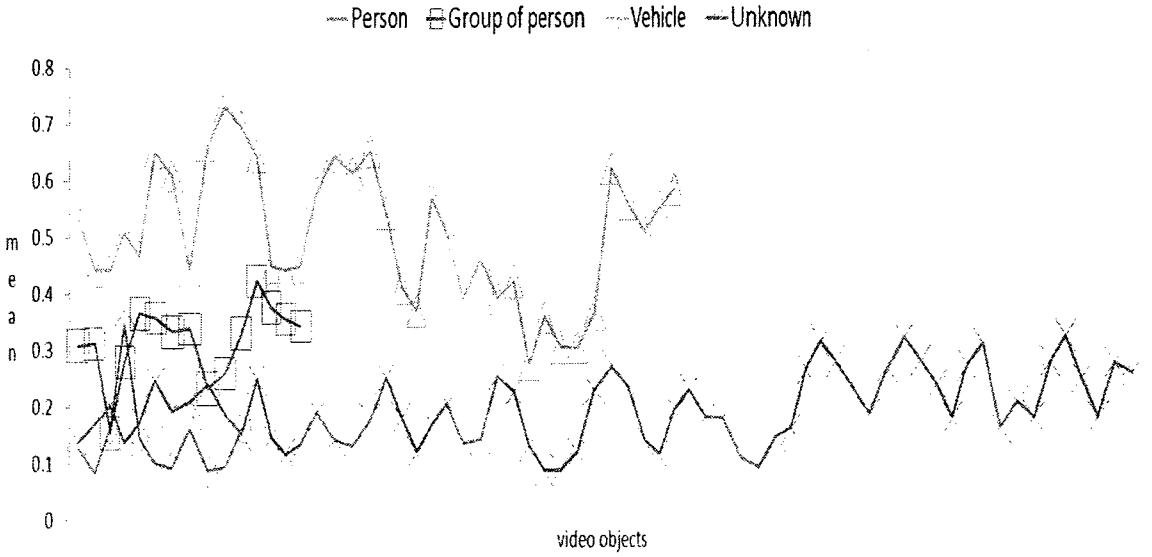


Figure 5.13: Mean of video objects in different classes for *C-MP7* with low- dimensional chaos. Video objects in *vehicle*, *unknown* and *group_of_persons* classes show clear separation on the average location of objects.

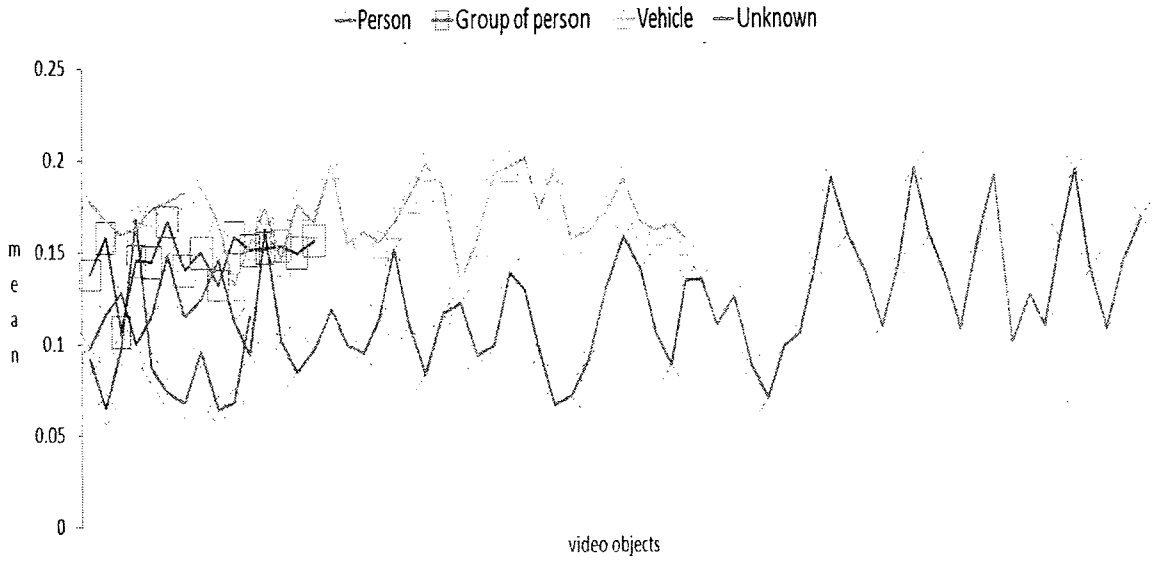


Figure 5.14: Mean of video objects in different classes for *C-MP7* with high- dimensional chaos. Video objects in *person*, *vehicle* and *unknown* classes show distinctive separation on the average location of objects.

5.3.5 Multi-class Discriminant Evaluation

Multi-class Fisher criteria [139] maximizes the distance for inter-class scatter over that for intra-class scatter in different class of video objects. Considering the overall distribution of multiple classes, the weighted mean of two-class Fisher criteria contributes on the selection of the best feature for multi-classes. The μ-class ($\mu > 2$) Fisher criterion can be decomposed into w two-class Fisher criteria (where $w =^{\mu} C_2$), here C indicates the combination of σ-classes as pairs. The inter-class scatter distance f_{AB} between a pair of class, A with E video objects Υ_A and class B with F video objects Υ_B, is,

$$f_{AB} = \frac{\mid \frac{1}{E}\sum_{A=1}^{E}\Upsilon_A - \frac{1}{F}\sum_{B=1}^{F}\Upsilon_B \mid}{\sqrt{\frac{\sum_{A=1}^{E}(\Upsilon_A - \frac{1}{E}\sum_{A=1}^{E}\Upsilon_A)^2}{E-1} + \frac{\sum_{B=1}^{F}(\Upsilon_B - \frac{1}{F}\sum_{B=1}^{F}\Upsilon_B)^2}{F-1}}} \tag{5.8}$$

In a μ-dimensional space, the distance measure is defined as [139],

$$DIS = min(\beta_u) + \lambda\frac{1}{w}\sum_{u=1}^{w}(\beta_u) \tag{5.9}$$

where β_u denotes the u^{th} binary class (e.g., f_{AB}) Fisher criterion among μ classes which is decomposed into w, and λ is an empirical factor. Higher distance *DIS* in Eq. 5.9 implies more discriminant feature vector for the corresponding multi-classes. Fig. 5.15 shows that the minimum inter-class distance, in the four classes we consider, is higher for *C-MP7* (with *high-dimensional* chaos) than that for *MPEG-7* and for *C-MP7* (with *low-dimensional* chaos). Following the suggestions by Fig. 5.11 in Section 5.3.4. Fig. 5.15 here confirms that *high* dimensional chaotic series is preferable for feature excitation in the proposed feature binding over that with *low* dimensional chaotic series.

In Fig. 5.16, Fig. 5.17. and Fig. 5.18 we use multidimensional scaling (Section 2.3) to visualize relative feature vector distances for different video objects in different

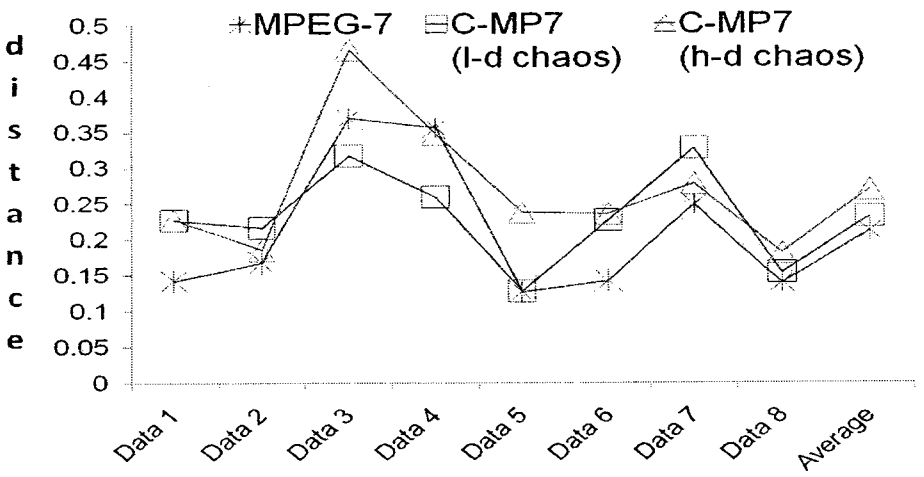


Figure 5.15: Multi-class Fisher criteria for four classes. Minimum inter-class distance is higher for *C-MP7* (with *high-dimensional* chaos) than that for *MPEG-7*, and for *C-MP7* (with *low-dimensional* chaos).

classes for training data set, with *MPEG-7*, *C-MP7* (low-dimensional chaos), and *C-MP7* (high-dimensional chaos), respectively. The classes are same as in the data sets (see Section 5.3.1): *has_person* (p), *has_group_of_persons* (g), *has_vehicle* (v), and *has_unknown* (u). In Fig. 5.16, the video objects of all the classes are overlapped, no clear clustering is visible. In Fig. 5.17 and Fig. 5.18, *C-MP7*, respectively with *low* dimensional and *high* dimensional chaotic series simulation, offers much better clustering for video objects in different classes. The class separation is well for *vehicle* objects in all data sets. However, in Fig. 5.17, few outlier video objects from p and g class overlap in the v cluster. Whereas, p, g and u class clustering is evident but overlapped in Fig. 5.18. The multi-dimensional scaling for 2D visualization in the feature space in Fig. 5.16, Fig. 5.17, and Fig. 5.18. indicate the excellence of *C-MP7* over *MPEG-7* for clustering video objects. This visualization is coherent with the observation in Fig.5.15. However, whether *C-MP7* with *low* dimensional or *high* dimensional chaos is better as a feature vector in specific applications such as video

object classification, that question remains to be answered in Chapter 6.

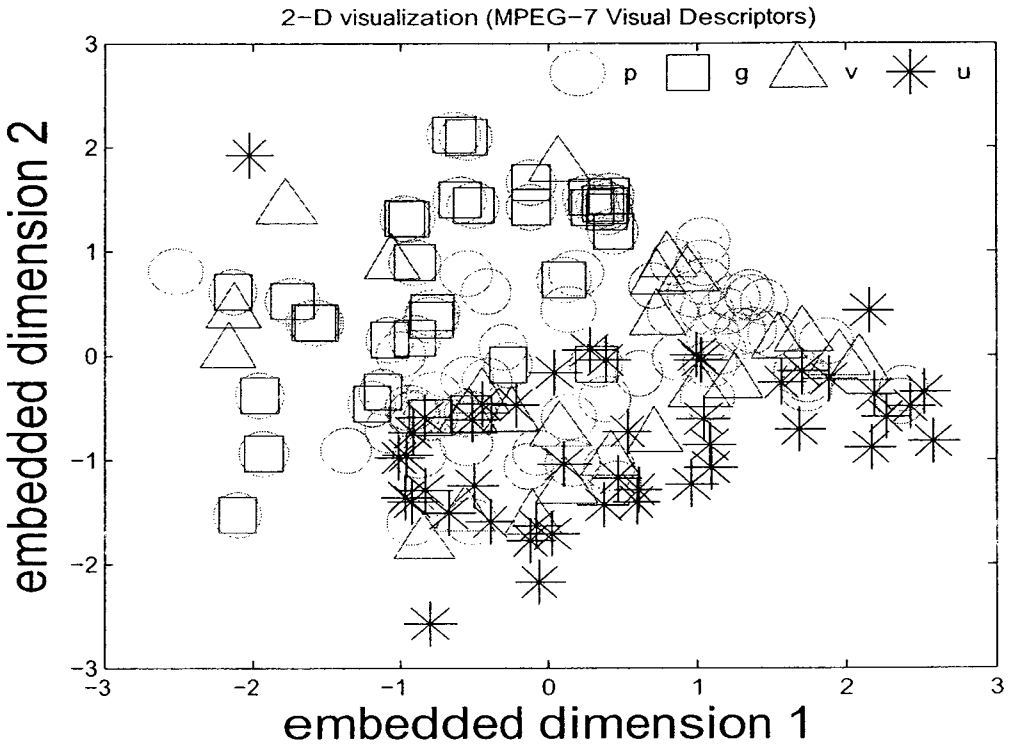


Figure 5.16: Multidimensional scaling to visualize relative feature vector distances for different video objects in different classes for training data set 16, with *MPEG-7* feature vector. Video objects in all classes are overlapped, no clear clustering is visible.

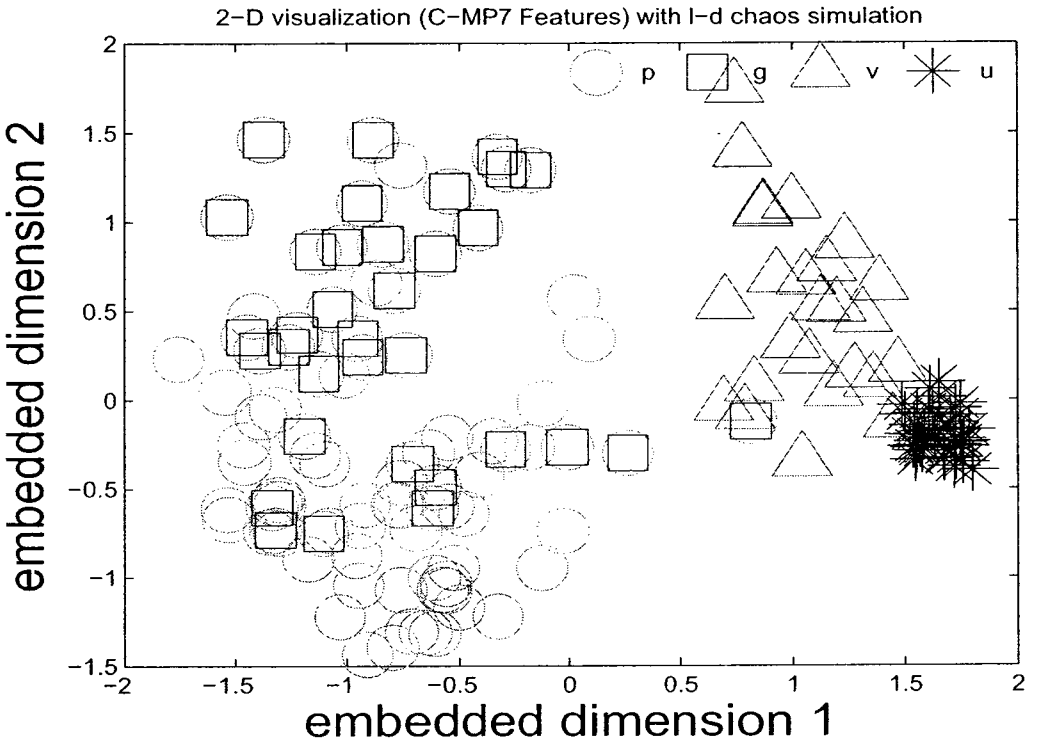


Figure 5.17: Multidimensional scaling for training data set 16, with *C-MP7* (*low-dimensional* chaos). Video objects in all classes are clustered. The class separation is well for *vehicle* objects. Few outlier video objects from p and g class overlap in the v class.

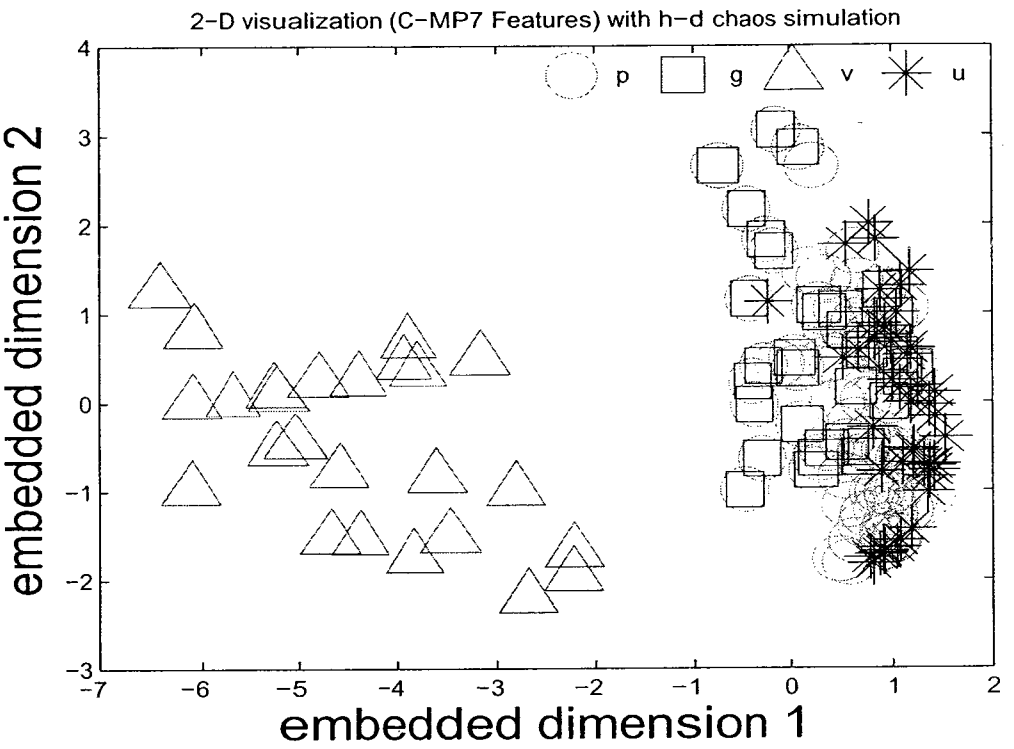


Figure 5.18: Multidimensional scaling (Section 2.3), again, for training data set 16, with *C-MP7* (*high- dimensional* chaos). For p, g and u classes, clustering is evident but overlapped. Best class separation for v class.

5.4 Summary

This chapter presented the proposed method which applies feature binding on coefficients of different *MPEG-7* visual descriptors for video objects. The main contribution of the proposed feature binding is to locate feature coefficients which capture the inherent pattern of video objects for a group or class. A layered multi-dimensional space *binding space*, is assumed which can incorporate different feature spaces of different *MPEG-7* visual descriptors. Each feature coefficients in visual descriptors is simulated, with chaotic series coefficents. The choice of chaotic series is limited to one dimensional finite-difference, and delay differential equation involving single variable. Neighborhood interaction of different chaotic series is calculated with Coupled Map Lattice [24]. Statistical property (e.g., *mean*) of the chaotic series coefficents is then mapped to replace the original *MPEG-7* feature coefficients. A histogram-based clustering is performed on the *means* of these mapped chaotic feature coefficients, to locate indexes of feature coefficents those belong to the largest cluster. The *MPEG-7* feature vector coefficients are finally mapped back to the located indexes. The rest of the feature coefficients at indexes outside the largest cluster are discarded in the new feature vector. We name the new *MPEG-7* compliant feature vector as *C-MP7*, which is supposed to offer discriminancy among video objects in different classes. Two most significant steps in the proposed feture binding method are, *neighborhood attractor interaction* and *histogram analysis*. In Section 6.10.1 we further explain how these steps contribute most to generate robust output feature vector.

In Section 5.3, the statistical analysis of the proposed feature binding method shows, *C-MP7* is *MPEG-7* compliant, and offers dynamic feature reduction from *MPEG-7* visual descriptor based feature vector. The chaotic feature binding reduces descriptor coefficients of video objects in different classes. The mean of *C-MP7* for dif-

ferent video objects show better discriminancy among different classes. The variance in *C-MP7* is higher than that of *MPEG-7* for video objects. Higher variance in *C-MP7* indicates that more diverse video objects can be described in a class. Higher variance in *C-MP7* is advantageous for video objects in surveillance system where challenging image-regions (e.g., incomplete, occluded, different orientations, different resolutions) usually refer to the same video object with same tracking *id* (due to similar motion) in successive frames. Multiple Fisher criteria shows higher minimum class separation between video objects in binary classes with *C-MP7* (high-dimensional chaos) than that with *MPEG-7*.

Chapter 6

Application to Video Object Classification

This chapter reports evaluation of the new feature vector *C-MP7*, with both *low*-dimensional and *high*-dimensional chaotic series, applied to video object classification.

6.1 Introduction

If any generic feature vector describe video objects in different classes successfully, any classifier is supposed to provide some indication ,e.g., high classification accuracy, on the feature vector excellence. To evaluate the usefulness of the new feature vector *C-MP7* in video object classification, different training and test data sets can be used in a multiple binary classifier framework.

For video object classification we use the same pre-labeled data sets, *has_person* (p), *has_group_of_persons* (g), *has_vehicle* (v), and *has_unknown* (u) as described in Section 5.3.1. In surveillance scenes, usually changes in video objects in successive frames are not significant. In cases, where the segmentation is sufficiently good, the

training and testing data sets may share almost similar image regions in successive frames. We select one pruned data set (data set 16) for training, i.e., similar video objects are skipped in each class to reduce any bias which may over fit the training of classifiers. For testing, randomly 80% of each test data set is selected. The classification accuracy is checked against previously labeled (human observation) video objects in test data sets. In the following sections we verify classification accuracy of *C-MP7* for pre-defined classes of video objects when compared to *MPEG-7*.

6.2 Multiple Binary Classifiers

Fig. 6.1 shows the classification setup, where, the input video goes through a video analysis module [132] that produces a list of segmented and tracked video objects. These video objects are first pre-labeled and grouped in different classes. The feature binding is done from the extracted *MPEG-7* visual descriptors in each video object. The feature binding then, derives feature vector *C-MP7* for video objects of each class. Multiple binary classifiers are deployed then for video object classification.

Multiple binary classifiers are used (i.e., one against all) for multiple classes as shown in Fig. 6.2. Here, *MPEG-7* visual descriptors are extracted for video objects as available from segmentation and tracking. *MPEG-7* descriptor coefficients, along with frame id (f_id), object id (o_id) are used to form video object matrix for different training classes. These video object matrix are denoted as YYs in Fig. 6.2. Then chaotic feature binding is applied on each YY. The feature binding produces *C-MP7* matrix for different training classes. Multiple binary classifier is trained with each *C-MP7* matrix. Cross-validation is performed on randomly selected 60% video objects in each *C-MP7* matrix. Then post classification techniques (see Section 6.2.2) are applied to determine class labels for each video object. In cases of new test data

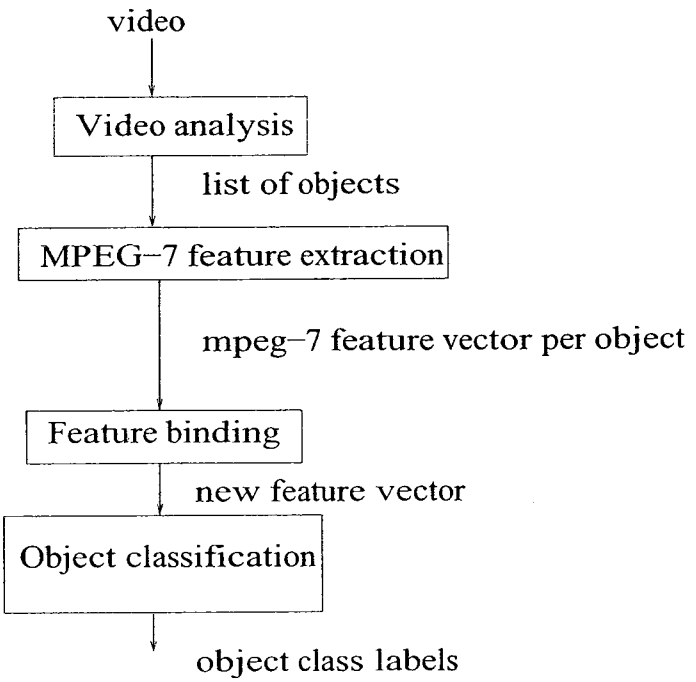


Figure 6.1: Video object classification framework after proposed feature binding.

sets, *C-MP7* matrix is generated using feature binding for multiple video objects in a set of frames. Only the video objects which share the same video object id are used in chaotic feature binding to produce *C-MP7* matrix for those specific video objects. Then the *C-MP7* matrix for each video object in a set of consecutive frames, is tested against earlier generated *C-MP7* matrix of training video objects. The computational overhead for generating *C-MP7* is sharply reduced during testing, because, very few video objects share the same object id in a set of consecutive frames. Thus the video objects matrix YY in chaotic feature binding during testing is much smaller than that during training.

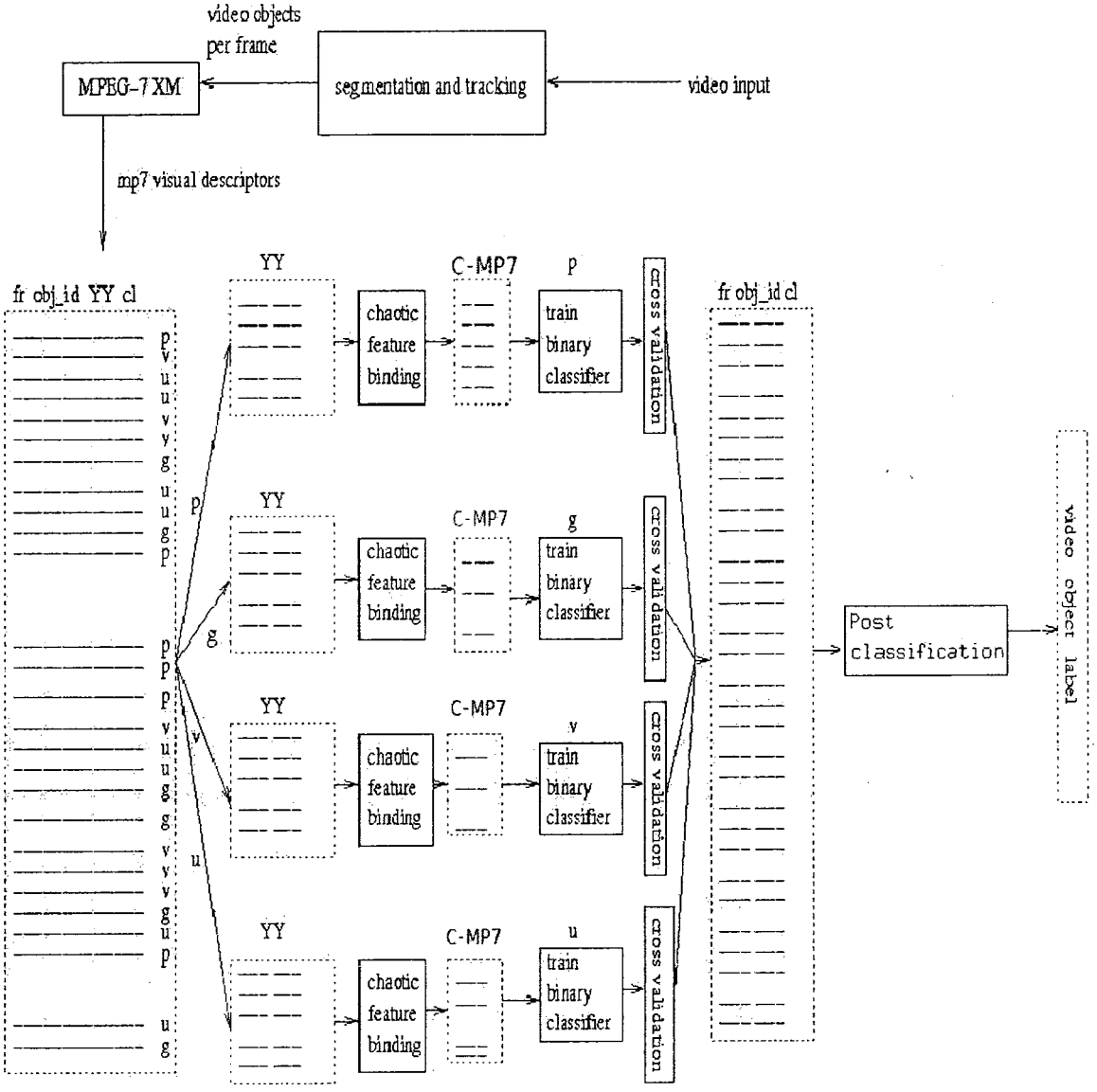


Figure 6.2: Multiple binary classifier design for training and testing.

6.2.1 Choice Of Classifier

Recent work (e.g., [116] [140]) on video object classification shows no specific preferences on the choice of classifiers. Here, we discuss different classifier choices before we decide on the choice of classifier in Fig. 6.2. k-Nearest Neighbor, kNN is commonly used in video object classification [97]. Among other classifiers, decision-tree based classifiers do not perform optimally with *high* dimensional feature vector, and difficulty arises in the manual process of designing the tree which can be time consuming [116]. Bayesian network requires prior knowledge about underlying distribution of the training data, which is difficult to define for incomplete video objects in surveillance. Use of probabilistic modeling, such as, Hidden Markov Models and Gaussian Mixture Models need very large training data, and is very sensitive to object segmentation error and object occlusion. For patterns in sparse *high* dimensional data, Support Vector Machine, SVM performs better than kNN and Naive Bayes classifiers [141]. However, SVM perform badly with many irrelevant features [142]. [113] shows appropriate feature selection can improve accuracy in SVM. The SVM is usually preferred over other classifiers for its high generalization performance. Also SVM does not require prior knowledge and offers better systematic hyperplane margins for non-linear non separable data. Boosting approaches, e.g., hierarchical boosting, lack the correlation among feature coefficients in high dimension. One variant of this approach, AdaBoost, can be a robust option to evaluate feature importance. But, it is usually effective with *low* dimensional feature vector (11 features in [121], 4 features in [131]), where as combination of *MPEG-7* visual features can produce a very *high* dimensional feature vector (e.g., 130 dimensional in our work). The other concern against using AdaBoost with *MPEG-7* visual descriptors is that, the index of coefficients in *MPEG-7* XML schema for object description will be lost. With neu-

ral network, it is tough to make a desired combination of feature sets in a problem domain with diverse training samples [121], e.g., diverse image regions for the same video objects in successive frames.

We verify the classification accuracy for *C-MP7* with three most commonly used classifiers: SVM, kNN, and Back propagation neural network. When compared with *MPEG-7*, *C-MP7* is expected to provide improved classification accuracy irrespective of the choice of classifiers. Different classifiers are only meant to be used as comparison tools. The SVM, kNN and back propagation neural network classifiers are implemented from [143]. A back propagation network with stochastic learning is used with *'5'* hidden units, *'0.1'* convergence criterion and *'0.1'* convergence rate. For the kNN, the choice of k is *'k=3'* and *'k=21'*. For the the kernel choice in the SVM, Radial Basis Function (RBF) is chosen with parameter *'0.5'*, and *'Perceptron'* solver type.

6.2.2 Post-Classification

The tracked segmented image regions from frames after tracking are made available as video objects. The classification of these image regions in a set of successive frames may hints to decide on the corresponding video object's class-label. The difference between image object classification and video object classification is as follow, a video object classifier has to recover a video object from possible segmentation and tracking errors (e.g., occlusions, incompleteness) in successive frames. Whereas in image classification the accuracy can be measured against each image region in each frame. As shown in Fig. 6.3-a, image object regions in frame 3 and 4 may be not trivial to an image classifier to be classified as *person*. But the video classifier has to correctly classify the video object as *person*, if the same tracking id for the video object is

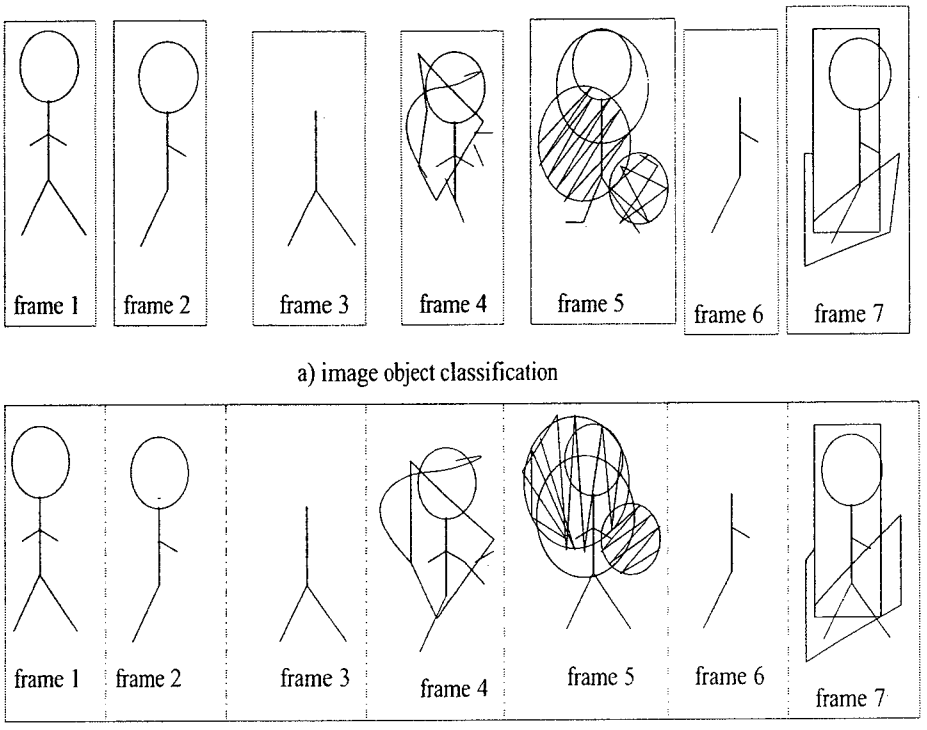


Figure 6.3: Object classification for, a) image and b) video. The accuracy for image object regions in frame 3 and 4 may be not trivial to any image classifier to be labeled as *person*. The video classifier suppose to correctly classify the video object as *person*, if the same tracking *id* for the video object is available from frame 1 through frame 7

available from frame 1 through frame 7.

Considering poor segmentation of video objects in training and test data, e.g., , two post-classification strategies 1) *item label*, during cross-validation and testing and 2) *tracking consistency*, in testing are deployed in multiple binary classifiers, as post-classification correction. Firstly, the result of binary image region classification for each video object in each frame is revisited. Temporary item labels are assigned by each binary classifier. Constant item labels are assigned when image regions solely (i.e., mutually exclusive) belong to a class (e.g., p), as well as does not belong to any other classes (e.g., not g, not v, and not u). For, mutually not exclusive item

Table 6.1: Cross-validation accuracy with 80% data for testing and 20% data for training.

Feature	Back propagation	kNN= 1	kNN= 3	SVM
MPEG-7	57.6%	75.6%	81.1%	78.6%
	on average	73.2%		
C-MP7 (Logistic Map)	56.3%	93.5%	90.0%	92.0%
C-MP7 (Mackey Glass)	89.1%	96.4%	92.9%	96.0%
	on average	87.6%		

labels, temporary item labels are kept from each binary classifier. Video objects with same tracking id (available from tracking) are sub-grouped and are labeled with a constant item label based on majority voting of temporary item-labels from all binary classifiers. Secondly, during testing, tracking consistency is checked for each video object in a set of successive frames before confirming the class label of a video object. The tracking consistency is performed by again applying majority voting of constant item-labels from different frames in the set of successive frames. The set of frames can be formed with number of frames which match with the frame rate of the surveillance system, e.g., 25 frames in a set where frame rate is 25 fps.

6.3 Cross-validation

First for cross-validation, we select 1192 video objects from the ground truth. The hold-out (80% data for testing and 20% data for training) cross validation accuracy for *C-MP7* is compared for each classifier (back propagation neural network, kNN, and SVM) with that of the *MPEG-7* visual descriptor based feature vector. Table 6.1 shows improved (up to 96.4%) cross-validation accuracy with kNN and

SVM classifiers for *C-MP7* when compared to that of the *MPEG-7* feature vector. The cross-validation is shown for two different *C-MP7* created from low-dimensional chaotic series, Logistic map, and from high-dimensional chaotic series, Mackey-Glass equation. Back propagation results, up to 89.1%, show lower improvements (for both *C-MP7* and *MPEG-7*) compared to the same with kNN and SVM. The cross validation accuracy with the *C-MP7* for different classifiers, i.e., back propagation, kNN, and SVM, improves, 87.6% on average, compared to the low, 73.2% on average, validation accuracy with the *MPEG-7* feature vector for video objects in different classes in different data sets.

6.4 Classification Accuracy

In Section 6.3, we show that, better cross-validation accuracy can be achieved using *C-MP7* over *MPEG-7*, with either *high-dimensional* or *low-dimensional* chaotic series. In surveillance scenes, usually changes in video objects in successive frames are not significant. Thus, in cross-validation, the training and testing data sets may share almost similar image regions in successive frames. Thus, further to cross-validation, verification on the classification accuracy is needed here. We present classification accuracy with training and test data sets from completely different scenes, and again observe the performance of *C-MP7* with *high-dimensional* and *low-dimensional* chaotic series. Randomly, one pruned (no repeated object to avoid over fitting in training) data set with few video objects (i.e., 167) is selected for training. In test data sets, repeated objects from subsequent frames are kept.

The new feature *C-MP7*, with *high-dimensional* chaotic series, provide higher classification accuracy than *MPEG-7*. Figures 6.4, 6.5, 6.6, and 6.7 show the classification accuracy for SVM, kNN with k=3, kNN with k=21 and Back Propagation

Neural Network, respectively. Fig. 6.4 shows improved, on average 83% in SVM, accuracy for *C-MP7*, compared to that, on average 62% in SVM, of the *MPEG-7*. In figure 6.7, Back propagation results show poor and inconsistent performance for both *C-MP7* and *MPEG-7*. The average accuracy in SVM with *C-MP7*, *low-dimensional* chaotic series, is reduced to 39.6%. Similar performance reduction is observed in kNN for *k=3* (Fig. 6.5) and *k=21* (Fig. 6.6). In both SVM and kNN, *C-MP7* perform well with *high-dimensional* chaotic series.

In addition to over all classification performance in all classes, above 99% accuracy is achieved for only *vehicles* against other objects in binary SVM classifier, for *C-MP7* with *high-dimensional* chaotic series. The corresponding accuracy for *MPEG-7* is on average 84%. The said *vehicle* accuracy complements the well clustered feature vector description of *vehicles* in Fig. 5.16, Fig. 5.17, and Fig. 5.18 in Section 5.3.5.

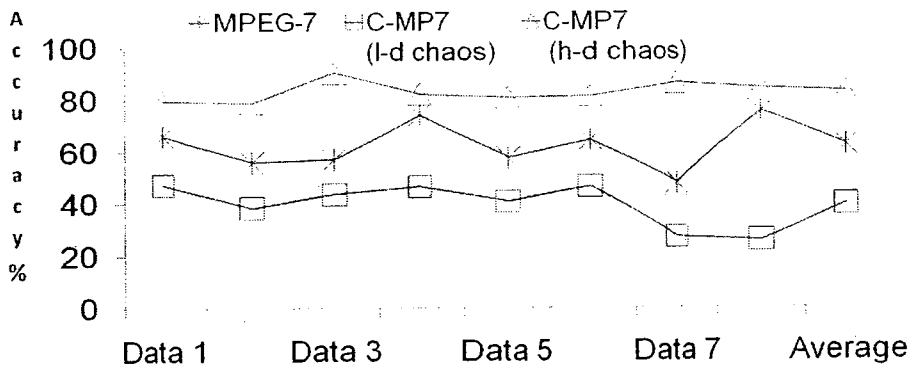


Figure 6.4: Classification accuracy with SVM. On average 83% accuracy improved for *C-MP7* with *h-d* chaotic series, compared to, on average 62% with *MPEG-7*

The classification accuracy offered by *C-MP7*, in this thesis, is sufficiently high despite the inclusion of challenging objects. Compared to *C-MP7*, related work, with similar class of video objects, either reports poor accuracy, or does not include challenging objects when reports higher accuracy. Challenging video objects include, video objects with poor segmentation, lack of invariance in scale, lighting, and ori-

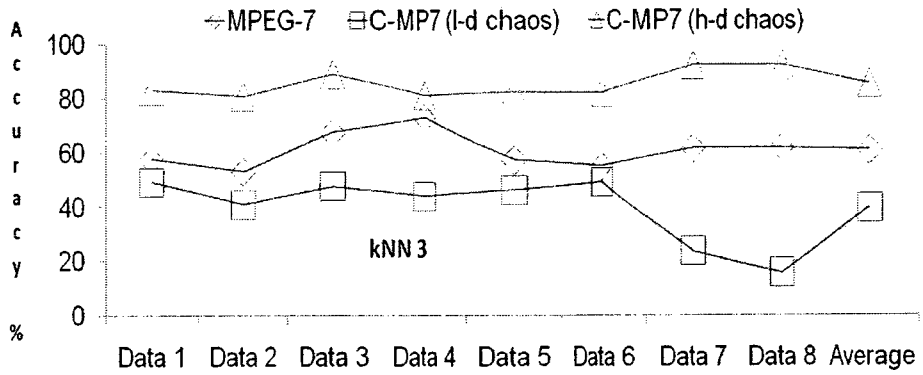


Figure 6.5: Classification accuracy with kNN, where k=3. *C-MP7* perform well with *h-d* chaotic series over *MPEG-7*. However, *C-MP7* performs poor with *l-d* chaotic series

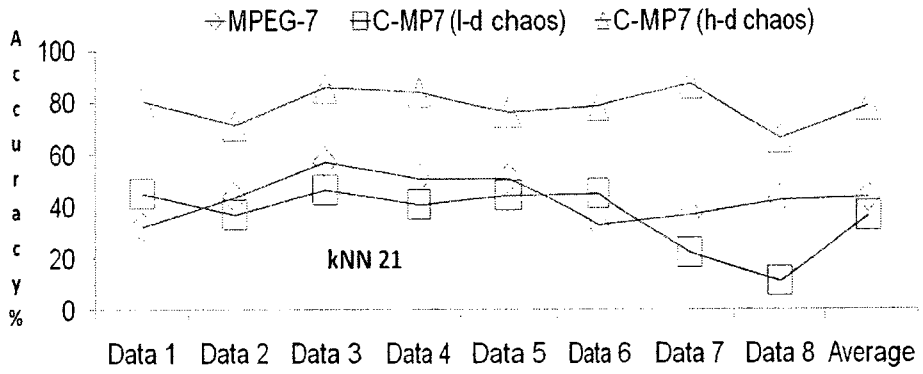


Figure 6.6: Classification accuracy with kNN, where k=21. Again, *C-MP7* provides better accuracy with *h-d* chaotic series compared to that with *MPEG-7*

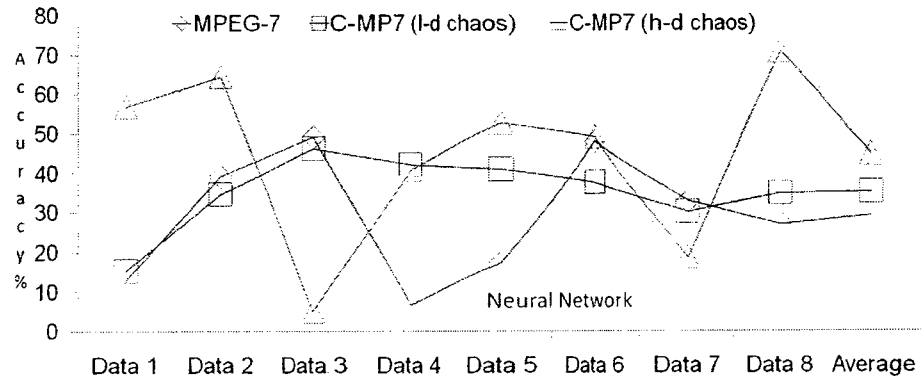


Figure 6.7: Classification accuracy with Back propagation neural network. Poor and inconsistent accuracy for both *C-MP7* and *MPEG-7*

entation of objects, existence of parts of background or other video objects inside segmented objects, and existence of parts of background or other video objects inside segmented objects (see Section 1.2). [124] uses Multi-block Local Binary Pattern (LBP) feature for data sets with good segmentation, and reports accuracy in different classes as follow: *bugs* 60%, *persons* 85%, *cars* 92%, *vans* 76%, *trucks* 71%, *bikes* 78%, *group of persons* 75%. The average accuracy considering all classes of video objects is 76.71%. [97, 115] reports, 98% accuracy for vehicle and human class. This accuracy is achieved using arbitrary camera viewpoints in training to distinguish humans from vehicles. Non-*MPEG-7* features on shape, motion and periodicity are used [97, 115] with video objects which do not undergo merge/split, occlusion, or do not lie on image border. Four camera views are used for testing data where video objects from other camera view is used in training. [120], uses blob region distance parameters as feature in a hierarchical multi-level neural network, and reports high accuracy for *car* 98%, but poor accuracy for *cycles* 9%, *pedestrian* 11%. [144] reports very high (above 97.23%) accuracy for *people* and *not people* binary classification. [144] uses posture features from image blob silhouette or human skeleton. Objects in *person* class used in training is the same as testing, but with different clothing and in different conditions. The skeleton is extracted from the blob by means of morphological operations and processed using a HMM framework, where the accuracy is very sensitive to segmentation errors, in particular to occlusions.

On the use of MPEG-7, [41] attains 91% cross validation accuracy to classify one class of objects, i.e., *ships* in coastline surveillance with *MPEG-7* region-based shape descriptor. [113] reports over 90% accuracy to classify video shots with *MPEG-7* region shape, homogeneous texture, color layout, color structure, and edge histogram visual descriptors along with other non-*MPEG-7* features. In this thesis, we select the set of visual descriptors which offer fast extraction, interoperability, are of compact

size, and are scale and resolution invariant. The selected descriptors are color layout, edge histogram, region shape, and contour shape (see Chapter 4 for more details).

6.5 Which Classifier Gives Better Accuracy?

The multiple binary classifier design in Fig. 6.2 can use any classifier. We verify the classification accuracy for *C-MP7* with three most commonly used classifiers: SVM, kNN, and Back propagation neural network. To find out which classifier gives better accuracy, we observe the binary classifier performance for different classifiers for p, g, v, and u classes. Figures. 6.8, and 6.9 show different classifier's cross-validation accuracy in data set 16 for binary classes (i.e., p and *not* p, g and *not* g, v and *not* v, and, u and *not* u). Post-classification strategies mentioned in Section 6.2, are applied on the outputs of these multiple binary classifiers before finalizing the class labels of video objects. The classifiers which perform best for binary classes are as follow, kNN 21 for p, kNN 21 for g, SVM for v, and SVM for u. This observation opens a scope to use hybrid classifiers in the multiple classifier design in Fig. 6.2. Fig. 6.10 shows the result with such hybrid classifiers, after applying post classification strategies. The hybrid classifier design perform similar as SVM and kNN, k=3 (Figs 6.4- 6.5).

To decide which classifier gives better performance in the multiple binary classifier framework, in Fig. 6.11, we present the cross-validation accuracy in data set 16 for different classifiers with *MPEG-7*, *C-MP7* (high-dimensional) chaotic series, It shows, the accuracy is on average above 80%, with *C-MP7* (high dimensional chaotic series) with SVM and kNN3.

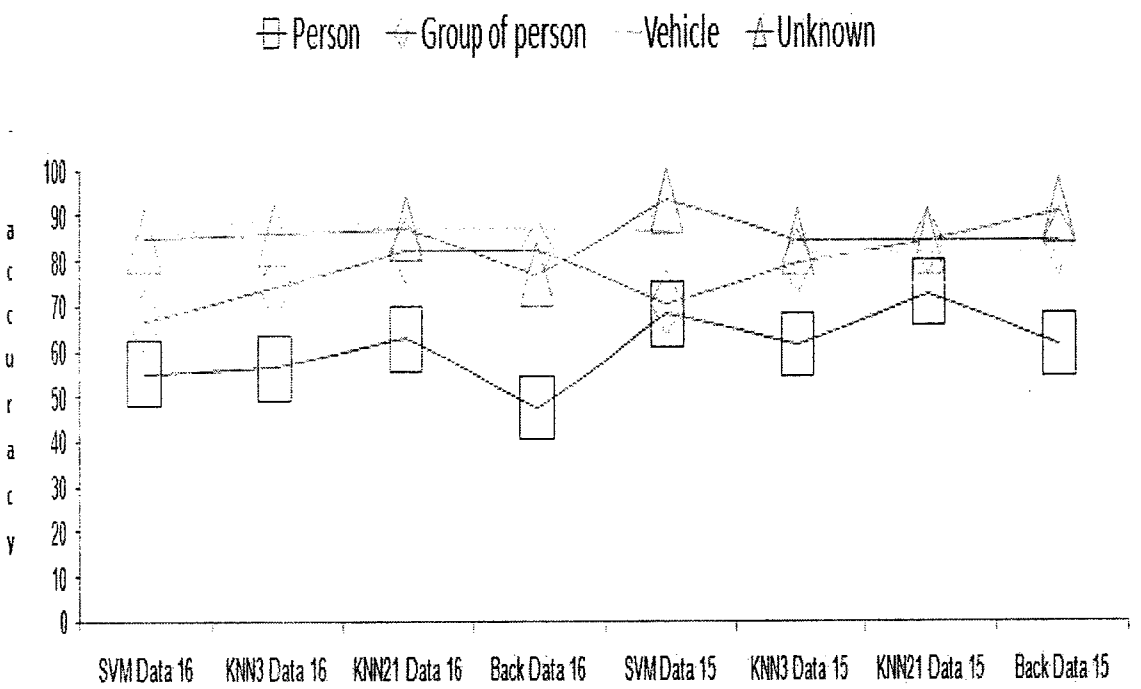


Figure 6.8: Cross validation accuracy in binary classifiers with C-MP7 (l-d chaos) for SVM, kNN, and Back propagation neural network classifiers. Here, binary classification is done on: p and *not* p, g and *not* g, v and *not* v, and, u and *not* u.

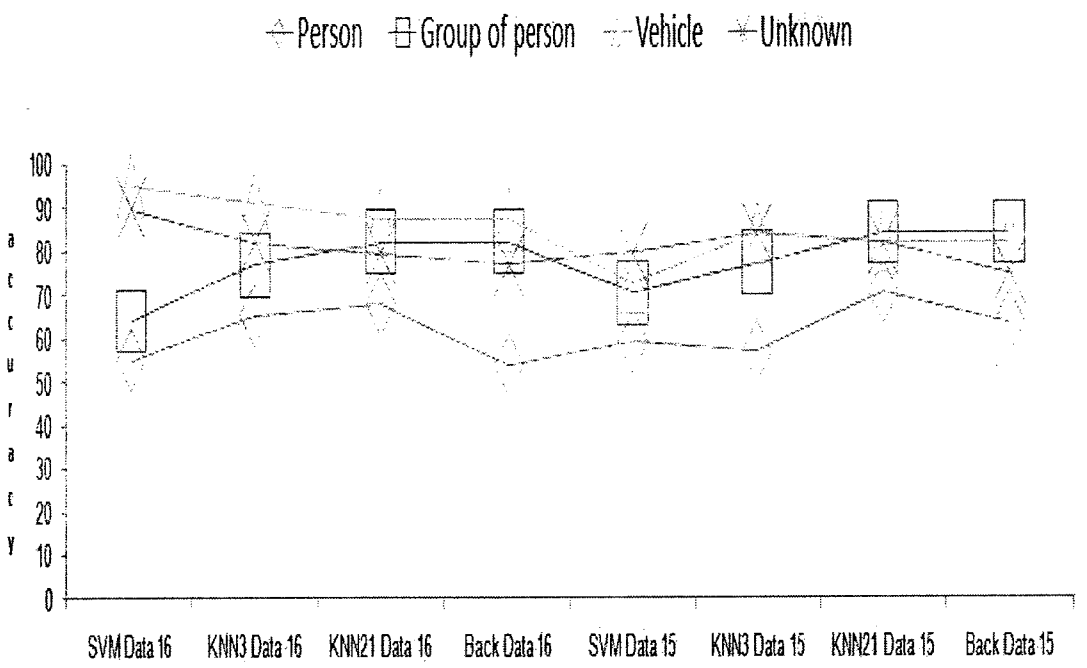


Figure 6.9: Cross-validation accuracy in binary classifiers with C-MP7 (h-d chaos) for SVM, kNN, and Back propagation neural network classifiers. kNN, k=21, performs well for p, kNN 21 for g, SVM for v, and SVM for u class.

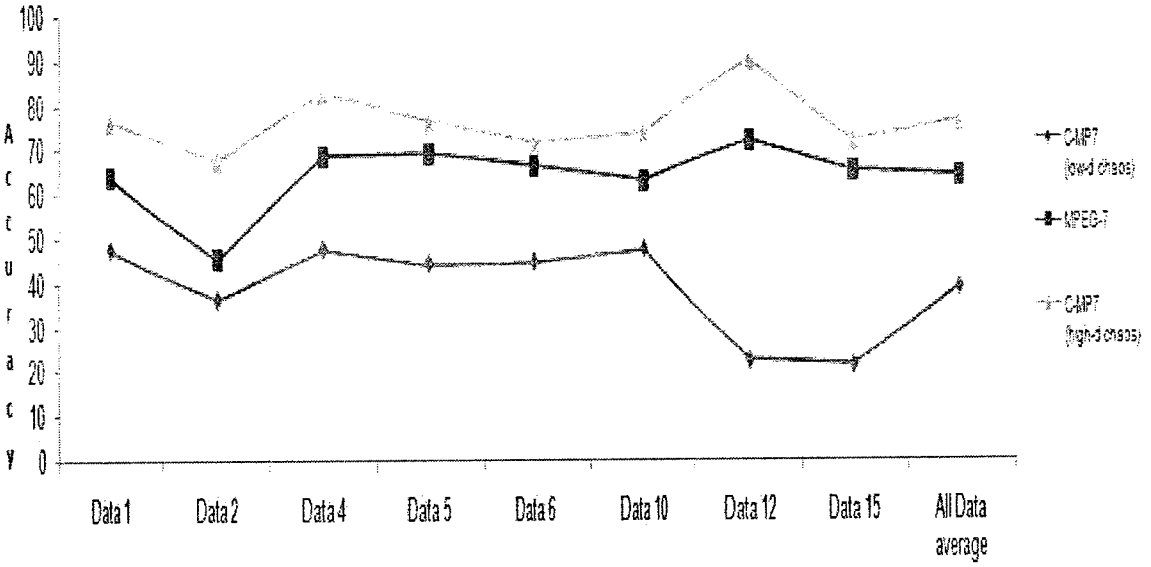


Figure 6.10: Cross-validation accuracy in hybrid classifiers. The hybrid classifier design perform similar as SVM and kNN (k=3), see Figs 6.4- 6.5.

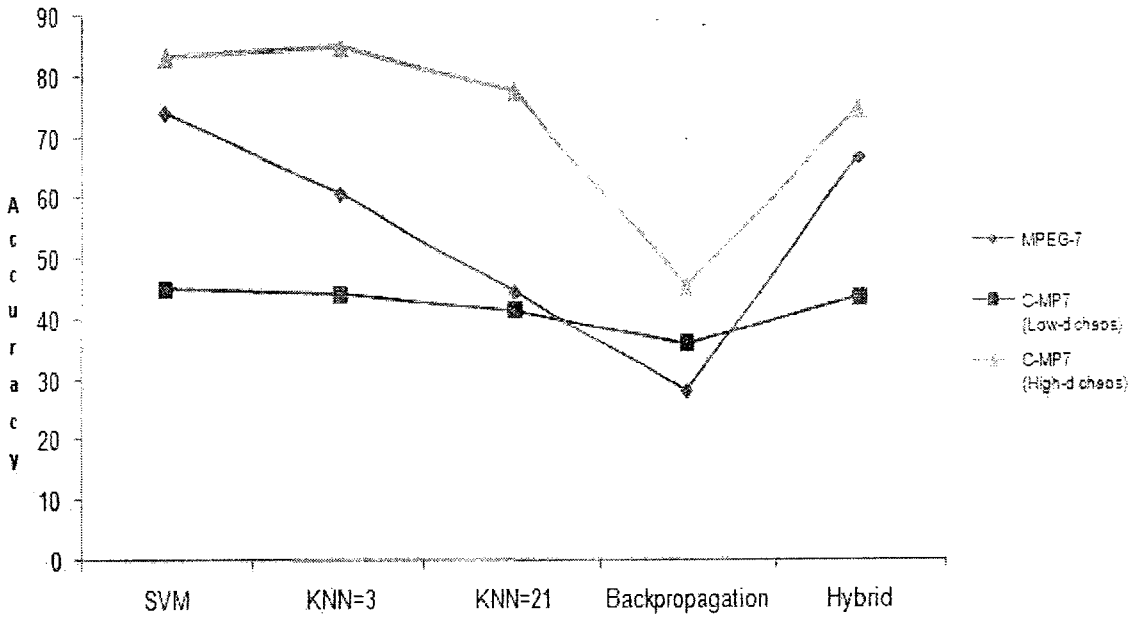


Figure 6.11: Cross-validation accuracy in data set 16 for different classifiers with *MPEG-7* and *C-MP7* (high-dimensional) chaotic series. The accuracy is on average above 80%, for *C-MP7* with high dimensional chaotic series in SVM and kNN, k=3.

Table 6.2: Continuous outdoor shots (in **bold**), and indoor shots. The shots are not shown to the classifier during training.

data set	details
data set 10 (training)	video objects (147): p=27, g=16, v=38, u=66
data set 12 (training)	video objects (500): p=129, g=35, v=210, u=126
data set 13 (training)	video objects (698): p=243, g=85, v=123, u=247
data set 16 (training)	video objects (159): p=55, g=34, v=25, u=45
data set 18 (testing)	frames (930) and video objects: **Bishop2entrance**=403, comm2org=227, **Road2**=300
data set 19 (testing)	frames (2438)and video objects: **concordia1**=282, ekrlb=327, hall=300, **July25PeopleBetter5**=587, **road3**=342, sit=600

6.6 Continuous Video Shot Testing

To further verify the classification accuracy of *C-MP7* in multiple classifier design, new video sequences are tested. These sequences are not shown to the classifier during training. A pruned data set, data set 16, is used for training. Video objects for the test data sets are created from the video analysis of successive frames of new sequences. In Fig. 6.12, we present classification accuracy of video objects in these scenes. These scenes are taken from successive frames of random day surveillance video, and also from offline videos, as shown in Table. 6.2. In successive frames, video objects with same tracking ID, are sub-grouped for generating *C-MP7*, and then testing. Fig. 6.12 shows higher accuracy for *CMP-7* with high-dimensional chaotic series over that with *MPEG-7* .

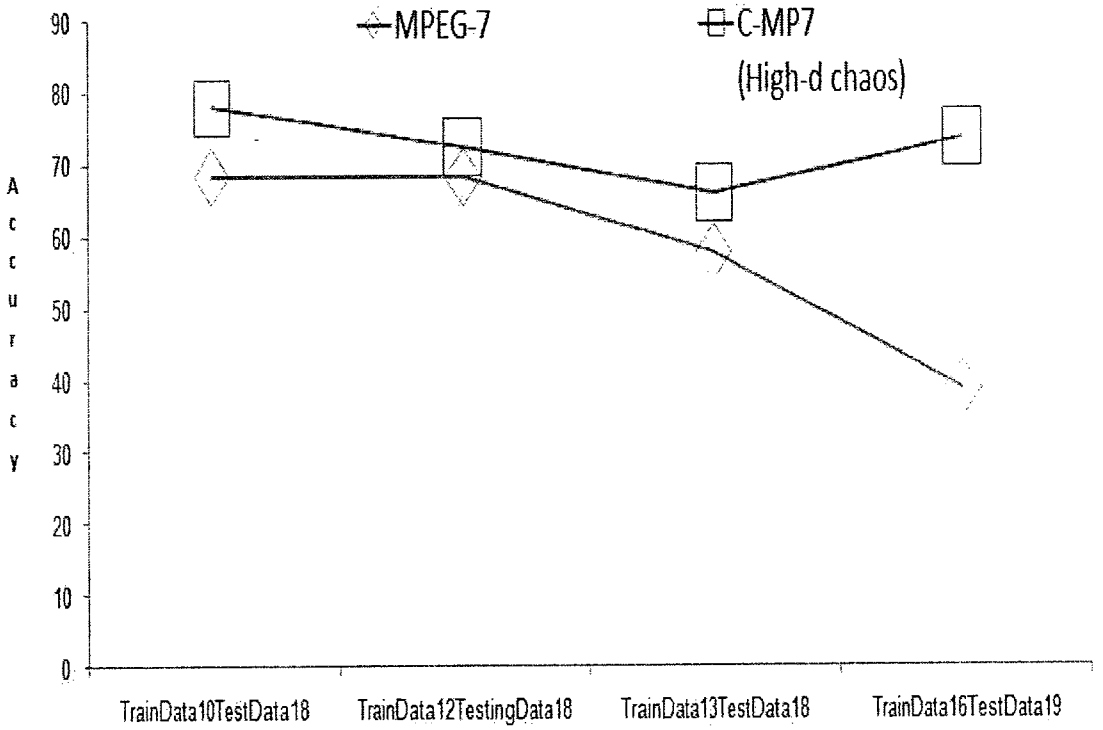


Figure 6.12: Test on video objects from successive frames. As the number of frames increases (test data set 19), higher classification accuracy is achieved for *CMP-7* with high-dimensional chaotic series over that with *MPEG-7*.

6.7 Accuracy With PCA Of *MPEG-7* ?

There are two techniques for dimensionality reduction in a feature vector [137]: one is to select a limited set of features out of the total set and the other is to extract a smaller set of features as linear or nonlinear functions of the original set of features using discriminant analysis, e.g., Principal Component Analysis (PCA) [138]. We follow the former technique because, the video objects of interest are often partially specified, which makes the task of obtaining transformed features using discriminant analysis a non-trivial exercise [113]. This is specially true for surveillance applications where segmentation and tracking usually output incomplete video objects. The use of a linear or non-linear function to reduce the set of *MPEG-7* visual descriptor coefficients during feature binding, adds the risk of loosing useful intel in describing different video objects from the available descriptors. This possible risk is implied in Fig. 6.13, which shows that, in SVM, the PCA of *MPEG-7* provides inconsistent and poor classification accuracy in different data sets. This is specifically expected because, when PCA is used for classification, it does not account for class separability, it makes no use of class labels of the underlying features. There is no guarantee that the direction of maximum variance (as calculated in PCA) will contain good feature for discrimination [137].

The poor accuracy with PCA, compliments the usefulness of the proposed chaotic feature binding. Also, with the use of PCA, the index of the *MPEG-7* coefficients in the XML description for video objects is lost.

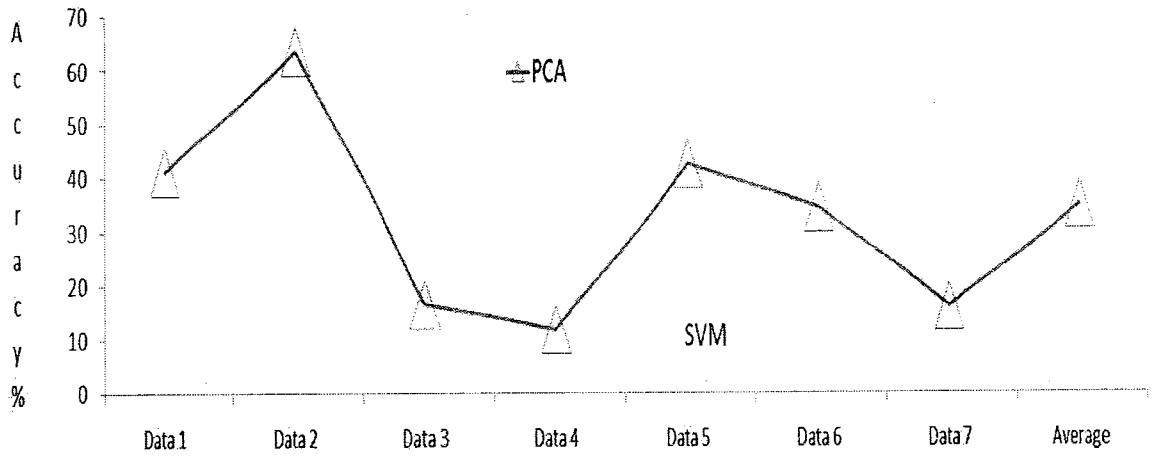


Figure 6.13: Cross-validation accuracy in SVM, with PCA of *MPEG-7*. Poor and inconsistent classification accuracy in different data sets. This is expected because, when PCA is used for classification, it does not account for class separability.

6.8 Sub-group Classification?

In this thesis we focus on generalized descriptions of primary video contents for video object classification in four commonly [21, 22] used classes in surveillance shots, namely, *has_person*, *has_group_of_person*, *has_vehicle*, and *has_unknown*. The scopes for sub-classification for sub-set of video objects in each class with *C-MP7*, are yet to be explored. Due to specific interests to classify *male* and *female* video objects [140,145], in this section, we include an initial observation on such sub-group classification. The *male* and *female* sub-set of video objects are considered from the data sets of *has_person* class. We also introduce a by-product feature vector Z as an output from the proposed feature binding method, which is different than *C-MP7*. We report subjective evaluations for these sub-set of video objects with *MPEG-7*, *C-MP7*, and Z.

Z is generated by not replacing the MPEG-7 feature coefficients, but keeping the chaotic feature coefficients after the *'histogram-based clustering'* step in Fig. 5.4 in Chapter 5. The difference between *C-MP7* and Z is in their feature coefficients. In *C-MP7*, the indexes of the largest cluster of chaotic coefficients is located in the video object array of corresponding class, and these indexes are mapped back to original *MPEG-7* coefficients The rest of the indexes, other than the largest cluster in histogram bins, are made null. Thus, *C-MP7* contains *MPEG-7* feature coefficients However, instead of mapping back to the *MPEG-7* feature coefficients, if the chaotic feature coefficients are kept as feature vector, and the rest of the indexes in the video object array, other than the largest cluster in histogram bins, are made null, then we get another new feature vector Z. This is a by-product feature vector, which is not *MPEG-7* complaint, and formed from the chaotic feature coefficients in the *'reconstruction of attractors'* step (see Section 5.2.4 in Chapter 5). We find that, the

classification accuracy for video objects with Z does not improve over *MPEG-7* as can be achieved for the same with *C-MP7* (h-d chaotic series). Thus, Z is not a good candidate for generalized descriptions of video objects.

However, subjective observations on same video object in different frames in a shot with Z shows similar pattern for *male* or *female* subset of video objects. Z with *l-d* chaotic series shows similar pattern for a subset of video objects, e.g., *male*s, and simulteniously, the pattern is distinctive than the other subset of video objects, e.g., *female*s. Figs. 6.14, 6.15, 6.16, and 6.17 shows such descriptions for *male* and *female* objects from different frames in different shots. We present, *C-MP7* with *l-d* chaotic series in Fig. 6.14, Z with *l-d* chaotic series in Fig. 6.15, *C-MP7* with *h-d* chaotic series in Fig. 6.16, and Z with *h-d* chaotic series in Fig. 6.17.

Interpreting Fig. 6.15 and Fig. 6.17, we see, in Fig. 6.15, Z using *l-d* chaotic series exhibits set of close to identical patterns for individual descriptions of *male* video objects, and another set of close to identical patterns for individual descriptions of *female* video objects. However, the patterns are distinctive between two different sets of *male* and *female* video objects. The patterns in Fig. 6.15 are more specialized than Z using *h-d* chaotic series in Fig. 6.17. The patterns in Fig. 6.14 and Fig. 6.16 for *C-MP7* with *l-d* chaos and *h-d* chaos are more generalized, and represents both *male* and *female* set of video objects as subsets of *has_person* class.

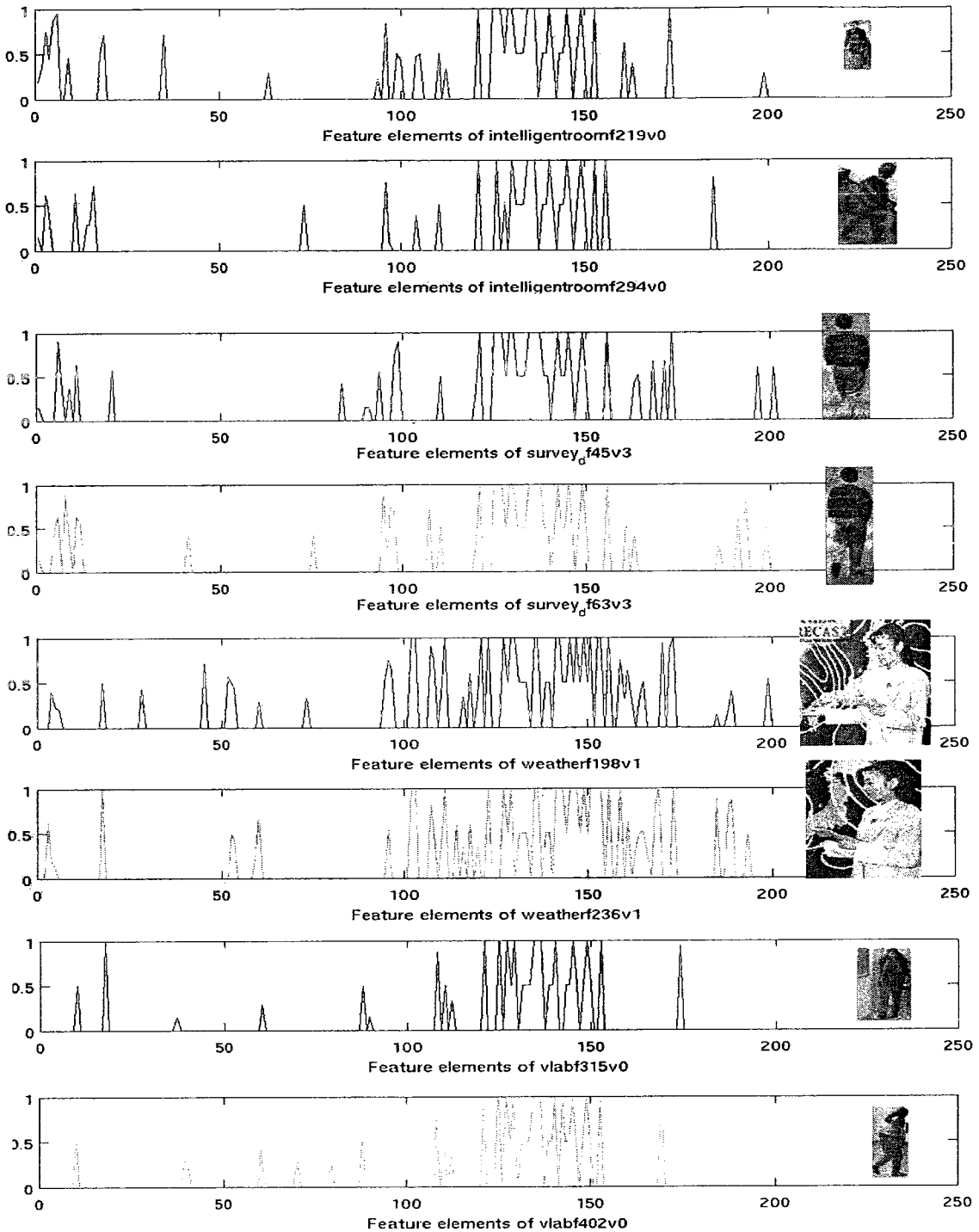


Figure 6.14: *C-MP7* (with *l-d* chaos) for same video objects in different frames from different shots. *Male* object set is shown from *intelligent room* (1,2 from top) and *survey* (3, 4 from top) shots. *Female* object set is shown from *weather* (5,6 from top) and *vlab* (7, 8 from top) shots.

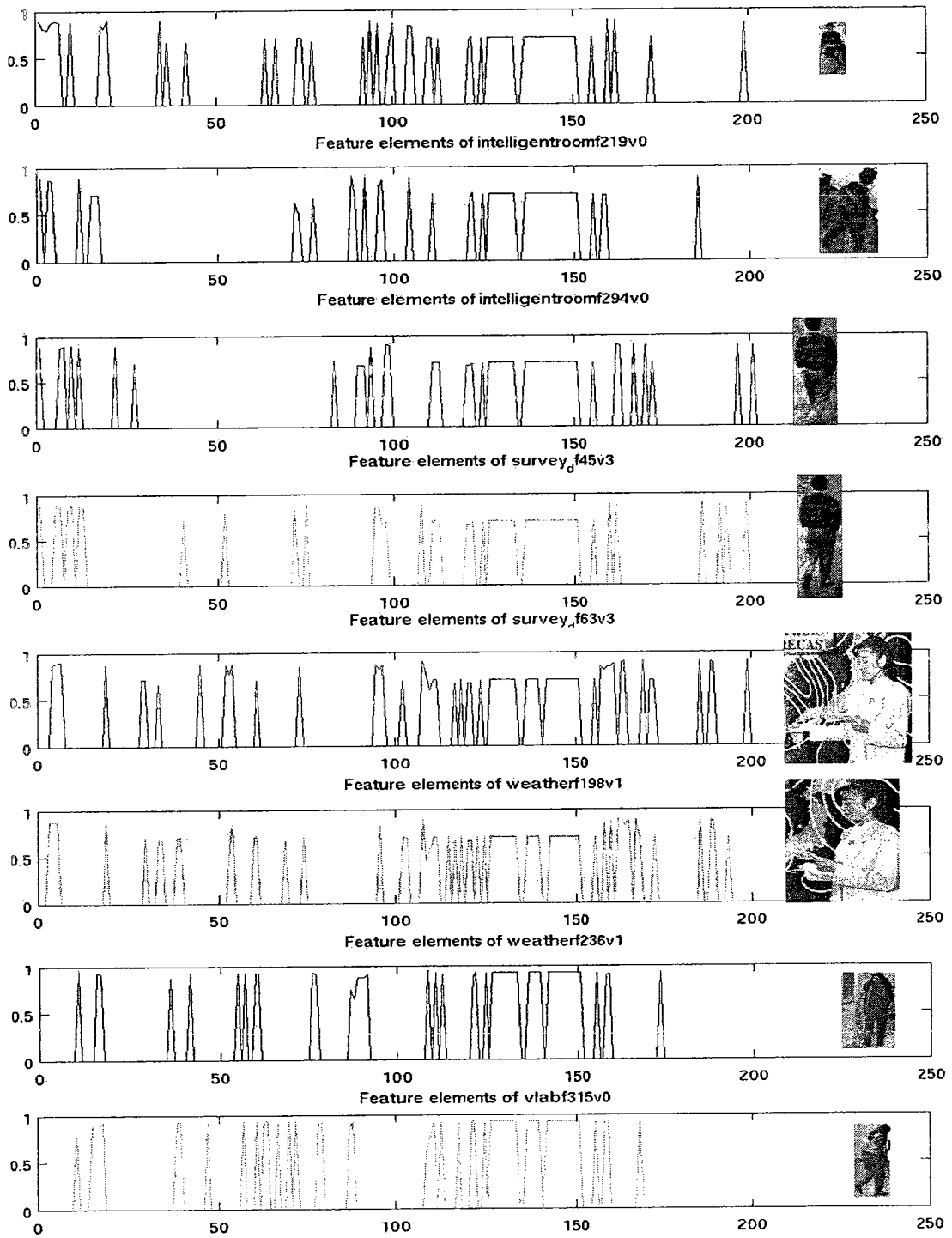


Figure 6.15: Z (with *l-d* chaos) for same video objects in different frames from different shots. *Male* object set is shown from *intelligent room* (1,2 from top) and *survey* (3, 4 from top) shots. *Female* object set is shown from *weather* (5,6 from top) and *vlab* (7, 8 from top) shots.

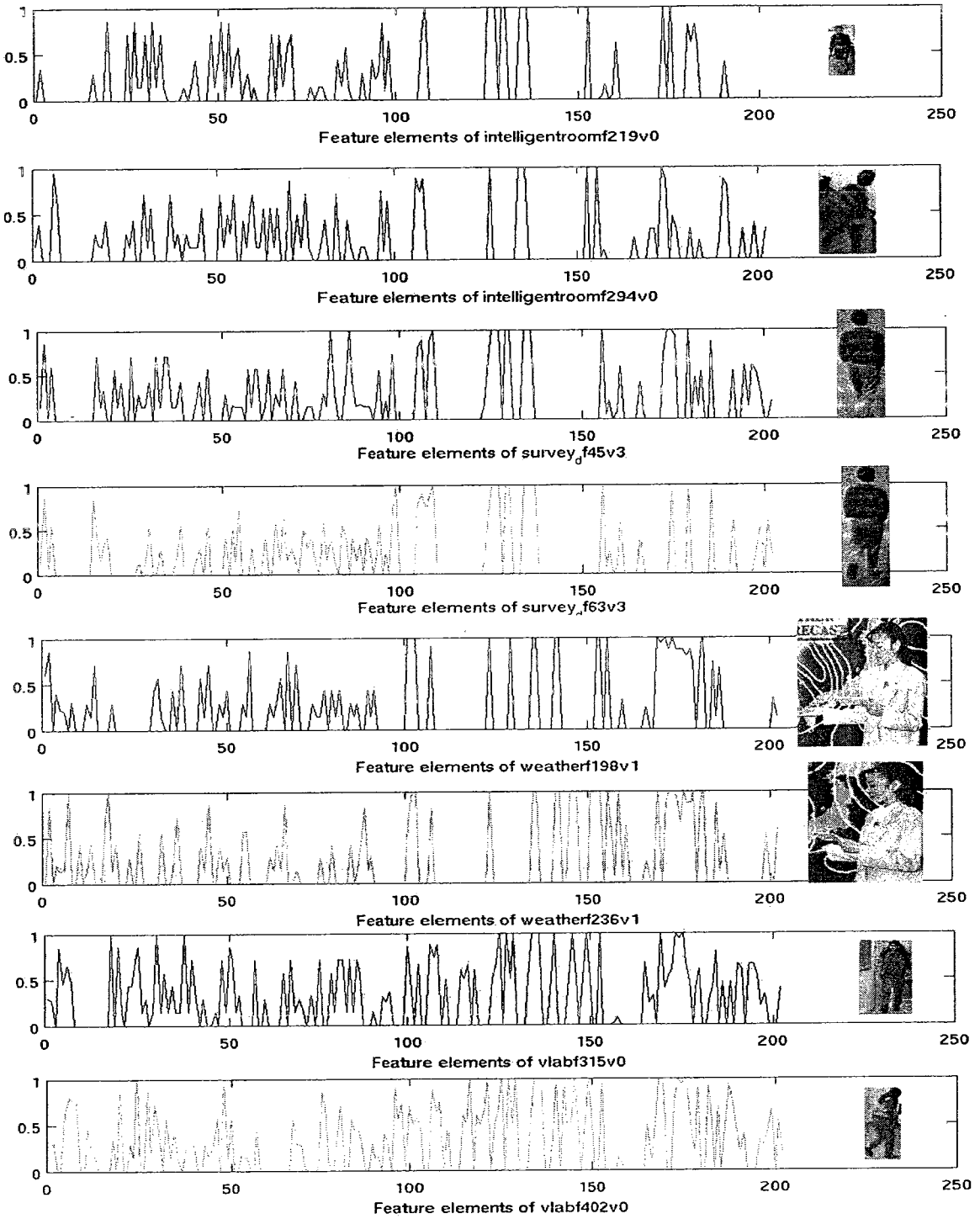


Figure 6.16: *C-MP7* (with *h-d* chaos) for same video objects in different frames from different shots. *Male* object set is shown from *intelligent room* (1,2 from top) and *survey* (3, 4 from top) shots. *Female* object set is shown from *weather* (5,6 from top) and *vlab* (7, 8 from top) shots.

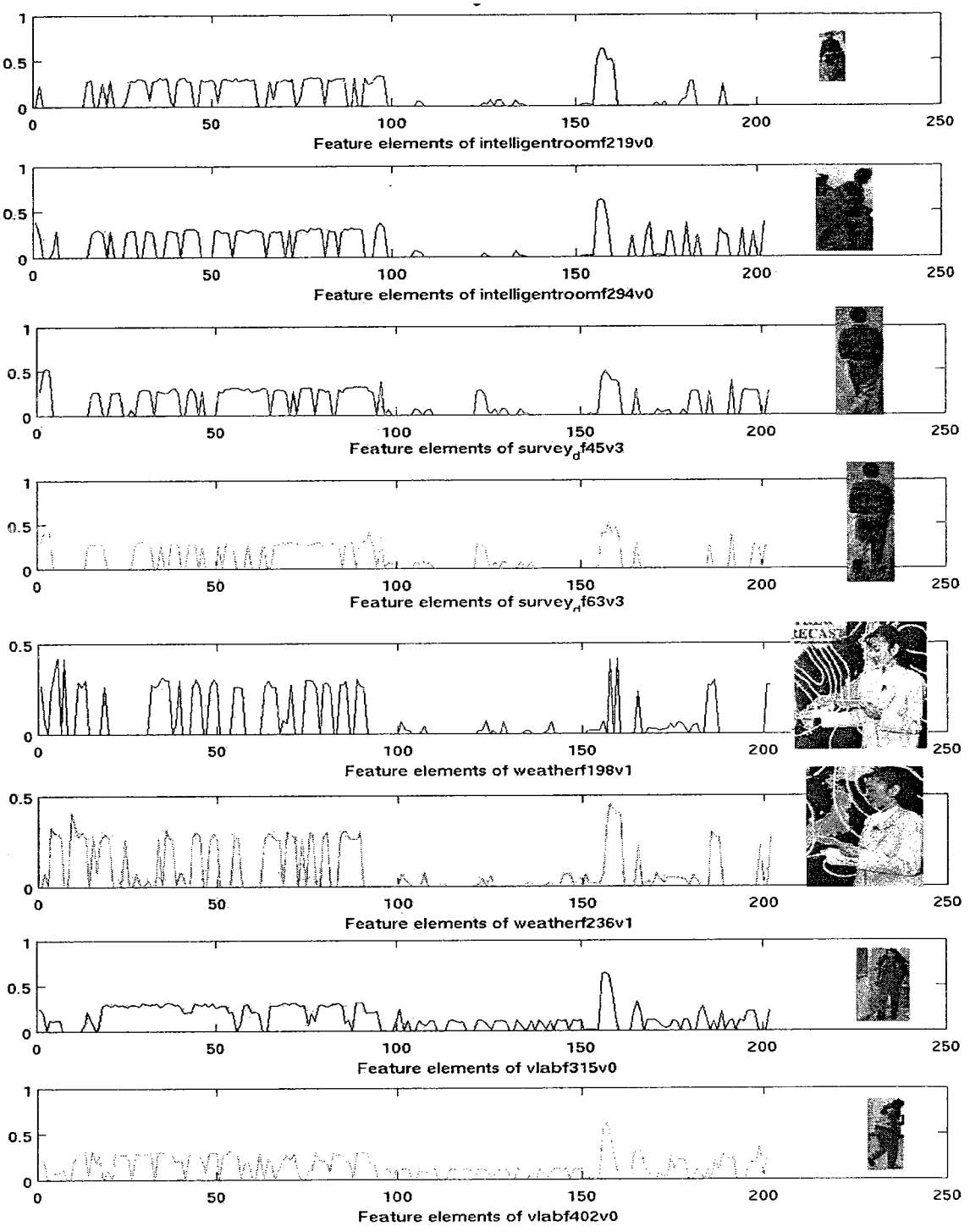


Figure 6.17: Z (with h-d chaos) for same video objects in different frames from different shots. *Male* object set is shown from *intelligent room* (1,2 from top) and *survey* (3, 4 from top) shots. *Female* object set is shown from *weather* (5,6 from top) and *vlab* (7, 8 from top) shots.

6.9 Robustness In *C-MP7*

We test the robustness of *C-MP7* as a feature vector in classification in three scenarios. First, for both *C-MP7* and *MPEG-7*, we observe the classification accuracy with diverse test data sets which include poorly segmented video objects. Secondly, we observe the classification accuracy, for both *C-MP7* and *MPEG-7*, with random training data sets and random test data sets. Finally, we observe the effect of bias in feature vector due to the existence of similar repeated video objects in both training and test data sets. Very similar video objects often appear in successive frames of any surveillance shot. These similar video objects offer minimal discriminant feature in corresponding class. In all three scenarios above, we find, the *C-MP7* is more robust than *MPEG-7*. Specifically, the *C-MP7* from high-dimensional chaotic series simulation is more robust than *MPEG-7*, however, the *C-MP7* from low-dimensional chaotic series simulation is not robust. In the following subsection we discuss our observation on robustness of *C-MP7*.

6.9.1 Accuracy Under Poor Segmentation

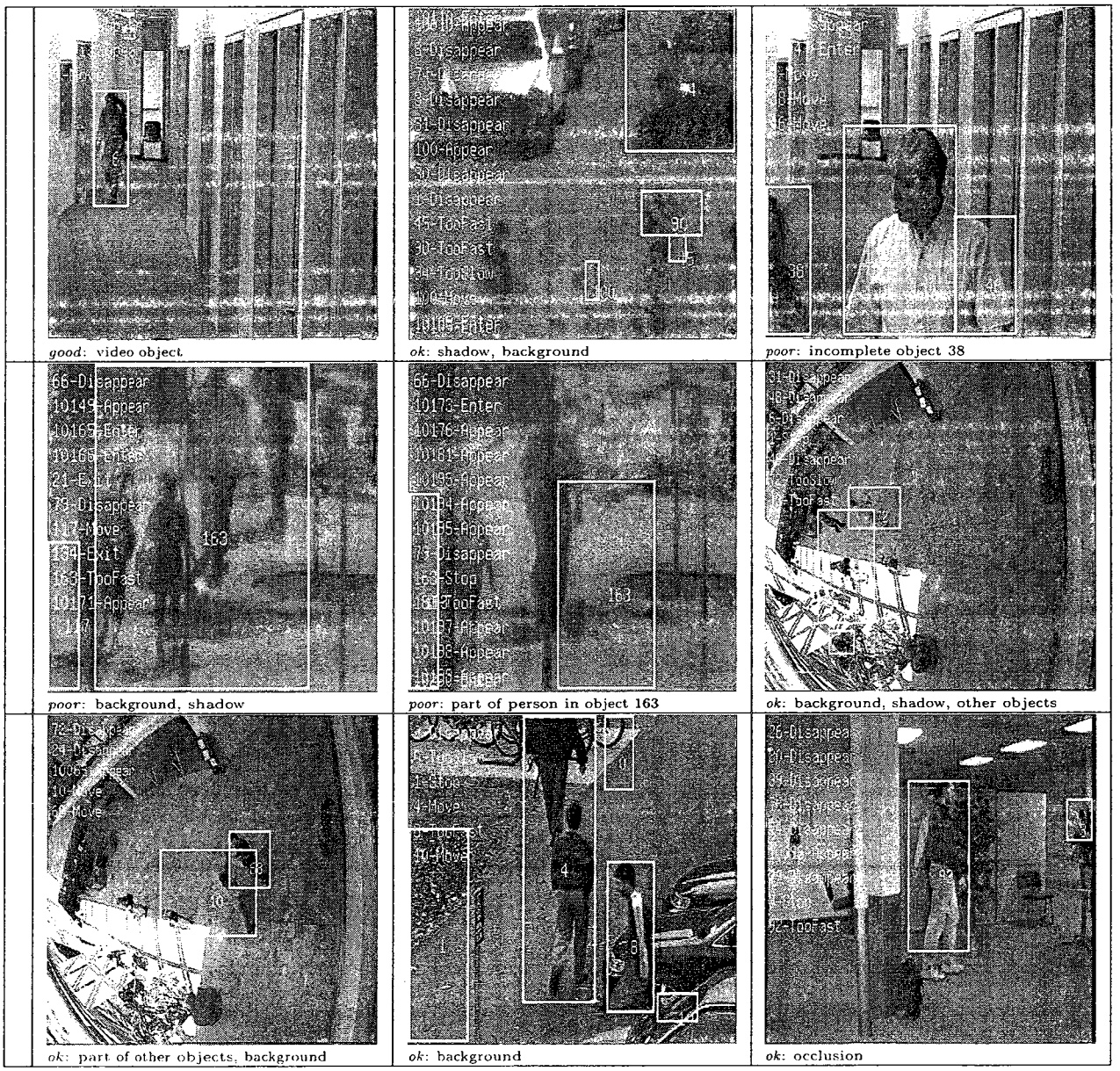
When we create the data sets, we include diverse video objects (e.g., poorly segmented objects) in test data sets. Thus, the test data sets include video objects of different orientations, different resolutions, and different scales in each class. Due to poor segmentation, some video objects in different data sets turn out to be challenging to classify. Here *challenging* means video objects which include: parts of objects, parts of background, parts of other objects, parts of shadow, and occlusions. We randomly group video objects in different data sets for corresponding class with *poor-*, *ok-*, and *good*-segmentation. The random grouping is done subjectively from the manually pre-labeled data sets in Section 5.3.1. Some examples of these challenging video

objects (i.e, good-, ok- and poor-segmentation) are shown in Table 6.3. In Section 6.4, we show that for all diverse test data sets, the classification accuracy is higher with *C-MP7* from high-dimensional chaotic series than that with *MPEG-7*. Thus, *MPEG-7* fails to capture the discriminancy among diverse video objects in different classes, when *C-MP7* successfully classify the video objects. In Section 5.3.5 we show that, for all test data sets, *C-MP7* from high-dimensional chaotic series offers higher minimum inter class distance than that for *MPEG-7*. The minimum inter class distance is calculated using multi-class Fisher criteria [139].

6.9.2 Accuracy Under Random Training

Fig. 6.18 shows SVM classification accuracy for *C-MP7* (from high-dimensional chaotic series) and *MPEG-7* with random training by the pruned data set 16 and random testing by other data sets, e.g., data set 5. The pruned data set 16 does not contain any repeated video objects but contains dissimilar video objects. In different epochs, i.e., classifier runs, the accuracy with *C-MP7* is higher and consistent, but with *MPEG-7* the accuracy is lower and inconsistent. However, if we use the *same* set of video objects (from data set 16) in training, and *random* video objects (from data set 5) in testing, Fig. 6.19 shows that, with *C-MP7* the accuracy is still higher (i.e., slighty higher in corresponding classifier runs as reported in Fig. 6.4 for data set 5) than that with *MPEG-7*, but the accuracy is consistent with both. Fig. 6.18 implies that despite training with random diverse video objects, *C-MP7* describes well the generic signature of video objects in corresponding classes as a feature vector, while *MPEG-7* performs not well in some classfier runs and performs poor in some classifier runs. So, MPEG-7 does not capture the generic signature of video objects in corresponding class. This observation supports the 2D visualization by multidimensional

Table 6.3: Challenging video objects in different data sets from both indoor/outdoor surveillance shots. Data sets are manually pre-labeled for video objects with *poor*-, *ok*-, and *good*-segmentation.



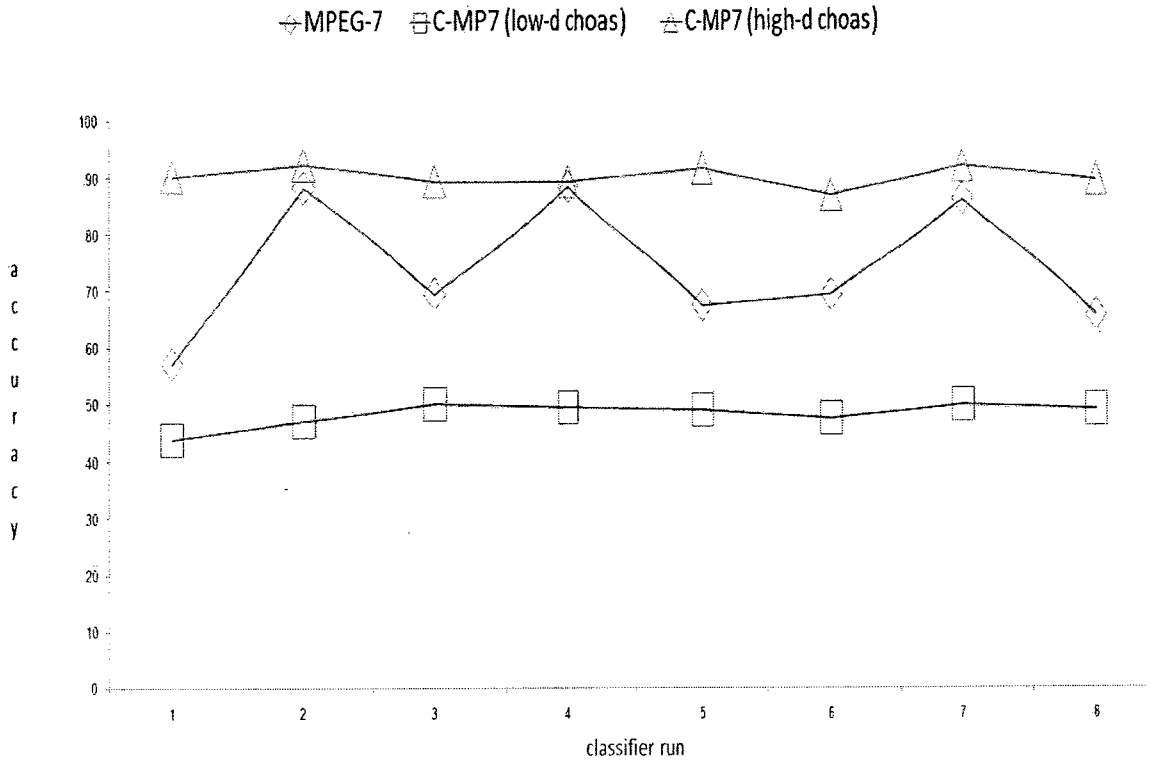


Figure 6.18: SVM classification accuracy for *C-MP7* and *MPEG-7* with random training with pruned data set 16, and random testing on other data sets, e.g., data set 5. The accuracy is inconsistent in *MPEG-7* but consistent in *C-MP7* in different epochs.

scaling for different video objects in different classes in Fig. 5.16 and Fig. 5.18. Thus, *C-MP7* is more robust as a feature vector for diverse video objects than *MPEG-7*.

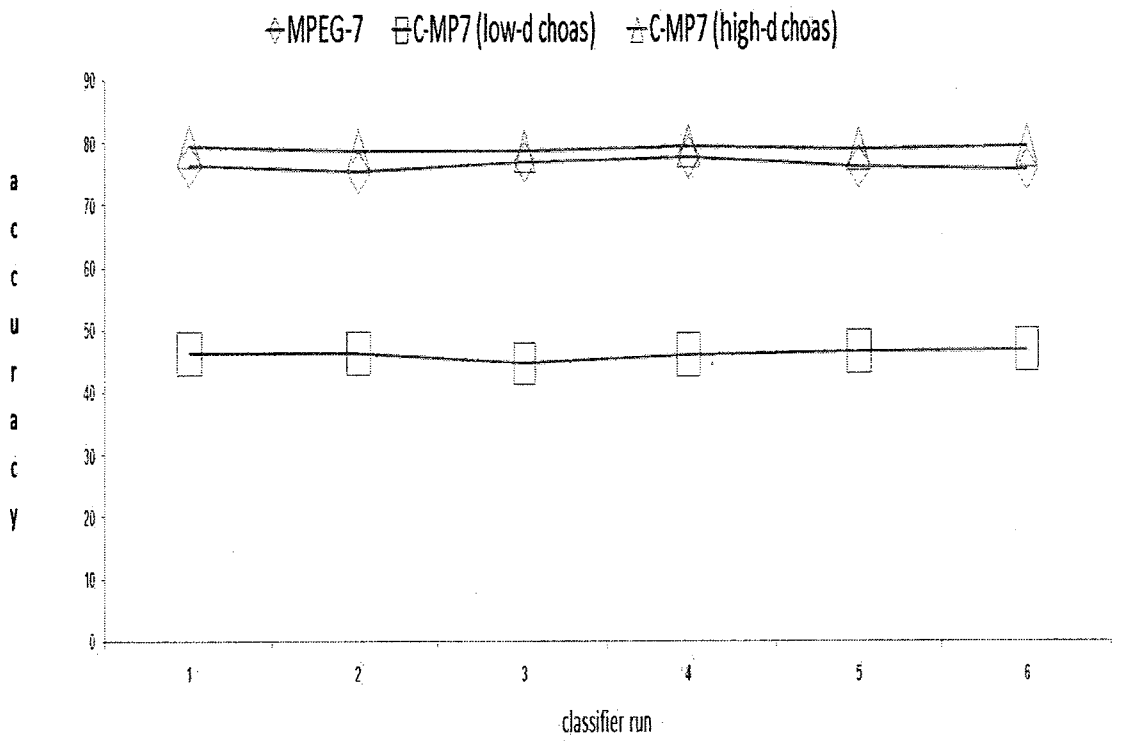


Figure 6.19: SVM classification accuracy for same set of video objects from data set 16 in training and random video objects from data set 5 in testing. Both *MPEG-7* and *C-MP7* gives consistent accuracy.

6.9.3 Reduced Feature Vector Bias

Surveillance shots often contain similar repeated video objects segmented from successive frames. Repeated video objects in training can bias the feature vector and the discriminancy of corresponding classes. Repeated video objects are thus not desired in training data sets for any classifier, and are dropped in the pruned data set 16. We use data set 16 as our training data set.

In *C-MP7*, the effect of similar repeated video objects are well compansated for (see Table 6.5) than *MPEG-7* by the histogram analysis step (see Fig. 5.4) in the proposed feature binding. The histogram analysis groups the largest cluster of the *mean* of chaotic series coefficients. The chaotic series coefficients are simulated from each feature coefficient. In presence of similar repeated video objects, the largest cluster, ignores coefficients from dissimilar but fewer video objects, and keeps coefficients from less discriminant but higher in numbers similar video objects. Table 6.4 shows the effect of similar repeated video objects with *C-MP7* from high-dimensional chaotic series, and with *MPEG-7*. In Table 6.4, feature coefficients of similar *vehicle* video objects form a less generic signature for the *vehicle* class than that in Table 6.5. In Table 6.5 repeated *vehicle* video objects are dropped. The subjective observation in Table 6.5 shows that the *vehicle* class signature is more generic. In Table 6.4, even if there are few other dissimilar *vehicle* video objects, feature coefficients of these other *vehicle* video objects are ignored during the histogram analysis.

If similar video objects are not dropped then training is done with the biased *C-MP7*. As testing data sets also include similar repeated video objects, the cross validation accuracy in Section 6.3 turns out high with both *C-MP7* and *MPEG-7*. Even with the bias, *C-MP7* cross validation accuracy is higher over that of *MPEG-7* (see Table 6.1). Section 6.4 presents the classification accuracy, where the discrim-

inancy in training video objects is preserved with the absence of similar repeated video objects in the pruned data set 16. Figures 6.4- 6.7 show, in such cases, the classification accuracy is higher with *C-MP7* (high-dimensional chaotic series) over that with *MPEG-7*. The classification accuracy is decreased with both the *MPEG-7* and the *C-MP7* than the corresponding cross validation accuracy. This happens as repeated video objects are no more in training data sets but still exist in test data sets. However, the decrease is more severe in *MPEG-7* (drops from 73.2% to 65%) than in *C-MP7* (drops from 87.6% to 83%). *MPEG-7* feature vector is directly used for classification without the proposed feature binding. When simillar video objects are dropped in training, the effect of feature vector bias is compansated more in the *C-MP7* than in *MPEG-7*. Thus, *C-MP7* is more reliable as a feature vector than *MPEG-7*.

Table 6.4: Effect of similar repeated video objects with *C-MP7* for high-dimensional chaotic series. In 2^{nd} to 3^{rd} columns, x-axis shows number of feature coefficients and y-axis shows coefficient values. Here, feature coefficients of similar *vehicle* video objects bias the *C-MP7* during histogram analysis, and form a less generic signature for the *vehicle* class.

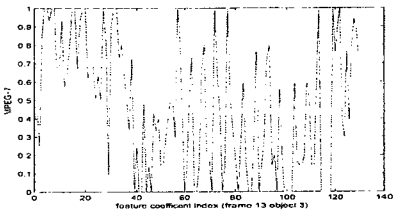
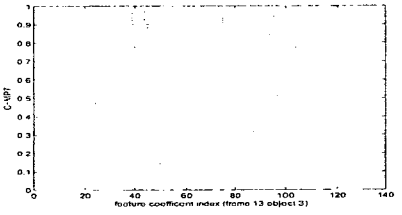
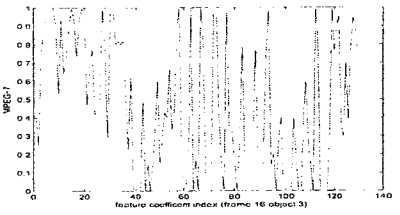
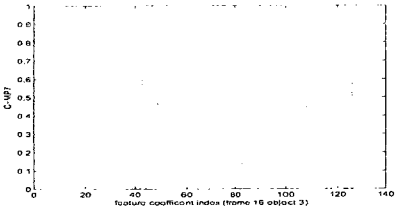
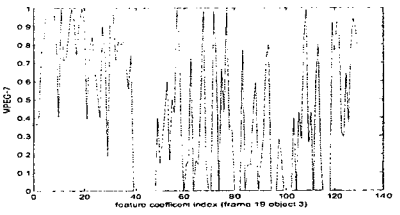
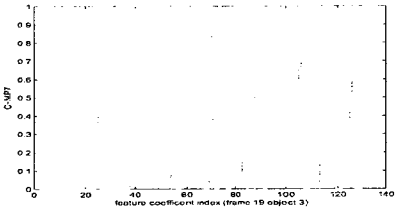
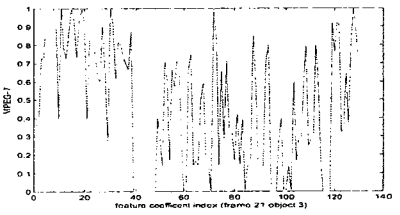
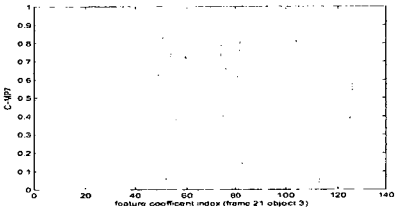
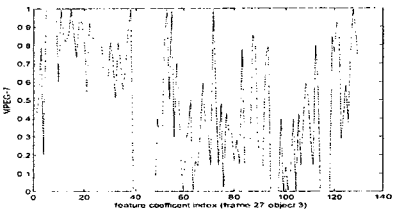
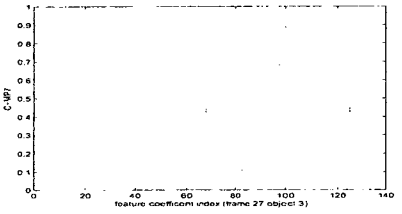
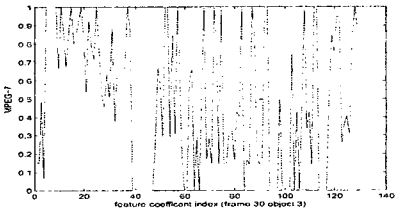
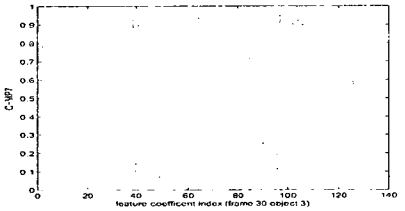
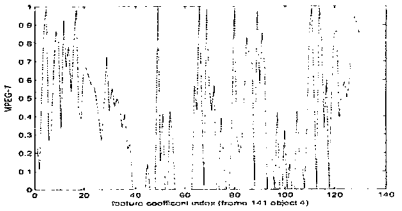
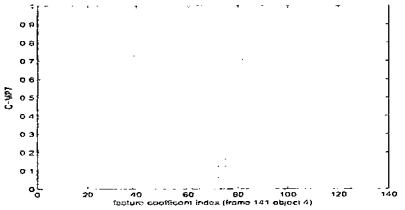
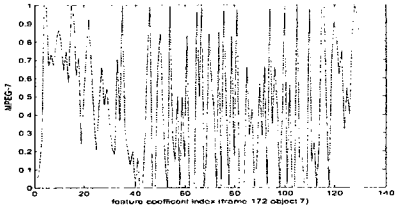
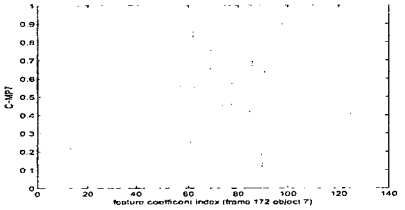
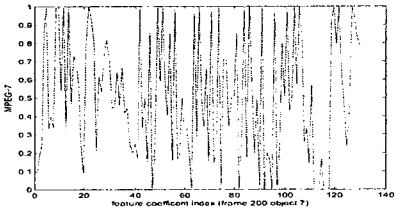
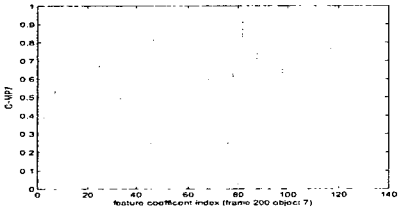
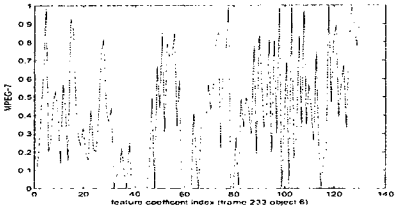
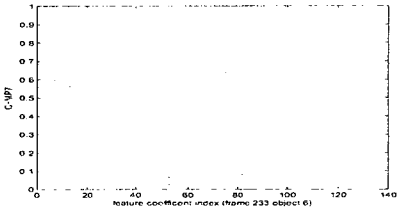
video object	*MPEG-7*	*C-MP7 h-d* chaotic series
a) 		
b) 		
c) 		
d) 		
e) 		

Table 6.5: Repeated *vehicle* video objects are dropped. The *vehicle* video objects are more diverse, and *vehicle* class signature is more generic with *C-MP7* for high-dimensional chaotic series. In 2^{nd} to 3^{rd} columns, x-axis shows number of feature coefficients and y-axis shows coefficient values. Dropping repeated objects creates more generic signature with *C-MP7* than *MPEG-7*, for diverse video objects in corresponding class.

video object	*MPEG-7*	*C-MP7 h-d* chaotic series
a) 		
b) 		
c) 		
d) 		
e) 		

6.10 Why *C-MP7* Is Robust?

In this section, we analyze the proposed feature binding method to understand, why the output feature vector *C-MP7* is more robust than *MPEG-7*. The video objects of interest in surveillance shots are often partially specified, meaning, segmentation and tracking can output incomplete video objects in corresponding class. The proposed feature binding excels in capturing discriminant feature coefficients from challenging video objects in a class despite poor segmentation and tracking. This is accomplished by eliminating any feature coefficient which is not needed to generalize the feature vector for different classes of video objects. Theoretically, any low-level feature can be used, if the feature offers discriminancy for video objects among different classes. We use a set of selected *MPEG-7* visual descriptors (see Chapter 4). In the following subsection, we re-visit two most significant steps in the proposed feature binding, i.e., neighborhood attractor interaction, and histogram analysis. The purpose of this subsection is to have a closer look into unique mechanisms which are offered in the proposed feature binding method. Then in the other subsection, we focus on why *C-MP7* simulated high-dimensional chaotic attractors is offering robustness, and, if there exists any unique characteristics of high-dimensional chaotic attractors to capture the pattern of video objects in a class.

6.10.1 What Happens In Feature Binding?

In the proposed feature binding, initially, a set of video objects are manually pre-labeled as a group or class. The *MPEG-7* descriptor coefficients are extracted and concatenated to form a feature vector for each video object in a class. *MPEG-7* feature coefficients for all video objects in corresponding class thus form a 2D video object array with video objects as *rows* and feature coefficients as *columns*. One

dimensional chaotic series coefficients are generated for each feature coefficient in the video object array. The video object array then becomes a 3D video object matrix.

At this point, we want to locate those feature coefficients in the video object array which belong to the largest cluster of chaotic series coefficients in the video object matrix. We compute the statistical *mean* of chaotic series coefficients (simulated from each feature coefficient) to define a scalar property of underlying distribution in the corresponding chaotic series. This scalar property computation will map back the 3D video object matrix to a 2D modified-video object array. In the modified-video object array, each feature coefficient is replaced with the corresponding chaotic series *mean*. Histogram analysis of this modified-video object array, creates clusters of chaotic series *means* in different histogram bins. The indexes in the modified-video object array, which belong to the largest histogram bin, can be located. Chaotic series *means* in these located indexes of the modified-video object array can be mapped back to corresponding *MPEG-7* feature coefficients, as available from the initially formed 2D video object array. *MPEG-7* feature coefficients in the rest of the indexes in the initially formed 2D video object array is discarded as null. The final output is a new 2D video object array (i.e., *C-MP7*) for corresponding class, with reduced *MPEG-7* feature coefficients.

One significant step in the proposed feature binding is the *neighborhood attractor interaction* step in Fig. 5.4. The neighborhood interaction using Coupled Map Lattice (CML) [24] is applied on video object matrix for corresponding class. The neighborhood interaction couples chaotic series coefficients as simulated from feature coefficients. We assume feature coefficients as dynamical systems in Section 1.3 and in Section 5.2.2. So the neighborhood interaction eventually couples those dynamical systems. The coupling, in this step, allows the proposed feature binding to generate patterns which are pertinent in corresponding class of video objects, even when the

video objects are corrupted from poor segmentation. For example, two car objects, *obj1* and *obj2*, can have features coefficient similar or different. In any case, chaotic series coefficients are generated, where these feature coefficients are seeds to chaotic series coefficient simulation. The chaotic series coefficients are sensitive to these seeds. By applying CML in the video object matrix, the sensitivity is compansated to make video object matrix more homogeneous by modifying chaotic series coefficients. Due to the CML, the proposed feature binding generates *C-MP7* for corresponding class, irrespective of same chaotic series coefficients at some indexes of the video object matrix. Another example scenario can be where, *obj1* is a *car*, *obj2* is a *person*. Both *obj1* and *obj2*, in this case, may have similar or different feature coefficients. The chaotic series coefficients from each objects are pre-labeled in separate groups (Section 5.3.1). Thus, the sensitivity dependency on initial seeds for same coefficients in these two objects are made independent of each other.

The other significant step in the proposed feature binding is the *histogram analysis* step in Fig. 5.4. The largest cluster of histogram bin, from the chaotic series *means* in the 2D modified-video object array, locates feature coefficients which contribute the most in defining a generic signature for video objects in corresponding class. In Section 1.3 and in Section 5.2.2, we assume, feature coefficients as dynamical systems similar to neurons in human brain, and these dynamical systems are simulated by chaotic series coefficients. The largest cluster of feature coefficients, as located finally in the 2D video object array, indicates those simulated dynamical systems which contribute the most in defining the generic signature for video objects in corresponding class. Finding the largest cluster from histogram analysis, makes the proposed feature binding method not constrained by any threshold, unlike rule based video content description approaches in literature [35, 98, 99].

The proposed feature binding is designed to achieve dynamic feature coefficient

reduction, and increases discriminancy among different groups of video objects. The dynamic coefficient reduction is performed by throwing out different feature coefficients from different video objects in a class to form a generic feature vector. The word *dynamic*, here, refers to the fact that, the video object array with the new feature vector, *C-MP7*, contains different feature coefficients (in columns) for a set of video objects (in rows) at different indexes. The not-eliminated feature coefficient indexes thus, vary for each video object in corresponding class, yet pertain the generic signature of the class. The remaining coefficients in the video object array can belong to any features, i.e., any *MPEG-7* visual descriptors.

Due to the presence of feature coefficients at varying indexes in the video object array, the proposed feature binding does not focus on individual visual descriptor importance. The pattern of video objects in corresponding class is defined by grouping feature coefficients in the video object array from all the video objects in the class.

6.10.2 Sensitivities To Initial Seeds

A chaotic series must be sensitive to initial conditions [33]. One reason for the excellence of *C-MP7* with *high* dimensional chaotic series, can be due to the sensitivities of initial seed x_0 (see Section 2.4 in Chapter 2), either, in the chaotic series, or in the chaotic attractor. Firstly, in Fig. 6.20, up to *n=13*, *low* dimensional chaotic series (Eq. 2.5) exhibits same series with initial seeds of 10^{-6} difference, i.e., 0.123423 and 0.123425. As n increases, chaotic series drift apart. In 2D phase space, $x(n)$ vs $x(n+1)$, both the series, however, stay on the same attractor, see Fig. 6.21. Secondly, in case of *high* dimensional chaotic series (Eq. 2.11), at higher iteration, e.g., *n=1900*, the said two seeds show drift in Fig. 6.22, but stay on same attractors in Fig. 6.23. However, for initial seeds of 10^{-1} difference, *high* dimensional chaotic attractors show

significant drift in Fig. 6.23. The non-linear structure (i.e., dynamics in trajectory) in *low* dimensional chaotic series is easily detectable and is hardly detectable in *high* dimensional chaotic series.

In Section 5.2.2, we use 500 iteration for any chaotic series. To verify, the dominance of drift or iteration length in video object classification accuracy, apart from the simulation in Section 5.3, we re-simulate *C-MP7* with *low* dimensional chaotic series at 10 iteration, where there is no drift in the series, and with *high* dimensional chaotic series at 2500 iteration, where there exists significant drift in the series. As in Figs. 6.4- 6.7, for SVM, the *C-MP7* with *low* dimensional chaotic series again shows poor accuracy, while, with *high* dimensional chaotic series achieves increased accuracy, on average 84.58%. It is implied that drift in chaotic attractors, not iteration length in chaotic series are the reason for *C-MP7* robustness in video object classification. The drifts in *high* dimensional chaotic attractors, for change in initial seeds in the range of 10^{-1}, is due to transient. In Fig. 6.23, this transient is the initial delay before the *high* dimensional chaotic series rides on the attractor, and is caused by asymptotic stability of fixed points in the attractor [33]. Transient is practically insignificant in *low* dimensional chaotic series for change in initial seeds in the range of 10^{-1} as different chaotic series in Fig. 6.21 ride the same chaotic attractor without any drift. Changes in *MPEG-7* feature coefficients in the range of 10^{-1} are pertinent in our data sets (see Appendix A). Thus, irrespective of the iteration length N and drift in either *low-* or *high-* dimensional chaotic series, the transient in *high* dimensional chaotic series allows the new *C-MP7* to capture subtle variations of *MPEG-7* descriptor coefficients for diverse video objects in surveillance video. The higher variance in *C-MP7* with *high-dimensional* chaotic series as shown in Fig. 5.11 in Section 5.3.4 for different video objects. manifests the said subtle variations.

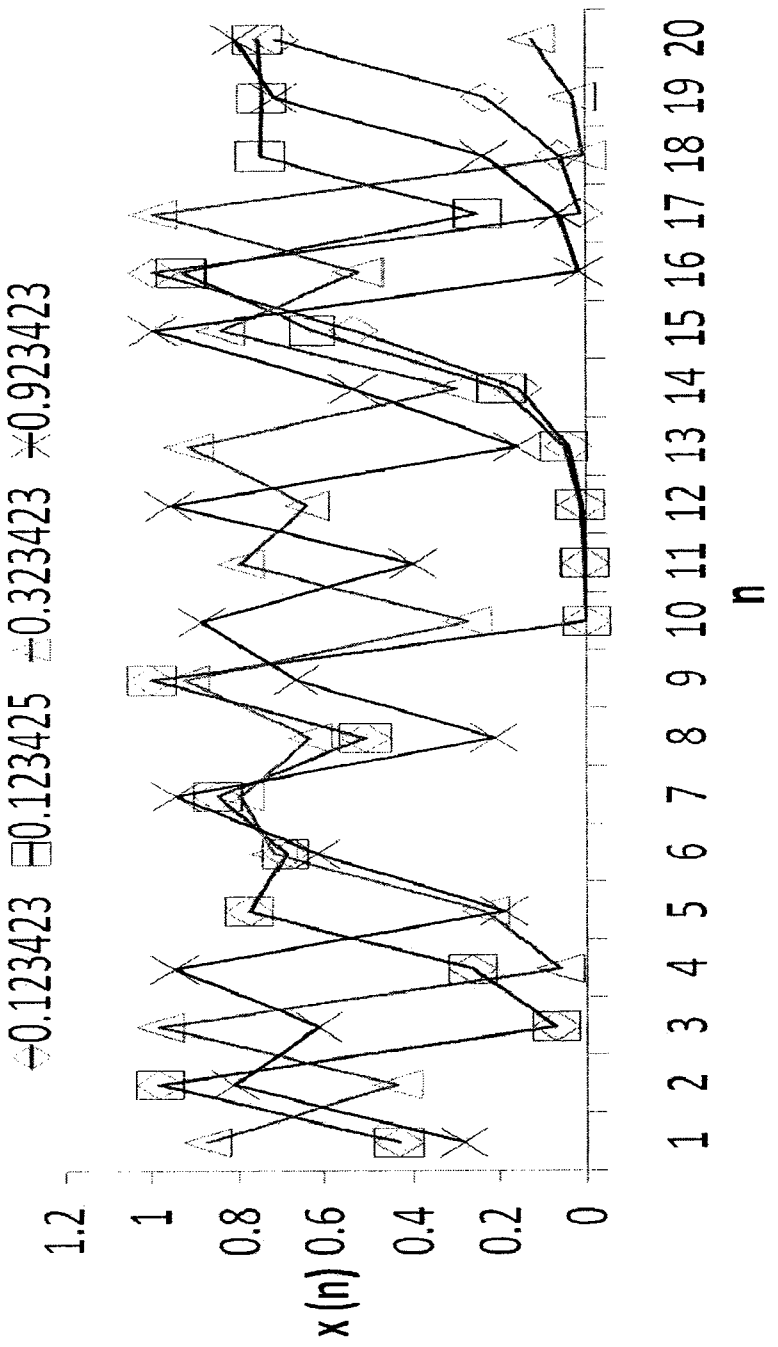


Figure 6.20: Drift in low-dimensional chaotic series with 50 iteration. Till n=13, no drift for 10^{-6} difference in seeds. However, for 10^{-1} difference significant drift is observed.

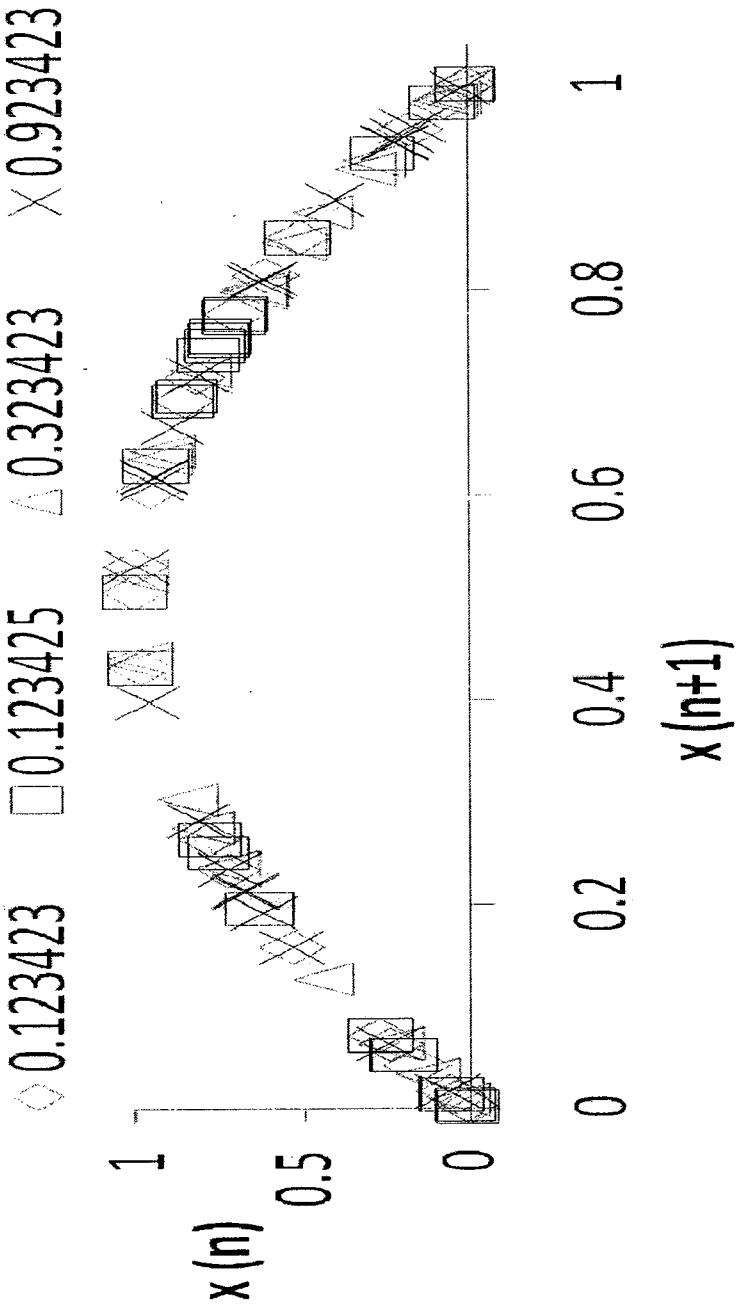


Figure 6.21: No drift in low-dimensional chaotic attractors with 100 iteration, no drift for 10^{-6} for 10^{-1} difference in seeds.

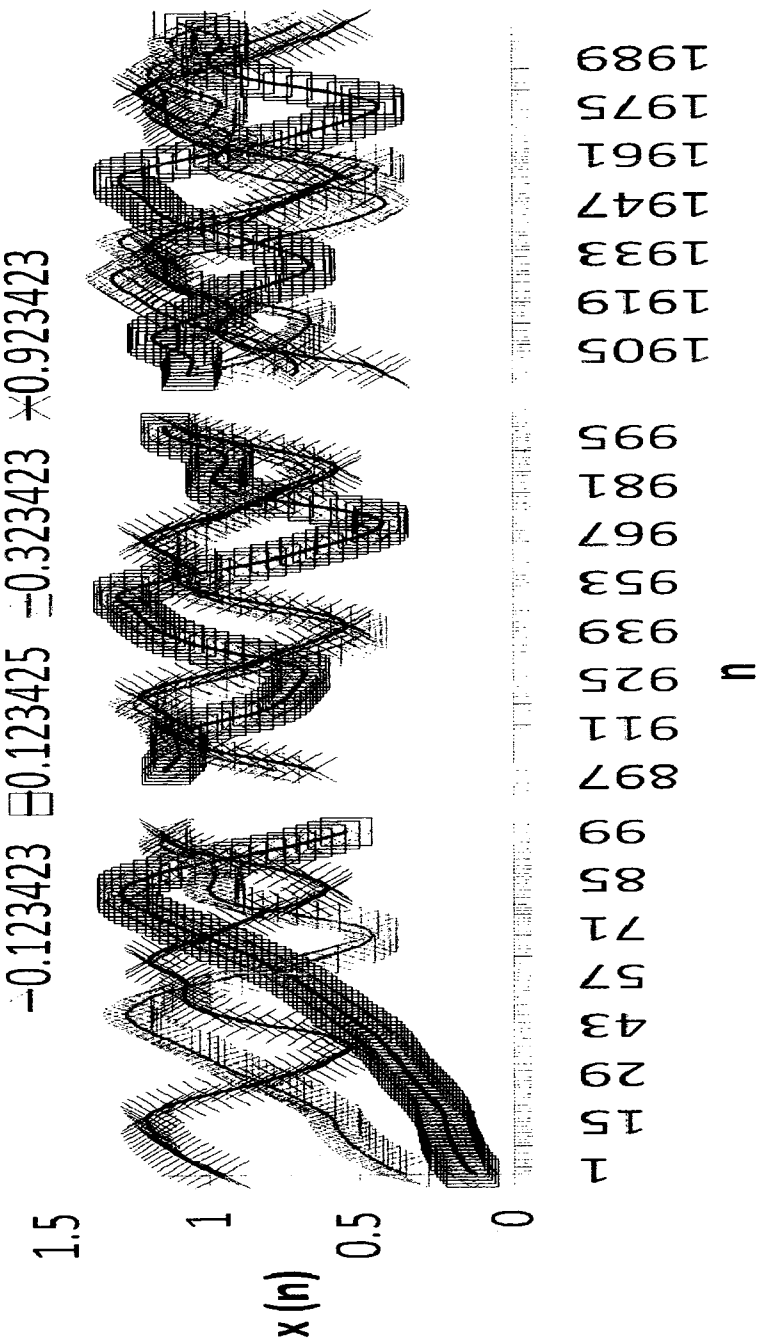


Figure 6.22: Drift in high-dimensional chaotic series with 3000 iteration. Till n=1900 no drifts for 10^{-6} difference in seeds. However, for 10^{-1} differences significant drift is observed.

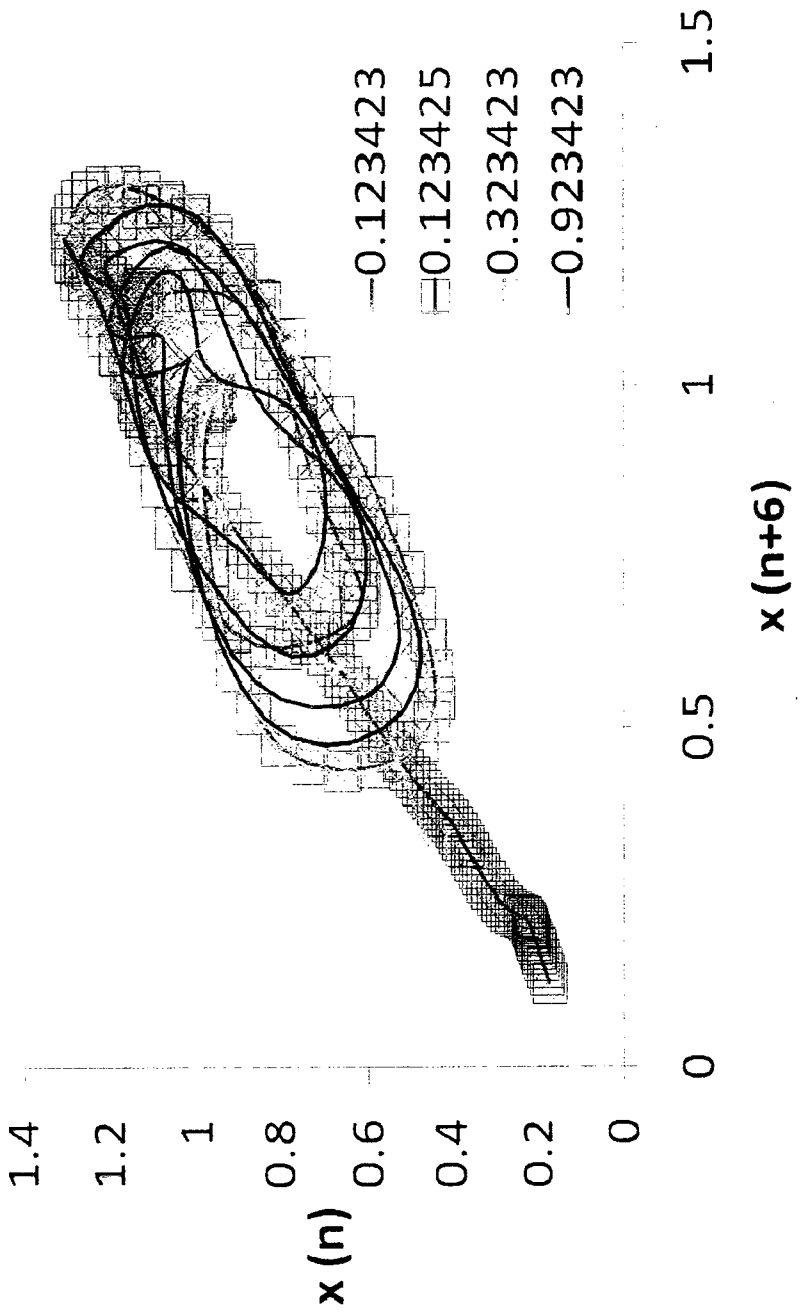


Figure 6.23: Drift in high-dimensional chaotic attractors with 3000 iteration. Significant drift for 10^{-1} difference in seeds.

6.11 Computational Cost

In this Section, we disscuss the computational cost of the proposed feature binding method to generate the new feature vector. In Section 3.3, related work on pattern classification mentions the computation of Lyapunov exponents, as a common property based on which chaotic descriptors are created. There are as many Lyapunov exponents as there are dimensions in the state space of the system, but the largest Lyapunov exponent is usually the most important. If this exponent is negative, then the orbits converge in time, and the dynamical system is insensitive to initial conditions. However, if this exponent is positive, then the distance between nearby orbits grows exponentially (i.e., divergence) in time, and the system exhibits sensitive dependence on initial conditions. By definition, a chaotic series is sensitive to initial conditions (see Section2.4).

The number of dimensions are the same as the number of Lyapunov exponents in a chaotic system, so, computation of either one property can be considered to characterize the chaotic series in the chaotic system. The details on computational overhead in using Lyapunov exponents is not reported in pattern classification literature (see Section 3.3). However, for video content description, our earlier work [29] shows that, calculating dimensions [43] from chaotic attractors, are computationally expansive. In this thesis, we skip the need for calculating either dimensions or Lyapunov exponents to win over expensive computational overhead. Another reason for not using Lyapunov exponent is the use of Takens vector [42] in our proposed approach (see Section 5). As mentioned here, the characteristics exposed by Lyapounov exponent or the correlation dimension is similar in a chaotic series. The Takens vector is used to reconstruct a chaotic attractor with smaller iteration length chaotic attractor which maintains the original correlation dimension. Hence, calculating correlation

Table 6.6: Computational overhead of the proposed feature binding for data set 16 with *low* dimensional chaotic series.

class	objects	MP-7 in sec	chaotic in sec	CML in sec	Trajectory in sec	histogram in sec	sec/ object
p	89	5.6	91.0	4.2	54.1	0.14	1.7
g	34	4.5	65.9	2.1	35.3	0.03	3.1
v	25	3.2	49.0	1.4	26.1	0.01	3.1
u	45	5.6	89.8	3.2	47.8	0.01	3.2
p	365	53.7	660.2	171.4	495.3	0.30	3.7
g	111	17.5	173.2	16.6	145.8	0.12	3.1
v	361	55.3	690.2	175.6	467.9	0.10	3.8
u	175	23.5	257.3	34.1	205.3	0.06	2.9

dimension is preferred in the proposed feature binding approach than to calculate the Lyapunov exponent. The proposed feature binding, in this thesis, allows us to successfully reduce the computational overhead of [29], from up to 8 hours to 4 seconds. Table 6.6 and Table 6.7 show the computational overhead in seconds for the generation of *C-MP7* for video objects using feature binding (data set 16 and data set 4) with *low*- and *high*-dimensional chaotic series, respectively. Processing time for different modules, i.e., *MPEG-7* visual descriptor extraction, *low*- or *high*-dimensional chaotic series simulation, CML interaction, trajectory re-construction, and histogram analysis are presented. On average it takes *2* to *4* seconds per video object to generate *C-MP7*. We use a standalone 1.6GHz intel centrino laptop with 512 MB RAM and Matlab 7.1.0.183 (R14) to calculate the overhead.

Table 6.7: Computational overhead of the proposed feature binding for data set 16 with *high* dimensional chaotic series.

class	obj ects	MP-7 in sec	chaotic in sec	CML in sec	Trajectory in sec	histogram in sec	sec/ object
p	89	7.2	68.6	4.512	59.4	0.08	1.5
g	34	4.2	42.1	2.155	36.5	0.03	2.5
v	25	3.0	31.2	1.425	27.0	0.01	2.5
u	45	5.3	56.4	3.302	48.7	0.01	2.5
p	365	36.9	827.1	134.8	412.4	1.1	3.8
g	111	16.3	235.7	14.3	124.5	0.04	3.5
v	361	46.9	1009.0	140.6	555.2	0.11	4.8
u	175	43.8	454.9	33.8	208.5	0.10	4.2

6.12 Summary

This chapter presented evaluation of *C-MP7* on video object classification. Diverse video objects containing abrupt changes in lighting, shaking trees, shadow, and occlusions are randomly grouped into different data sets (see Table 5.1), and pre-labeled in four different classes: *has_person* (p), *has_group_of_persons* (g), *has_vehicle* (v), and *has_unknown* (u). Irrespective of the choice of classifiers, *C-MP7* with high-dimensional chaotic series offer higher classification accuracy compared to *MPEG-7* for the same data sets Hold-out cross validation, test with different data sets, and test with unseen successive sequences are reported. *C-MP7* (with *high-dimensional* chaotic series) offers higher classification accuracy, and is more robust than the same with *MPEG-7*. The effect of transient in high-dimensional chaotic series is identified as the reason for excellence of *C-MP7* as feature vector which justified the low accuracy with *low-dimensional* chaotic series. The transient in high-dimensional chaotic series allows more subtle variations in attractors from slightest difference in feature coefficients to capture the discriminancy of diverse image regions of the same video object in successive frames. We also include an initial observation on sub-group classification of *male* and *female* video objects in *has_person* class. The pattern of *male* and *female* video objects with a by-product chaotic feature vector Z (*low-dimensional* chaos) are more specialized in *has_person* class than that with *MPEG-7*. The computational cost on average per video object is 2 to 4 seconds, which is still high for the generation of *C-MP7* in a real time surveillance system.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has proposed a feature binding method to generate a new feature vector, *C-MP7*, for video object description. Our work is inspired by the simulation of chaotic series in functional responses of neurons in human brain. The proposed method considers *MPEG-7* descriptor coefficients as dynamical systems in an abstract multi-dimensional feature space. Dynamical systems are excited (similar to neuronal excitation) with either *low-* or *high*-dimensional chaotic series, and then chaotic series coefficients undergo histogram-based clustering. We find, *high* dimensional chaotic series is preferable as a choice over *low* dimensional chaotic series for feature excitation in the proposed feature binding. The *C-MP7*, simulated from *high* dimensional offers higher variance as a feature vector than *MPEG-7*. *C-MP7* also provides better discriminancy in different video objects description. The multi-class discriminant analysis of *C-MP7* with Fisher-criteria shows increased binary class separation for clustered video objects over that of *MPEG-7*. *C-MP7*, specifically provides good clustering of *vehicle* objects against other video objects.

To verify the excellence of *C-MP7* over *MPEG-7* as a feature vector multiple binary classifiers are used. *C-MP7* trains different classes of video objects in each binary classifier. Testing of diverse video objects show improve classification accuracy with the *MPEG-7* feature coefficients excited by *high* dimensional chaotic series, over that with the original *MPEG-7* visual descriptors. Using diverse video objects, including objects from poor segmentation, *C-MP7* (from high-dimensional chaotic series simulation) offers more robustness than *MPEG-7* as a feature vector for classification. *C-MP7* reduce the feature vector bias better than *MPEG-7*, at the absence of repeated video objects in data sets. However, *C-MP7* from low-dimensional chaotic series simulation is not as robust as *MPEG-7*. Longer transient in the *high* dimensional chaotic series is identified as the reason of its excellence over the excitation with *low* dimensional chaotic series. The transient is insignificant in the *low* dimensional chaotic series. The variance in *C-MP7* with *high* dimensional chaotic series, is observed to be higher than that with *low* dimensional chaotic series.

High classification accuracy is achieved with C-MP7, from high-dimensional chaotic series simulation, despite poor segmentation of video objects. Video objects of different scale, resolution, orientation and occlusion are mixed in the test data set from both indoor and outdoor surveillance scenes. *Vehicle* objects are clustered well, which leads to above 99% accuracy for only *vehicles* against other objects in all data sets.Thus, using diverse video objects, including objects from poor segmentation, *C-MP7* offer more robustness than *MPEG-7*. *C-MP7* reduce the feature vector bias better than *MPEG-7*, at the absence of repeated video objects in training data sets. Due to specific interests to classify *male* and *female* video objects in *has_person* class, we also include an initial observation on such sub-group classification. We find, the pattern of *male* and *female* video objects with a by-product chaotic feature vector Z (*low-dimensional* chaos) are more specialized in *has_person* class than that with *MPEG-7*.

The scopes for sub-classification for sub-set of video objects in each class are yet to be explored.

7.2 Future work

Future work include, 1) optimization of chaotic series properties, e.g., effect of different chaotic series parameters, effect of different coupling coefficients, and histogram window length parameters, 2) study of similar feature binding method for finding patterns in other video contents, e.g., concepts, events, 3) study sub-group classification of video objects, as oppose to generalized video object classification, 4) integration of *MPEG-7* temporal descriptors, and 5) study of similar feature binding method in future application development, such as biometrics integration in surveillance video.

7.2.1 Chaos Parameter Optimization

We have not included any optimization on the characteristics of the chaotic series, e.g., effect of different chaotic series parameters, effect of different coupling coefficients, and histogram window length parameters. An interesting observation can be to verify the performance of each step in the proposed feature binding method (see Section 5.4). This verfication can confirm theoritical explainations in Section 6.10.1 on, a) which step plays key role in the proposed method, and b) the effect of major steps for robustness in the proposed method.

In this thesis, we use one-dimensional chaotic series, i.e., one variable in nonlinear difference equation and first order differential equation. Discrete form of one-dimensional chaotic series is adopted for both low-dimensional chaos equation, e.g., Logistic map, and a high-dimensional chaos equation, e.g., Mackey Glass equation. There a number of differential equations which exhibit chaotic series characteristics.

Observation on additional high- dimensional chaotic series (other than Mackey Glass equation, e.g., KIII equations [12]) can be explored.

7.2.2 Sub-group Classification Of Video Contents

In this thesis, we use the output of the proposed feature binding to create generalized descriptions of primary video contents for video object classification in four commonly used classes in surveillance shots, namely, *has_person*, *has_group_of_person*, *has_vehicle*, and *has_unknown*.. The scopes for specialized video content description with *C-MP7* for sub-set of video objects in a class, is yet to be explored. In Section 6.8, we present initial observations on *male* and *female* sub-group classification with, chaotic feature vector Z, which is a by-product output of the proposed feature binding. We find Z is not a good candidate for generalized descriptions of video objects in different groups (i.e., *has_person*, *has_group_of_person*, *has_vehicle*, and *has_unknown*). However, subjective observations on same video object in different frames in a shot with Z show unique patterns for *male* or *female* sub-set of video objects in has_person class. Such unique patterns are not dominant in the subjective evaluation for the same two sub set of video objects with *C-MP7* (see Figs. 6.14, 6.15, 6.16, and 6.17). More quantitative study is needed to formulate a conclusive decision on the use of the proposed feature binding for such specialized sub-group classification of video objects.

7.2.3 *MPEG-7* Description Schemes For Objects And Events

The visual descriptors are structured and related within a common framework based on description schemes in *MPEG-7*. These description schemes define a model of the description using the descriptors according to different rules. Examples are event

description scheme for 'enter', object description schemes for 'john' or 'person' [146]. The description schemes generate human readable XML description [147,148]. Future work can be initiated to generate description schemes from successfully classified objects and events in a video surveillance scene.

7.2.4 *MPEG-7* Temporal Descriptors

Our future work includes the investigation of temporal *MPEG-7* visual descriptors (e.g., motion activity, trajectory) to further strengthen the chaotic feature binding to yield more distinctive chaotic descriptor for video contents. In the proposed approach we extract a set of *MPEG-7* visual descriptors which provide spatial information about the video object. Temporal descriptors are not considered as they are not suitable to be compared in the same spatial domain of frame analysis. For example, motion activity descriptor is not used as they integrate in the temporal domain, and can be comparable only if the descriptors used (e.g., color, texture, and shape) would be aggregated over time [10].

It is plausible to use *MPEG-7* temporal descriptors in a shot-based object and event description. All the descriptor coefficients will need to be aggregated over the shot (e.g., set of 30 frames/second), and corresponding statistical property (e.g., mean) of each feature element can be taken as input for the proposed histogram-based feature binding. Observation on occlusion is not performed in this work, as description of occluded object is much easier with temporal descriptors. Temporal descriptors will enable description of occluded objects in successive frames before and after occlusion. In the current work, occlusion detection is handled during tracking in video analysis. We focus on video content description, and use list of video objects as available from tracking. Inclusion of temporal descriptors will require shot-based

classifier design of with new training and test data.

7.2.5 Biometrics Integration In Surveillance Video

In the post-9/11 era of reality, the research in surveillance video is moving towards the integration of multi-modal biometrics, and is gaining momentum in homeland security applications. Biometrics, by definition, identify a person based on some feature of his/her biological characteristics. These feature can be behavioral in nature, such as, the way a person writes a signature, or types in keyboard, or these features can be physiological, such as a palm print or finger print, the arrangement of blood vessels in the retina, the patterns in the iris, voice print, and face response of emotional expressions [149–151]. Traditional biometric applications deal with image-based input in constrained environments and co-cooperativeness from the person in question. Major advancement is achieved in recognizing an individual person, using image-based traditional biometrics [152]. More needs to be done for unconstrained surveillance scenarios, 1) to identify/recognize an individual, and 2) to predict an incident/crime before it happens.

The future surveillance system needs to integrate biometric [8, 153–156]. A biometric surveillance system needs to be multi-modal, flexible, and scalable. [157] uses fusion of visual and thermal modalities for person authentication. For both the above mentioned scenarios 1) and 2) in this section, fusion [158] of multi-modal biometric features and low resolution traditional surveillance camera feeds, needs to be studied with newer feature processing tools.

MPEG-7 provides a set of robust, scale and resolution invariant, visual descriptors based on human visual systems [9]. On the other hand, neuroscience experiments have established a link in explaining the brain activity of human and chaos theory [11, 13].

[159] uses *MPEG-7* based color and texture features for face recognition. [160, 161] uses chaos to encrypt watermark of face images for biometric verification. For person identification, [162] uses wavelet decomposition to extract invariant features from spatially enhanced fused images of face and palm print images. Fractal dimension feature of iris is used in [163] for personal identification. Chaos theory is also explored in biometric applications [161, 164]. Both the *MPEG-7* and chaos theory can be further studied separately for multi-modal biometric feature fusion. Another possible avenue can be to use the proposed feature binding method, in this thesis, to create or modify biometric features for person identification/recognition, and crime/incident prevention.

Bibliography

[1] T. List, J. Bins, R. Fisher, D. Tweed, and K. Thorisson, "Two Approaches to a Plug and Play Vision Architecture CAVIAR and Psyclone," in *AAAI workshop on Modular Construction of Human-like Intelligence, Pittsburgh*, Jul 2005.

[2] D. Mayisela P. Dagba M. Ghazal D. Ostheimer, S. Lemay and A. Amer, "A Modular Distributed Video Surveillance System Over IP," in *IEEE Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada*, May 2006, pp. 1001–1004.

[3] F. Mokhtarian and M. Z. Bober, *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*, Springer, 2003.

[4] A. Amer, *Object and Event Extraction for Video Processing and Representation in On-Line Video Applications*, Ph.D. thesis, INRS-Telcommunications, University of Quebec, Montreal, Quebec, 2001.

[5] *PETS 2001 Benchmark Data*, Online, 2001.

[6] R. Kozma and W.J. Freeman, "Chaotic Resonance - Methods and Applications for Robust Classification of Noisy and Variable Patterns," *Int. Journal:Bifurcation and Chaos*, vol. 11, no. 6, pp. 1607–1629, 2000.

[7] A. Amer and C. Regazzoni, "Introduction to the special issue on video object processing for surveillance applications," *Real-Time Imaging*, vol. 11, no. 1-5, 2005.

[8] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Transcation:Sytems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334 – 352, August 2004.

[9] B.S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7*, John Wiley and Sons, Ltd, 2002.

[10] H. Eidenberger, "How Good Are The Visual MPEG-7 Features?," in *SPIE Visual Communications and Image Processing 2003, Edited by Touradj Ebrahimi and Thomas Sikora*, 2003, vol. 5150, pp. 476–488.

[11] E. Basar, *Chaos in Brain Function*, Springer-Verlag, 1990.

[12] W. J. Freeman, *How Brains Make Up Their Minds*, Columbia University Press, 2001.

[13] C. A. Skarda and W. J. Freeman, "How Brains Make Chaos in Order to Make Sense of the World," *Behav Brain Sci.*, vol. 10, pp. 161–195, 1987.

[14] M. Valera and S.A. Velastin, "Intelligent Distributed Surveillance Systems: A Review," in *IEE Proc. Vis. Image Signal Process.*, 2005, vol. 152, pp. 192–204.

[15] G. L. Foresti, P. Mahoen, and C. Regazzoni, *Multimedia Video-Based Surveillance System, Requirements, Issues and Soluions*, Kluwer Academic Publishers, 2002.

[16] C. Regazzoni, G. Fabri, and G. Vernazzza, *Advanced Video-based Surveillance System*, Kluwer Academic Publishers, 2002.

[17] P. Remagnio, G. Jones, N. Paragios, and C. Regazzoni, *Video-based Surveillance Systems, Computer Vision and Distributed Processing,* Kluwer Academic Publishers, 2002.

[18] B.L. Tseng, C. Lin, and J.R. Smith, "Using MPEG-7 and MPEG-21 for Personalizing Video," *IEEE Multimedia*, vol. 11, no. 1, pp. 42–52, 2004.

[19] O. Steiger, *Adaptive Video Delivery Using Semantics,* Ph.D. thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, 2005.

[20] A. J. Lipton, C. H. Heartwell, N. Haering, and D. Madden, "Critical Asset Protection, Perimeter Monitoring, and Threat Detection Using Automated Video Surveillance," *www.objectvideo.com/objects/pdf/products/onboard/*, 2009.

[21] Genetec, "Omnicast:Networked Video Surveillance Security Solution," *www.genetec.com/English/Documents/genetec-omnicast-video-surveillance-brochure-en.pdf*, 2009.

[22] Keeneo, "UnitControl:4D Automatic Advanced Counting," *http://keeneo.com/docs/*, 2009.

[23] F. T. Arecchi, "Chaotic Neuron Dynamics, Synchronization and Feature Binding," *Computational Neuroscience: Cortical Dynamics Lecture Notes in Computer Science*, vol. 3146, pp. 90–108, 2004.

[24] K. Kaneko, "Spatiotemporal Chaos in One- and Two- Dimensional Coupled Map Lattices," *Physica*, vol. 37 D. pp. 60–82, 1989.

[25] R. Kozma and W. Freeman, "Classification of EEG Patterns using Nonlinear Dynamics and Identifying Chaotic Phase Transitions," *Neurocomputing Published by Elsevier Science B.V.*, , no. PII: S0925-2312(02) 00429-0, 2002.

[26] A. Alkhatib and M. Krunz, "Application of Chaos Theory to the Modeling of Compressed Video," in *IEEE Int. Conf. on Communications*, 2000, vol. 2, pp. 836–840.

[27] T. Sun, L. Cui, S. Wang, Y. Chen, L. Chang, and C. Hsu, "Research on Technology of Chaos Secrecy Communications in Digital Watermarking," *IEEE Pacific Rim Conference on Multimedia*, vol. 2532, pp. 105–111, 2002.

[28] H. Azhar and A. Amer, "Chaotic Synchronization of MPEG-7 Descriptors for Interpretation in Surveillance Video," in *IEEE International Symposium on Multimedia (ISM)*, December 2006, number ISBN 0769527469, pp. 356–362.

[29] H. Azhar and A. Amer, "Chaos-Synchronization Based Representation of Objects and Events From MPEG-7 Low-Level Descriptors," in *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, April 2007, pp. 356–362.

[30] D. Jean-Baptiste, H. Azhar, and Aishy Amer, "MPEG-7 Descriptor Integration for On-line Video Surveillance Interface," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 2007, vol. 3, pp. 308–312.

[31] H. Azhar and A. Amer, "Chaos and MPEG-7 Based Feature Vector for Video Object Classification," in *IEEE International Conference on Image Processing, ICIP*, 2008, pp. 432–437.

[32] H. Azhar and A. Amer, "High Dimensional versus Low Dimensional Chaos in MPEG-7 Feature Binding for Object Classification," *Lecture Notes in Computer Science, Image and Signal Processing, Editors: A. Elmoataz, et. al. Springer Berlin / Heidelberg*, vol. 6134, pp. 315–323, 2010.

[33] D. Kaplan and L. Glass, *Understanding Nonlinear Dynamics*, Springer-Verlag New York Berlin Heidelberg, 1998.

[34] R.L. Devaney, *Chaos, Fractals, and Dynamics*, Addison-Wesley, 1990.

[35] A. Amer, E. Dubois, and A. Mitiche, "Real-time System for High-level Video Representation: Application to Video Surveillance," *Elsevier Journal for Real-Time Imaging*, vol. 11, no. 3, pp. 244–256, 2005.

[36] J. L. Crowley, "Things that See: Context-Aware Multi-modal Interaction," Tech. Rep., Laboratoire GRAVIR, INRIA Rhne Alpes, 2004, http://www-prima.inrialpes.fr/Prima.

[37] F. BreAmond and M. Thonnat, "Issues of Representing Context Illustrated by Video-Surveillance Applications," *Int'l J. Human-Computer Studies, Special Issue on Context*, vol. 48, no. 3, pp. 375–391, 1998.

[38] ISO/IEC, "Information Technology Multimedia Content Description Interface Part 6: Reference Software," Tech. Rep., ISO/IEC JTC 1/SC 29 N 4475, 2002.

[39] S. Baeg, J. Park, J. Koh, K. Park, and M. Baeg, "An Object Recognition Scheme Based on Visual Descriptors for a Smart Home Environment," in *16th IEEE International Conference on Robot and Human Interactive Communication*, August 2007.

[40] Mathworks, "Matlab," 2003, www.mathworks.com.

[41] L. Qiming, T.M. Khoshgoftaar, and A. Folleco, "Classification of Ships in Surveillance Video," in *IEEE International Conference on Information Reuse and Integration*, September 2006, pp. 432–437.

[42] F. Takens, "Dynamical Systems and Turbulence," in *Lecture Notes in Mathematics*, p. 898. Springer Verlag, Berlin, 1981.

[43] P. Grassberger and I. Procaccia, "Characterization of Strange Attractors," *Phys. Rev. Lett.*, vol. 50, pp. 346–349, 1983.

[44] B. Van der Pol and J. Van der Mark, "Frequency Demultiplication," *Nature*, vol. 120, pp. 363–364, 1927.

[45] D.H. Hubel, *Eye, Brain, and Vision*, W. H. Freeman; 2nd edition, 1995.

[46] R. Pashaie and N.H. Farhat, "Self-Organization in a Parametrically Coupled Logistic Map Network: A Model for Information Processing in the Visual Cortex," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 597 – 608, 2009.

[47] AM Albano GC deGuzman NN Greenbaum PE Rapp, ID Zimmerman and TR Bashore, "Experimental Studies of Chaotic neural Behavior: Cellular Activity and Electroencephalographic Signals," *HG Othmer (ed), Nonlinear Oscillations in Biology and Chemistry*, pp. 175–205, 1985, Springer, Berlin Heidelberg New York.

[48] A. Babloyantz and A. Destexhe, "An Instance of Low-dimensional Chaos in Epilepsy," *National Academy of Science, USA.* , no. 83, pp. 3513–3521, 1986.

[49] WJ Freeman and G Viaana Di Prisco, "EEG Spatial Pattern Differences with Discriminated Odors Manifest Chaotic and Limit Cycle Attractors in Olfactory Buld of Rabbits," *G Palm, and A Aertsen (eds), Brain Theory*, pp. 97–119, 1986, Springer, Berlin Heidelberg New York.

[50] G. Meyer-Kress, "Application of Dimension Algorithms to Experimental Chaos," *Hao Bai Lin (ed), Directions in Chaos*, pp. 122–147, 1987, World Scientific, Singapore.

[51] C. Nicolis A. Babloyantz and M. Salazar, "Evidence of of Chaotic Dynamics of Brain Activity During the Sleep Cycle," *Physics Letter (A)*, vol. 111, pp. 152–156, 1985.

[52] C. A. Skarda and W. J. Freeman, "Chaos and the New Science of the Brain," *Concepts in Neuroscience*, vol. 1, no. 2, pp. 275–285, 1990.

[53] H. Leung, S. Shanmugam, N. Xie, and S. Wang, "An Ergodic Approach for Chaotic Signal Estimation at Low SNR With Application to Ultra-wide-band Communication," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1091 – 1103, 2006.

[54] X. Liu and H. Leung, "Through the Wall Imaging using Chaotic Modulated Ultra Wideband Synthetic Aperture Radar," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2007, pp. III–1257 – III–1260.

[55] R. Dogaru, I. Dogaru, K. Hyongsuk, S. Sungsik, and G. Oubong, "Binary Synchronization of Chaos in Hybrid Cellular Automata for Low Complexity Image Compression and Transmission ," in *12th International Workshop on Cellular Nanoscale Networks and Their Applications, CNNA*, 2010, pp. 1 – 7.

[56] R. Dogaru, I. Dogaru, and K. Hyongsuk, "Chaotic Scan: A Low Complexity Video Transmission System for Efficiently Sending Relevant Image Features ," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 2, pp. 317 – 321, 2010.

[57] S. Chen and H. Leung, "A Chaotic Authentication Technique for Digital Video Surveillance ," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2005, pp. 533 – 536.

[58] K.S. Ntalianis and S.D. Kollias, "Chaotic video objects encryption based on mixed feedback, multiresolution decomposition and time-variant S-boxes," in *IEEE International Conference on Image Processing, ICIP*, 2005, pp. II – 1110–13.

[59] J. Peng, D. Zhang, and X. Liao, "Design of a Novel Image Block Encryption Algorithm Based on Chaotic Systems." in *8th IEEE International Conference on Cognitive Informatics, ICCI*, 2009, pp. 215 – 221.

[60] L. Min and H. Lu, "Design and Analysis of a Novel Chaotic Image Encryption," in *Second International Conference on Computer Modeling and Simulation, ICCMS*, 2010, pp. 517 – 520.

[61] C. Chen, C. Lin, and Y. Chiu, "Data Encryption Using ECG Signals with Chaotic Henon Map ." in *International Conference on Information Science and Applications (ICISA)*, 2010, pp. 1–5.

[62] H. Nakano and T. Saito, "Grouping Synchronization in a Pulse-coupled Network of Chaotic Spiking Oscillators," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1018 – 1026, 2004.

[63] L. Zhao, de Carvalho, and Z. Li, "Pixel Clustering by Adaptive Pixel Moving and Chaotic Synchronization," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1176 – 1185, 2004.

[64] Hanif Azhar, Khan Ifthekharuddin, and Robert Kozma, "Biology Inspired Automatic Segmentation of Brain Images with Chaos Synchronization," in *2nd Int. Conf. on Computational Harmonic Analysis, Vanderbilt University, USA*, http://www.math.vanderbilt.edu/ futamuf/abstracts.pdf, 2004.

[65] Hanif Azhar, Khan Ifthekharuddin, and Robert Kozma, "A Chaos Synchronization-Based Dynamic Vision Model for Image Segmentation," in *Proceedings of IEEE Intematonal Joint Conference on Neural Networks, Montreal, Canada*, 2005, pp. 3075–3080.

[66] X. Zhang and C. Zhang, "Fast Image Segmentation Based on Chaos Optimization and Recurring for 2-D Tsallis Entropy," in *International Symposium on Computer Network and Multimedia Technology, CNMT*, 2009, pp. 1 – 4.

[67] S. Yang and Z. Li, "Classification of Ship-radiated Signals Via Chaotic Features," *Electronics Letters*, vol. 39, no. 4, pp. 395 – 397, 2003.

[68] S. Yang, "Nonlinear Signal Classification in the Framework of High-Dimensional Shape Analysis in Reconstructed State Space ," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 8, pp. 512 – 516, 2005.

[69] I. Kokkinos and P. Maragos, "Nonlinear Speech Analysis Using Models For Chaotic Systems," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 6, pp. 1098 – 1109, 2005.

[70] D. Calitoiu, B.J. Oommen, and D. Nussbaum, "Desynchronizing a Chaotic Pattern Recognition Neural Network to Model Inaccurate Perception ," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 3, pp. 692 – 704, 2007.

[71] W.A. Chaovalitwongse, Y. Fan, and R.C. Sachdeo, "On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 6, pp. 1005 – 1016, 2007.

[72] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, "Mixed-Band Wavelet-Chaos-Neural Network Methodology for Epilepsy and Epileptic Seizure Detection." *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 9, pp. 1545 – 1551, 2007.

[73] O.H. Kocal, E Yuruklu, and I. Avcibas, "Chaotic-Type Features for Speech Steganalysis ," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 651 – 661, 2008.

[74] L. Zhao, M. Sun, J. Cheng, and Y. Xu, "A Novel Chaotic Neural Network With the Ability to Characterize Local Features and Its Application," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 735 – 742, 2009.

[75] S. Kacimi and S. Laurens, "The Correlation Dimension: A Robust Chaotic Feature for Classifying Acoustic Emission Signals Generated in Construction Materials," *Journal of Applied Physics*, vol. 106, no. 2, pp. 024909 – 024909–8, 2009.

[76] A. Yao and J. Jin, "The Development of A Video Metada Authoring and Browsing System in XML," ACM Intl. Conf. Proc. Series, Selected papers from the Pan-Sydney workshop on Visualisation, vol 2, Sydney, Australia, 2000, pp. 39–46.

[77] N. Guimaraes M. Costa, N. Correia. "Annotations as Multiple Perspectives of Video Content," in *ACM Multimedia, Juan-les-Pins, France*, 2002, pp. 283–286.

[78] W. Bailer, H. Mayer, H. Neuschmied, W. Haas, M. Lux, and W. Klieber, "Content-based Video Retrieval and Summarization using MPEG-7," in *SPIE and IST, Internet Imaging V, San Jose*, 2004, pp. 1–12.

[79] P.Fonseca and F.Pereira, "Automatic Video Summarization Based on MPEG-7 Descriptions," *Signal Processing: Image Communication*, vol. 19, no. 8, pp. 658–699, 2004.

[80] M. Karaman L. Goldmann and T. Sikora, "Human Body Posture Recognition Using MPEG-7 Descriptors," in *Proc. SPIE Visual Communications and Image Processing*, and Bhaskaran Vasudev Sethuraman Panchanathan. Ed.. 2004, vol. 5308, pp. 177–188.

[81] M. Lux, W. Klieber, and M. Granitzer, "Caliph and Emir: Semantics in Multimedia Retrieval and Annotation," in *19th International CODATA Conference, The Information Society: New Horizons for Science*, http://www.semanticmetadata.net/features/, 2004.

[82] M. Lux and M. Granitzer, "Retrieval of MPEG-7 based Semantic Descriptions," in *BTW Workshop on WebDB Meets IR in context of the GI Fachtagung fur Datenbanksysteme in Business, Technologie and Web*, http://www.semanticmetadata.net/features/, 2005.

[83] R. Klamma, M. Spaniol, and M. Jarke, "MECCA: Hypermedia Capturing of Collaborative Scientific Discourses about Movies," *The International Journal of an Emerging Discipline, N. Sharda (Ed.): Special Series on Issues in Informing Clients using Multimedia Communications*, vol. 8, pp. 3–38, 2005.

[84] IBM Research, "MARVEL: MPEG-7 Multimedia Search Engine," Tech. Rep., IBM, http://www.research.ibm.com/marvel/, 2006.

[85] IBM Research, "VideoAnnEx Annotation Tool," Tech. Rep., IBM, http://www.research.ibm.com/VideoAnnEx/, 2006.

[86] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event Detection and Analysis from Video Streams," *IEEE Pattern Anal. and Machine Intell.*, vol. 23, no. 8, pp. 873–889, 2001.

[87] X. Li and F.M. Porikli, "A Hidden Markov Model Framework for Traffic Event Detection Using Video Features," IEEE Int. Conf. on Image Processing ICIP, 2004, pp. 2901–2904.

[88] N. Haering, *A Framework for the Design of Event Detectors*, Ph.D. thesis, School of Computer Science, University of Central Florida, Orlando, USA, 1999.

[89] J. Fernyhough, A.G. Cohn, and D. C. Hogg, "Constructing Qualitative Event Models Autometically from Video Input," *Image Vis. Comput.*, vol. 18, no. 9, pp. 81–103, 2000.

[90] Y. Zhai, Z. Rasheed, and M. Shah, "Semantic Classification of Movie Scenes Using Finite State Machines," *IEE Proc. on Vis. Image Signal Process.*, vol. 152, no. 6, pp. 896–901, 2005.

[91] R. Vezzani R. Cucchiara, A. Prati, "An Intelligent Surveillance System for Dangerous Situation Detection in Home Environments," *Intelligenza Artificiale*, vol. 1, no. 1, pp. 11–15, 2004.

[92] M. M. Yeung, *Analysis, Modeling and Representation of Digital Video*, Ph.D. thesis, Electrical Engineering, Princeton University, USA, 1996.

[93] S. Park, *A Hierarchical Graphical Model for Recognizing Human Actions and Interactions in Video*, Ph.D. thesis, The University of Texas at Austin, USA, 2004.

[94] P. Partsinevelos, *Detection and Generation of Spatio-Temporal Trajectories for Motion Imagery*, Ph.D. thesis, Spatial Information Science and Engineering, The University of Maine, USA, 2002.

[95] S. Gong, J. Ng, and J. Sherrah, "On the Semantics of Visual Behaviour, Structured Events and Trajectories of Human Action," *Image and Vision Computing*, vol. 20, no. 12, pp. 873–888, Oct. 2002.

[96] D. Makris, *Learning an Activity-Based Semantic Scene Model*, Ph.D. thesis, School of Engineering and Mathematical Sciences Information Engineering Centre, City University, London, 2004.

[97] A. Hampapur, L. Brown, J. Connell, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C. Shu, and Y. Tian, "S3 R1: the IBM Smart Surveillance System-release 1," in *ACM SIGMM Workshop on Effective Telepresence*, 2004, pp. 59–62.

[98] D. Ayers and M. Shah, "Monitoring Human Behavior from Video Taken in an Office Environment," *Image and Vision Computing*, vol. 19, pp. 833–846, 2001.

[99] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time Surveillance of People and Their Activities," *IEEE Pattern Anal. Machine Intell.*, vol. 22, pp. 809–830, 2000.

[100] T. List, J. Bins, R. Fisher, and D. Tweed, "A Plug-and-Play Architecture for Cognitive Video Stream Analysis," in *IEEE Int. Workshop on Computer Architecture for Machine Perception*, Palermo, Italy, 2005, pp. 67–72.

[101] S. Venkatesh G. West, M. Lazarescu and T. Caelli, "On the automated interpretation and indexing of American football," in *IEEE Intl. Conf. on Multimedia Computing and Systems*, 1999, vol. 1, pp. 802–806.

[102] J. Kittler K. Messer W.J. Christmas, B. Levienaise-Obadia and D. Koubaroulis, "Generation of Semantic Cues for Sports Video Annotation," in *IEEE Intl. Conf. on Image Processing*, 2001, vol. 3, pp. 26–29.

[103] A. Hirano N. Babguchi S. Miyauchi and T. Kitahashi, "Collaborative Multimedia Analysis for Detecting Semantical Events from Broadcasted Sports Video," in *IEEE 16th Intl. Conf. on Pattern recognition*, 2002, vol. 2, pp. 1009–1012.

[104] S. Krishnan S. G. Quadri and L. Guan, "Indexing of NFL Video Using MPEG-7 Descriptors and MFCC features," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05)*, 2005, vol. 2, pp. 429–432.

[105] I. Lin, *Video Object Plane Extraction and Representation: Theory and Application*, Ph.D. thesis, Electrical Engineering, Princeton University, USA, 2000.

[106] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Semantic Video Analysis for Adaptive Content Delivery and Automatic Description," *IEEE Circuits and Systems for Video Technology*, vol. 15, pp. 1200–1209, 2005.

[107] M. Bonnet P. Piscaglia, A. Cavallaro and D. Douxchamps, "High Level Description of Video Surveillance Sequences," in *Multimedia Applications, Services and Techniques ECMAST, 4th European Conference*, H. Leopold and N. Garcia, Eds. 1999, vol. 1629, pp. 316–331, Springer-Verlag.

[108] A. J. Perrott, A. T. Lindsay, and A. P. Parkes, "Real-time Multimedia Tagging and Content-based Retrieval for CCTV Surveillance Systems," *Proc. SPIE Internet*

Multimedia Management Systems III, John R. Smith; Sethuraman Panchanathan; Tong Zhang; Eds., vol. 4862, pp. 40–49, 2002.

[109] J. Annesley and J. Orwell, "On the Use of MPEG-7 for Visual Surveillance," in *IEEE International Workshop on Visual Surveillance*, May 2006, pp. 501–504.

[110] M. R. Naphade, *A Probabilistic Framework for Mapping Audio-Visual Features To High-Level Semantics in Terms of Concepts and Context*, Ph.D. thesis, Electrical Engineering, University of Illinois at Urbana-Champaign, USA, 2001.

[111] C. Lin, B. L. Tseng, M. Naphade, A. Natsev, and J. R. Smith, "VideoAL: A Novel End-to-End MPEG-7 Video Automatic Labeling System," in *IEEE International Conference on Image Processing*, 2003, pp. III–53–6.

[112] R. Vezzani, C. Grana, D. Bulgarelli, and R. Cucchiara, "A Semi-Automatic Video Annotation tool with MPEG-7 Content Collections," *IEEE International Symposium on Multimedia*, vol. 1, no. 1, pp. 742–745, 2006.

[113] A. Mittal and L. Cheong, "Addressing the Problems of Bayesian Network Classification of Video Using High-Dimensional Features," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 230 – 244, February 2004.

[114] M. Shah, O. Javed, and K Shafique, "Automated Visual Surveillance in Realistic Scenarios," *IEEE Multimeda Magazine*, 2007.

[115] L. M. Brown, "View Independent Vehicle/Person Classification," in *ACM Second International Workshop on Video Surveillance and Sensor Networks*, 2004.

[116] R. Yan, J. Zhang, J. Yang, and A.G. Hauptmann, "A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2006.

[117] M. Soysal and A. A. Alatan, "Combining MPEG-7 Based Visual Experts for Reaching Semantics," in *8th Int. Workshop on Visual Content Processing and Representation, Madrid*, 2003, pp. 66–75.

[118] P. Viola, M. J. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," in *9th IEEE International Conference on Computer Vision*, 2003.

[119] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[120] G.L. Foresti, C. Micheloni, and L. Snidaro, "Event Classification for Automatic Visual-based Surveillance of Parking Lots," in *17th International Conference on Pattern Recognition*, 2004.

[121] M. Tsuchiya and H. Fujiyoshi, "Evaluating Feature Importance for Object Classification in Visual Surveillance," in *18th International Conference on Pattern Recognition*, 2006, pp. 978–981.

[122] A. Negre, H. Tran, N. Gourier, D. Hall, A. Lux, and J.L. Crowley, "Comparative Study of People Detection in Surveillance Scenes," *Lecture Notes in Computer Science on Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin / Heidelberg*. pp. 100 – 108, 2006.

[123] M. K. Leung and Y. H. Yang, "First Sight: A Human Body Outline Labeling System," *IEEE Pattern Anal. Machine Intell.*, vol. 17, no. 4, pp. 359–37, April 1995.

[124] L. Zhang, S.Z. Li, X. Yuan, and S Xiang, "Real-time Object Classification in Video Surveillance Based on Appearance Learning," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[125] S. Boragno, B. Boghossian, D. Makris, and S. Velastin, "Object Classification for Real-Time Video-Surveillance Applications," in *5th International Conference on Visual Information Engineering*, Aug 2008, pp. 192–197, Xian China.

[126] X. Li, G. Chen, Q. Ji, and E. Blasch, "A Non-cooperative Long-range Biometric System for Maritime Surveillance." in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[127] L. Xin and T. Tan, "Ontology-Based Hierarchical Conceptual Model for Semantic Representation of Events in Dynamic Scenes," in *2nd Joint IEEE Int. Workshop on VS-PETS. Beijing*, Oct 2005, pp. 57– 64.

[128] M. Barais, T. Caljon, V. Enescu, and H. Sahli, "A Framework for Integrating MPEG-7 Knowledge Templates into Video Surveillance Applications," in *IEEE Workshop on Multimedia Signal Processing*, October 2006, pp. 501–504.

[129] G. Strang. "Wavelet Transforms Versus Fourier Transforms," *Bulletin of the American Mathematical Society*, vol. 28, pp. 288–305, 1993.

[130] C. S. Won, D. K. Park, and S. Park, "Efficient Use of MPEG-7 Edge Histogram Descriptor." *ETRI Journal. Daejeon*, vol. 24, pp. 23–30, Feb 2002.

[131] J. Renno, D. Markis, and G.A. Jones, "Object Classification In Visual Surveillance Using Adaboost," in *Seventh International Workshop on Visual Surveillance*, 2007.

[132] A. Amer, "Voting-based Simultaneous Tracking of Multiple Video Objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 1448–1462, November 2005.

[133] T. Geest, L. F. Olsen, C. G. Steinmetz, R. Larter, and W. M. Schaffer, "Nonlinear Analyses Of Periodic And Chaotic Time Series From The Peroxidase-Oxidase Reaction," *The Journal of Physical Chemistry*, vol. 97, no. 32, pp. 8431â8441, 1993.

[134] C. V. Leeuwen, "Coupled Map Lattices as Models for Visual Information Processing," in *7th International Conference on Neural Information Processing*, 2000.

[135] *Video Object Samples For This Thesis*, H. Azhar, 2010.

[136] *TV Anywhere project*, www.tvanywhere.org, 2010.

[137] P.A. Devijver and J. Kittler, *Pattern Recognition: A statistical approach*, Pretice Hall International, 1982.

[138] N. Wyse, R. Dubes, and A.K. Jain, "A Critical Evaluation of Intrinsic Dimensionality Reduction Algorithms," *Pattern Recognition in Practice*, pp. 415–425, 1980.

[139] L. Jack H. Guo and A. Nandi, "Feature Generation Using Genetic Programming with Application to Fault Classification," *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, vol. 35, no. 1, 2005.

[140] R. Verschae, J. Ruiz del Solar, and M. Correa, "A Unified Learning Framework For Object Detection and Classification Using Nested Cascades of Boosted Classifiers," *Lecture Notes in Computer Science on Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin / Heidelberg*, vol. 19, no. 2, pp. 85–103, January 2008, ISSN:0932-8092.

[141] A. Noulas, "Visual Object Class Recognition," M.S. thesis, Artificial Intelligence, School of Informatics University of Edinburgh, 2005.

[142] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, V. Vapnik, and T. Poggio, "Feature Selection for SVMs," *Neural Information Processing Systems*, pp. 668–674, 2000.

[143] D. G. Stork and E. Yom-Tov, *Computer Manual in MATLAB to accompany Pattern Classification*, Wiley Interscience, 2004.

[144] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic Posture Classification for Human-Behavior Analysis," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 35, no. 1, pp. 42–54, Jan 2005.

[145] B. Moghaddam and M. Yang, "Gender Classification with Support Vector Machines," *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, vol. 19, no. 2, pp. 306, 2000.

[146] Z. Li and M. S. Drew, *Fundamentals of Multimedia*, Pearson Education, Inc., Prentice Hall, 2004.

[147] P. Salembier, "Overview of the MPEG-7 Standard and of Future Challenges for Visual Information Analysis," in *EURASIP Journal on Applied Signal Processing*, 2002, vol. 2002, pp. 343–353.

[148] Int.Organisation for Standardisation Organisation ISO/IEC / JTC1 / SC29 / WG11 N. Day, "MPEG-7 Projects and Demos," Tech. Rep., ISO/IEC JTC1/SC29/WG11 N4034, Singapore, http://www.chiariglione.org/mpeg, 2001.

[149] J. Bullington, "âAffectiveâ Computing and Emotion Recognition Systems: The Future of Biometric Surveillance?," in *2nd Annual Conference on Information Security Curriculum Development*, 2005, pp. 95–99.

[150] R. Wang and B. Fang, "Affective Computing and Biometrics Based HCI Surveillance System," in *International Symposium on Information Science and Engieering*, 2008, pp. 192–195.

[151] A. Ariyaeeinia, "Editorial Special Issue on Biometric Recognition," in *IET Signal Processing*, 2009, pp. 233–235.

[152] R. Chellappa and G. Aggarwal, "Video Biometrics," in *14th International Conference on Image Analysis and Processing*, 2007, pp. 363–370.

[153] A. Khalifa, K. Sundaraj, Z. Ibrahim, and V. Retnasamy, "Complex Background Subtraction For Biometric Identification," in *International Conference on Intelligent and Advanced Systems*, 2007.

[154] P. Kumar, A. Mittal, and P. Kumar, "Study of Robust and Intelligent Surveillance in Visible and Multi-modal Framework," in *Informatica*, 2008, pp. 63–77.

[155] T. Ko, "A Survey on Behavior Analysis in Video Surveillance for Homeland Security Applications," in *37th IEEE Applied Imagery Pattern Recognition Workshop*, 2009, pp. 1–8.

[156] G. Garibotto, "Video Surveillance and Biometric Technology Applications," in *Advanced Video and Signal Based Surveillance*, 2009.

[157] O. Arandjelovic, R. Hammoud, and R. Cipolla, "On Person Authentication by Fusing Visual and Thermal Face Biometrics," in *IEEE International Conference on Video and Signal Based Surveillance*, 2006.

[158] R. Raghavender, R. Jillela, and A. Ross, "Adaptive Frame Selection for Improved Face Recognition in Low-Resolution Videos," in *International Joint Conference on Neural Networks*, 2009.

[159] M. Hahnel, D. Kliinder, and K. Kraiss, "Color and Texture Features for Person Recognition," in *IEEE International Joint Conference on Neural Networks*, 2004, p. 652.

[160] KK Muhammad, J. Zhang, and L. Tian, "Protecting Biometric Data for Personal Identification," in *Lecture Notes in Computer Science, Advances in Biometric Person Authentication, Springer Berlin / Heidelberg*, 2005, pp. 629–38.

[161] X. Li, Z. Qi, Z. Yang, and J. Kong, "A Novel Hidden Transmission of Biometric Images Base on Chaos and Image Content." in *First International Workshop on Education Technology and Computer Science*, 2009, pp. 21–25.

[162] D. R. Kisku, A. Rattani, P. Gupta, and J. K. Sing, "Biometric Sensor Image Fusion for Identity Verification: A Case Study with Wavelet-based Fusion Rules and Graph Matching," in *IEEE Conference on Technologies for Homeland Security*, 2009, pp. 433–439.

[163] W. Chen and S. Yuun, "A Novel Personal Biometric Authentication Technique Using Human Iris Based on Fractal Dimension Features," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, pp. 201–4.

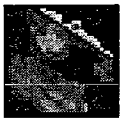
[164] F. Han, X. Yu, and J. Hu, "A New Way of Generating Grid-Scroll Chaos and its Application to Biometric Authentication," 2005.

Appendix A

XML descriptions of *MPEG-7* Visual Descriptors

This appendix presents sample XML description of MPEG-7 visual descriptors in different video objects.

Table A.1: MPEG-7 Dominant Color descriptor coefficents for different video objects.



```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001"
xmlns:xml = "http://www.w3.org/XML/1998/namespace"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7 = "urn:mpeg:mpeg7:schema:2001" xsi:schemaLocation
= "urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "DominantColorType">
<SpatialCoherency>0</SpatialCoherency>
<Value><Percentage>17</Percentage>
<Index>0 0 0 </Index></Value>
<Value><Percentage>2</Percentage>
<Index>6 5 5 </Index></Value>
<Value><Percentage>7</Percentage>
<Index>11 9 9 </Index></Value>
<Value><Percentage>3</Percentage>
<Index>17 16 16 </Index></Value>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_person'

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001"
xmlns:xml = "http://www.w3.org/XML/1998/namespace"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7 = "urn:mpeg:mpeg7:schema:2001" xsi:schemaLocation
= "urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "DominantColorType">
<SpatialCoherency>0</SpatialCoherency>
<Value><Percentage>21</Percentage>
<Index>0 0 0 </Index></Value>
<Value><Percentage>0</Percentage>
<Index>11 12 18 </Index></Value>
<Value><Percentage>3</Percentage>
<Index>28 29 30 </Index></Value>
<Value><Percentage>1</Percentage>
<Index>17 18 21 </Index></Value>
<Value><Percentage>1</Percentage>
<Index>6 7 8 </Index></Value>
<Value><Percentage>1</Percentage>
<Index>18 19 18 </Index></Value>
<Value><Percentage>1</Percentage>
<Index>11 13 12 </Index></Value>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_group_of_persons'

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001"
xmlns:xml = "http://www.w3.org/XML/1998/namespace"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7 = "urn:mpeg:mpeg7:schema:2001" xsi:schemaLocation
= "urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "DominantColorType">
<SpatialCoherency>0</SpatialCoherency>
<Value><Percentage>13</Percentage>
<Index>0 0 0 </Index></Value>
<Value><Percentage>9</Percentage>
<Index>5 4 3 </Index></Value>
<Value><Percentage>4</Percentage>
<Index>14 12 10 </Index></Value>
<Value><Percentage>2</Percentage>
<Index>9 4 3 </Index></Value>
<Value><Percentage>1</Percentage>
<Index>22 20 18 </Index></Value>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_vehicle'

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001"
xmlns:xml = "http://www.w3.org/XML/1998/namespace"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7 = "urn:mpeg:mpeg7:schema:2001" xsi:schemaLocation
= "urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "DominantColorType">
<SpatialCoherency>0</SpatialCoherency>
<Value><Percentage>13</Percentage>
<Index>0 0 0 </Index></Value>
<Value><Percentage>10</Percentage>
<Index>7 7 8 </Index></Value>
<Value><Percentage>7</Percentage>
<Index>14 14 14 </Index></Value>
<Value><Percentage>0</Percentage>
<Index>25 25 28 </Index></Value>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_unknown'

Table A.2: MPEG-7 Color Layout descriptor coefficents for different video objects.

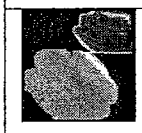
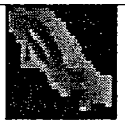
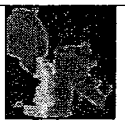
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ColorLayoutType"><YDCCoeff>7</YDCCoeff> <CbDCCoeff>30</CbDCCoeff> <CrDCCoeff>33</CrDCCoeff> <YACCoeff5>9 18 6 21 18 </YACCoeff5> <CbACCoeff2>16 16 </CbACCoeff2> <CrACCoeff2>17 16 </CrACCoeff2> </Descriptor></DescriptionUnit></Mpeg7>
	'has_person'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ColorLayoutType"><YDCCoeff>7</YDCCoeff> <CbDCCoeff>33</CbDCCoeff> <CrDCCoeff>29</CrDCCoeff> <YACCoeff5>14 14 9 16 12 </YACCoeff5> <CbACCoeff2>16 16 </CbACCoeff2> <CrACCoeff2>17 17 </CrACCoeff2> </Descriptor></DescriptionUnit></Mpeg7>
	'has_group_of_persons'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ColorLayoutType"><YDCCoeff>11</YDCCoeff> <CbDCCoeff>25</CbDCCoeff> <CrDCCoeff>35</CrDCCoeff> <YACCoeff5>20 7 8 10 6 </YACCoeff5> <CbACCoeff2>15 18 </CbACCoeff2> <CrACCoeff2>16 16 </CrACCoeff2> </Descriptor></DescriptionUnit></Mpeg7>
	'has_vehicle'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ColorLayoutType"><YDCCoeff>13</YDCCoeff> <CbDCCoeff>32</CbDCCoeff> <CrDCCoeff>29</CrDCCoeff> <YACCoeff5>20 18 5 17 7 </YACCoeff5> <CbACCoeff2>15 15 </CbACCoeff2> <CrACCoeff2>16 16 </CrACCoeff2> </Descriptor></DescriptionUnit></Mpeg7>
	'has_unknown'

Table A.3: MPEG-7 Edge Histogram descriptor coefficents for different video objects.



```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "EdgeHistogramType">
<BinCounts>4 4 3 7 1 0 4 0 6 1 0 0 0 2 0 0 0 0 0 0
5 1 1 7 1 2 1 0 7 3 2 1 3 5 0 0 0 0 0 0 0 1 0
1 0 2 3 1 7 0 2 5 1 7 0 3 1 0 3 0 0 0 0 0 1 0 0
0 1 1 1 2 3 2 0 3 1 1 1 0 </BinCounts>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_person'



```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "EdgeHistogramType">
<BinCounts>3 1 1 3 1 3 0 0 1 1 0 0 0 0 0 0 0 0 0 0
4 0 1 3 1 6 1 3 3 1 1 3 4 4 1 0 0 0 1 0 1 0 0
2 1 6 0 2 4 1 3 2 5 5 2 2 1 1 2 2 1 1 0 0 1 5 2
2 4 1 3 0 1 3 1 2 0 1 4 1 </BinCounts>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_group_of_persons'

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "EdgeHistogramType">
<BinCounts>3 2 6 0 1 2 6 5 5 1 0 7 2 4 0 3 1 1 7 0
6 1 5 3 0 0 0 0 0 0 0 2 1 0 0 4 0 1 5 0 5 0 1
7 0 0 0 1 0 0 1 2 1 1 0 6 0 0 3 0 0 3 0 7 1 0 7
1 4 0 0 7 0 1 0 2 4 6 3 0 </BinCounts>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_vehicle'

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "EdgeHistogramType">
<BinCounts>3 3 2 1 0 0 3 0 2 0 0 3 2 0 0 2 5 1 1 0
4 4 5 5 0 0 2 5 0 0 1 1 3 0 0 1 2 0 0 0 4 4 5
2 1 0 5 5 1 0 1 1 1 3 0 0 1 0 0 0 2 4 3 3 1 0 7
0 1 0 0 4 0 0 0 1 3 1 0 0 </BinCounts>
</Descriptor></DescriptionUnit></Mpeg7>
```

'has_unknown'

Table A.4: MPEG-7 Region Shape descriptor coefficents for different video objects.

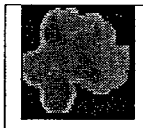
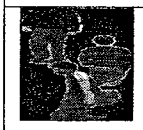
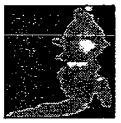
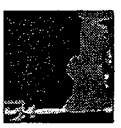
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "RegionShapeType"> <MagnitudeOfART>14 15 0 0 1 13 12 3 1 0 0 14 13 9 0 0 0 9 9 9 0 0 0 7 6 6 0 0 0 6 7 6 0 0 0 </MagnitudeOfART> </Descriptor></DescriptionUnit></Mpeg7>
	'has_person'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "RegionShapeType"> <MagnitudeOfART>12 15 1 2 1 15 15 11 2 2 0 14 14 11 3 2 0 7 1 8 3 1 0 11 11 4 3 0 0 10 11 9 3 0 0 </MagnitudeOfART> </Descriptor></DescriptionUnit></Mpeg7>
	'has_group_of_persons'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "RegionShapeType"> <MagnitudeOfART>14 15 1 1 4 13 10 1 4 2 2 14 13 9 3 0 3 6 7 8 2 1 1 7 6 9 1 0 1 5 5 6 1 1 0 </MagnitudeOfART> </Descriptor></DescriptionUnit></Mpeg7>
	'has_vehicle'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "RegionShapeType"> <MagnitudeOfART>14 15 9 11 12 15 13 11 7 3 6 13 12 7 0 5 7 13 12 7 6 4 4 2 1 1 0 2 2 9 8 6 5 4 2 </MagnitudeOfART> </Descriptor></DescriptionUnit></Mpeg7>
	'has_unknown'

Table A.5: MPEG-7 Contour Shape descriptor coefficents for different video objects.

	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001" xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation = "urn:mpeg:mpeg7:schema:2001 schema/Mpeg7-2001.xsd"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ContourShapeType"><GlobalCurvature>24 7 </GlobalCurvature> <PrototypeCurvature>2 8 </PrototypeCurvature> <HighestPeakY>38</HighestPeakY> <Peak peakX = "28" peakY = "3"/> <Peak peakX = "42" peakY = "5"/> <Peak peakX = "1" peakY = "5"/> <Peak peakX = "14" peakY = "6"/> <Peak peakX = "8" peakY = "6"/> <Peak peakX = "20" peakY = "6"/> <Peak peakX = "54" peakY = "5"/> <Peak peakX = "14" peakY = "5"/> <Peak peakX = "29" peakY = "7"/> </Descriptor></DescriptionUnit></Mpeg7>
	'has_person'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001" xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation = "urn:mpeg:mpeg7:schema:2001 schema/Mpeg7-2001.xsd"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ContourShapeType"><GlobalCurvature>39 7 </GlobalCurvature> <PrototypeCurvature>3 11 </PrototypeCurvature> <HighestPeakY>39</HighestPeakY> <Peak peakX = "38" peakY = "5"/> <Peak peakX = "48" peakY = "2"/> <Peak peakX = "21" peakY = "7"/> <Peak peakX = "53" peakY = "6"/> <Peak peakX = "9" peakY = "6"/> <Peak peakX = "15" peakY = "6"/> <Peak peakX = "43" peakY = "6"/> <Peak peakX = "29" peakY = "7"/> <Peak peakX = "6" peakY = "5"/> </Descriptor></DescriptionUnit></Mpeg7>
	'has_group_of_persons'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001" xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation = "urn:mpeg:mpeg7:schema:2001 schema/Mpeg7-2001.xsd"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ContourShapeType"><GlobalCurvature>63 7 </GlobalCurvature> <PrototypeCurvature>11 26 </PrototypeCurvature> <HighestPeakY>46</HighestPeakY> <Peak peakX = "24" peakY = "2"/> <Peak peakX = "44" peakY = "7"/> <Peak peakX = "20" peakY = "5"/> <Peak peakX = "53" peakY = "6"/> <Peak peakX = "10" peakY = "6"/> <Peak peakX = "45" peakY = "5"/> <Peak peakX = "52" peakY = "6"/> <Peak peakX = "35" peakY = "5"/> <Peak peakX = "42" peakY = "7"/> </Descriptor></DescriptionUnit></Mpeg7>
	'has_vehicle'
	<?xml version='1.0' encoding='ISO-8859-1' ?> <Mpeg7 xmlns = "urn:mpeg:mpeg7:schema:2001" xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation = "urn:mpeg:mpeg7:schema:2001 schema/Mpeg7-2001.xsd"> <DescriptionUnit xsi:type = "DescriptorCollectionType"> <Descriptor xsi:type = "ContourShapeType"><GlobalCurvature>9 8 </GlobalCurvature> <PrototypeCurvature>3 8 </PrototypeCurvature> <HighestPeakY>23</HighestPeakY> <Peak peakX = "17" peakY = "5"/> <Peak peakX = "24" peakY = "2"/> <Peak peakX = "59" peakY = "4"/> <Peak peakX = "20" peakY = "7"/> <Peak peakX = "57" peakY = "7"/> <Peak peakX = "11" peakY = "6"/> <Peak peakX = "3" peakY = "6"/> <Peak peakX = "22" peakY = "6"/> <Peak peakX = "25" peakY = "6"/> </Descriptor></DescriptionUnit></Mpeg7>
	'has_unknown'