1-1-2003

Object extraction in video sequences based on spatiotemporal independent component analysis

Zhenhe Chen
Ryerson University

OBJECT EXTRACTION IN VIDEO SEQUENCES BASED ON SPATIOTEMPORAL INDEPENDENT COMPONENT ANALYSIS

by

ZHENHE CHEN
Bachelor of Engineering
South China University of Technology, China, 1996

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2003

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Zhenhe Chen, Master of Applied Science, Electrical and Computer Engineering, Ryerson University.

Video object extraction is one of the most important areas of video processing in which objects from video sequences are extracted and used for many applications such as surveillance systems, pattern recognition etc.

In this research work, an object-based technique based on the spatiotemporal independent component analysis (stICA) is developed to extract moving objects from video sequences. Using the stICA, the preliminary source images containing moving objects in the video sequence are extracted. These images are processed using wavelet analysis, edge detection, region growing and multiscale segmentation techniques to improve the accuracy of the extracted objects. A novel compensation method is applied to deal with the nonlinear problem caused by the application of the stICA directly to the video sequences. The recovered objects are indexed by the singular value decomposition (SVD) and linear combination analysis. Simulation results demonstrate the effectiveness of the stICA-based object extraction technique in content-based video processing applications.

Acknowledgment

I have enjoyed and benefited from the pleasant and stimulating research environment at Department of Electrical and Computer Engineering, Ryerson University.

I am grateful to my supervisor, Dr. Xiao-Ping Zhang, for having guided me with an open but practical mind, for being always willing to answer questions or discuss problems and for infusing me with his intellectual honesty. I also thank my colleagues in Communications and Signal Processing Applications Lab (CASPAL), in particular Hua, for the interesting discussions and for the active help to enhance this document. My deep gratitude is also expressed to Karthikeyan, Yuhong and Kan who supplied encouraging supports for my research.

My special thanks go to each member of the dissertation jury for having accepted to evaluate this thesis. I have sincerely appreciated your comments and questions.

I would like to express my warm appreciations to Bonnie for caring and being patient during the past few years. Her understanding provided the strongest motivation to finish this writing.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Review of Previous Works 1

1.3 Objectives 3

1.4 Proposed Approaches and Methodologies 4

1.5 Overview of the Thesis 5

2 Principle Component Analysis and Independent Component Analysis **7**

2.1 Principle Component Analysis, Singular Value Decomposition, and Whitening 7

2.1.1 Principle Component Analysis 7

2.1.2 Singular Value Decomposition 8

2.1.3 Whitening 10

2.2 Independent Component Analysis 10

2.3 Comparison of PCA, Whitening and ICA 12

2.4 The ICA Estimation Methods 15

2.5 Spatiotemporal ICA 18

2.6 Summary 21

3 Formulation of the stICA Model for Video Sequences **23**

3.1 Formulation of the stICA Model for Video Sequences 23

3.2 The stICA Based Video Segmentation Approach 26

3.3 Simulation of the stICA applied to Video Processing in the First Iteration 28
3.4 Summary 30

4 Post-processing in the First Iteration 31
4.1 Using Wavelet Analysis to Locate Regions of Interest 31
4.2 Image Edge Detection with Region Growing 35
4.3 Multiscale Image Segmentation 40
4.4 Simulations of the Post-processing Techniques in the First Iteration 42
4.4.1 Simulation of Wavelet Analysis to Locate ROIs 43
4.4.2 Simulation of Edge Detection with Region Growing 46
4.4.3 Simulation of Multiscale Image Segmentation 49
4.5 Summary 52

5 A Compensation Approach of stICA for Practical Video Sequences 53
5.1 A Compensation Approach of stICA 53
5.2 Frame Object Indexing Approach 56
5.3 Simulations 57
5.3.1 Simulation of Compensation Approach of stICA 58
5.3.2 Simulation of the Frame Object Indexing Approach 59
5.4 Summary 67

6 Conclusion 69
6.1 Contribution 69
6.2 Possible Extension 70

Bibliography 72

Chapter 1

Introduction

1.1 Motivation

THE increasing popularity of video processing is due to the high demand for video in entertainment, security related applications, education, tele-medicine, database and new wireless telecommunications. Recently, interesting research topics such as automated and efficient video content-based techniques are attracting much attention.

Video content-based techniques are aimed at achieving significant data reduction of video by applying suitable transformation on video sequences based on their content. This data reduction has two main advantages: video databases work efficiently for searching content-based videos, and processing cost reduces dramatically. The content-based video presentation is an essential need for broadcasting services, Internet and security applications. This thesis develops a framework for automated content-based video processing based on the spatiotemporal independent component analysis (stICA). Both theoretical derivation and simulation results are provided to illustrate the effectiveness of the presented methods.

1.2 Review of Previous Works

The essence of this thesis is in applying the stICA technique to extract the objects in video sequences. A brief review of some of the works done in these fields is covered in

this section.

Raw video clips are usually binary streams that are not well organized. To represent their contents, video clips must be decomposed into objects so analysis can be performed. The object-based technique is one way of analyzing the video clips and it is gaining importance in achieving compression and performing content-based video retrieval.

Recently, there have been many video object segmentation techniques to extract or track the objects, such as transition-based [1] [2] and key frame estimation [3] [4]. The transition-based methods (also named scene change detection) look at the grayscale value difference between two image frames being considered. This process identifies any pixel as either being a "changed" or "unchanged" pixel when a function of its grayscale value difference is respectively greater than or smaller than a certain predetermined decision threshold. This kind of method often suffers from noise due to global thresholding and inaccurate moving object boundaries due to occlusion areas. Moreover, this method is very reliable for abrupt changes but not so effective for gradual changes.

A video key frame is the frame that can represent the salient content of a video shot. Key frames provide an abstraction for video processing. One important class of the methods is shot boundary based approach [3]. Another important class is unsupervised clustering based approach [4]. However, most of the key frame estimation methods perform object segmentation based on low-level image features and other readily available information instead of semantic primitives of video, such as objects of interest, actions and events. Thus it cannot satisfy the requirements of a video surveillance system.

All the above object segmentation approaches are frame-based techniques. In this thesis, we introduce a novel statistical analysis method based on the stICA. The stICA model is used to formulate the spatial and temporal independence of the different moving objects. The solution of the stICA model can therefore identify these objects.

In recent years, the independent component analysis (ICA) based techniques are getting much attention in video processing. The ICA based techniques have been applied in many areas of signal processing, medical application, neural networks, information theory

and telecommunications. The ICA can be used in two complementary ways to decompose an image sequence into a set of images and a corresponding set of time-varying image amplitudes. The spatial ICA (sICA) [5] finds a set of mutually independent component (IC) images and a corresponding set of unconstrained time courses, whereas the temporal ICA (tICA) [6] finds a set of IC time courses and a corresponding set of unconstrained images. However, the sICA and tICA can only seek either the ICs of images (frames) or the time courses, respectively. As shown by McKeown [5], the sICA extracts the independent images but these images' corresponding temporal sources could be highly correlated. This is undesirable for object-based video sequence analysis, since the corresponding time courses for the independent objects should be independent as well. The stICA, the generalization of the classic ICA, can blindly separate the independent sources from their spatial and temporal mixtures. It was initially developed in functional magnetic resonance imaging (fMRI) [8].

1.3 Objectives

The presented research focuses on the video sequences taken with a still camcorder. We assume that there is a stationary background in each frame. The objects and background can be considered the spatial ICs and the corresponding time courses can be considered the temporal ICs.

The following are the objectives of our proposed framework in this thesis:

1. To verify/assess the applicability of the stICA model for video sequences. This involves segmenting objects of interest from a stationary background in every video frame.

2. To deal with the limitations of the stICA model on video sequence applications, since objects of interest and their background are not linear combination.

3. To show that the algorithms proposed in this system are effective.

The novelty of the proposed methods in this thesis lies in the extraction of semantic moving objects through a background separation technique in a complex environment and in the processing of every independent frame of the video sequences.

The contributions of this thesis consist of

1. A new method of analyzing video sequences by the stICA model.

2. A novel compensation method to deal with the nonlinear combination problem in the stICA model for video sequences.

3. The integrated post-processing techniques based on wavelet analysis, edge detection with region growing and multiscale segmentation approaches.

1.4 Proposed Approaches and Methodologies

To achieve the goals mentioned above, the proposed system involves the following modules as stated in Figs. 1.1, 3.2, and 5.1 [9] [10]:

- The stICA is applied to the video frames to separate the spatial and temporal signals.

- The signals obtained after the stICA are further processed in the first iteration, where wavelet analysis, edge detection with region growing, and multiscale image segmentation techniques are employed to improve the accuracy.

- In the second iteration, a compensation approach is introduced to deal with the nonlinear combination problem of the stICA. A frame object indexing technique is then performed to reconstruct the sequence of frames containing only the objects. More precise video objects are extracted in this iteration.

1.5 Overview of the Thesis

This thesis will summarize the ICA and other related technologies in chapter 2. In chapter 3, the stICA model is used to formulate video sequences. Chapter 4 and chapter 5 elaborate on all the methodologies applied in the proposed two-iteration approach. There are simulation results and summaries from chapter 3 to chapter 5. Finally, chapter 6 summarizes all work included in this thesis and points out some possible future work that might improve the current stICA model.

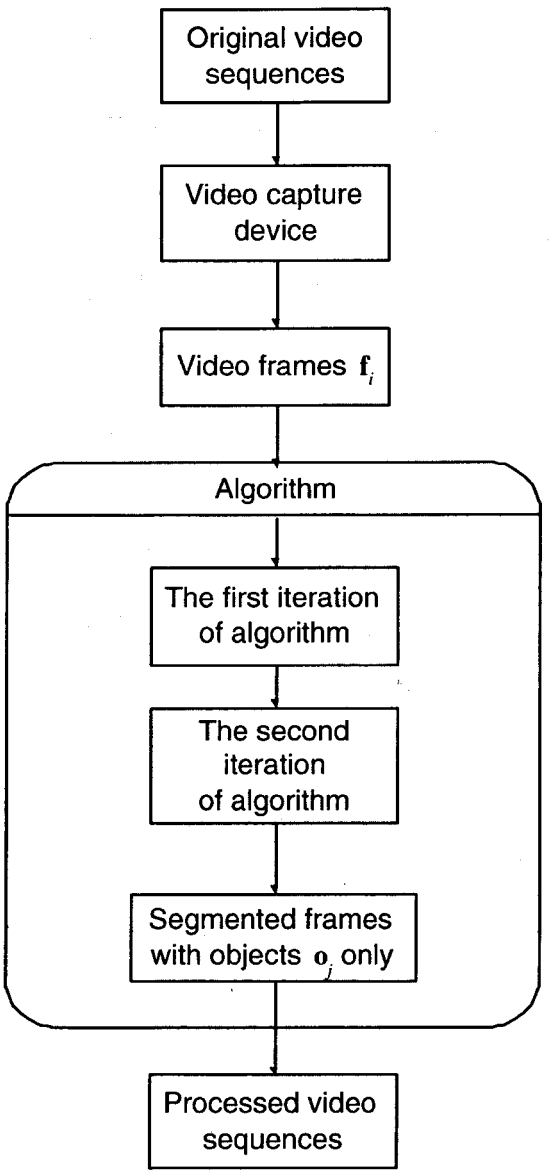

Figure 1.1: Block diagram of the framework. i and j are the indices of frames and objects respectively.

Chapter 2

Principle Component Analysis and Independent Component Analysis

IN this chapter, the basic concepts of principle component analysis (PCA), singular value decomposition (SVD), whitening, ICA and stICA are introduced. This chapter is a summary of the work stated in [11] [12] [8].

2.1 Principle Component Analysis, Singular Value Decomposition, and Whitening

2.1.1 Principle Component Analysis

The PCA is potentially valuable for applications involving reduction of the dimension of multivariate data. We suppose that $\mathbf{x}=[x_1, \cdots, x_n]^T$ is a zero-mean vector, and $\mathbf{m_x}$ is a vector with its mean values. $\mathbf{C_x}$ is the covariance matrix of $\mathbf{x}$ such that

$$\mathbf{C_x} = E\{(\mathbf{x} - \mathbf{m_x})(\mathbf{x} - \mathbf{m_x})^T\}. \tag{2.1}$$

Since the mean of vector $\mathbf{x}$ is zero, i.e. $\mathbf{m_x}=\mathbf{0}$, $\mathbf{C_x}$ is given by the correlation matrix

$$\mathbf{C_x} = E\{\mathbf{x}\mathbf{x}^T\}. \tag{2.2}$$

The goal of the PCA is to find an $n \times n$ orthogonal matrix $\mathbf{W}=[\mathbf{w}_1, \cdots, \mathbf{w}_n]$ that determines a linear transform of $\mathbf{x}$, i.e. $\mathbf{y}=\mathbf{W}^T\mathbf{x}$. It can be proven that such an orthogonal transform does not change the total variance of $\mathbf{x}$. This is true because the orthogonal

transform changes neither the angles between $\mathbf{x}$ and $\mathbf{y}$ nor the vectors' lengths, which means [12]

$$\{\text{total variance of } x_1, \cdots, x_n\} = \{\text{total variance of } y_1, \cdots, y_n\} = \lambda_1 + \cdots + \lambda_n, \quad (2.3)$$

where λ_j (j=1, $\cdots$, n) are the eigenvalues of $\mathbf{C_x}$.

The solution to the PCA is given by the unit-length eigenvectors $\mathbf{e}_1$, $\cdots$, $\mathbf{e}_n$ of $\mathbf{C_x}$. Thus we have $\mathbf{w}_1$=$\mathbf{e}_1$, $\cdots$, $\mathbf{w}_n$=$\mathbf{e}_n$. The detailed solution of eigen decomposition can be found in [13].

Compared with the SVD, eigen decomposition is only valid for a given square matrix [14] while the SVD is valid for any given $m\times n$ matrix [12]. Thus in practical applications, the SVD is the main tool used to perform the PCA. In the next subsection, the general idea of the SVD will be introduced.

2.1.2 Singular Value Decomposition

The SVD is one of the most widely used matrix factorizations in applied linear algebra. The SVD of $\mathbf{A}$ involves an $m\times n$ "diagonal" matrix Σ of the form

$$\Sigma = \begin{bmatrix} \mathbf{D} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (2.4)$$

where $\mathbf{D}$ is an $r\times r$ diagonal matrix for some r not exceeding the smaller of m and n.

Let $\mathbf{A}$ be an $m\times n$ matrix with rank r. Then there exists an $m\times n$ matrix Σ as in Eq. (2.4), where the diagonal entries in $\mathbf{D}$ are in the first r singular values of $\mathbf{A}$, $\sigma_1\geq \cdots\geq\sigma_r>0$, and there exists an $m\times m$ orthogonal matrix $\mathbf{U}$ and an $n\times n$ orthogonal matrix $\mathbf{V}$ such that [12]

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T. \quad (2.5)$$

Any factorization $\mathbf{A}$=$\mathbf{U}\Sigma\mathbf{V}^T$, with $\mathbf{U}$ and $\mathbf{V}$ orthogonal and Σ as in Eq. (2.4), is called an SVD of $\mathbf{A}$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are not unique, but the diagonal entries of Σ are necessarily the singular values of $\mathbf{A}$.

Since $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{A}^T\mathbf{A}$ is symmetric and can be orthogonally diagonalized [12]. Let $[\mathbf{v}_1, \cdots, \mathbf{v}_n]$ be an orthonormal basis for n-dimensional space ($\Re^n$) consisting of eigenvectors of $\mathbf{A}^T\mathbf{A}$. Let $\lambda_1, \cdots, \lambda_n$ be the associated eigenvalues of $\mathbf{A}^T\mathbf{A}$. Then, for $1 \leq i \leq n$,

$$\begin{aligned} \|\mathbf{A}\mathbf{v}_i\|^2 &= (\mathbf{A}\mathbf{v}_i)^T\mathbf{A}\mathbf{v}_i = \mathbf{v}_i^T\mathbf{A}^T\mathbf{A}\mathbf{v}_i \\ &= \mathbf{v}_i^T(\lambda_i\mathbf{v}_i) \\ &= \lambda_i(\mathbf{v}_i^T\mathbf{v}_i) = \lambda_i, \end{aligned} \tag{2.6}$$

where $\mathbf{v}_i$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$, and $\lambda_i\mathbf{v}_i = \mathbf{A}^T\mathbf{A}\mathbf{v}_i$. The singular values of $\mathbf{A}$ are the square roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$, denoted by $\sigma_1, \cdots, \sigma_n$. That is, $\sigma_i = \sqrt{\lambda_i}$, for $1 \leq i \leq n$, the singular values of $\mathbf{A}$ are the lengths of the vectors $\mathbf{A}\mathbf{v}_1, \cdots, \mathbf{A}\mathbf{v}_n$.

There is a theorem concerning about the rank and the singular values [12]: if an $m \times n$ matrix $\mathbf{A}$ has r nonzero singular values, $\sigma_1 \geq \cdots \geq \sigma_r > 0$ with $\sigma_{r+1} = \cdots = \sigma_n = 0$, then the rank of $\mathbf{A}$ is equal to r.

The SVD is based on the property of the ordinary diagonalization that can be imitated for rectangular matrices. Let us denote the symmetric matrix $\mathbf{A}^T\mathbf{A}$ by $\mathbf{B}$. The eigenvalues of $\mathbf{B}$ determine how much of the energy of $\mathbf{B}$ is distributed along the directions specified by the eigenvectors. If $\mathbf{B}\mathbf{x} = \lambda\mathbf{x}$ and $\|\mathbf{x}\| = 1$, then

$$\|\mathbf{B}\mathbf{x}\| = \|\lambda\mathbf{x}\| = |\lambda|. \tag{2.7}$$

If λ_1 is the eigenvalue with the greatest magnitude, then the corresponding unit eigenvector $\mathbf{v}_1$ identifies the direction along which the stretching effect of $\mathbf{B}$ is the greatest. This is, the length of $\mathbf{B}\mathbf{x}$ is maximized when $\mathbf{x} = \mathbf{v}_1$, and $\|\mathbf{B}\mathbf{v}_1\| = |\lambda_1|$, by Eq. (2.7).

Lay [12] describes the relationship between the PCA and the SVD as follows: if $\mathbf{C}$ is an $m \times n$ matrix of observation with zero mean, and if $\mathbf{A} = (1/\sqrt{n-1})\mathbf{C}^T$, then $\mathbf{A}^T\mathbf{A}$ is the covariance matrix of $\mathbf{C}$. The squares of the singular values of $\mathbf{A}$ are the eigenvalues of $\mathbf{A}^T\mathbf{A}$, and the singular vectors of $\mathbf{A}$ are the unit eigenvectors of $\mathbf{C}$. Through the SVD, the unit eigenvectors of the image matrices can be obtained. Thus the SVD is widely used to perform the PCA.

2.1.3 Whitening

Whitening is a useful preprocessing technique in signal processing. The term "white" comes from the fact that the power spectrum of white noise is constant over all frequencies, somewhat like the spectrum of white light contains all colors. A zero-mean random vector $\mathbf{y}=[y_1, \cdots, y_n]^T$ is said to be white if its elements y_i are uncorrelated and have unit variances:

$$E\{y_i y_j\} = \delta_{ij}, \tag{2.8}$$

Generally, the objective of whitening is: Given a random vector $\mathbf{x}$ with n elements, find a linear transformation $\mathbf{V}$ into another vector $\mathbf{y}$ such that

$$\mathbf{y} = \mathbf{V}\mathbf{x} \tag{2.9}$$

has elements that are uncorrelated and have unit variances.

Let us denote the covariance matrix of $\mathbf{x}$ by $\mathbf{C_x}$. Let $\mathbf{E}=[\mathbf{e}_1, \cdots, \mathbf{e}_n]$ be the matrix whose columns are the unit-norm eigenvectors of $\mathbf{C_x}$. Let $\mathbf{D}=\text{diag}(\lambda_1, \cdots, \lambda_n)$ be the diagonal matrix of the eigenvalues of $\mathbf{C_x}$. Then a linear whitening transform is [13] [11]

$$\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T. \tag{2.10}$$

It is easily proven that the matrix $\mathbf{V}$ of Eq. (2.10) is indeed a whitening transformation. In fact, $\mathbf{C_x}$ can be written in terms of its eigenvector and eigenvalue matrices $\mathbf{E}$ and $\mathbf{D}$ as $\mathbf{C_x}=\mathbf{EDE}^T$ [13], where $\mathbf{E}$ is an orthogonal matrix satisfying $\mathbf{E}^T\mathbf{E}=\mathbf{E}\mathbf{E}^T=\mathbf{I}$. It holds that:

$$E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{V}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{V}^T = \mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{E}\mathbf{D}\mathbf{E}^T\mathbf{E}\mathbf{D}^{-1/2} = \mathbf{I}. \tag{2.11}$$

The covariance of $\mathbf{y}$ is the unit matrix, hence $\mathbf{y}$ is white.

2.2 Independent Component Analysis

Imagine that there are two people speaking simultaneously in a room. Two microphones record these voices and give two time signals that can be denoted as $x_1(t)$ and $x_2(t)$. Each

of these recorded signals is a weighted sum of the speech signals $s_1(t)$ and $s_2(t)$ given by the two speakers, respectively. Usually, we express them as linear combination:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{2.12}$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \tag{2.13}$$

where the a_{ij} with i,j=1,2 are the parameters that depend on the distances of the microphones from the speakers. It would be very useful and challenging if we could restore the original signals $s_1(t)$ and $s_2(t)$, by using only the recorded signals $x_i(t)$. This is called the "cocktail-party problem".

This seems to be an impossible task since we know neither a_{ij} nor $s_i(t)$. One relatively new tool to estimate both a_{ij} and $s_i(t)$ relies on the use of the statistical information of the signals $s_i(t)$. This tool is named independent component analysis (ICA). In the ICA, the observed random vector $\mathbf{x}$ is modelled as

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2.14}$$

where the mixing matrix $\mathbf{A}$ is assumed to be square, i.e. the number of ICs is equal to the number of observed mixtures; and $\mathbf{s}$ is the original signal vector. This model can also be written as

$$\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i s_i \tag{2.15}$$

where $\mathbf{a}_i$ is the column vector of $\mathbf{A}$ and n is the total number of ICs.

By definition, elements s_i are statistically mutually independent (zero mean) random variables such that

$$p(\mathbf{s}) = \prod_{i=1}^{n} p_i(s_i). \tag{2.16}$$

Eq. (2.14) is the basic ICA model. The ICA model is a derivative model, which means it describes how the observed data are generated by a process of mixing the components s_i. The ICs s_i are latent variables, meaning that they cannot be directly observed. Also the mixing coefficients a_{ij} are assumed to be unknown. The ICA problem now becomes

the estimation of both the ICs $\mathbf{s}$ and the mixing matrix $\mathbf{A}$ using only the observation $\mathbf{x}$. This is a type of blind model identification.

There are some assumptions underlying the ICA method.

1. The source signals are assumed to be statistically independent.

2. The ICs must have non-Gaussian distributions.

There are two ambiguities in the ICA model in Eq. (2.14):

1. The variances(energies) of the ICs cannot be determined.
   Since both $\mathbf{s}$ and $\mathbf{A}$ are unknown, any scalar multiplier in one of the sources s_i could always be cancelled by dividing the corresponding column $\mathbf{a}_i$ of $\mathbf{A}$ by the same scalar.

2. The order of the ICs cannot be determined.
   This is also due to the indeterminacies because both $\mathbf{s}$ and $\mathbf{A}$ being unknown. We can freely change the order of the terms in Eq. (2.15), and call any of the ICs the first one.

2.3 Comparison of PCA, Whitening and ICA

To transform some given random variables into uncorrelated variables, whitening or the PCA is the straightforward method. However, whitening or the PCA cannot recover the ICs from these given random variables.

Two random variables y_1 and y_2 with zero mean are uncorrelated if their covariance is zero:

$$\text{cov}(y_1, y_2) = E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0. \tag{2.17}$$

Since their mean values are zero, $E\{y_1\}$=$E\{y_2\}$=0. In this case, the covariance is equal to the correlation corr(y_1,y_2)=$E\{y_1 y_2\}$, and uncorrelatedness is the same thing as zero correlation [11].

If two random variables are independent, they must be uncorrelated. Furthermore, for any variables derived from certain functions of these two variables, they must be uncorrelated as well. Suppose we have two functions f_1 and f_2 for two independent random variables y_1 and y_2 respectively, we have:

$$\begin{aligned} E\{f_1(y_1)f_2(y_2)\} &= \int\int f_1(y_1)f_2(y_2)p(y_1,y_2)dy_1dy_2 \\ &= \int\int f_1(y_1)f_2(y_2)p_1(y_1)p_2(y_2)dy_1dy_2 \\ &= \int f_1(y_1)p_1(y_1)dy_1 \int f_2(y_2)p_2(y_2)dy_2 \\ &= E\{f_1(y_1)\}E\{f_2(y_2)\}, \end{aligned} \tag{2.18}$$

which verifies that the variables $f_1(y_1)$ and $f_2(y_2)$ must also be uncorrelated.

However, on the other hand, uncorrelatedness does not imply independence. For example, suppose that (y_1,y_2) are discrete values and follow a distribution such that the probability of the pair being equal to any of the following values: (1,0), (0,1), (-1,0) and (0,-1) is $\frac{1}{4}$. Obviously y_1 and y_2 both have zero mean values. In this specific example, y_1, y_2 are uncorrelated, based on the calculation as follows:

$$\text{cov}\{y_1,y_2\} = \text{corr}\{y_1,y_2\} = \sum_{i=1}^{4} y_{1i}y_{2i}p(y_{1i},y_{2i}) = \sum_{i=1}^{4} 0\cdot p(y_{1i},y_{2i}) = 0. \tag{2.19}$$

On the other hand,

$$E\{y_1^2y_2^2\} = \sum_{i=1}^{4} y_{1i}^2y_{2i}^2p(y_{1i},y_{2i}) = \sum_{i=1}^{4} 0\cdot p(y_{1i},y_{2i}) = 0 \tag{2.20}$$

$$E\{y_1^2\}E\{y_2^2\} = \sum_{i=1}^{4} y_{1i}^2p(y_{1i}) \sum_{i=1}^{4} y_{2i}^2p(y_{2i}) = \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4} \tag{2.21}$$

$E\{y_1^2y_2^2\}\neq E\{y_1^2\}E\{y_2^2\}$. It violates the condition in Eq. (2.18), so the variables cannot be independent.

Whiteness is a slightly stronger property than uncorrelatedness. Whitening random vector $\mathbf{y}$ with zero mean will make its components uncorrelated and their variances equal unity. As shown in Eq. (2.8), the covariance matrix of $\mathbf{y}$ is $E\{\mathbf{y}\mathbf{y}^T\}=\mathbf{I}$. Whitening can be done by using eigenvalue decomposition of the covariance matrix as well as the SVD.

How far is the whitened data from being independent? Hyvärinen *et al.* showed that "whitening is only half ICA" [11]. Suppose that the data in the ICA model is whitened. Whitening transforms the mixing matrix $\mathbf{A}$ into a new one, $\tilde{\mathbf{A}}$. We have from Eqs. (2.9) and (2.14)

$$\mathbf{y} = \mathbf{VAs} = \tilde{\mathbf{A}}\mathbf{s}. \tag{2.22}$$

Since $\tilde{\mathbf{A}}$=$\mathbf{VA}$ is orthogonal, $E\{\mathbf{y}\mathbf{y}^T\}$=$\tilde{\mathbf{A}}E\{\mathbf{s}\mathbf{s}^T\}\tilde{\mathbf{A}}^T$=$\mathbf{I}$, which means the searching for the mixing matrix can be restricted to the space of orthogonal matrices. Instead of having to estimate the n^2 parameters that are the elements of the orthogonal matrix $\mathbf{A}$, we only need to estimate an orthogonal mixing matrix $\tilde{\mathbf{A}}$. This orthogonal matrix has $n(n\text{-}1)/2$ degrees of freedom. The complexity of the ICA problem is reduced partially, "ICA is solved on the half way" [11].

The following example shows the fact that only non-Gaussian variables are accepted in the ICA, which is explained by whitening. Assume that the joint distribution of two ICs, s_1 and s_2, is the standard Gaussian distribution. Their joint probability density function (pdf) is [11]

$$p(s_1, s_2) = \frac{1}{2\pi}\exp\left[-\frac{s_1^2 + s_2^2}{2}\right] = \frac{1}{2\pi}\exp\left[-\frac{\|\mathbf{s}\|^2}{2}\right]. \tag{2.23}$$

Furthermore, let us assume that the mixing matrix $\mathbf{A}$ is orthogonal. For example, we could assume that this is so because the data has been whitened. Using the classic formula of transforming pdf's in [11], and noting that for an orthogonal matrix $\mathbf{A}^{-1}$=$\mathbf{A}^T$ holds, we get the joint pdf of the mixtures x_1 and x_2

$$p(x_1, x_2) = \frac{1}{2\pi}\exp\left[-\frac{\|\mathbf{A}^T\mathbf{x}\|^2}{2}\right]|\det\mathbf{A}^T|. \tag{2.24}$$

Because of $\mathbf{A}$'s orthogonality, we have $\|\mathbf{A}^T\mathbf{x}\|^2$=$\|\mathbf{x}\|^2$ and $|\det\ \mathbf{A}|$=1. Note that if $\mathbf{A}$ is orthogonal, so is $\mathbf{A}^T$. Thus

$$p(x_1, x_2) = \frac{1}{2\pi}\exp\left[-\frac{\|\mathbf{x}\|^2}{2}\right]. \tag{2.25}$$

The orthogonal mixing matrix has no effect on Gaussian pdf, because it does not appear in the pdf. Both the original and mixed distributions are identical. The reason for such un-

identity is due to the fact that uncorrelated Gaussian random variables are independent. This tells us that the information of the ICs does not exceed that of whitening.

2.4 The ICA Estimation Methods

- The ICA by Negentropy

  A fundamental result of the information theory is that a Gaussian variable has the largest entropy among all random variables of equal variance [15] [16]. From this conclusion, two hints can be obtained:

  1. Entropy can be used as a measure of non-Gaussianity, and
  2. Gaussian distribution is the "most random" or the least structured among all distributions.

  Let us define negentropy J as a measure of non-Gaussianity that is zero for a Gaussian variable and always non-negative:

  $$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}), \tag{2.26}$$

  where $\mathbf{y}_{\text{gauss}}$ is a Gaussian random variable of the same correlation (and covariance) matrix as $\mathbf{y}$, and H is the entropy. This negentropy is always non-zero and is zero if and only if $\mathbf{y}$ has a Gaussian distribution.

  Using negentropy as a measure of non-Gaussianity has its advantages. It is well justified by statistical theory [11] [17]. However, the computational complexity is very high.

- The ICA by Minimization of Mutual Information

  Another approach for the ICA estimation, inspired by information theory, is minimization of mutual information. One can discover the fundamental relationship between mutual information and negentropy.

The definition of mutual information I comes from differential entropy. Here we denote m random variables y_i, such that:

$$I(y_1, \cdots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}), \tag{2.27}$$

where i=1, $\cdots$, m and $\mathbf{y}$ is the vector containing y_1, $\cdots$, y_m. Mutual information is a natural measure of the dependence between random variables. In fact, it is equivalent to the Kullback-Leibler divergence [11] between the joint density $f(\mathbf{y})$ and the product of its marginal densities; a very natural measure for independence. It is always non-negative, and zero if and only if the variables are statistically independent.

To show the relationship between mutual information and negentropy, an important property of mutual information is that if an invertible linear transformation $\mathbf{y}$=$\mathbf{W}\mathbf{x}$ exists then Eq. (2.27) can be expressed as [17]

$$I(y_1, \cdots, y_m) = \sum_{i} H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{W}|. \tag{2.28}$$

Let us assume that y_i is whitened (y_i is uncorrelated and has unit variance) - $E\{\mathbf{y}\mathbf{y}^T\}$=$\mathbf{W}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T$=$\mathbf{I}$. We can get

$$\begin{aligned} \det \mathbf{I} &= 1 = \det(\mathbf{W}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T) \\ &= (\det \mathbf{W})(\det E\{\mathbf{x}\mathbf{x}^T\})(\det \mathbf{W}^T) \end{aligned} \tag{2.29}$$

and this implies that det$\mathbf{W}$ must be constant since det$E\{\mathbf{x}\mathbf{x}^T\}$ does not depend on $\mathbf{W}$. Moreover, for y_i of unit variance, entropy and negentropy differ only by a constant and the sign, as can be seen in Eq. (2.26). Thus we have

$$I(y_1, \cdots, y_m) = C - \sum_{i} J(y_i). \tag{2.30}$$

where C is a constant that does not depend on $\mathbf{W}$. This derivation shows that the ICA estimation by minimization of mutual information is equivalent to maximizing the sum of non-Gaussianities of the estimates, when the estimates are constrained

to be uncorrelated. This constraint can simplify the computation considerably. Thus mutual information gives another rigorous justification for finding maximally non-Gaussian directions [11] [17].

- The ICA by Maximum Likelihood Estimation

Starting from the density p_x of the ICA model in Eq. (2.14), we can get [11]

$$p_x(\mathbf{x}) = \mid \det \mathbf{W} \mid p_s(\mathbf{s}) = \mid \det \mathbf{W} \mid \prod_i p_i(s_i), \tag{2.31}$$

where $\mathbf{W}=\mathbf{A}^{-1}$, and the p_i are the densities of the ICs. If we denote $\mathbf{W}=[\mathbf{w}_1, \cdots, \mathbf{w}_n]^T$, we have

$$p_x(\mathbf{x}) = \mid \det \mathbf{W} \mid \prod_i p_i(\mathbf{w}_i^T \mathbf{x}). \tag{2.32}$$

If there are K observations of $\mathbf{x}$, denoted by $\mathbf{x}(1), \cdots, \mathbf{x}(K)$. Then the likelihood can be obtained as the product of this density evaluated at the K points. This is denoted by L and considered as a function of $\mathbf{W}$ [18]:

$$L(\mathbf{W}) = \prod_{t=1}^{K} \prod_{i=1}^{n} p_i(\mathbf{w}_i^T \mathbf{x}(t)) \mid \det \mathbf{W} \mid . \tag{2.33}$$

Because many density functions contain an exponential function, it is more convenient to deal with the log-likelihood function

$$\log L(\mathbf{W}) = \sum_{t=1}^{K} \sum_{i=1}^{n} \log p_i(\mathbf{w}_i^T \mathbf{x}(t)) + K \log \mid \det \mathbf{W} \mid . \tag{2.34}$$

To simplify notation, we can denote the sum over the sample index t by an expectation operator, and divide the likelihood by K to obtain

$$\frac{1}{K} \log L(\mathbf{W}) = E\{\sum_{i=1}^{n} \log p_i(\mathbf{w}_i^T \mathbf{x})\} + \log \mid \det \mathbf{W} \mid \tag{2.35}$$

This expectation is not the theoretical expectation, but an average computed from the observed sample.

Gradient methods are the simplest algorithms to maximize the likelihood. The Bell-Sejnowski algorithm [6] is one of the most popular maximum likelihood estimation

techniques. Bell *et al.* showed that the gradient of the log-likelihood in Eq. (2.35) is:

$$\frac{1}{K}\frac{\partial \log L}{\partial \mathbf{W}} = [\mathbf{W}^T]^{-1} + E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T\}. \tag{2.36}$$

Here $\mathbf{g}(\mathbf{W}\mathbf{x})$=[$g_1(\mathbf{w}_1^T\mathbf{x})$, $\cdots$, $g_n(\mathbf{w}_n^T\mathbf{x})$] is a component-wise vector function that consists of the score function g_i of the distribution of s_i, which is defined as

$$g_i = (\log p_i)' = \frac{p_i'}{p_i}. \tag{2.37}$$

This gives the following algorithm for maximum likelihood estimation:

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T\}. \tag{2.38}$$

As it is a stochastic version of this algorithm, the expectation is omitted. In each step of the algorithm, only one data point is used:

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + \mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T. \tag{2.39}$$

Due to the inversion of the matrix $\mathbf{W}$ that is needed in every step, this algorithm converges slowly. The convergence can be improved by using whitening [6].

2.5 Spatiotemporal ICA

The stICA is the generalization of the classic ICA. The distinction between the ICs and the mixing matrix is completely abolished. Considering the data with n observed vectors as its columns: $\mathbf{X}$=[$\mathbf{x}_1$, $\cdots$, $\mathbf{x}_n$], and likewise for the ICs $\mathbf{S}$=[$\mathbf{s}_1$, $\cdots$, $\mathbf{s}_n$]. The ICA model can be expressed as

$$\mathbf{X} = \mathbf{A}\mathbf{S}. \tag{2.40}$$

Taking a transpose of this equation, we have

$$\mathbf{X}^T = \mathbf{S}^T\mathbf{A}^T. \tag{2.41}$$

Now we find that the matrix $\mathbf{S}$ is like a mixing matrix, with $\mathbf{A}^T$ giving the realizations of the "ICs". In the conventional ICA model Eq. (2.14), the difference between $\mathbf{s}$ and

$\mathbf{A}$ is due to the statistical assumptions made on $\mathbf{s}$. Now for the stICA, the independent constraints are made on both $\mathbf{A}$ and $\mathbf{S}$.

The stICA was initially developed for fMRI that is a form of magnetic resonance imaging (MRI) of the brain that registers blood flow to functioning areas of the brain [7]. The fMRI signal associated with a given voxel is affected by a subject's general arousal levels, the experimental task being executed, drifting sensor outputs, and noise. Thus the signal at each voxel consists of a mixture of underlying source signals (Fig. 2.1). Stone uses the stICA to separate signal mixtures into a set of statistically independent signals [8]. He describes a matrix containing a sequence of n fMRI mixtures $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$. Each image $\mathbf{x}_i$ is an $m \times 1$ vector. A linear decomposition into k modes is defined by a matrix factorization like Eq. (2.40)

$$\mathbf{X} = \mathbf{S}\Lambda\mathbf{T}^T, \tag{2.42}$$

where $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_k]$, $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_k]$ and Λ is a diagonal matrix of scaling parameters. The independent image vectors $\mathbf{s}_i$ are the columns of spatial images $\mathbf{S}$ and the corresponding independent time courses $\mathbf{t}_i$ are the columns of $\mathbf{T}$.

Using the SVD [12], fMRIs are decomposed into two parts, eigenimages $\mathbf{U}$ and corresponding eigensequences $\mathbf{V}$:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T = (\mathbf{U}\Sigma^{1/2})(\mathbf{V}^T\Sigma^{1/2}) = \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T, \tag{2.43}$$

where $\mathbf{U}$ is an $m \times k$ matrix of $k \leq m$ eigenimages, $\mathbf{V}$ is an $n \times k$ matrix of $k \leq n$ eigensequences, and Σ is a diagonal matrix of singular values. In order to determine the ICs $\mathbf{S}$ and $\mathbf{T}$, two $k \times k$ unmixing matrices $\mathbf{W}_S$ and $\mathbf{W}_T$ are assumed to exist such that

$$\mathbf{S} = \widetilde{\mathbf{U}}\mathbf{W}_S, \tag{2.44}$$

and

$$\mathbf{T} = \widetilde{\mathbf{V}}\mathbf{W}_T. \tag{2.45}$$

where $\widetilde{\mathbf{U}} = \mathbf{U}\Sigma^{1/2}$ and $\widetilde{\mathbf{V}} = \mathbf{V}\Sigma^{1/2}$. Now we have

$$\mathbf{X} = \mathbf{S}\Lambda\mathbf{T}^T = \widetilde{\mathbf{U}}\mathbf{W}_S(\widetilde{\mathbf{V}}\mathbf{W}_T)^T = \widetilde{\mathbf{U}}\mathbf{W}_S\mathbf{W}_T^T\widetilde{\mathbf{V}}^T. \tag{2.46}$$

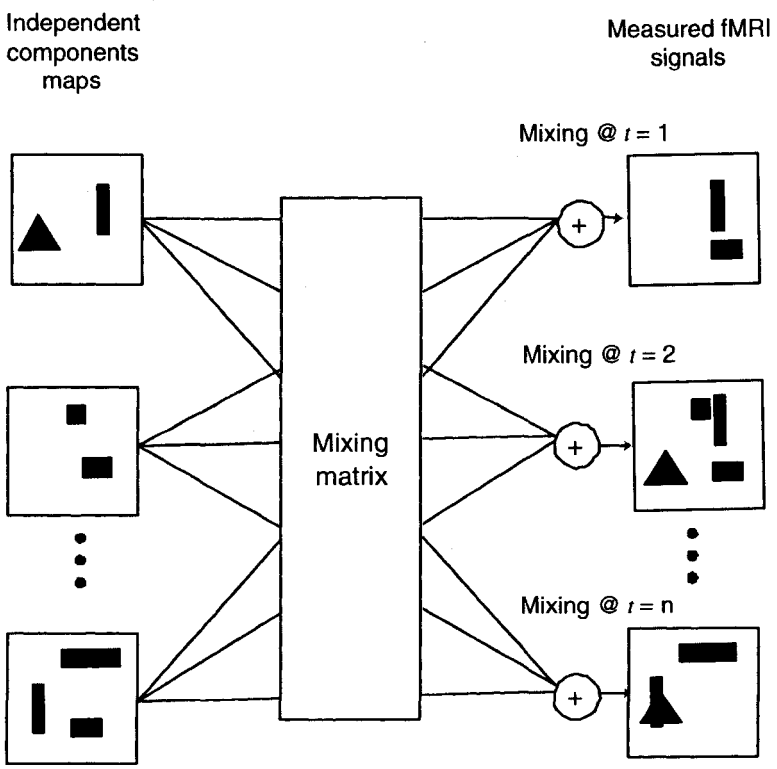

Figure 2.1: Illustration of fMRI mixing.

Given that $\mathbf{X}=\widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T=\mathbf{S}\Lambda\mathbf{T}^T$, it can be shown that $\mathbf{W}_T=(\mathbf{W}_S^{-1})^T(\Lambda^{-1})^T$.

To find the unmixing matrices $\mathbf{W}_T$ and $\mathbf{W}_S$, it is necessary to simultaneously maximize a function h_{ST} of the spatial entropy

$$h_S = H(\sigma(\widetilde{\mathbf{U}}\mathbf{W}_S)), \tag{2.47}$$

and temporal entropy

$$h_T = H(\tau(\widetilde{\mathbf{V}}\mathbf{W}_T)), \tag{2.48}$$

where σ and τ approximate the cumulative density function (cdf) of each of the spatial source signals and temporal signals, respectively. The function h to be maximized is defined as:

$$h_{ST}(\mathbf{W}_S) = \alpha h_S + (1-\alpha)h_T, \tag{2.49}$$

where α is a weighting factor given to spatial and temporal entropy. To optimize these two entropies by maximum likelihood estimation [6], their notations need to be changed to:

$$h_S = \log \mid \mathbf{W}_S \mid + \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{k} \log \sigma_i'(s_{ij}), \tag{2.50}$$

and

$$h_T = \log \mid \mathbf{W}_T \mid + \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{k} \log \tau_i'(t_{ij}), \tag{2.51}$$

where s_{ij} and t_{ij} are the corresponding elements of $\mathbf{S}$ and $\mathbf{T}$ in Eq. (2.42). σ_i and τ_i are the cdfs of the spatial and temporal signals, respectively. Their derivatives σ_i' and τ_i' are the corresponding pdfs.

One can recover the spatial signals and the time courses at the same time using maximum likelihood estimation, which is similar to the conventional ICA [11] approximation techniques.

2.6 Summary

Let us review the procedures to find the ICs from the mixed observed data. The basis of this approach is that if the model in Eq. (2.14) holds, then the ICs corresponding to the uncorrelated one-dimensional projections are maximally non-Gaussian. For an observed random vector $\mathbf{x}$, a vector $\mathbf{w}_i$ is sought such that

$$\hat{s}_i = \mathbf{w}_i^T \mathbf{x} \tag{2.52}$$

have a maximally non-Gaussian distributions and are mutually uncorrelated $E\{\hat{s}_i \hat{s}_j\}$=0, when $i \neq j$.

A simple way to do this is to whiten the data, and then seek orthogonal, non-Gaussian projections. This is justified since uncorrelated projections in the original data correspond to orthogonal projections in the whitened data, and vice versa. Thus, a two-step process is used to estimate the ICs:

1. The observed vector $\mathbf{x}$ is transformed by a whitening process $\mathbf{y}$=$\mathbf{V}\mathbf{x}$ such that the elements of $\mathbf{y}$ are of unit variance and uncorrelated, i.e. $E\{\mathbf{y}\mathbf{y}^T\}$=$\mathbf{I}$.

2. An orthogonal matrix $\mathbf{W}$ that maximizes the non-Gaussianity of the elements of $\hat{\mathbf{s}}=\mathbf{W}\mathbf{y}$ can be obtained.

For the stICA, there are more constraints on both $\mathbf{A}$ and $\mathbf{s}$, so the notations are changed to $\mathbf{S}$ and $\mathbf{T}$, respectively. The algorithm for maximizing the independence on $\mathbf{S}$ and $\mathbf{T}$ is the same as the ICA. Through the stICA approach, the ICs in $\mathbf{S}$ and $\mathbf{T}$ can be found.

Chapter 3

Formulation of the stICA Model for Video Sequences

FROM the last chapter, we can see that the ICA is an ideal tool for data analysis, especially for source data separation. In this chapter, the ICA is employed to extract objects of interest from video sequences.

3.1 Formulation of the stICA Model for Video Sequences

Let us denote a video sequence with n frames as $\hat{\mathbf{F}}=[\hat{\mathbf{f}}_1, \cdots, \hat{\mathbf{f}}_N]$, where $\hat{\mathbf{f}}_i$ is an $M \times 1$ column vector representing a frame that contains M pixels. These image vectors are constructed by taking the column-wise elements from the frame images. Thus the dimension of matrix $\hat{\mathbf{F}}$ is $M \times N$. The mutual independent objects of interest are denoted as $\mathbf{O}=[\mathbf{o}_1, \cdots, \mathbf{o}_K]$, where $\mathbf{o}_i$ is constructed in the same way as $\hat{\mathbf{f}}_i$ and $K \leq N$. The dimension of the object vector $\mathbf{o}_i$ should be the same as $\hat{\mathbf{f}}_i$, $M \times 1$. Thus the dimension of $\mathbf{O}$ is $M \times K$. If the video sequence is captured by a fixed camera, for example in the surveillance security system, the background is a constant vector. To simplify the work, the stationary background can be considered as a vector of $\mathbf{O}$, say $\mathbf{o}_K$. The independent temporal signals time courses $\mathbf{A}=[\mathbf{a}_1, \cdots, \mathbf{a}_K]$ affect the objects on every time unit. Again, we use the same method to construct the time course column vector $\mathbf{a}_i$. In every time unit there should be time courses affecting each object and the dimension of any time course vector $\mathbf{a}_i$ should be

equal to the number of video frames, i.e. $N\times 1$. This means that each column of $\mathbf{A}$ is the time signature for the corresponding objects in $\mathbf{O}$. The dimension of $\mathbf{A}$ is $N\times K$, where $K\leq N$. Because the background is stationary, the corresponding time course vector $\mathbf{a}_K$ has no effect on it, which means all elements of vector $\mathbf{a}_K$ have value 1. We have

$$\hat{\mathbf{F}} = \mathbf{O}\mathbf{A}^T. \tag{3.1}$$

To find out each object's effect on the video frames, we expand the matrices:

$$\begin{aligned}
[\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \cdots, \hat{\mathbf{f}}_N] &= [\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_K][\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_K]^T \\
&= \begin{bmatrix} o_{11} & o_{12} & \cdots & o_{1K} \\ \vdots & \vdots & \cdots & \vdots \\ o_{M1} & o_{M2} & \cdots & o_{MK} \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{N1} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1K} & a_{2K} & \cdots & a_{NK} \end{bmatrix} \\
&= \begin{bmatrix} a_{11}o_{11} & a_{21}o_{11} & \cdots & a_{N1}o_{11} \\ \vdots & \vdots & \cdots & \vdots \\ a_{11}o_{M1} & a_{21}o_{M1} & \cdots & a_{N1}o_{M1} \end{bmatrix} + \begin{bmatrix} a_{12}o_{12} & a_{22}o_{12} & \cdots & a_{N2}o_{12} \\ \vdots & \vdots & \cdots & \vdots \\ a_{12}o_{M2} & a_{22}o_{M2} & \cdots & a_{N2}o_{M2} \end{bmatrix} \\
&+ \cdots + \begin{bmatrix} a_{1K}o_{1K} & a_{2K}o_{1K} & \cdots & a_{NK}o_{1K} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1K}o_{MK} & a_{2K}o_{MK} & \cdots & a_{NK}o_{MK} \end{bmatrix}.
\end{aligned} \tag{3.2}$$

A function g is assumed to describe the object $\mathbf{o}_i$'s contribution to $\hat{\mathbf{F}}$. From the above matrices expansion, we can see:

$$g(\mathbf{o}_i) = \mathbf{o}_i\mathbf{a}_i^T, \text{ where } i = 1, \cdots, K. \tag{3.3}$$

These equations reveal the fact that $\mathbf{a}_i$ is the time signature for the corresponding object $\mathbf{o}_i$. We can rewrite Eq. (3.1) in vector format as:

$$\hat{\mathbf{F}} = \sum_{i=1}^{K} \mathbf{o}_i\mathbf{a}_i^T. \tag{3.4}$$

To find the element construction in jth video frame $\hat{\mathbf{f}}_j(j=1, \cdots, K)$, we need to utilize the linear combination relationship between the spatial elements o_{ik} and the time sequence signals a_{jk} from previous assumptions (Eqs. (3.1) and (3.2)):

$$\hat{f}_{ij} = \sum_{k=1}^{K} o_{ik}a_{jk}, \tag{3.5}$$

where i=1, $\cdots$, M. This equation reveals that an element at a specific location in a frame is the linear combination of the elements at the same locations of all the independent spatial objects at a certain time moment i; i.e. the ith element in the jth video frame is the linear combination of all the ith elements in all the independent object vectors $\mathbf{o}_1$, $\cdots$, $\mathbf{o}_K$ at ith moment.

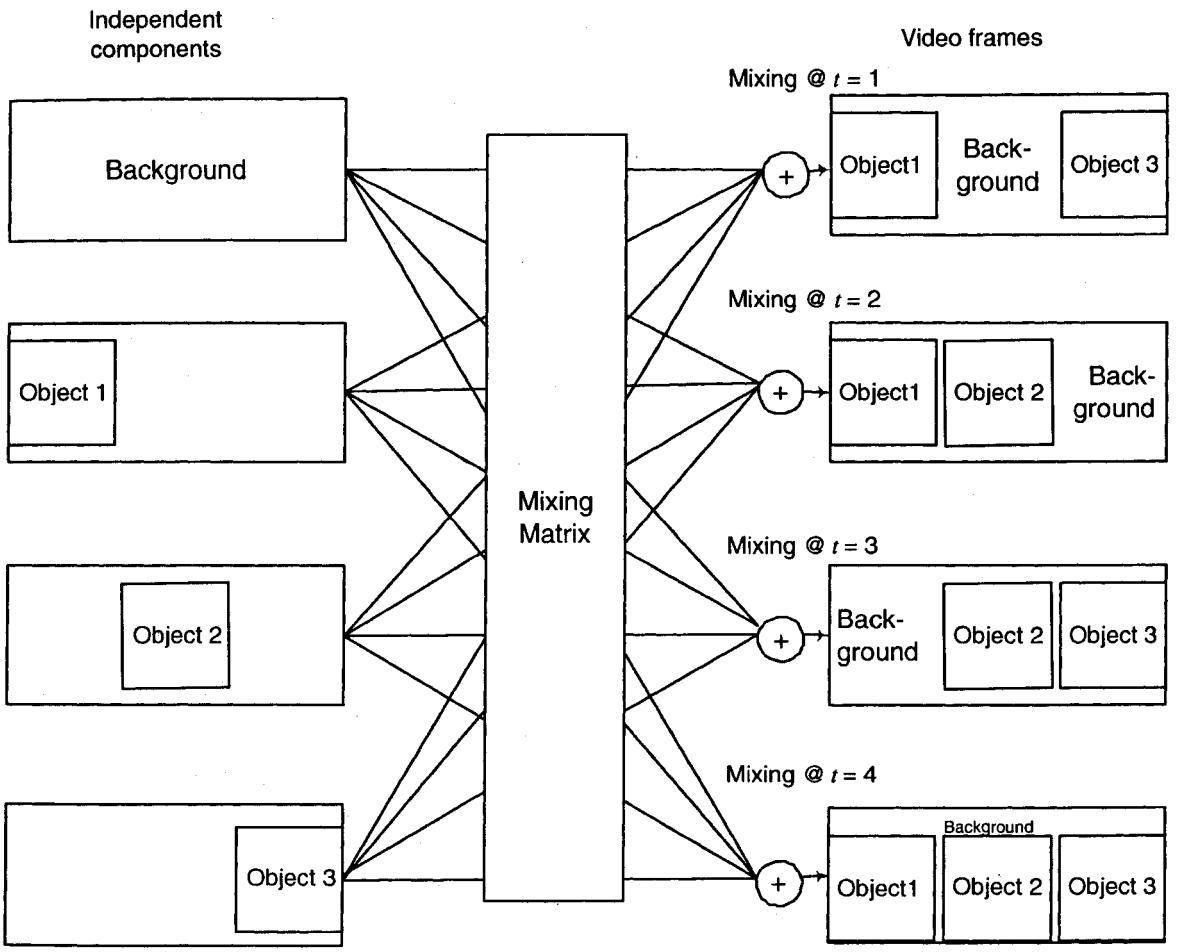

Figure 3.1: Illustration of video frame construction by mixing objects.

Fig. 3.1 demonstrates how the stICA model is applied to video frames. At a certain time moment, a video frame consists of a linear combination of all the objects, including the background.

For example at t=1, video frame 1 is obtained by the linear combination of the spatial

ICs on the left-hand side of Fig. 3.1. Frame 1 carries the information of the background, object 1 and object 3. In this way, different video frames are constructed.

Please note that the video frames cannot be the linear combination of the ICs that we want because some of the background is blocked by the moving objects in the video frames. This condition violates the stICA assumption. Thus we need to compensate for the background information that is lost due to object blocking. In this way, the assumption of linear combination may hold so that the stICA requirements are met. Here we denote the ideally blocked background information by $\hat{\Delta}_i$ in ith frame $\mathbf{f}_i$, such that

$$\hat{\mathbf{f}}_i = \mathbf{f}_i + \hat{\Delta}_i, \tag{3.6}$$

where the dimension of $\hat{\Delta}_i$ is also $M \times 1$ and i=1, $\cdots$, N.

Between the practical video frame model in Eq. (3.6) and the fitting model in Eq. (3.4), there is a gap $\hat{\Delta}_i$ that affects the accuracy of the stICA approach on video sequences. This problem is dealt with by our proposed methods in the following chapters.

3.2 The stICA Based Video Segmentation Approach

In this section, we will introduce an stICA based iterative approach, which can segment semantic video objects without any human intervention. To deal with the nonlinear combination problem (shown in Eq. (3.6)), we set up a two-iteration scheme (Fig. 1.1). In the first iteration (block diagram in Fig. 3.2), the stICA model is applied to the captured video frames. The maximum likelihood estimation is employed on both spatial and temporal signals. The Bell-Sejnowski algorithm is implemented to find the unmixing matrices, where the ICs $\mathbf{o}_i$ and $\mathbf{a}_i$ are substituted for both $\mathbf{s}_i$ and $\mathbf{t}_i$ in fMRI (Eqs. (2.42)-(2.46)). However, since the video frames cannot be the linear combination of the objects and their background, the recovered spatial signals $\mathbf{o}_i$ are still coarse representation of the objects (Fig. 3.3). Among all the recovered spatial signals, only the background image is clear. We can subtract it from all original video frames to get the preliminarily processed images which only contain objects (Fig. 3.5). Post-processing techniques are

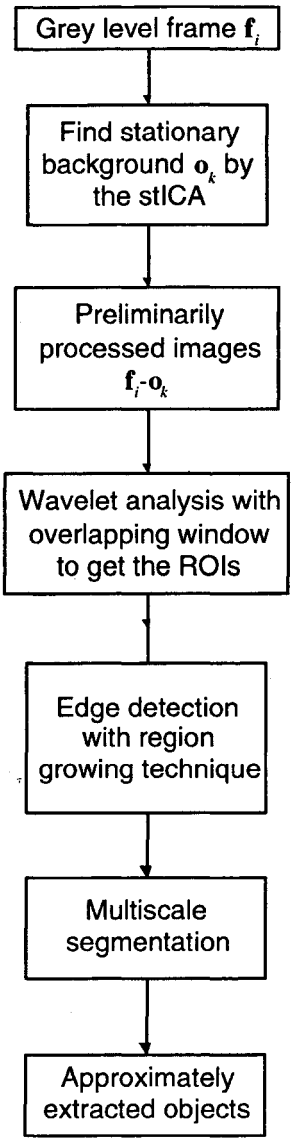

Figure 3.2: Block diagram of the first iteration.

then required to refine the object segmentation, which will be introduced in the next chapter.

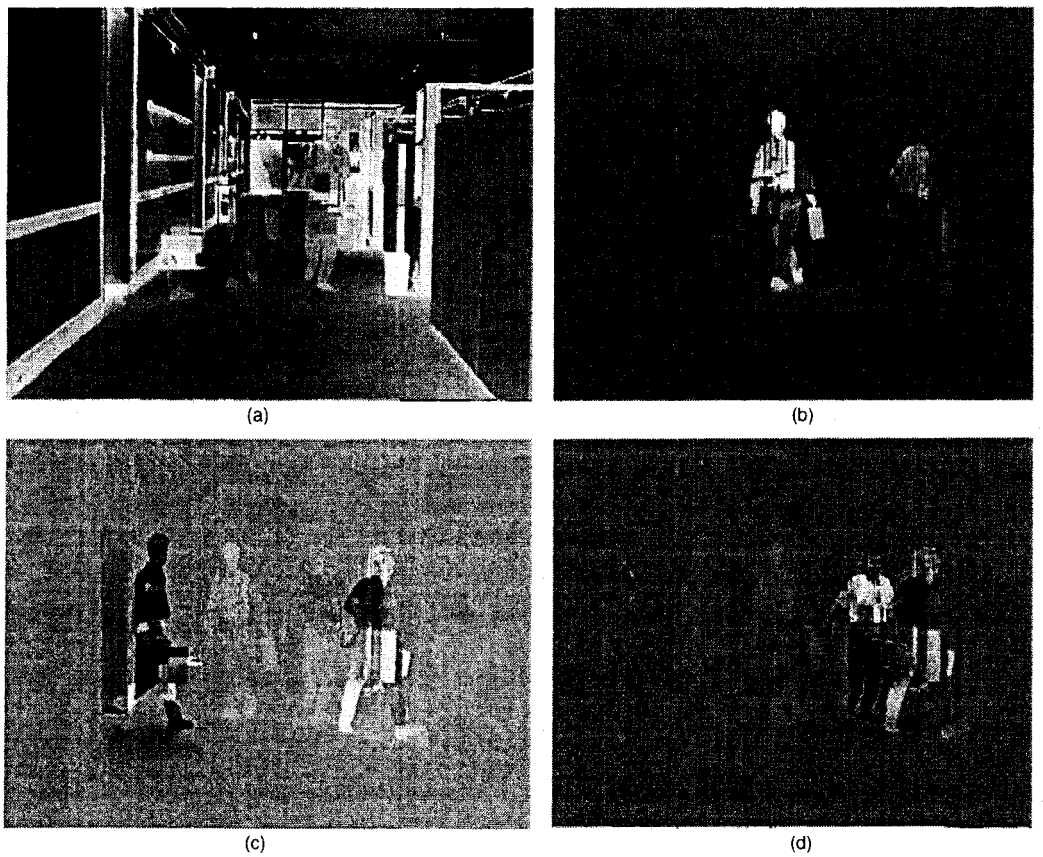

Figure 3.3: Spatial source signals from the first stICA processing.

3.3 Simulation of the stICA applied to Video Processing in the First Iteration

In our experiments, if without further notice, the proposed system is applied to the video sequence "Hall Monitor" with 9.28-second duration. There are altogether 280 frames, each of which has 240×360 pixels and 256 grayscale levels, i.e. there are 280 images generated. We suppose that every video frame contains at least one object of interest. This means there are no pure "background" images.

A set of frame is selected from these 280 frames for further processing. To avoid interference between close objects, frames are selected from the sequence at a constant

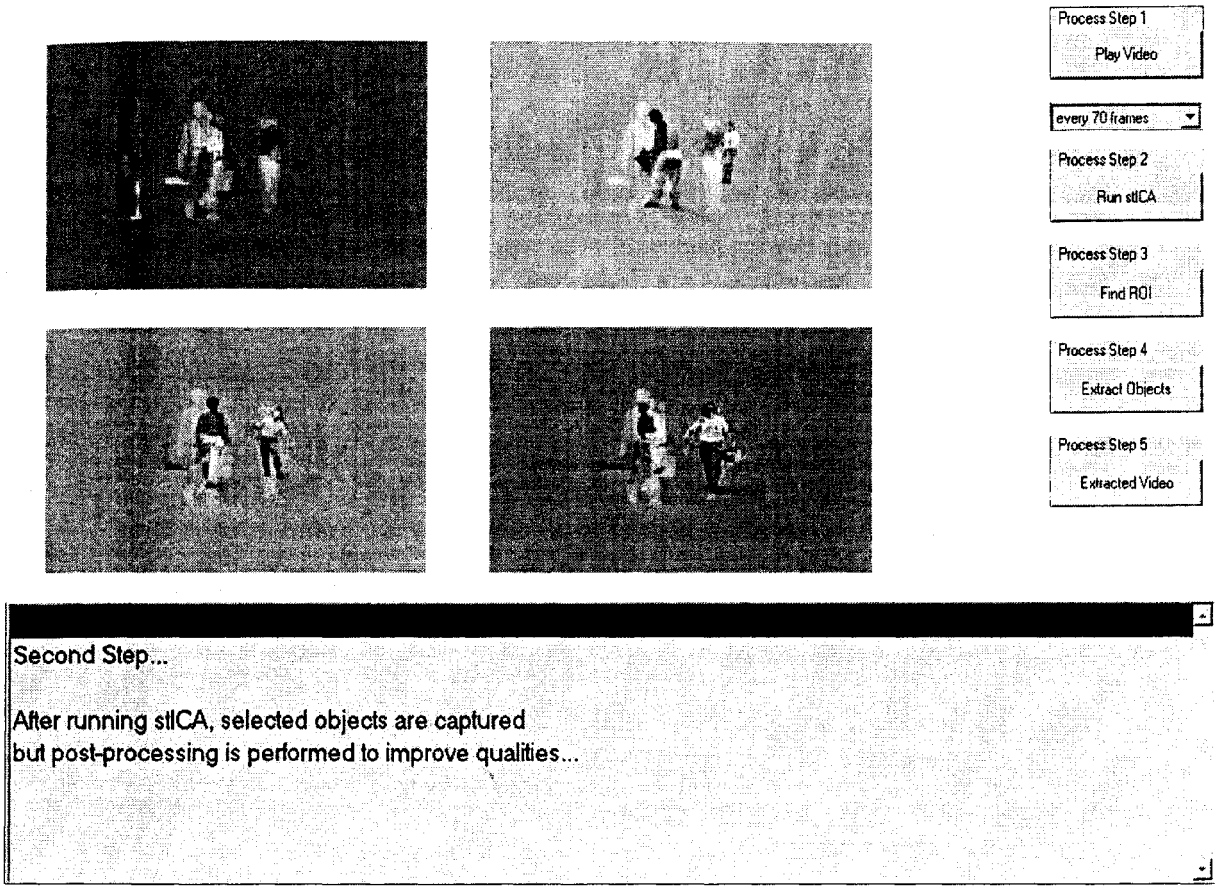

Figure 3.4: A GUI for the stICA based object extraction in video sequences.

interval. We set up a GUI (graphic user interface) that can show the processing details step by step (Fig. 3.4). The program allows one to define a frame selection interval. Based on the frame selection rate, a number of frames is selected from the 280 frames and the stICA model is applied to them. Through the stICA processing, we obtain the same number of spatial output images as input frames.

Among the output images in Fig. 3.3, only the background image is clear. Meanwhile, there are a number of undesirable output images (Fig. 3.3(b)-(d)). The reason is that the pixels representing objects in the video frames are not the linear combination of the pixels representing objects and the background in recovered image signals. In other words, these video frames are not a linear mixture of all the independent sources, namely the objects and background. Since the background image is clear among all the outputs, we can

subtract it from all original video frames to get the preliminarily processed images which only contain objects as shown in Fig. 3.5.

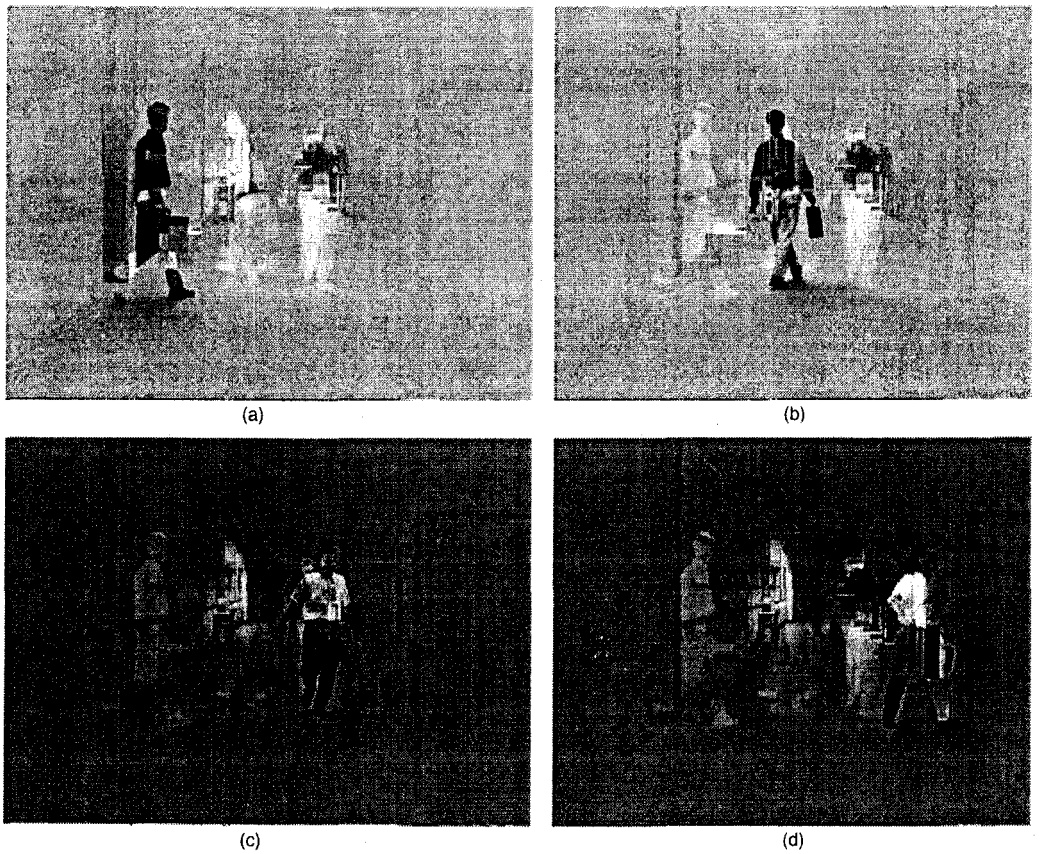

Figure 3.5: Preliminarily processed images from the first stICA processing subtraction.

In these images, we can see extensive noise. Post-processing techniques, which will be presented in the next chapter, are thus required to refine the object segmentation.

3.4 Summary

In this chapter, an stICA model is formulated for video sequences. A two-iteration approach is proposed to segment moving objects from a video sequence. In the next chapter, the post-processing techniques will be presented. Based on the first iteration results, the nonlinear combination problem will be dealt with in the second iteration.

Chapter 4

Post-processing in the First Iteration

THE stICA approach alone cannot provide a satisfactory object segmentation result. Some post-processing techniques are required to fine tune the output images. In this chapter, we introduce post-processing techniques based on wavelet analysis, edge detection, region growing and multiscale image segmentation. These methods are applied sequentially to segment the object of interest.

4.1 Using Wavelet Analysis to Locate Regions of Interest

As an ideal tool of image analysis, the wavelet transform performs well in characterizing singularities [19] [20]. In other words, large coefficients represent edge transitions in the wavelet domain.

The wavelet transform decomposes an image into three wavelet subspaces (LH, HL and HH) and one scaling subspace (LL). A single level of 2D wavelet decomposition is shown in Fig. 4.1. The three wavelet subspaces capture image details along the vertical, horizontal and diagonal directions, respectively.

We use the HL subspace to detect the horizontal boundaries of the image objects and the LH subspace to detect the vertical boundaries. First let us focus on the HL subspace to make an illustration. As we know, image object boundaries are represented by large coefficients in the wavelet domain. Thus in the HL subspace, image horizontal edges are

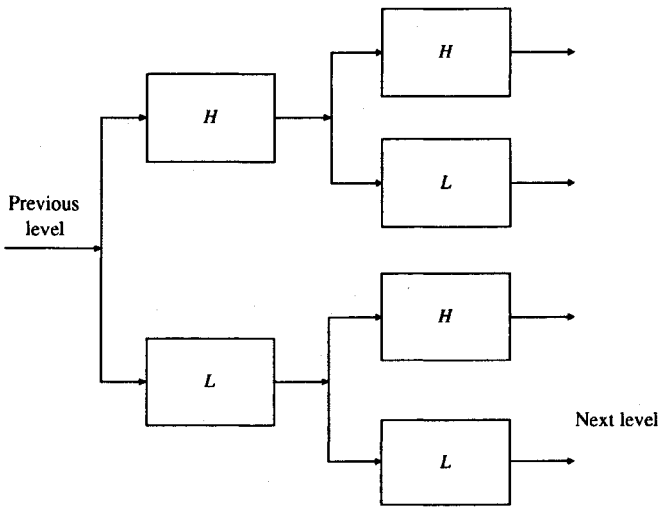

Figure 4.1: Illustration of one stage of 2D wavelet decomposition.

represented by large coefficients. For other image areas where there are no horizontal edges, basically there are no large coefficients in such areas in the HL subspace. Thus we can apply a slide window in the HL subspace and let it slide from one side of the image to the other side horizontally. While it is sliding, we observe the coefficients along each column and use the coefficient with the largest absolute value to represent such column. According to the wavelet features we explained before, if this coefficient has a very small value, this indicates that there are no image horizontal edges along this column. If the value is large, it means there are some horizontal edges. Through this method we can define the horizontal scope of the image objects in the HL subspace, which can bring us the horizontal region of interest (ROI) ($\mathrm{ROI_{HL}}^{\mathrm{horizontal}}$) in the wavelet domain.

For any spatial signal after the stICA processing, we define $\mathbf{W}$ as the HL subspace at the Nth level of the wavelet decomposition and w_{ij} is the coefficient in that subspace, where i, j are the indices of rows and columns of $\mathbf{W}$, respectively. We also use a vector $\Psi=\{\psi_1, \cdots, \psi_q\}$ to represent the ensemble of those largest coefficient values in the HL subspace, where $\psi_j=\max_i|w_{ij}|$ is the largest absolute coefficient value of column j in the HL subspace. Here q is the total number of columns in the subspace, which is decided by the level of the wavelet decomposition. For example, if the dimension of an image is $r\times r$,

then

$$q = \left(\frac{1}{2}\right)^N \times r. \tag{4.1}$$

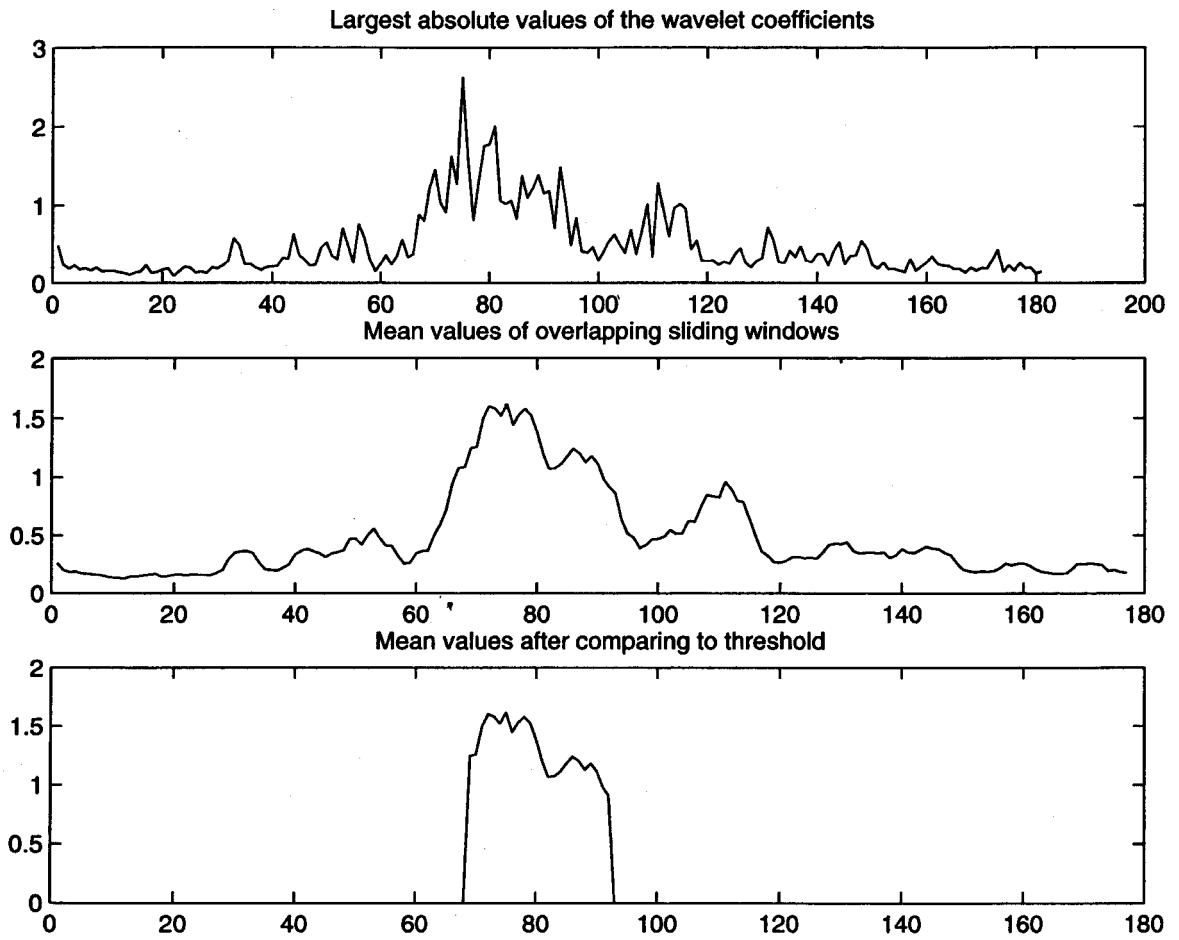

Figure 4.2: (From top to bottom) **a.** Maxima of absolute wavelet coefficients; **b.** Mean values of the maxima in the overlapping windows; **c.** Mean values after thresholding.

The method stated above to detect a horizontal image edge is based on the detection of large coefficients that represent image edges. The method requires a successive set of large coefficients to detect a single horizontal edge. However, if there are some small coefficients (below threshold) existing among these large ones, we may detect two or more edges where the object only has one. For example, in Fig. 4.2(a), there exist some valleys

between peaks, which without further processing would give erroneous results.

To avoid wrong detection of multiple edges, we apply an overlapping sliding window in the HL subspace. While it is sliding from the left to the right, we calculate the mean value of the largest absolute values within the window and use the mean value to decide whether there is an edge or not. Using this method reduces the likelihood of erroneous results. The overlapping sliding window has two important parameters to control the sliding properties. One is the width of the overlapping sliding window frame and the other is the sliding step. We define the width parameter as l and the step parameter as 1. At the last several steps of window sliding, the number of ψ_i is less than l, so there is a total of q-l+1 steps for the window to slide. We calculate the mean value of ψ_i within the window at each sliding step as follows:

$$m_k = \frac{\sum_{i=k}^{k+l-1} \psi_i}{l}, \; k = 1, \cdots, q - l + 1. \tag{4.2}$$

The processing result is shown in Fig. 4.2(b). Now the object edges are represented by some large mean values and the image background is represented by some small mean values. To distinguish these two classes, we need to define some thresholds.

A threshold detector is set up by comparing the mean values and the global absolute maximum value of the HL subspace wavelet coefficients:

$$m_k \geq \max\{\psi_1, \cdots, \psi_q\} \times \alpha = \max_i\{\max_j \mid w_{ij} \mid\} \times \alpha, \tag{4.3}$$

where α is an empirical constant, k = 1, $\cdots$,q+l-1, and i,j = 1, $\cdots$,r. We compare the mean values to the threshold. Once the first mean value that is greater than or equal to the threshold is observed, the corresponding position in HL subspace is recorded as a, the beginning of the edge. We continue to compare values until we observe a mean value that is less than the threshold. At this point, the position in HL subspace is recorded as b, the end of the edge. In HL subspace, the wavelet coefficient ensemble can stand for the region containing the object horizontal edges if these coefficients meet the criteria such that (Fig. 4.2(c)):

$$\mathrm{ROI_{HL}}^{\mathrm{horizontal}} = \{i \mid a \leq i \leq b\}, \tag{4.4}$$

where i is the column index. We continue comparing values and this manner all regions containing object edge can be detected.

Applying the same method in the LH subspace, the vertical ROI can be defined:

$$\mathrm{ROI_{LH}}^{\mathrm{vertical}} = \{j \mid c \leq j \leq d\}, \tag{4.5}$$

where j is the row index; c, d are the starting and ending points of vertical edges, respectively. Thus the rectangular ROIs that contain the objects in the wavelet domain are obtained:

$$\mathrm{ROI_{wavelet}} = \{i, j \mid i \in \mathrm{ROI_{HL}}^{\mathrm{horizontal}}, j \in \mathrm{ROI_{LH}}^{\mathrm{vertical}}\}. \tag{4.6}$$

The corresponding ROI in the stICA processed images can be located by using the inverse calculation in Eq. (4.1).

The purpose of segmenting a ROI is to decrease computational complexity for later post-processing and to reduce noise so that edge detection techniques and region-based segmentation approaches can achieve better results. Moreover, the object indexing approach that will be introduced in chapter 5 also needs the ROI to detect true objects. The ROI technique can also be applied to the original video frames as the video object tracking method.

In the following two sections, two post-processing approaches are applied sequentially: one is edge detection with region growing and the other is multiscale image segmentation.

4.2 Image Edge Detection with Region Growing

The ROIs detected by the presented object detection method based on the stICA represent areas of the objects of interest. However, the ROIs do not contain exact boundary information of the detected objects. The Canny edge detection technique is applied to these rectangular ROIs. This operation renders a binary image, in which 1s stand for the object (foreground) and 0s for the background.

From the binary images, we have obtained prospective object regions from edge detection. However, not all the obtained regions are objects of interest. In the ROIs, the target

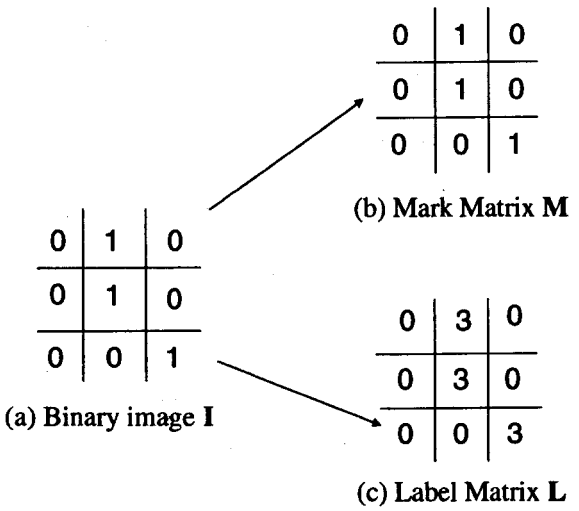

Figure 4.3: Region growing technique to label connected pixels. a. Binary edge pixel neighbourhood; b. Mark pixel neighbourhood; c. Label pixel neighbourhood.

objects are generally larger than the other isolated regions. Thus we can discriminate the target objects from those unwanted regions through the comparison of their sizes. Here a region growing technique is introduced to calculate the connected region size. To perform this region growing operation, we fill the interior regions inside the closed edge with the value 1. These closed-edge detections are performed by the Canny technique.

Marshall *et al.* [21] introduced a region growing approach that has the following operating procedures:

1. An initial set of small areas are iteratively merged according to similar constraints.
2. Start by choosing an arbitrary seed pixel and compare it with neighbouring pixels.
3. The region is grown from the seed pixel by adding in neighbouring pixels that are similar, increasing the size of region.
4. This whole process continues until all pixels belonging to a region are processed.

In a digital image, if two pixels have similar grayscale values and they are in their neighbours of eight, they are deemed in the same region. In our case, the processed

images are binary images. If two adjacent pixel values are equal to 1, they are considered to be in the same region. We assume in a binary image **I** with $r \times r$ size, its pixels $i_{i,j}$ and $i_{p,q}$ are in the same region if

$$i_{i,j} = i_{p,q} = 1, \text{ where } i, j, p, q \leq r, \mid p - i \mid \leq 1, \text{ and } \mid q - j \mid \leq 1. \tag{4.7}$$

We define two matrices with the same dimensions as **I**. All pixel values in these two matrices are initialized to zero. One matrix is named "Mark Matrix" and the other is named "Label Matrix". The flag with value 1 is assigned to a certain pixel in the Mark Matrix **M** to indicate that this pixel has been processed to avoid repeated processing. The Label Matrix **L** is used to assign a unique labelling integer to each isolated region. Thus the isolated regions can be distinguished by the different labelling integers. The total number of each labelling integer indicates the region size. For example, in Fig. 4.3(a), a seed pixel $i_{i,j}$ is randomly chosen, the values of its eight neighbours are checked in a clock-wise order. In this case, pixels $i_{i-1,j}$ and $i_{i+1,j+1}$ are equal to 1, therefore they are in the same region as $i_{i,j}$. For each pixel that is in the same region as $i_{i,j}$, its corresponding element in the Mark Matrix **M** is set to 1 to indicate that it has been processed such that (Fig. 4.3(b))

$$M_{i,j} = M_{i-1,j} = M_{i+1,j+1} = 1. \tag{4.8}$$

Meanwhile, a labelling integer, say 3, is assigned to corresponding elements in the Label Matrix **L** to represent that region (Fig. 4.3(c))

$$l_{i,j} = l_{i-1,j} = l_{i+1,j+1} = 3. \tag{4.9}$$

After the labelling of **L** is completed, the sizes of all isolated regions can be easily calculated. In Fig. 4.3(c) the total number of labelling integers 3 in matrix **L** represents the size of this region.

Let us explore the proposed region growing approach depicted in Fig. 4.4. First of all, in the binary image **I**, a seed pixel $i_{i,j}$ is selected, which must satisfy two criteria:

1. Pixel value must be 1: $i_{i,j}$=1;

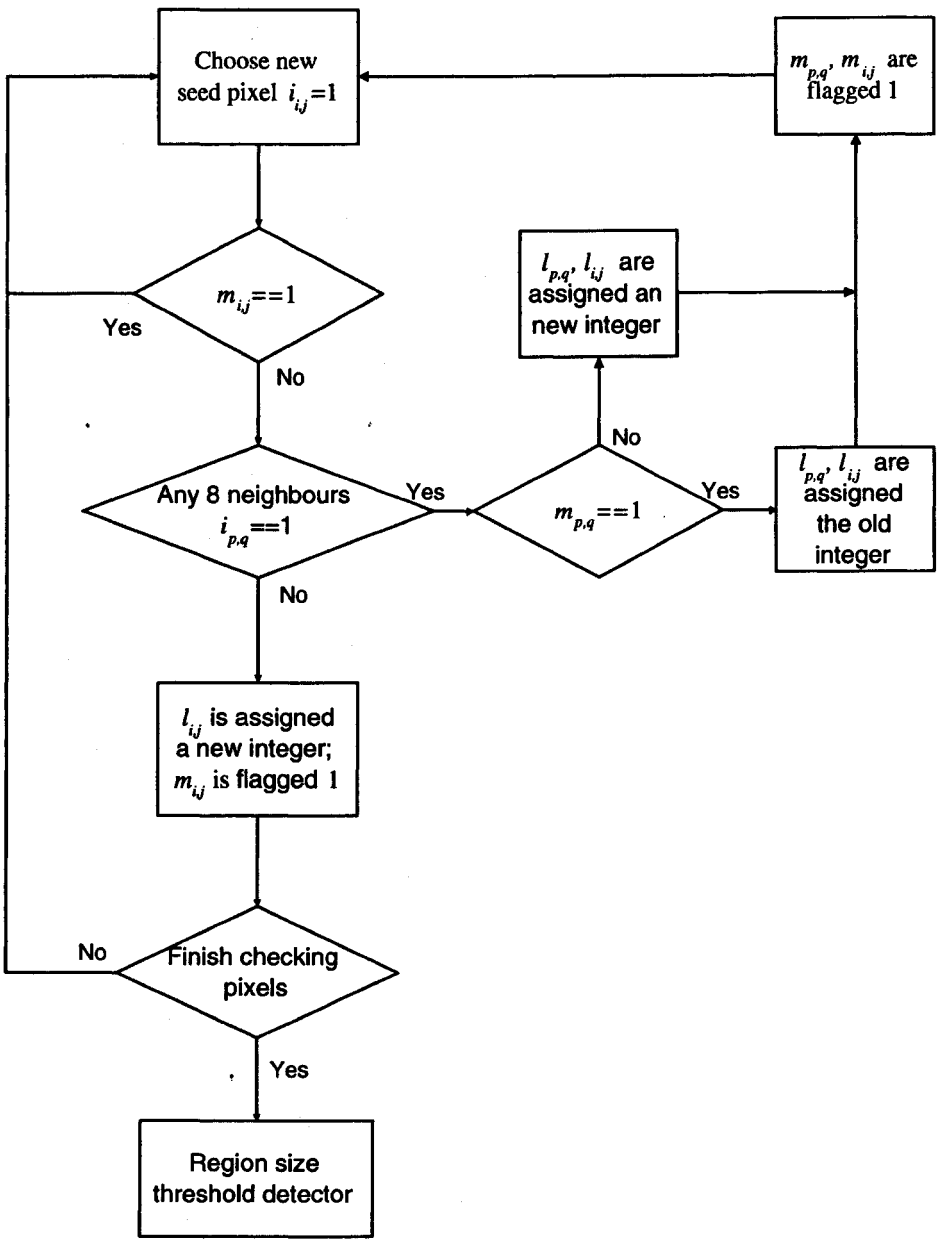

Figure 4.4: Block diagram of region growing technique.

2. The Mark Matrix element value cannot be 1: $M_{i,j} \neq 1$. Otherwise $i_{i,j}$ has been processed.

Once a new seed pixel $i_{i,j}$ is chosen, its eight neighbours $i_{p,q}$ ($| p - i | \leq 1$, $| q - j | \leq 1$)

are examined. There are two underlying possibilities.

1. If $i_{p,q}$=1 holds for any p,q ($\mid p-i \mid \leq 1$, $\mid q-j \mid \leq 1$), the value of the corresponding element in the Mark Matrix **M**, $M_{p,q}$ should be checked. There are two possibilities under this condition:

   (a) $M_{p,q}$=1: this indicates that the pixels corresponding to $M_{p,q}$ and $i_{p,q}$ have been processed. Thus $i_{i,j}$ belongs to the same region as $i_{p,q}$, and $l_{i,j}$ is assigned the same value as $l_{p,q}$.

   (b) $M_{p,q}$=0: this implies $i_{p,q}$ has not been processed. If all the neighbours of $i_{i,j}$ have not been processed, $l_{p,q}$ and $l_{i,j}$ are both assigned a new labelling integer.

2. If there is no value 1 pixel in the seed pixel $i_{i,j}$'s neighbourhood, i.e. $i_{p,q}$=0, this means $i_{i,j}$ is the only one pixel in its region. $M_{i,j}$ is flagged to 1 and $l_{i,j}$ is assigned a new labelling integer.

In this way, all $i_{i,j}$'s neighbours $i_{p,q}$ with value 1 are identified. Their Mark Matrix elements $M_{i,j}$, $M_{p,q}$ are marked flag 1 after they have been processed. The corresponding Label Matrix elements $l_{i,j}$ and $l_{p,q}$ are assigned the same labelling integer.

This recursive computing method is employed on every unmarked seed pixel. After the seeking is finished, all isolated regions are assigned different labelling integers by the Label Matrix **L**. A region size threshold detector is used to distinguish the objects of interest from any smaller size regions, which are not the objects of interest and are subsequently removed.

This region growing algorithm is simple, easy to implement, and reliable. It employs two ancillary matrices, which are efficient in processing. The Mark Matrix eliminates unnecessary processing, and the Label Matrix makes calculation of the region size easy. Such implementation does not change any pixel in the binary image **I**.

4.3 Multiscale Image Segmentation

Edge detection techniques such as the Sobel method and Canny method work efficiently on sharp edges. However, the processed images after the stICA do not possess such sharp edges. This leads to some false edges that affect further processing. In the last section, we apply the Canny edge detection technique to the rectangular ROIs and then exploit the region growing method to remove small regions that are not objects of interest. This gives us the approximate regions of objects, which are called the object regions.

In Fig. 4.5, the objects of interest are obtained by edge detection with region growing to remove the small regions that are disconnected with the objects. However, this approach cannot remove the regions that are connected to the objects. For simplicity, the connected regions are given a new name: connected component. Because of the false edges generated by edge detection, the region growing method cannot accurately identify the edges. Thus a multiscale region-based still image segmentation method [22] [23] [24] is employed on the object regions in post-processing. Note that here the term "multiscale" means the scales of the grayscale variance in a region. A region in this method is defined by measuring grayscale similarity and each region is labelled with a unique integer. The result of region growing shown in Fig. 4.9(c) is combined with the original image in the ROI in Fig. 4.8(a) giving the object regions in Fig. 4.10(a). Multscale segmentation is then performed on the object regions giving the multiscale segmented regions shown in Fig. 4.10(c).

Apparently, segmentation of regions with similar grayscale generally does not segment the objects of interest in images. A grayscale region may contain multiple objects, or one object may be divided to several grayscale regions. If an image has complex structure, it is difficult to find correspondence between each closed homogeneous region and a specific object. Fig. 4.5 is an illustration of the proposed approach. In Fig. 4.5(c), an object and its connected component are divided to four regions (R_1, R_2, R_3 and R_4) according to their grayscale similarities. In this case, Regions R_1 R_2 and R_3 belong to the object of interest. However, we cannot segment R_1, R_2 and R_3 from R_4 if only using multiscale segmentation.

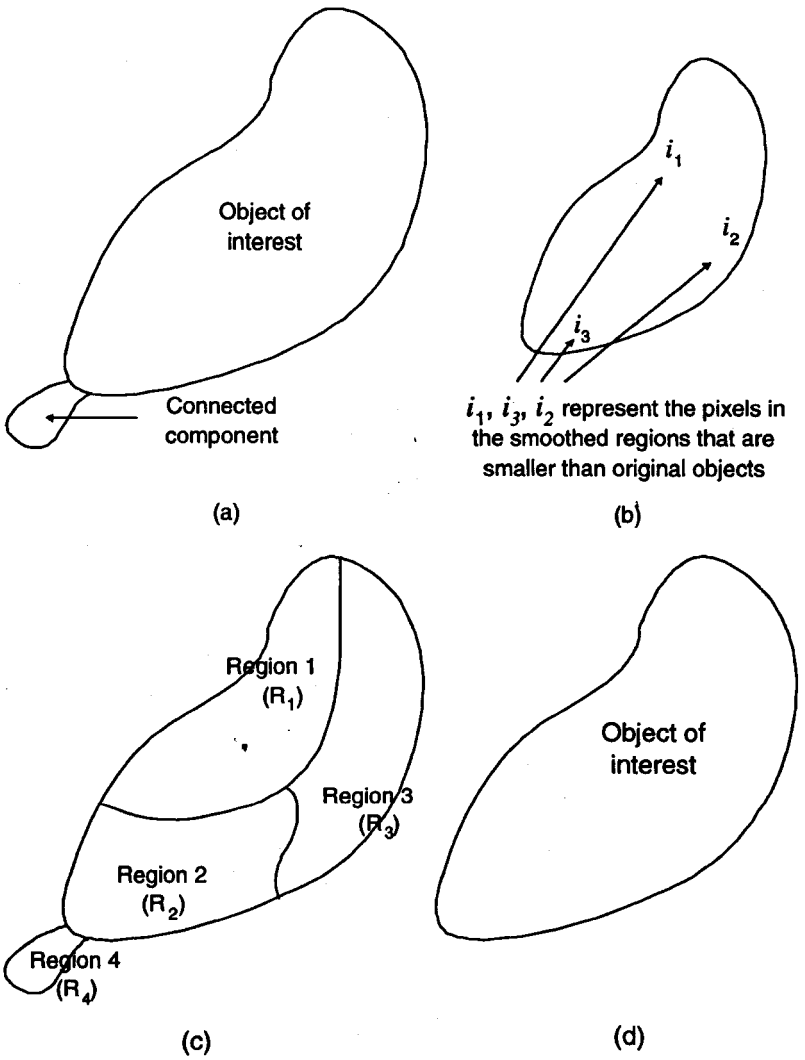

Figure 4.5: Illustration of the procedures of incorporating edge detection and multiscale segmentation. a. Regions obtained by edge detection and region growing; b. Smoothed regions obtained by Matlab Image Processing Toolbox; c. Regions obtained by multiscale segmentation; d. Objects obtained by the projecting operation between (**b**) and (**c**).

If we apply a smoothing and projecting approach on the multiscale segmentation results, the objects of interest can be identified. This indicates that we need to distinguish which regions in Fig. 4.5(c) should belong to the object of interest. The first step is to smooth all the connected regions (R_1, R_2, R_3 and R_4) in the object regions. The

smoothed results are shown in Fig. 4.5(b). The purpose of smoothing is to reduce the object region and make sure that there are no extra pixels outside of the objects of interest. The second step involves the removal of the connected component by the projecting operation. This involves mapping the pixels in the smoothed image (Fig. 4.5(b)) to the corresponding pixels in the segmented image (Fig. 4.5(c)). Then the regions labelled by the corresponding pixels in the segmented image are regarded as the desired parts of the object. The theoretical basis for the approach is that the connected components are relatively small and so that smoothing will effectively remove them. After smoothing, their pixels and relevant segmentation labelling information will be removed in the smoothed image plane. Thus the smoothed image only contains the pixels belonging to the object. For example, utilizing the location information of pixels i_1, i_2 and i_3 in Fig. 4.5(b) can correspondingly indicate that R_1, R_2 and R_3 in Fig. 4.5(c) belong to the object of interest. Fig. 4.5(d) shows the segmented object of interest that contains R_1, R_2 and R_3 only.

In this way, by utilizing wavelet analysis, edge detection, region growing and multiscale image segmentation approaches on the stICA outputs, objects with shape and boundaries can be approximately extracted.

4.4 Simulations of the Post-processing Techniques in the First Iteration

The first iteration is illustrated in Fig. 3.2. In the last chapter, the inputs for post-processing are the preliminarily processed images obtained by subtracting the recovered background from original video frames. The preliminarily processed images are processed by wavelet analysis to locate the rectangular ROIs. The ROIs can track the objects of interest, however they cannot describe the exact object boundaries. Thus edge detection and region growing approaches are necessary. They are used to outline the edges and remove the isolated small size regions. After the edge detection and the region growing, there may still be some connected components (e.g. R_4 in Fig. 4.5). Connected components and object are given a new name: object regions (e.g. R_1-R_4 in Fig. 4.5). To remove

the connected components from an object, the multiscale segmentation technique is applied to the object regions. Through the smoothing and projecting approaches, multiscale segmented regions belonging to the object can be identified.

4.4.1 Simulation of Wavelet Analysis to Locate ROIs

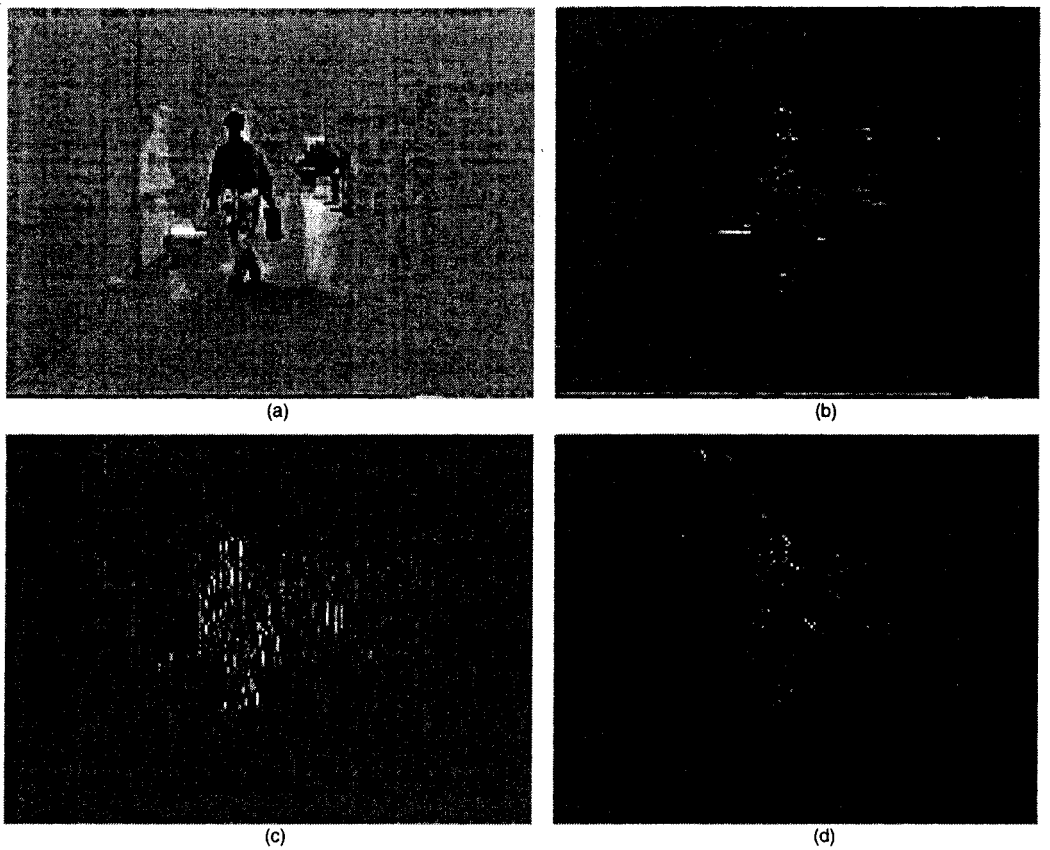

Figure 4.6: An example of 2D wavelet decomposition. a. LL scaling subspace; b. LH subspace; c. HL subspace; d. HH subspace.

After the subtraction of the recovered background, the preliminarily processed images contain object but with extensive noise (e.g. Fig. 3.5(a)-(d)). The discrete wavelet transform decomposes an image into four subspaces: three wavelet subspaces(LH,HL and HH) and one scaling subspace(LL). A scaling subspace (LL) example is shown in Fig. 4.6(a). It is a low frequency approximation of the original image. The other three subspaces LH, HL and HH are shown in Fig. 4.6(b)- 4.6(d). We can see that as we stated

above, LH, HL and HH describe image details along three directions: vertical, horizontal and diagonal directions, respectively. Thus LH and HL subspaces are used to locate the vertical and horizontal edges.

HL subspace wavelet coefficients are used to detect the horizontal edges. An overlapping sliding window approach is then applied. The subspace wavelet coefficients w_{ij} have q columns such that:

$$q = \left(\frac{1}{2}\right)^N \times r = \left(\frac{1}{2}\right)^1 \times 360 = 180. \tag{4.10}$$

Thus vector Ψ has 180 elements $[\psi_1, \cdots, \psi_{180}]$. Each element ψ_j is the largest absolute coefficient value of column j in matrix $\mathbf{W}$. The 60th frame in the video sequence is selected to demonstrate the proposed method. The graph of vector Ψ is shown in Fig. 4.2(a). The edge detection technique is based on a set of large coefficients to detect a single horizontal edge. However, some small coefficients (below threshold) may exist among these large ones. From the curve, decision-making of edge detection might not work because of some valleys between peaks.

To minimize this adverse effect, an overlapping sliding window method is employed. This method has two important parameters: a window width of 4 and a sliding step of 1 were found to provide good results in the experiments. Then the number of mean values is

$$q - l + 1 = 180 - 4 + 1 = 177. \tag{4.11}$$

This is also the number of steps for the window to slide from the left to the right of HL subspace. Fig. 4.2(b) demonstrates the application of Eq. (4.2). The rough curve in Fig. 4.2(a) is smoothed and is shown in Fig. 4.2(b). It clear that the result emphasizes the edges on the horizontal direction. Now the object edges are represented by some large mean values and the image background is represented by some small mean values. To distinguish the two classes, we need to define a threshold.

The threshold is determined by comparing each mean value with the global absolute maximum value of the HL subspace wavelet coefficients. The empirical constant α in Eq. (4.3) is set at 0.685. This constant proves to be efficient in all test images. The

mean values m_1 to m_{177} are compared with the threshold sequentially. Whenever a mean value is found to be greater than the threshold, the corresponding position in the HL subspace is recorded. In the selected frames, $a = 68$. Comparison continues until a mean value is found to be lower than the threshold, which indicates that the horizontal edges end and this position in the HL subspace is recorded as $b = 96$. The results are shown in Fig. 4.2(c). The mean values out of the range (68:96) are set to be zero. Then the corresponding 68th-96th columns in the HL subspace can construct the horizontal edges of the ROI:

$$\mathrm{ROI_{HL}}^{\mathrm{horizontal}} = \{i \mid 68 \leq i \leq 96\}. \tag{4.12}$$

The horizontal ROI in the original domain is illustrated in Fig. 4.7(a). Note that the horizontal ROI is located between the 136th column and the 192nd column. The locations are acquired by the inverse calculation in Eq. (4.1). Applying the same method in the LH subspace leads to the detection of the vertical edges and the vertical ROI:

$$\mathrm{ROI_{LH}}^{\mathrm{vertical}} = \{j \mid 24 \leq j \leq 89\}. \tag{4.13}$$

Combining the $\mathrm{ROI_{HL}}^{\mathrm{horizontal}}$ and $\mathrm{ROI_{LH}}^{\mathrm{vertical}}$ yields a rectangular ROI:

$$\mathrm{ROI_{wavelet}} = \{i, j \mid 68 \leq i \leq 96, 24 \leq j \leq 89\}. \tag{4.14}$$

The post-processing begins with the wavelet analysis along the vertical and horizontal directions, which can accurately locate the ROIs that contain object of interest. This procedure can significantly reduce the searching range of the object and thus reduce the computational complexity. For example, in Fig 4.7(a) and (b), we only need to consider an area containing rows 48 to 178, and columns 136 to 192. This rectangular area is 131×57, which is only

$$\frac{131 \times 57}{240 \times 360} = 8.64\% \tag{4.15}$$

of the original image area. Fig. 4.7(b) also shows that the locations of the ROIs are very accurate and the object of interest is completely included within the rectangular ROI. Fig. 4.7(c) shows a "zoom in" video frame from its original size 240×360 to 131×57.

This "zoom in" operation reduces the computation complexity and makes the detection immune to the interferences generated by the stICA (consider reference to Fig. 4.6(a)).

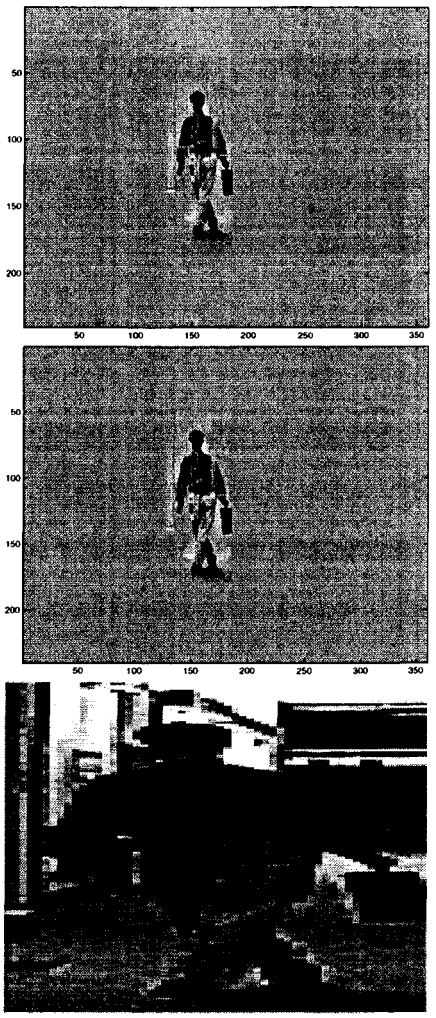

Figure 4.7: (From top to bottom) a. Horizontal ROI; b. A rectangular ROI after the horizontal and vertical wavelet analysis; c. The "zoom in" video frame 60.

4.4.2 Simulation of Edge Detection with Region Growing

The ROIs detected by the presented object detection method based on the stICA describe the areas of the object of interest, but the ROIs do not contain exact boundary information of the detected objects. The Canny edge detection technique is applied to these rectangular ROIs. This operation renders a binary image, in which 1s stand for the

object(foreground) and 0s for the background. Fig. 4.8(b) shows this operation. However, in this binary image of the ROI, not all the detected regions are object of interest. For example, in Fig. 4.8(b), besides the moving human object, there are some other regions included, such as the door. In the ROIs, the target objects are generally larger than the other isolated regions. Thus we can discriminate the target objects from those unwanted regions through the comparison of their sizes. For example, in Fig. 4.8(b), the size of the moving human is much larger than others'.

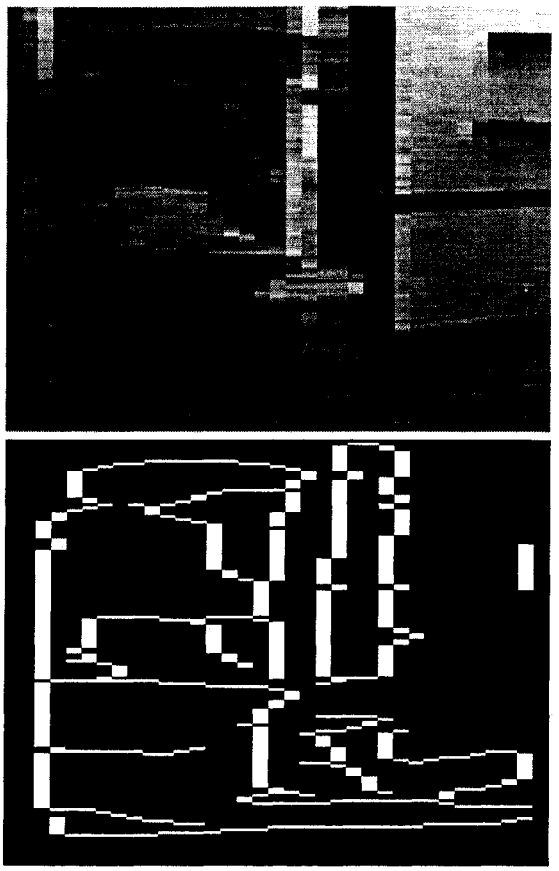

Figure 4.8: (From top to bottom) a. Original image in the ROI; b. Edge detection by the Canny detector.

The region growing approach can categorize these isolated regions. To apply this technique we fill the interior regions inside the closed edge with the value 1. These closed-edge detections are performed by the Canny technique. Fig. 4.9(a) shows these three isolated regions in white (1s), and the background in black (0s). Another two matrices

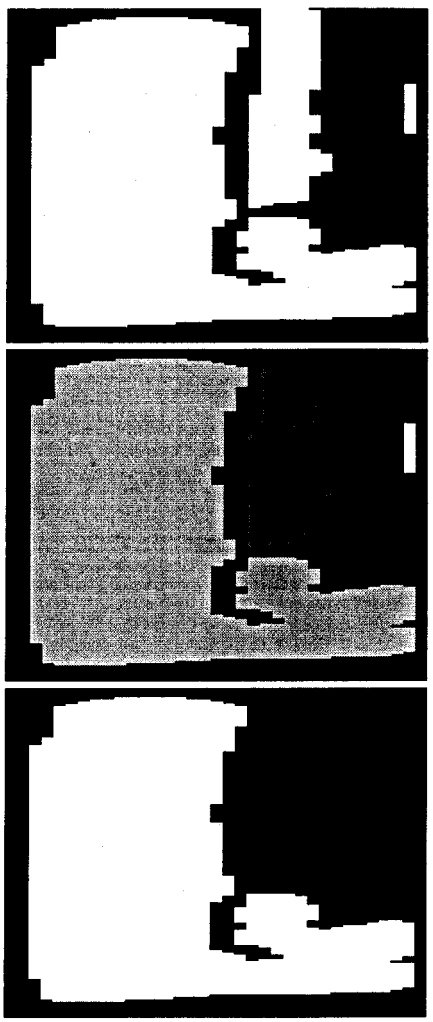

Figure 4.9: (From top to bottom) a. Filling regions after edge detection; b. Labelling regions with the same integer; c. Removing regions that are not of interest by threshold detection.

with the same dimensions are defined: the Mark Matrix **M** and the Label Matrix **L**. All pixel values in these two matrices are initialized to zero.

This region growing algorithm is a recursive computing method (Fig. 4.4). Its implementation is simple and reliable. In the binary image **I**, the first seed pixel $i_{i,j}$ that meets two criteria ($i_{i,j}$=1, $M_{i,j} \neq$1) is found by column-wise searching. Its eight connected neighbours $i_{p,q}$ ($\mid p-i \mid \leq 1$, $\mid q-j \mid \leq 1$) are then checked for both their pixel values and their Mark Matrix element $M_{p,q}$. After the checking is finished, the Mark Matrix is flagged and the same labelling integer is assigned to $l_{i,j}$, $l_{p,q}$. One of the neighbours $i_{p,q}$ with

value 1 is considered to be a new seed pixel on which the same operation is implemented. This operation continues until all pixels are processed. In this way, a region finishes its growing and its corresponding Label Matrix region is assigned a unique labelling integer. For example, Fig. 4.9(b) shows three connected regions that are assigned three labelling integers. The sizes of isolated regions are easily acquired by summing up the number of each labelling integer. We get the sums of labelling integers 0 through 3 as 2023, 457, 2643 and 22. Labelling integer 0 corresponds to the background; labelling integer 2 corresponds to the moving object; and labelling integers 1 and 3 correspond to the non-target regions.

Finally, the small regions corresponding to the labelling integers 1 and 3 are eliminated by a region size threshold detector (Fig. 4.9(c)). This threshold is set to 10% of the largest region size (except the background) in the whole binary image. In the test image, the threshold is set at $10\% \times 2643 \approx 264$. After threshold detection, only the approximate object of interest remains. Thus image quality improves.

4.4.3 Simulation of Multiscale Image Segmentation

In Fig. 4.9(c), the object regions are obtained by edge detection and region growing. However, this approach cannot remove the unwanted components that are connected to the objects. Such components are caused by the false edges resulting from edge detection.

The multiscale region-based still image segmentation method in [22] [23] [24] is employed on the object regions in post-processing. In the simulation, we apply this algorithm to the original frame in the area outlined by edge detection. Then we get the multiscale segmented regions in Fig. 4.10(c). Since the object of interest (the human) and its connected regions are segmented into several regions based on their grayscale similarities, extra information is required to distinguish which regions should be considered as parts of the object. This information comes from edge detection of the object regions.

However, edge detection can bring unnecessary connected components because of the false edges. Smoothing the edge detected regions can remove such unnecessary connected

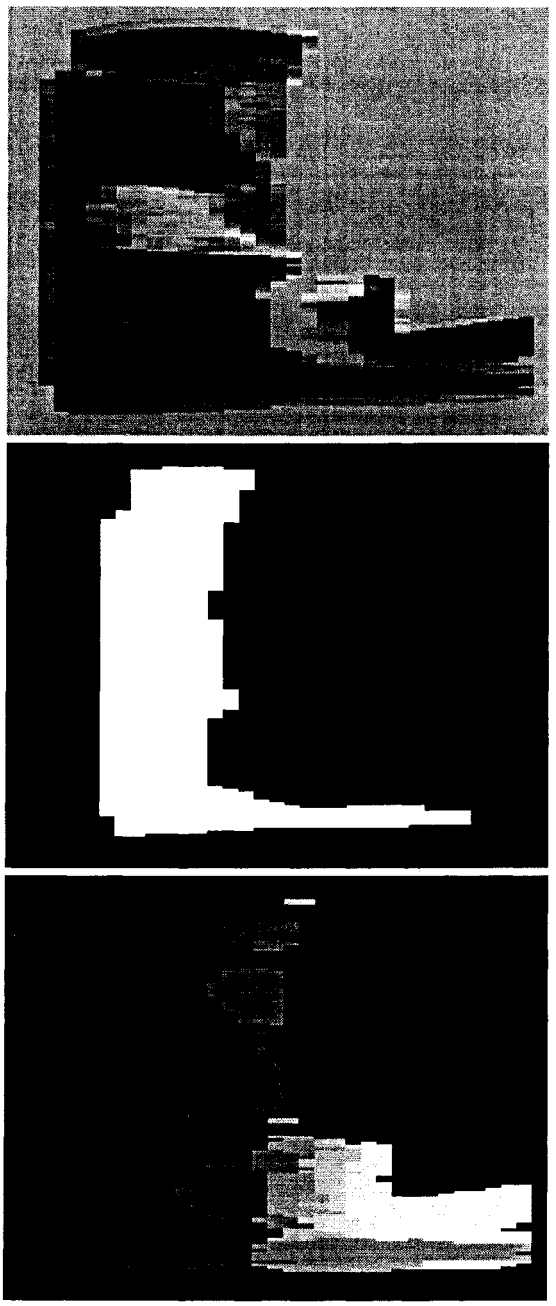

Figure 4.10: (From top to bottom) a. Object regions in the ROI; b. Smoothed regions from edge detection; c. Multiscale segmented regions.

components. After smoothing the regions in Fig. 4.9(c), a "slimmer" object is obtained and shown in Fig. 4.10(b). The major unnecessary connected components have been

removed. We project the pixels after the smoothing operation to the multiscale segmented image (Fig. 4.10(c)). The regions belonging to the object are identified. The reason for smoothing the binary regions is to make sure that no pixel is projected to the connected components. An example of the extracted object from the original image is illustrated in Fig. 4.11(b).

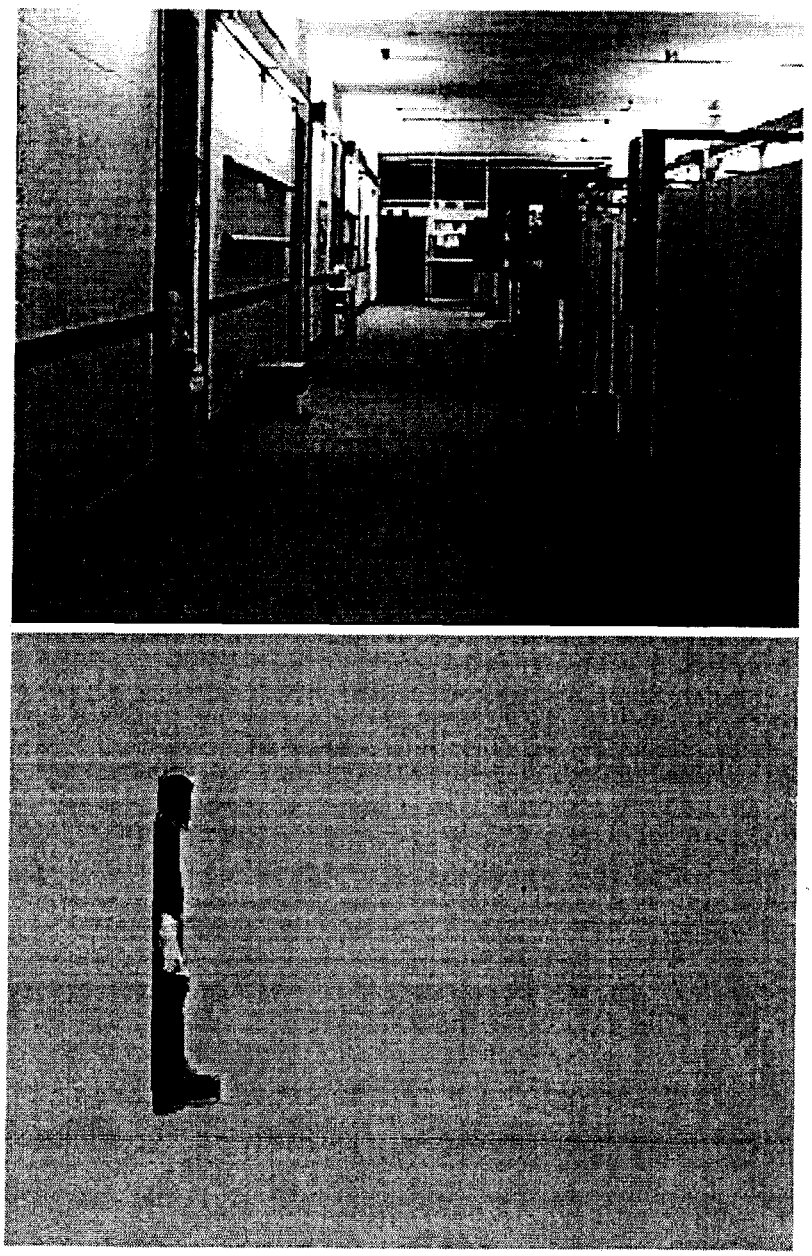

Figure 4.11: (From top to bottom) a. Original video frame 20; b. Extracted object.

4.5 Summary

In summary, the first iteration includes the following steps (as shown in Fig. 3.2):

1. Use the stICA to process the selected frames from a video sequence. The preliminarily processed images are obtained by subtracting the recovered background from original video frames.

2. The preliminarily processed images are processed by using the wavelet analysis followed by applying overlapping moving windows and a threshold detector to obtain the rectangular ROIs.

3. From the ROIs, edge detection of the extracted object is performed by using the Canny method. A recursive region growing technique is employed to remove the small size regions in the ROIs. The object regions are formed in this step.

4. Multiscale segmentation techniques are applied to the object regions with the smoothing/projecting approach to identify the regions belonging to the object.

In this way, by utilizing the wavelet analysis, edge detection, region growing and multiscale image segmentation approaches on the stICA outputs, the objects with specific shapes and boundaries can be approximately extracted.

Chapter 5

A Compensation Approach of stICA for Practical Video Sequences

THE post-processing procedures described in chapter 4 work effectively for the objects with relatively simple background. For the objects used in the tests, the background regions at the target object boundaries possess grayscale values that are sufficiently different to enable easy distinction. However, if both the background and the objects of interest have similar greyscale values, false regions may be identified as the objects of interest, as illustrated in Fig. 5.3(b) and (c). To deal with this problem and the nonlinear combination problem in the stICA model for video sequences, a "compensation" technique is applied to the stICA in the second iteration of our framework (Fig. 1.1). In the second iteration (Fig. 5.1), satisfactory object segmentation results are achieved by a compensation approach, a frame object indexing method and the post-processing techniques.

5.1 A Compensation Approach of stICA

The major obstacle of the stICA's application to video sequences is the nonlinear combination problem as shown in Eq. (3.6). The nonlinear property of video frames leads to the poor outputs from the stICA when it is applied directly to the video frames. Thus the complicated post-processing methods are required in the first iteration (Fig. 3.2). If we can determine the approximate region Δ_i that is blocked by the object in each frame

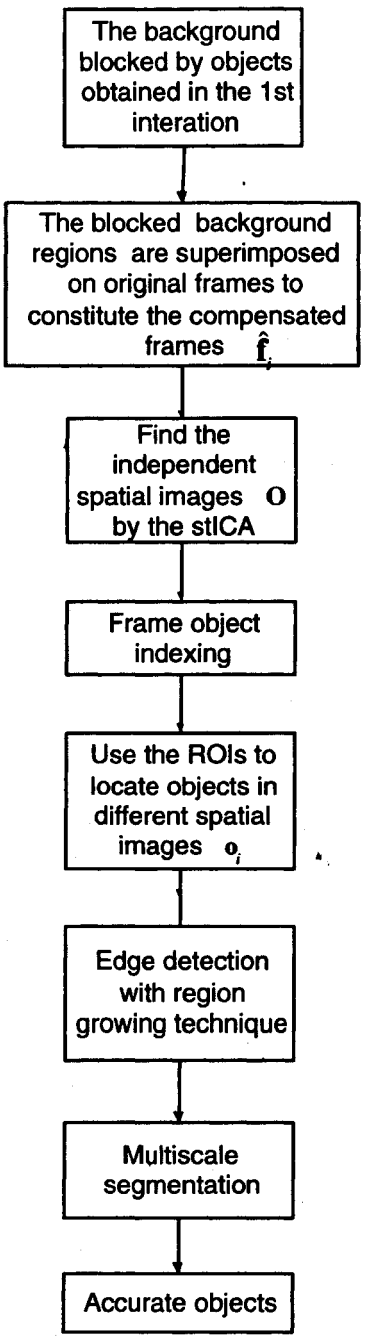

Figure 5.1: Block diagram of the second iteration.

$\mathbf{f}_i$ and "compensate" the blocked background back to each frame (Eq. (3.6)), then we can

obtain the ideal frames $\hat{\mathbf{f}}_i$ from:

$$\mathbf{f}_i + \Delta_i = \hat{\mathbf{f}}_i - \hat{\Delta}_i + \Delta_i = \hat{\mathbf{f}}_i + (-\hat{\Delta}_i + \Delta_i), \tag{5.1}$$

where Δ_i, $\hat{\Delta}_i$, $\mathbf{f}_i$ and $\hat{\mathbf{f}}_i$ are the $M \times 1$ column vectors as stated in chapter 3.

If Δ_i is ideally located, $-\hat{\Delta}_i + \Delta_i = \mathbf{0}$, which means the video frames can fit the stICA model. In fact, if we get the accurate blocked background information, we can outline the objects of interest and fulfil the video object segmentation task. However, we can only acquire the approximate blocked background information in the first iteration and use it for the stICA processing in the second iteration. The following steps are the procedures of the compensated frames for the stICA processing in the second iteration:

1. The blocked regions of the background are determined by the segmented objects in the first iteration. The blocked regions are used as binary masks (Fig. 5.4) and the masks are applied to the background image obtained in the first iteration to get the blocked background information Δ_i (Fig. 5.5).

2. Δ_i is superimposed onto its corresponding original video frame and the compensated frames are obtained (Fig. 5.6).

Comparing Eq. (3.5) and Eq. (5.1) we obtain

$$\mathbf{f}_i + \Delta_i = \hat{\mathbf{f}}_i + (-\hat{\Delta}_i + \Delta_i) \approx \hat{\mathbf{f}}_i, \tag{5.2}$$

where $-\hat{\Delta}_i + \Delta_i$ is the major factor that determines the accuracy of the stICA processing results.

We apply the stICA model (Eq. (2.46)) to the "compensated" frames (Fig. 5.6). The stICA estimation algorithm stated in chapter 3 is employed again. The accurate foregrounds (objects) and background are recovered from the "compensated" frames. Fig. 5.2(b)-(d) are the results of the stICA outputs. The edges of extracted objects are much clearer than those in the first application of the stICA (Fig. 3.3(b)-(d)). Their clear edges demonstrate an improvement in the object segmentation quality.

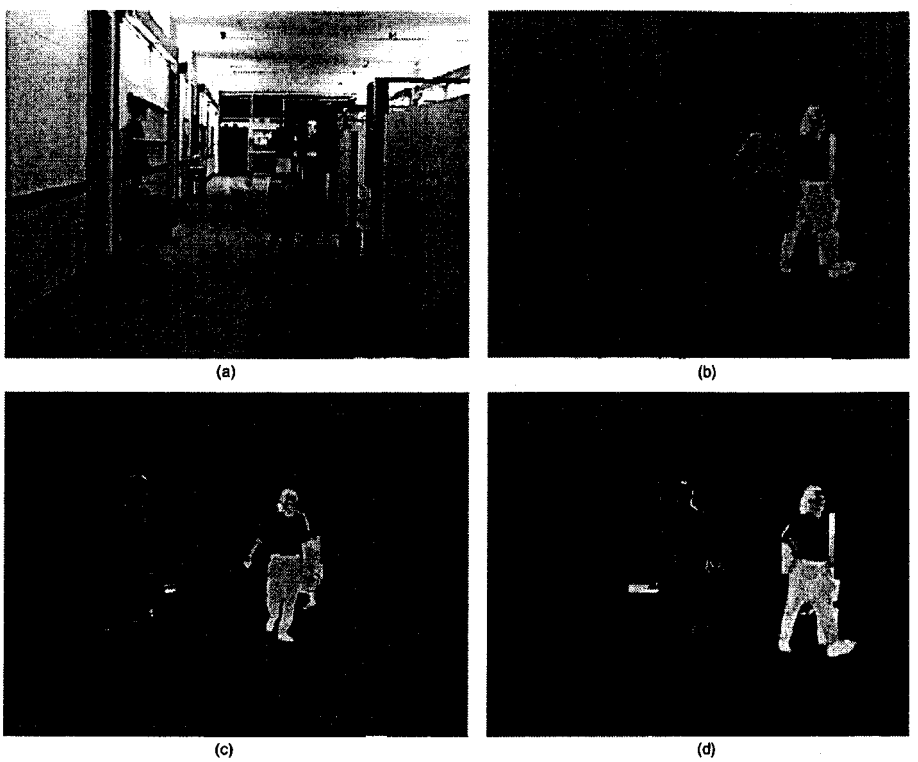

Figure 5.2: Spatial source signals from the second stICA processing: $\mathbf{o}_1$, $\mathbf{o}_2$, $\mathbf{o}_3$, $\mathbf{o}_4$.

5.2 Frame Object Indexing Approach

The stICA recovered video objects (spatial signals $\mathbf{O}$) are clear enough for edge detection. However, due to the ambiguities of the ICA [11], the order of the ICs cannot be determined. The order of the ICs is very important for reconstructing the video sequence containing only the objects. Thus, before edge detection, the recovered spatial objects $\mathbf{O}$ must be indexed according to the order of the video frame $\mathbf{F}$. We propose an indexing method based on the SVD [12] and the corresponding weighting matrices. Such a weighting matrix is a kind of linear combination.

We derive the frame object indexing from the compensated video sequence $\mathbf{F}$ and its SVD products $\mathbf{U}$ and $\mathbf{V}$ in Eq. (2.43) such that

$$\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T = (\mathbf{U}\Sigma^{1/2})(\mathbf{V}^T\Sigma^{1/2}) = \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T. \tag{5.3}$$

Since both $\mathbf{U}$ and $\mathbf{V}$ are orthogonal [8], we could make use of these two equations $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\widetilde{\mathbf{U}} = \mathbf{U}\Sigma^{1/2}$. We can get the following derivation results

$$\mathbf{FV} = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V} = \mathbf{U}\Sigma\mathbf{I} = \mathbf{U}\Sigma = \mathbf{U}\Sigma^{1/2}\Sigma^{1/2} = \widetilde{\mathbf{U}}\Sigma^{1/2}, \tag{5.4}$$

where Σ is a diagonal matrix with singular values. The multiplication of $\widetilde{\mathbf{U}}$ with $\Sigma^{1/2}$ can only change the amplitude of $\widetilde{\mathbf{U}}$ (eigenimages), but can not change the eigenimage indices. Let us suppose that $\mathbf{V}$ is a $k \times k$ weight matrix. Eigenimage $\mathbf{u}_i$ (i=1, $\cdots$,k) is most affected by the frame that has the largest absolute element in the corresponding column of $\mathbf{V}$.

Once we find the indexing relationship between $\mathbf{F}$ and eigenimages $\widetilde{\mathbf{U}}$, we can proceed to get the indexing relationship between $\widetilde{\mathbf{U}}$ and the independent spatial images $\mathbf{O}$. Referring to Eq. (2.44), we denote spatial ICs

$$\mathbf{O} = \widetilde{\mathbf{U}}\mathbf{W}_O, \tag{5.5}$$

where $\mathbf{W}_O$ is a $k \times k$ unmixing matrix. In the stICA model, $\mathbf{O}$ is generated by the multiplication of eigenimages $\widetilde{\mathbf{U}}$ and the unmixing matrix $\mathbf{W}_O$. The indexing relationship between $\widetilde{\mathbf{U}}$ and $\mathbf{O}$ can be found in the same manner as that used for $\mathbf{F}$ and $\widetilde{\mathbf{U}}$. Now the object index in $\mathbf{O}$ can be referred to the order of $\mathbf{F}$.

Here we still need to use the ROIs obtained from the first iteration to assure the quality of edge detection. Those post-processing techniques used in the first iteration, such as the edge detection with region growing and multiscale image segmentation, are applied to the indexed objects. In this way, the objects with accurate shape and boundaries are extracted.

5.3 Simulations

The methods we used in chapter 4 work effectively for the objects with a simple adjacent background, which means the greyscale of the background pixels are not similar to the target objects. Fig. 5.3(a) and (d) are in this category. However, if both the background

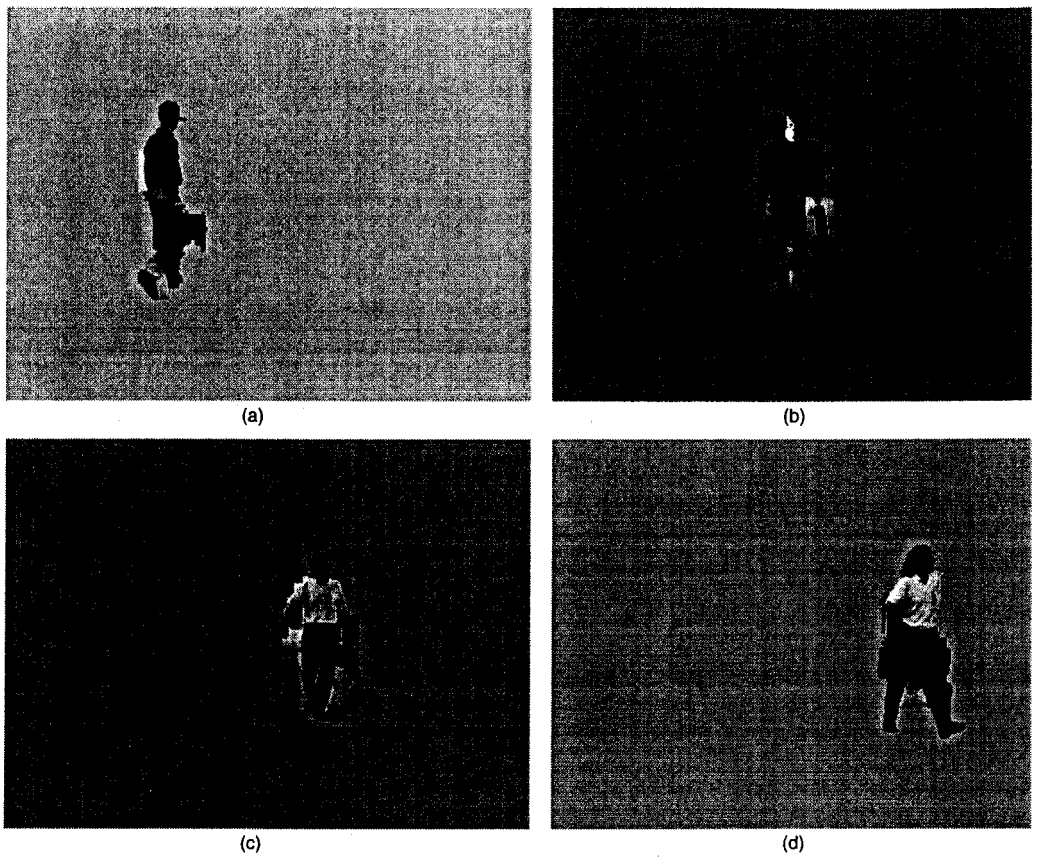

Figure 5.3: The output images from the first iteration.

and the object of interest have similar greyscale values, false regions may be identified as the objects of interest as shown in Fig. 5.3(b) and (c).

5.3.1 Simulation of Compensation Approach of stICA

Fig. 5.4(a)-(d) show the binary masks that are determined by the segmented objects from the first iteration. The blocked background regions are obtained by projecting the masks to the background we recovered in the first iteration so that the compensated video frames are the sum of original video frames and the corresponding blocked background regions. Fig. 5.6(a)-(d) are the examples of the compensated video frames.

Then the stICA model is applied to the compensated video frames (Fig. 5.6(a)-(d)). As expected, the second stICA processing detects the object edges accurately. Compared

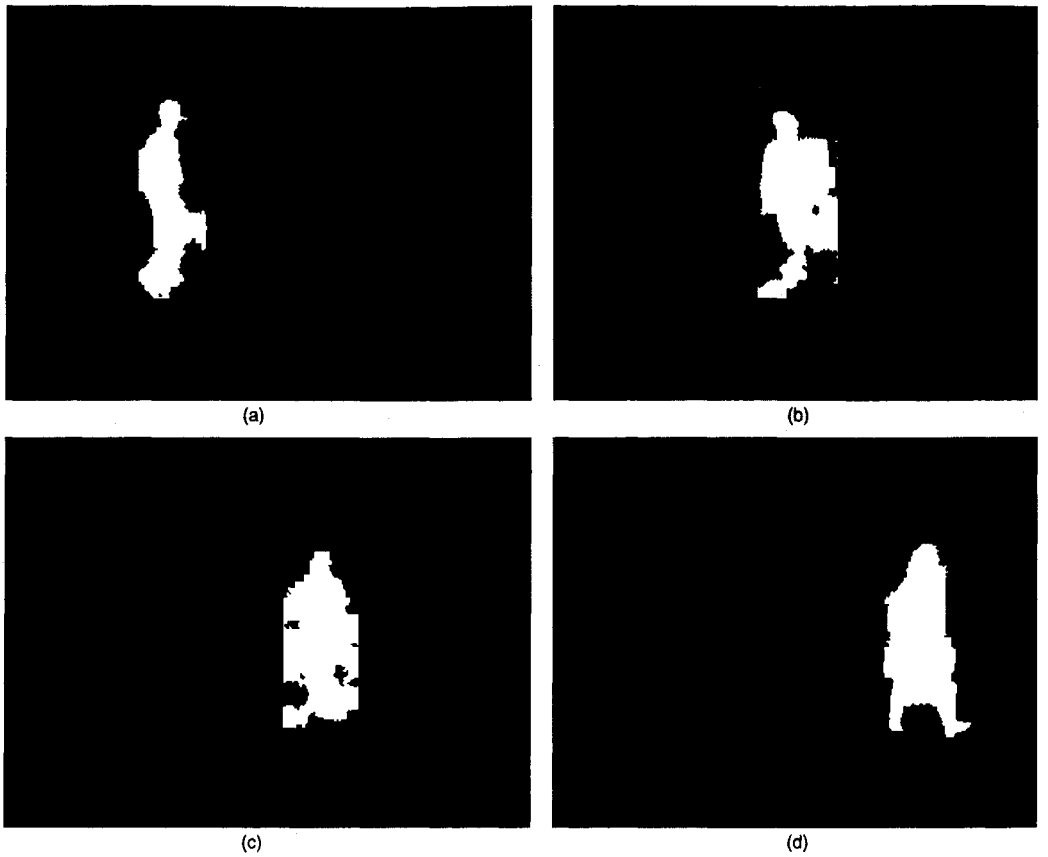

Figure 5.4: Binary masks determined by the first iteration.

with the results obtained in the first stICA processing (Fig. 3.3(b)-(d)), the edges of the recovered spatial ICs in the second stICA processing (Fig. 5.2(b)-(d)) are clearer and sharper. This represents an improvement the object segmentation quality.

5.3.2 Simulation of the Frame Object Indexing Approach

After the second stICA processing, the stICA recovered objects (spatial signals **O**) are clear and the edge detection results are more accurate than the first one. However, due to the ICA's ambiguities, the order of the ICs cannot be determined. The order of ICs is very important for reconstructing the video sequence containing only the objects.

In the simulation experiment, there are altogether four video frames defined as inputs to the stICA. Since the SVD is the pre-processing tool of the ICA, we first use the SVD to

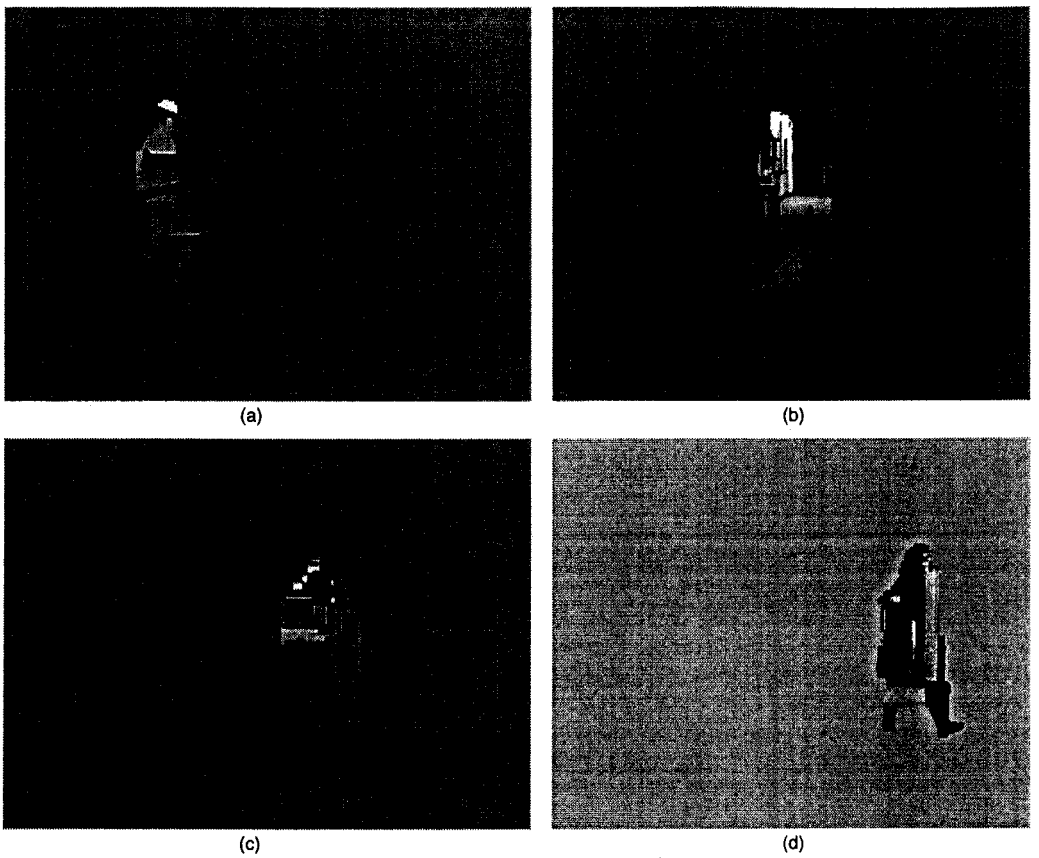

Figure 5.5: Blocked background regions determined by the binary mask.

find the indexing relationship between the video frames $\mathbf{F}$ and the eigenimages $\widetilde{\mathbf{U}}$. In the eigenimages matrix $\widetilde{\mathbf{U}}$, the first principle component $\mathbf{u}_1$ represents the strongest energy among all the principle components [12]. Among all the objects, the background has the strongest energy because it exists in every frame of the video sequence. Thus $\mathbf{u}_1$ should correspond to the background (a special object). Through the observation of the elements of the eigensequence matrix $\mathbf{V}$, the indices of other objects can be found.

In this case, each video frame contains only one object. So there are altogether four objects and one background to be indexed. Since we have a total of four eigenimages after the SVD, there should be more than one object to be indexed in a certain eigenimage.

We need to find the indexing information of the four objects and a background from these four eigenimages. The eigenimage $\mathbf{u}_1$ corresponds to the background. To determine

Figure 5.6: a.-d. Compensated video frames for the second stICA processing.

what the indices of the four objects are, the largest absolute coefficients in columns two to four of the eigensequence matrix $\mathbf{V}$ found.

$$\begin{bmatrix} 0.4899 & 0.4343 & \underline{-0.7464} & 0.0245 \\ 0.4820 & -0.1482 & 0.4756 & \underline{-0.7302} \\ 0.4948 & \underline{0.6290} & 0.4348 & 0.4129 \\ 0.5019 & 0.2218 & -0.1661 & \underline{-0.8002} \end{bmatrix}$$

As can be seen, the third coefficient of column two has the largest absolute value in that column, which means the object segmented from the second eigenimage $\mathbf{u}_2$ will be indexed as the third frame in the video sequence. This is true because the third frame corresponds to the third coefficient and the frame has the largest contribution to the formation of the second eigenimage and to the object in it. For the same reason, the object segmented from the third eigenimage $\mathbf{u}_3$ will be indexed as the first frame in the

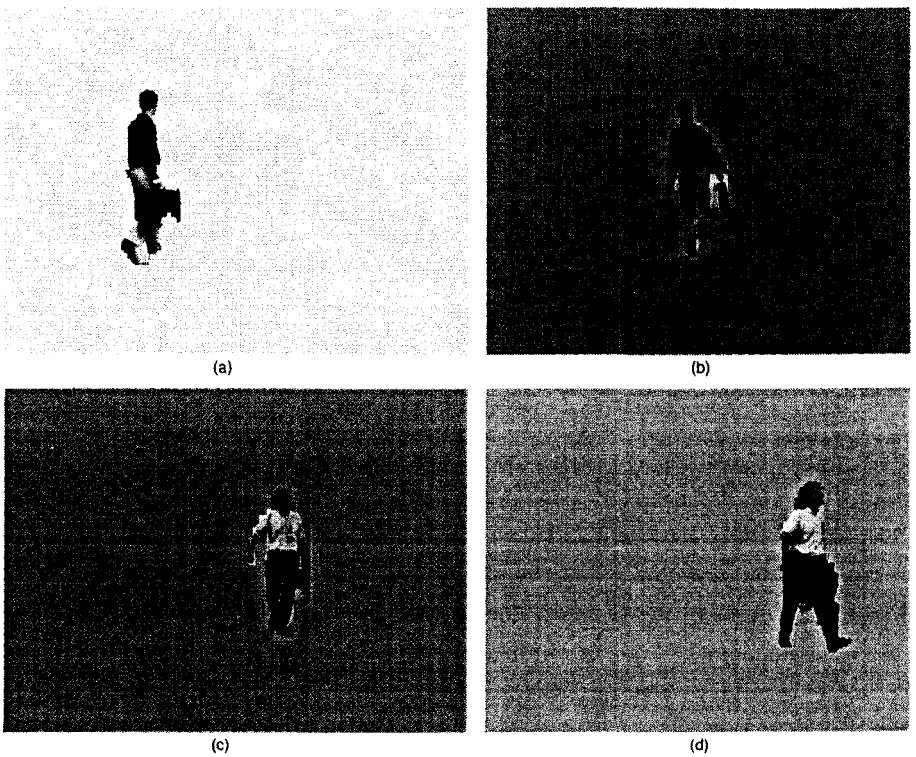

Figure 5.7: The output images from the second iteration.

video sequence. Finally, column four has two large coefficients at positions two and four, which means there are two objects to be segmented from the fourth eigenimage $\mathbf{u}_4$ and their indices in the video sequence will be the second and the fourth frames, respectively.

The indexing relationship between the Eigenimages $\widetilde{\mathbf{U}}$ and the video frames $\mathbf{F}$ can be described as follows:

$$\mathbf{u}_3 \longrightarrow \mathbf{f}_1$$

$$\mathbf{u}_4 \longrightarrow \mathbf{f}_2$$

$$\mathbf{u}_2 \longrightarrow \mathbf{f}_3$$

$$\mathbf{u}_1 \longrightarrow \mathbf{f}_4$$

Then we use the Bell-Sejnowski algorithm in the stICA to optimize the eigenimages $\widetilde{\mathbf{U}}$ and obtain the unmixing matrix $\mathbf{W}_O$ such that $\mathbf{O}=\widetilde{\mathbf{U}}\mathbf{W}_O$. In the experiment, $\mathbf{W}_O$ is a

Figure 5.8: The original video sequence frames.

4×4 matrix

$$\begin{bmatrix} \underline{-10.9408} & -0.8998 & 2.1762 & -1.4259 \\ -1.9929 & \underline{-38.6003} & 1.4613 & 2.8184 \\ -0.5246 & 0.2471 & \underline{-40.5752} & 2.8608 \\ -1.0995 & 8.3672 & 6.9683 & \underline{35.0712} \end{bmatrix}$$

For the same reason outlined above, the relationship between $\widetilde{\mathbf{U}}$ and $\mathbf{O}$ is

$$\mathbf{u}_1 \longrightarrow \mathbf{o}_1$$

$$\mathbf{u}_2 \longrightarrow \mathbf{o}_2$$

$$\mathbf{u}_3 \longrightarrow \mathbf{o}_3$$

$$\mathbf{u}_4 \longrightarrow \mathbf{o}_4$$

Thus we can map the relationship between $\mathbf{F}$ and $\mathbf{O}$ as follows:

$$\mathbf{o}_3 \longrightarrow \mathbf{f}_1$$

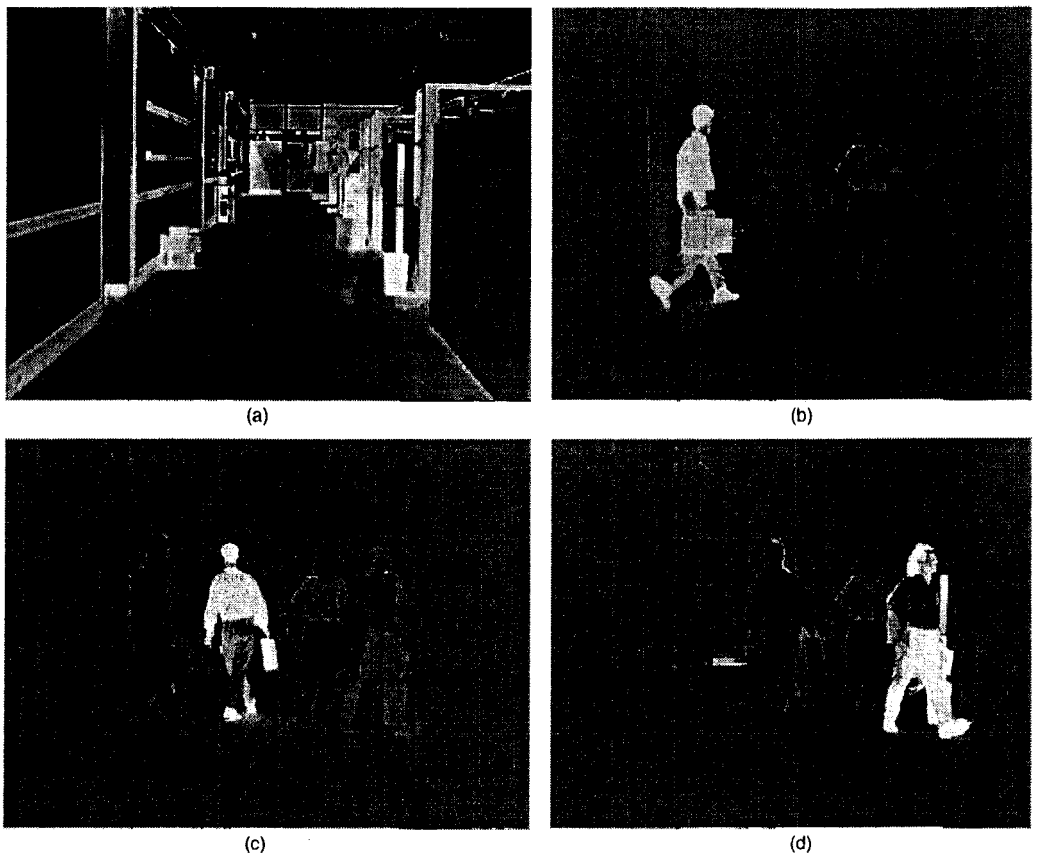

Figure 5.9: The eigenimages: $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$.

$$\mathbf{o}_4 \longrightarrow \mathbf{f}_2$$

$$\mathbf{o}_2 \longrightarrow \mathbf{f}_3$$

$$\mathbf{o}_4 \longrightarrow \mathbf{f}_4$$

The object indexing relationship from $\mathbf{F}$ to $\mathbf{O}$ through $\widetilde{\mathbf{U}}$ is illustrated in Fig. 5.10. In this way, the frame object order can be determined.

To compare the segmented image quality in these two iterations, the Peak Signal to Noise Ratio (PSNR) [25] [26] is employed. The PSNR is a standard criterion for objective noise measuring in video systems. For example, the image size is $M \times N$, $o_p(i,j)$ and $o_r(i,j)$ denote the pixel amplitudes of the processed and reference images, respectively, at the

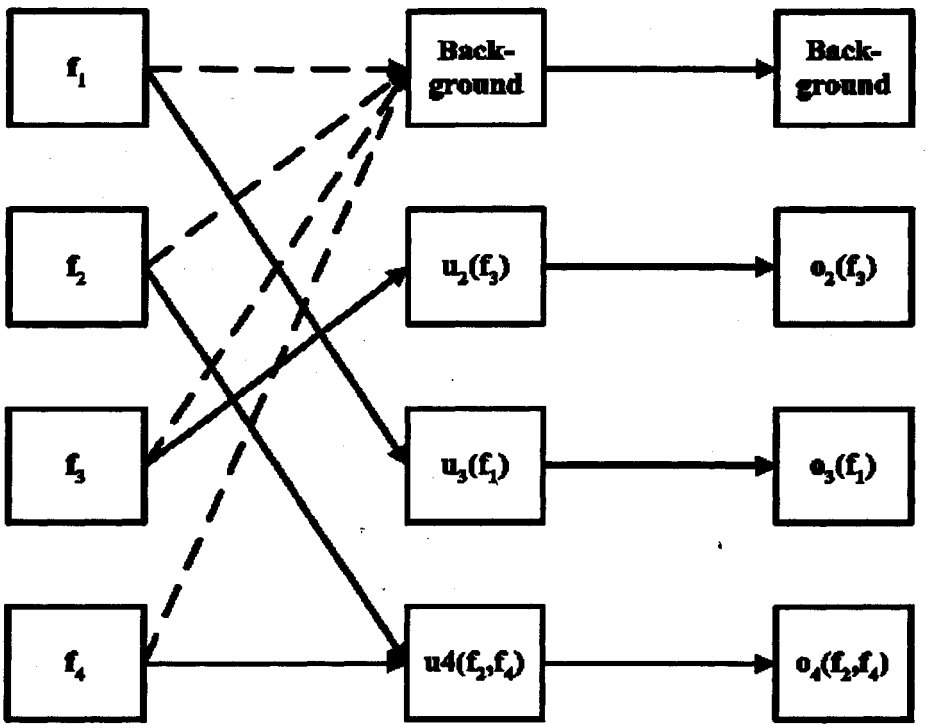

Figure 5.10: Illustration of the indexing relationship from **F** to **O** through $\tilde{\mathbf{U}}$.

position (i,j):

$$\text{PSNR} = 10 \cdot \log \frac{(255)^2}{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(o_p(i,j) - o_r(i,j))^2} \text{ dB.} \tag{5.6}$$

Table 5.1 compares the PSNR values (dB) of the segmented object images in the two iterations from the "Hall Monitor" sequence. It shows that the results obtained in the second iteration (Fig. 5.7) are superior to those in the first one. (Fig. 5.3).

Table 5.1: PSNR values (dB) of the segmented images in "Hall Monitor" sequence.

Iteration	Image (a)	Image (b)	Image (c)	Image (d)
First	30.25	27.43	26.12	34.71
Second	36.36	39.84	41.72	40.30

In another simulation experiment, the "Computer Lab" video sequence with 4.35-second duration is used. There are altogether 160 frames, each of which has 240×360

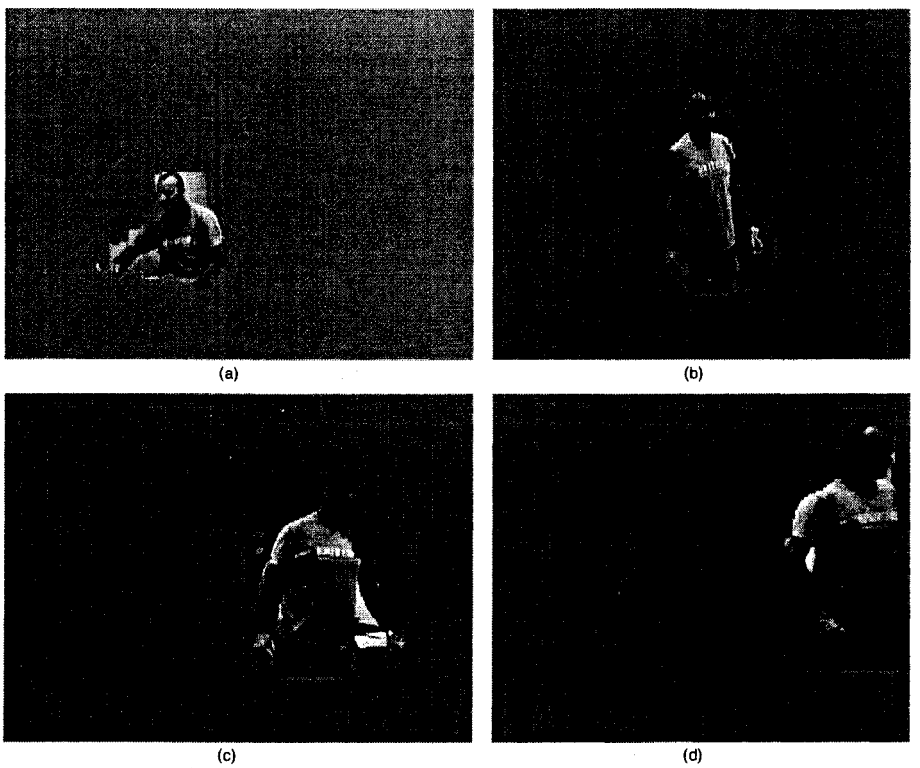

Figure 5.11: The output images from the first iteration of "Computer Lab" sequence.

pixels and 256 grayscale levels. We suppose that every video frame contains at least one object of interest. This means there are no pure "background" images. A set of frames are selected from these 160 frames for further processing in the proposed system. To avoid interference between close objects, frames are selected from the sequence at a constant interval 40. Thus there are 4 frames are selected to be processed by the system each time. The approaches in first and the second iterations (Figs. 3.2 and 5.1) are applied. The output images in the two iterations are shown in Figs. 5.11 and 5.12, respectively.

Based on the "Computer Lab" simulation experiment, Table 5.2 gives the comparison results of the PSNR values between the first iteration and the second iteration. The obtained results are quite similar to the results in Table 5.1. The results after the second iteration are better than the results after the first iteration. Moreover, the missing information on the object's face in Fig. 5.11(c) can be retrieved back in Fig. 5.12(c) by the proposed compensation method. This is because the compensated frames consist of

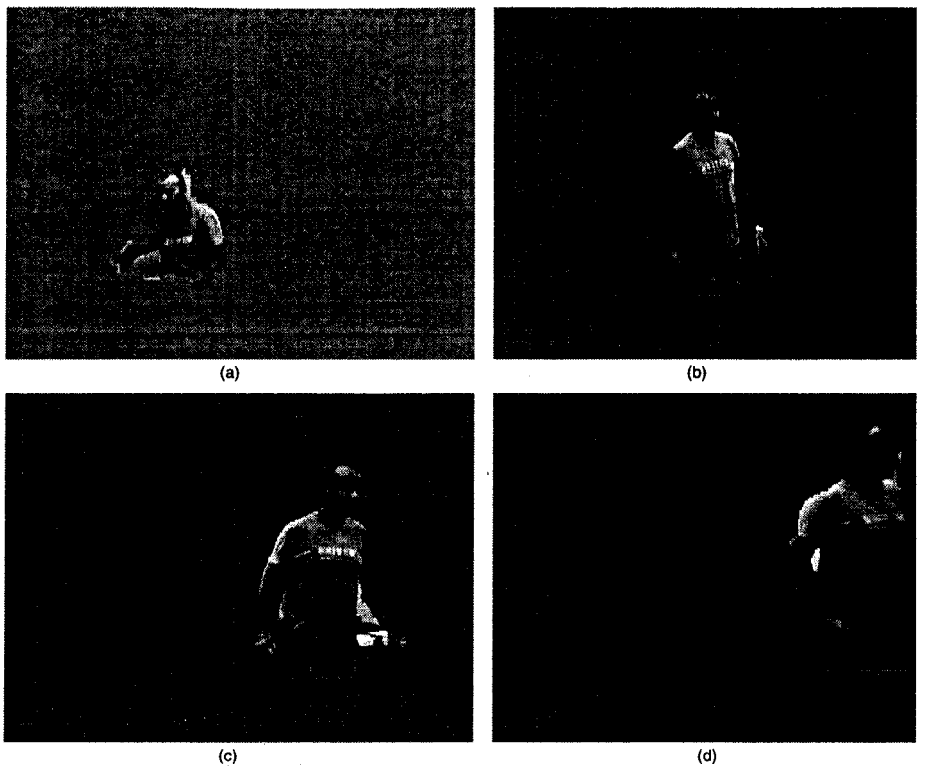

Figure 5.12: The output images from the second iteration of "Computer Lab" sequence.

both background information and object information inside the edges. Thus even some information is lost in the first iteration, it can still be retrieved by the proposed methods based on the stICA model. This is an advantage of applying the stICA to video frames.

Table 5.2: PSNR values (dB) of the segmented images in "Computer Lab" sequence.

Iteration	Image (a)	Image (b)	Image (c)	Image (d)
First	24.42	29.66	25.17	38.72
Second	26.67	31.21	31.54	40.28

5.4 Summary

To deal with the nonlinear combination problem in stICA model for video sequences, a novel compensation method is introduced. Spatial signals with clear shapes and edges are recovered from compensated video frames by the stICA. An object indexing method

is used to index the segmented objects in the original video sequence. The ROIs obtained in the first iteration are used to locate the objects in different spatial images (Fig. 5.2). Those post-processing techniques used in the first iteration, such as edge detection with region growing and multiscale image segmentation, are employed again in the second iteration to improve object segmentation accuracy.

In the second iteration, the processing approaches consist of (shown in Fig. 5.1):

1. Extracting the regions of background that are blocked by the objects whose boundaries are obtained in the first iteration.

2. Superimposing the regions of background that are blocked by the objects onto the original frames to obtain the compensated frames.

3. Employing the stICA to process the compensated frames to produce spatial signals with clearer edges.

4. Indexing the frame objects by the SVD and the weighting matrices.

5. Using the ROIs obtained in the first iteration to locate the objects in different spatial images.

6. Employing edge detection with region growing, multiscale image segmentation approaches to get accurate objects (shown in Fig. 5.7).

Simulation results reveal that the proposed approaches along with the post-processing techniques can segment the objects of interest accurately and effectively.

Chapter 6

Conclusion

IN this thesis, a new framework for high-level video object segmentation based on the stICA is presented. This chapter summarizes the work presented and suggest some possible research extensions.

6.1 Contribution

The main purpose of this thesis is to verify the efficacy of the stICA model for video sequences. Based on the similarity of the independence of spatial and temporal signals in fMRI and video sequences, an stICA model for video sequences is formulated. When the stICA model is applied directly to the video objects, a nonlinear combination problem will arise due to the absence of background information. To deal with the nonlinear combination problem, a novel two-iteration approach is presented.

In the first iteration, the stICA processing together with wavelet analysis, edge detection, region growing and multiscale segmentation techniques segment the objects from their backgrounds. However, some of the segmented objects cannot be extracted accurately, especially when objects have a complicated background. Thus the second iteration is necessary.

The nonlinear combination problem in Eq. (3.6) is the major obstacle for the stICA application for video sequences. The problem leads to rather poor outputs from the video frames processed by the stICA. To deal with this problem, we introduce a novel

blocked region compensation method. The stICA model is applied to the compensated frames and it recovers the spatial and temporal signals. Both theoretical derivation and simulation results show that this compensation technique is effective. After the second stICA processing, a frame object indexing approach is introduced to address the problem that is caused by the uncertain signal order of the ICA. This approach is based on the SVD and the weighting matrices used in the stICA algorithm. The ROIs obtained in the first iteration are used to locate the objects in different recovered spatial signals. The post-processing techniques utilized in the first iteration, such as edge detection with region growing, and multiscale segmentation are applied again. Simulation experiments show that the outputs from the second iteration are improved and the extracted objects are superior to those extracted in the first iteration.

The contributions of this thesis consists of

1. A new method of analyzing video sequences by the stICA model.

2. A novel compensation method to deal with the nonlinear combination problem in the stICA model for video sequences.

3. An integrated post-processing approach that consists of wavelet analysis, edge detection, region growing and multiscale segmentation techniques.

6.2 Possible Extension

There are some possibilities that may be explored in the future in order to enhance the performance of the proposed system and to extend its applicability.

1. The stICA optimization processing has the highest computational cost in the proposed system. The implementation of its algorithm can be optimized to allow faster execution of the whole system.

2. Object motion analysis. The proposed method in this thesis can effectively segment different moving objects from a video sequence. If two successive frames are

compared to obtain the change of locations of the moving objects, then the objects moving velocity and direction can be predicted. This application has a promising future in the image and video processing.

Bibliography

[1] D. Lelescu and D. Schonfeld, "Statistical Sequential Analysis for Real-time Video Scene Change Detection on Compressed Multimedia Bitstream," *IEEE Transaactions on Multimedia,* vol. 5, issue: 1, pages: 106-117, March 2003.

[2] U.R. Gargi and S. Antani, "Performance Characterization and Comparision of Video Indexing Algorithms," *Proceedings of SPIE Conference Storage and Retreval for Image and Video Databaseds VII,* San Jose, CA, pages: 290-301, 1999.

[3] P. Campisi and A. Neri, "Synthetic Summaries of Video Sequences Using a Multiresolution Based Key Frame Selection Technique in a Perceptually Uniform Color Space," *Proceedings of 2000 International Conference on Image Processing,* vol. 2, pages: 10-13, Septmeber 1997.

[4] H. J. Zhang, J.Y.A. Wang, and Y. Altunbasak, "Content-Based Video Retrieval and Compression: A Unified Solution," *Proc. Int. Conf. Image Processing.* vol. 1, pages: 13-16, Oct. 1997.

[5] M. Mckeown and M. Makeig, "Spatially Independent Activity Patterns in Functional Magnetic Resonance Imaging Data During the Stroop Color-Naming Task," *Proc. Natl. Acad. Sci. USA,* vol. 95, pages: 803-810, 1998.

[6] A.J. Bell and T.J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Computation.* Vol. 7, pages: 1129-1159, 1995.

[7] Hyper Dictionary. *functional Magnetic Resonance Imaging, http://http://www.hyperdictionary.com/dictionary/fMRI.*

[8] J. Stone,"Spatial, Temporal, and Spatiotemporal Independent Component Analysis of fMRI Data," *Proceedings of the 18th Leeds Statistical Research Workshop on Spatial-Temporal Modeling and Its Applications,* pages: 23-28, 1999.

[9] Z. Chen, X.-P. Zhang, "Video Sequences Processing Based on Spatiotemporal Independent Component Analysis," *Proceedings of 2003 Canadian Conference on Electrical and Computer Engineering,* Montreal, Canada, 2003.

[10] Z. Chen, X.-P. Zhang, "Object Extraction in Video Sequences Based on Spatiotemporal Independent Component Analysis," *Visual Communications and Image Processing 2003.* Lugano, Switzerland, 2003.

[11] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* John Wiley & Sons Inc., 2001.

[12] D.C. Lay, *Linear Algebra and Its Applications.* Addison-Welsely Publishing Company, Boston, 1993.

[13] K.I. Diamantaras and A.P. Kung. *Principal Component Neural Networks: Theory and Applications.* Welsely Publishing Company, 1996.

[14] Eric Weisstein's World of Mathematics (MathWorld). *Eigen Decomposition Theorem, http://mathworld.wolfram.com/EigenDecompositionTheorem.html.*

[15] T.M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley, 1991.

[16] A. Papoulis, *Probability, Random Variables, and Stochastic Process.* McGraw-Hill, 3rd edition, 1991.

[17] A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Neural Networks,* vol. 13, pages: 411-430, 2000.

[18] D.T. Pham, P. Garrat, and C. Jutten, "Separation of a Mixture of Independent Sources Through a Maximum Likelihood Approach," *Proceedings of European Signal Processing Conference(EUSIPCO),* pages: 771-774, 1992.

[19] T.-C. Hsung, D.P.-K. Lun, and W.-C. Siu, "Denoising by Singularity Detection," *IEEE Transactions on Signal Processing,* vol. 47, No. 11, November 1999.

[20] S. Mallat and W.L. Hwang, "Singularity Detection and Processing with Wavelets," *IEEE Transactions on Information Theory,* vol. 38, No. 2, March 1992.

[21] A.D. Marshall and R.R. Martin, *Computer Vision, Models and Inspection.* World Scientific Publishing Company, River Edge, NJ, USA, 1993.

[22] M. Tabb and N. Ahuja, "Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Transaction on Image Processing,* vol. 6(5), pages: 642-655, 1997.

[23] X.-P. Zhang, "Target Segmentation and Extraction from Geographic Images Based on Multiscale Analysis," *Proc. of 5th WSES/IEEE World Multiconference on Circuits, Systems, Communications & Computers,* Rethymnon, Crete, July 8-15, 2001.

[24] X.-P. Zhang, "Multiscale Tumor Detection and Segmentation in Mammograms," *Proc. of 2002 IEEE International Symposium on Biomedical Imaging,* Washington D.C., USA, July 2002.

[25] I.-M. Kim and H.-M. Kim, "A New Resource Allocation Scheme Based on a PSNR Criterion for Wireless Video Transmission to Stationary Receivers over Gaussian Channels," *IEEE Transactions on Wireless Communications,* vol. 1, issue. 3, pages: 393-401, July 2002.

[26] S. Saha and R. Vemuri, "An Analysis on the Effect of Image Features on Lossy Coding Performance," *IEEE Signal Processing Letters,* vol. 7, No. 5, May 2000.