DUBLIN CITY UNIVERSITY
SCHOOL OF ELECTRONIC ENGINEERING

MAIN CHARACTER DETECTION IN NEWS AND MOVIE CONTENT

By
Csaba Czirjék

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
M. ENG. IN ELECTRONIC ENGINEERING
AT
DUBLIN CITY UNIVERSITY
IN THE
SCHOOL OF ELECTRONIC ENGINEERING
2005

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of M. Eng. in Electronic Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ____________ ID No.: 50161512

Date: ____________

Abstract

Advances in multimedia compression standards, data storage, digital hardware technology and network performance have led to a considerable increase in the amount of digital content being archived and made available online. As a result, data organization, representation and efficient search and retrieval from digital video repositories has seen increased interest from the research community in recent years. In order to facilitate access to desired media segments, many indexing techniques have been employed. Automatic content structuring is one enabling technology used to aid browse/ retrieval. Scene-level analysis and sports summarization are two examples of active research in this area. Content structuring can be considered as the task of building an "index" and/or "table of contents" for events or objects that occur throughout a programme.

Our approach to content structuring is to build an index based on the reappearance of the main characters within the content. For news programmes, this can be used for temporal segmentation into individual news stories based on the fact that the anchor-person, the main "character" in this scenario signals the beginning of a news item. For movie content, this could provide enhanced random access browsing functionality to the end user. In this thesis we propose an approach to news story segmentation that uses low-level features and three different algorithms for temporal segmentation. We then extend this system to perform anchor-person detection using automatic face detection and clustering algorithms. An extensive manually marked up test set has been used to validate each component of our overall approach. Finally, we discuss how our approach could be extended to identify the main characters in movie content using similar classification techniques and directorial conventions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's world we are surrounded by information. Our very existence depends on the sensory information we acquire via a large variety of channels. *It's not money that rules the world, it's information.* The vast amount of information available in digital format could not be imagined decades ago. Newspapers, technical documents, e-libraries, online museum collections are just a few examples of day-to-day applications for industrial, commercial, educational, financial, governmental purposes. Storage, search and retrieval of large amounts of data is possible due to huge technical advances in digital circuit design, database management, network infrastructure, data mining techniques, artificial intelligence etc. The rapid advances in digital hardware technology has led to a constant increase of the performance-per-price ratio thus making possible higher storage capacity and high volume data transmission. We can say that the technical progress corroborated with scientific research made the digital era a reality. Organizing huge amount of information in an efficient manner is not a trivial task. The utility and usability of a digital library strongly depends on the organization and management of information, which leads to more or less efficient use. The initial blueprint or underlying structure has to be carefully chosen, having in mind that a library is expanding and therefore has to deal with ever larger amounts of data.

Newspapers and radio-television news broadcasts are a focal point in peoples' daily life. We can think of it as a summary of activities on a local or global scale which concern us and touch our lives directly or indirectly. The permanent exposure to information globalizes society and broadens people's views and knowledge. The impact of news on human activities is high. Just imagine the case of a company listed in a stock exchange; rumours about the company's situation can drive a company into bankruptcy or, in the fortunate case, to success due to rising share prices.

The World Wide Web has taken information exchange to another level. Ideas run through the Internet in an instant, making possible information transfer between people all around the globe. The Web has become the information highway and news channels possess a significant slice of the traffic and user interest as well. Broadband Internet connection in conjunction with hundreds of channels give on-line access to a large quantity of information from a variety of sources.

Our research is focused on digital video. The *presentation aspect* plays an important role in digital media retrieval. It is not possible to digest large volumes of data in raw format thus efficient retrieval of digital media is a must-have requirement for multimedia libraries in order to be practical. Searching through thousands of GBytes of unorganized data is a difficult and time consuming task from the user's point of view. In such cases, the desire is to minimize human intervention and automate the system as much as possible. On the producer's end, manual annotation is also a tedious burden and the demand for automatic multimedia indexing and retrieval systems is therefore substantial. The user queries tend to be very broad in scope. Example queries include: "find me a BBC programme about the outbreak of SARS and its spread" or "give me a summary of today's business report". User queries are formulated at the level of human understanding of information, addressing specific events, persons or objects. In contrast automatic audio-visual analysis deals with features such as colour, texture, shape, audio signal characteristics (frequency spectrum, pitch, zero crossing rate etc) etc., which can be extracted for any given programme by an automatic process. Thus the fundamental problem or difficulty in video indexing and retrieval is the gap between the user queries formulated at the *cognitive level* and the analysis carried out at *system level* [Hanjalic et al., 2000] ; in other words how humans perceive and understand the information and how audio-visual analysis tries to approximate semantics or meaning in video.

Many approaches have been reported in the literature which deal with the above problems. One common factor in learning-based approaches to semantic video indexing is *multi-modal feature fusion.* Hidden Markov Models, Bayesian Networks, Support Vector Machines, Neural Networks use feature vectors built by extracting relevant characteristics from the available information sources: visual cues, audio signal characteristics, closed captions, textual descriptions. Usually, a supervised or unsupervised classification takes place. Another category of approaches to video indexing by content are the *rule-based* methods. In this case, commonly observed patterns in the structure of video are used as cues to generate a set of rules or a grammar which approximates the temporal structure of the programme. A common characteristic of the above methods is that the set of rules is *adapted to genre.* Choosing a set of rules for each genre seems realistic (i.e. news, sports, documentaries) because we might want to constrain our algorithm to a specific type of programme.

There are advantages and disadvantages to each of the above approaches. A classifier needs to be trained therefore demands a manually annotated training corpus. Rules, on the other hand, are "made-up" and they can only estimate a fixed set of scenarios which can occur in real situations.

The research in this thesis is focused on semantic video segmentation adapted to genre. Given the large diversity of broadcasts encountered daily, we concentrate on two of the most frequently encountered type of programmes. First, news programmes are an important slice of the media today and this type of content is highly demanded. Automatically segmenting a news programme into its constituent stories is a challenging task and could provide vital information for users and content providers. We define the *major cast* in the context of news as the anchorperson because it is the leading character or actor throughout the news and has a particular significance. Therefore automatically segmenting news "on-the-fly" seems to be useful, given the amount of news coverage today. As a second experiment we target a broader type of programmes in which we define the main characters as the leading actors which occur throughout the movie. Motion pictures have a variable scope, audience, style and target. In this case, we try to focus our analysis on expanding the major cast detection technique to a genre where there are multiple actors appearing throughout the movie and classify them according to their degree of importance using content analysis based on movie-specific semantic units.

This thesis is organized as follows: Chapter 2 presents an overview on existing low-level audio-visual feature extraction techniques used for image and video indexing and retrieval. More emphasis lies on the visual side because the application developed and described in the following chapters predominantly rely on these. Chapter 3 gives an overview of mid-level visual features for video databases and describes a face detection and clustering technique which contribute to major cast detection for news and movies. Chapters 4 and 5 each give a description of major cast detection in news and movies respectively presenting existing systems and a number of experimental setups carried out. A summary and directions for future work are included in the concluding section of this thesis.

Chapter 2

Low-level Audio-Visual Analysis for Video Indexing

This chapter gives an overview of low-level audio-visual feature extraction techniques for video indexing and retrieval. Low-level features are extracted directly from the digital signal the audio-visual content is represented by and have little or nothing to do with human perception. There have been many low-level features proposed for image/video indexing and retrieval [Nagasaka and Tanaka, 1992, Rui and Huang, 1999, Zhang et al., 1997]. Broadly speaking, these features express characteristics like colour, shape, texture, motion etc. In video indexing and retrieval most of the analysis is employed at shot level as these are regarded the basic units, above the raw video frames in the visual index hierarchy. For this purpose techniques for determining shot boundaries in video are also presented in this chapter.

2.1 Temporal Video Segmentation

Due to the large amount of data involved, a video stream needs to be partitioned into manageable pieces. This process is termed *temporal video segmentation* or *shot boundary detection* and is usually the first step in video indexing. In this way, as illustrated in Figure 2.1, the representation of video at shot level can be regarded as an abstractization from the raw video data to homogeneous video segments possessing common low-level visual characteristics. Because levels 0,1 and 2 can be instantiated

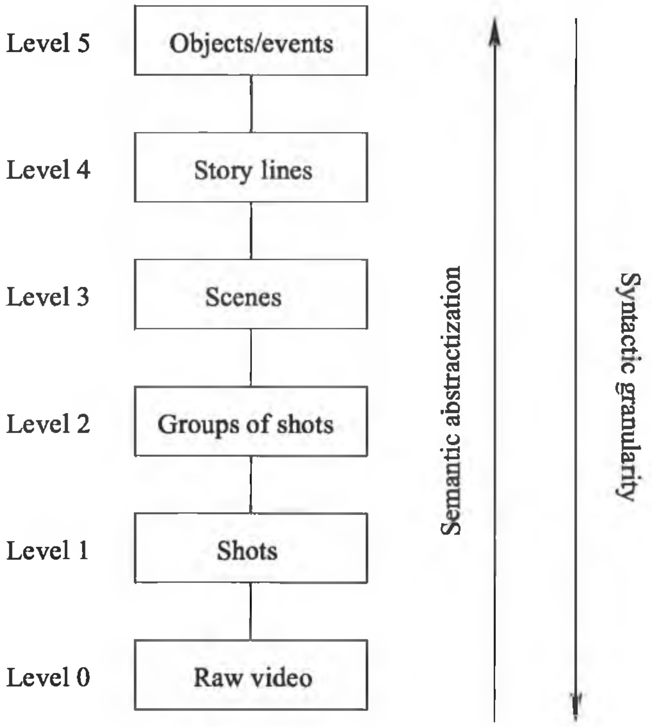


Figure 2.1: A hierarchical visual index

for any given video content they are termed low-level indexes. In the research community addressing video indexing and retrieval, shots are regarded as basic units for further analysis. In the next sections we focus on the level 0/1 transition.

2.1.1 Shot Boundary Detection

Shot boundary detection is a common preliminary step in video indexing. A *shot* is defined as a continuous stream of frames taken from the same camera. In order to determine the shot structure of the video, the transitions between shots need to be determined. There are two basic types of transitions which may occur between shots: *abrupt* and *gradual*. Abrupt transitions, often termed hard-cuts, appear due to switching of the video signal from one camera to another during broadcasting or in the editing process. Gradual transitions are effects introduced to combine two shots and are more difficult to detect because the transition between two shots occurs over a number of frames. Typical gradual transitions used in cinematography are fades and dissolves. Fades in/out consist of a stream of frames with gradual increase/decrease in intensity from/to a black frame. Superimposing two shots with increasing (respective decreasing) intensity constitutes a dissolve. More complex gradual transitions are wipes, windows, blocks, clock effects etc.

The main idea in shot cut detection is that frames surrounding the boundary exhibit discontinuities of visual features. Examples of visual features extracted to compute the degree of dissimilarity between video frames are pixel values, colour- and edge histograms and motion vector fields. A considerable amount of research has been done in this field [Boreczky and Rowe, 1996, Hanjalic et al., 2000, Nagasaka and Tanaka, 1992, Yeo and Liu, 1995, Zhang et al., 1993].

There are a variety of techniques for determining shot boundaries in video sequences but all can be grouped into two main categories, depending on the status of the video signal they operate on: *uncompressed-* and *compressed-domain* methods. In the first category the discontinuities of visual features are analyzed on the uncompressed video frames (pixel, histogram, edge tracking etc.) [Kikukawa and Kawafuchi, 1992, Yeo and Liu, 1995, Zabih et al., 1999]. Another more efficient way is to find shot cut occurrence patterns in the compressed or partly uncompressed video bitstream [Calic et al., 2002]. There is a trade-off between accuracy of prediction, given that the information held by uncompressed features is more comprehensive than compressed-domain features available in the MPEG bitstream. On the other hand, the second approach is necessary if real-time requirements are important. In the following we will present methods proposed in the literature for detecting video shot boundaries in the uncompressed- and compressed-domain.

2.1.1.1 Techniques for Visual Discontinuity Assessment

A. Pixel change ratio

To quantify the visual dissimilarity between successive frames within a video the average of pixel differences has been used in [Kikukawa and Kawafuchi, 1992]. For greyscale images this quantity is computed as:

$$Diff(i,i+1) = \frac{\sum_{x=1}^{M}\sum_{j=1}^{N}|P_i(x,y) - P_{i+1}(x,y)|}{M \cdot N} \tag{2.1.1}$$

For colour images the above equation becomes:

$$Diff(i,i+1) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\sum_{c\in\{Y,C_b,C_r\}}|P_i(x,y,c) - P_{i+1}(x,y,c)|}{M \cdot N} \tag{2.1.2}$$

In Equations 2.1.1 and 2.1.2 $P_i(x,y)$ and $P_i(x,y,c)$ denote the gray level intensity (respective colour value) of pixel (x,y) in frame F_i $M \times N$, whereas c represents the colour band index, in the above case $c \in \{Y, C_b, C_r\}$. If the difference value $Diff(i,i+1)$ exceeds a threshold a shot cut is declared between frames F_i and F_{i+1}. A slightly modified version [Zhang et al., 1993] counts the number of pixels whose intensity changes significantly over consecutive frames and a shot cut is identified if this quantity is greater that a second threshold:

$$Diff(i,i+1) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\Delta(i,i+1,x,y)}{M \cdot N} \tag{2.1.3}$$

where:

$$\Delta(i,i+1,x,y) = \begin{cases} 1, & \text{if } |P_i(x,y) - P_{i+1}(x,y)| > T_1 \\ 0, & otherwise \end{cases} \tag{2.1.4}$$

The disadvantage of the above methods is that they are sensitive to object and camera motion which produce many false positives.

B. Histogram Comparison

A popular and successful approach to shot boundary detection is based on histogram comparison over successive frames. Histograms are robust to object motion with constant background (as opposed to the previously described method) and small camera motion, therefore the visual content of successive frames taken from the same camera remains roughly the same. In the case of a shot change the histogram difference signals a significant change in the colour content of the scene. However, histograms are sensitive to camera operations such as panning or zooming. If H_i^c with $c \in \{Y, C_b, C_r\}$

is the colour histogram corresponding to frame i then a simple metric to compute inter-frame dissimilarity [Yeo and Liu, 1995] is:

$$Diff(i,i+1) = \sum_{j=1}^{N_{Bins}} \left(|H_i^Y(j) - H_{i+1}^Y(j)| + |H_i^{C_b}(j) - H_{i+1}^{C_b}(j)| + |H_i^{C_r}(j) - H_{i+1}^{C_r}(j)| \right) \tag{2.1.5}$$

The chi-square test χ^2 is another method used by [Zhang et al., 1993] and a slightly modified version in [Gunsel et al., 1996]:

$$\chi^2 = \sum_{j=1}^{N_{Bins}} \frac{|H_i(j) - H_{i+1}(j)|^2}{H_{k+1}(j)} \tag{2.1.6}$$

Other attempts exploit the invariance of the Hue component in the HSV colourspace to different lighting conditions [Arman et al., 1993].

C. Edge Change Ratio

Edge detection is often a pre-processing step for object detection or feature extraction in computer vision. Researchers have identified that the number of edge pixels entering/leaving frames provide clues for the shot boundary determination task. The edge change ratio is computed as:

$$\rho_n = \max\left\{\frac{\rho_n^{in}}{\sigma_n}, \frac{\rho_{n-1}^{out}}{\sigma_{n-1}}\right\} \tag{2.1.7}$$

where ρ_n^{in} and ρ_{n-1}^{out} represent the number of edge pixels entering/leaving frame $n/n-1$ and σ_n denotes the total number of edge pixels in frame n. [Zabih et al., 1999] use such methods for detecting and classifying production effects. They establish a cut point by examining the edge change fraction $\rho = max(\rho_{in}, \rho_{out})$, where ρ_{in} and ρ_{out} represent the percentage of entering (respective exiting) edge pixels. Peaks in the ρ curve tend to correspond to shot cuts. However, a special case arises when credits or cast list are displayed at the beginning or end of a movie, for example. Open-captions display strong vertical and horizontal gradient [Chen et al., 2004], thus the edge pixel percentages ρ_{in} and ρ_{out} change significantly, but no shot break occurs in the video.

D. Motion Vector Field Discontinuity

[Akutsu et al., 1992] use information provided by motion vectors to compute the inter-frame motion discontinuity as an indicator of shot breaks. The absolute difference between motion vectors of macroblocks in the same location in two consecutive pairs of frames are calculated. The discriminator is the ratio between the number of macroblocks with high motion vector difference from frame i to $i+1$ and frames $i+1$ and $i+2$ respectively. The inverse of this ratio represents the motion smoothness along frames.

2.1.1.2 Compressed Domain Analysis

Highly efficient shot boundary detection algorithms in the MPEG compressed domain are described in [Calic et al., 2002, Meng et al., 1995, Pei and Chou, 1999]. The techniques which fall under this category (double thresholding, DCT coefficient clustering) examine the temporal distribution of macroblock types in the MPEG bitstream and try to establish a pattern for shot cuts. When analyzing the MPEG *Group of Pictures (GOP)*, if there are many intra-coded and forward- respective backward-predicted macroblocks in B-pictures it is highly possible that a shot-cut will happen in the next respective frames (in coding order or within the GOP). A more complex analysis is required for detecting gradual shot changes which sometimes can be similar to scenes with high motion. A comprehensive review of compressed domain features for audio-visual analysis for video indexing can be found in [Wang et al., 2003a].

2.1.2 Keyframe Extraction

A *key frame* is a compact representation of a video shot. Key frames play the role of visual summaries. It is natural for humans to recollect the most memorable image(s) from a video segment (clip) so the scope of keyframe extraction is to approximate this human ability in an automated way. Clearly there is a subjective factor as to how people perceive the most significant aspects in video. For stationary shots, one key frame is sufficient most of the time. For shots with high visual activity (e.g. camera motion, object motion, objects entering/leaving) this task becomes more difficult. Multiple key frames which approximate the visual variation within a

shot are a solution in this case. It should be mentioned that keyframe extraction is strongly dependent on the accuracy of the shot boundary detection algorithm. Shot boundary detection accuracy can always be evaluated when having a ground truth to compare to. Unfortunately, in the case of keyframe extraction, this is not generally possible.
Keyframe extraction techniques can be grouped into three categories:

1. shot-based
2. visual content-based
3. clustering-based

Preliminary work on keyframe extraction techniques which fall into the first category use simply the first frame of a shot as representative [Zhang et al., 1997]. Choosing the frame just after a shot boundary is not always a recommended option; this frame might be the part of a dissolve and the keyframe image quality will suffer as a result.

[DeMenthon et al., 1998] represent a video sequence as a curve in high-dimensional space and, using techniques for planar curve simplification, extract the junctions between different curve segments as keyframes for video summarization. Depending on the level of simplification of the curve, keyframes represent, at the corresponding levels, perceptually significant density points along the video sequence. Based on the dissimilarity between the histogram of the frame under investigation and the average histogram of the previous N frames, [Gunsel et al., 1996] extract the representative keyframe if this discontinuity value surpasses a threshold.

Motion information has been used as an aide for key-frame selection. [Wolf, 1996] computes an inter-frame dissimilarity measure based on the optical flow calculated for each frame. Keyframe selection is performed by examining the local minima of this metric. Without using explicitly motion vector information, [Gresle and Huang, 1997] extract the representative keyframe at the minima of the shot activity curve computed on the basis of intra- and reference histogram differences as an activity indicator.

Non-sequential keyframe extraction techniques using unsupervised clustering have been developed by [Girgensohn and Boreczky, 2000, Hanjalic et al., 2000, Zhuang et al., 1998].

2.2 Low-level Audio-Visual Feature Extraction

2.2.1 Colour Features

Color features, such as color histogram, have proved to be effective in image and video indexing and retrieval. Often colours are sometimes converted from RGB values to other perceptual colour spaces, such as HSV, La*b*, HMMD etc., because they are closer to human colour perception models. In the MPEG video coding standard, colors are converted to YCbCr components where luminance and chrominance information are separated. The Cb and Cr components comprise the chrominance information and each of these are subsampled considering the characteristics of the human visual system [Rao and Hwang, 1996].

The first-order gray level histogram $P(I)$ is defined as:

$$P(I) = \frac{\text{number of pixels having gray level I}}{\text{total number of pixels in the image}} \tag{2.2.1}$$

Histograms are useful descriptors of the global colour distribution in an image. To compare video frames, a comparison is based on their colour signature. Highly visually similar frames will contain similar amounts of colours which is a successful image retrieval technique. Even if they don't provide any clue to what objects appear in the scene, histograms are robust to camera operations and object motion, characteristics which make them useful in video indexing and retrieval.

In the MPEG-7 [Manjunath et al., 2001] multimedia description standard there are a number of proposed colour-descriptors based on histograms which characterize local, spatial or global colour distributions in images. A few examples are:

- Colour Structure Descriptor - provides local colour features.
- Scalable Colour Descriptor - gives a global colour description.
- Colour Layout Descriptor - captures the spatial distribution of colour in an image.

2.2.2 Edge Features

Edges are boundaries between objects and background or between regions within an object and are manifested by high gradient values. Edge detection is implemented by convolving the signal with some form of linear filter which approximates a first or second derivative operator. If there is a discontinuity in the intensity function or a steep intensity gradient, the edges are found by computing the derivative of the intensity and locating the points where the derivative is a maximum.

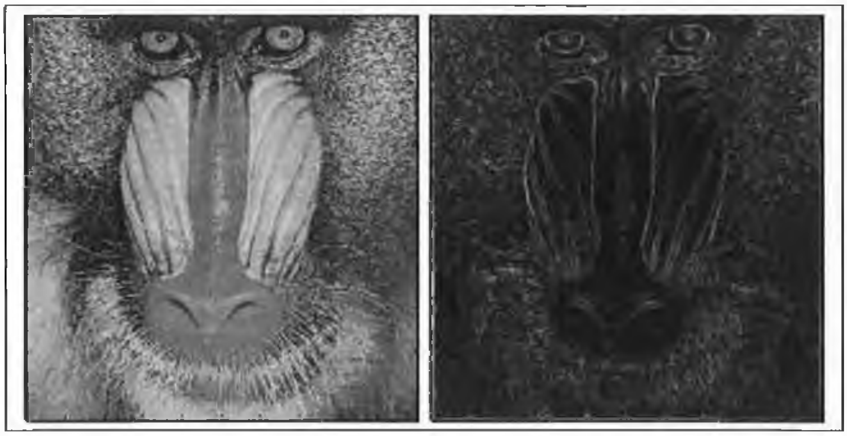


Figure 2.2: Test image and edge detector output

Considering the image function as I, the gradient is given by the vector:

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

The magnitude of the gradient is given by:

$$\sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

and its direction by:

$$tan^{-1}\left(\frac{\partial I/\partial y}{\partial I/\partial x}\right)$$

Examples of gradient operators are Prewitt, Roberts, Sobel. The Sobel convolution mask for $\partial I/\partial x$ is given by:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

To find both horizontal and vertical edges the second derivative or Laplacian is used:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

The Canny edge detector determines the maxima of:

$$\frac{\partial^2}{\partial n^2}(G \otimes I)$$

where $n = \frac{\nabla G \otimes I}{|\nabla G \otimes I|}$. G represents a Gaussian which smooths the image to reduce de effect of noise in the output edge map.

The MPEG-7 standard [Manjunath et al., 2001] defines the Edge Histogram Descriptor (EHD) which captures the spatial distribution of edges in the image by tessellating the image into 16 × 16 non-overlapping blocks and computing 5 edge directions for each block having the result quantized into a 5-bin histogram. A comprehensive review of edge detection techniques can be found in [Ziou and Tabbone, 1998]. A highly efficient edge extraction technique from compressed images is described in [Shen and Sethi, 1996].

2.2.3 Texture Description

2.2.3.1 Statistical Texture Description

Texture is an important feature that is considered in image and video analysis. Texture can be described by the number and types of primitives and by their spatial relationships; by texture primitives we understand the *texels* which constitute the building blocks of a texture. A number of texture description tools are [Pratt, 1991]:

- autocorrelation function

- co-occurrence matrices
- edge frequency
- primitive length
- mathematical morphology
- fractals

Using autocorrelation the linear spatial properties of texture primitives are evaluated by the correlation coefficients. These coefficients are given by:

$$C(p,q) = \frac{M \cdot N}{(M-p)(N-q)} \cdot \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} f(i,j) f(i+p, j+q)}{\sum_{i=1}^{M} \sum_{j=1}^{N} f^2(i,j)} \tag{2.2.2}$$

where *M,N* are the width and height of the image, *p* and *q* represent the position difference in the *i,j* direction and *f(i,j)* is the pixel value at *i,j*.

The lengths of texture primitives in different directions can be used for texture description. A coarse texture has a large number of neighboring pixels possessing the same gray level (run length); the opposite is valid for fine textures.

2.2.3.2 Haar Wavelets

Wavelets are widely used for texture description [Theodoridis and Koutroumbas, 1999]. In general, starting from the mother wavelet function $\Phi(t)$ we define a family of shifted and stretched scaling functions $\{\Phi_{k,n}(t)\}$ given by:

$$\begin{aligned} \Phi_{k,n}(t) &= 2^{-\frac{k}{2}} \Phi\left(2^{-k} t - n\right) && (2.2.3) \\ &= 2^{-\frac{k}{2}} \Phi\left(\frac{1}{2^k}\left(t - n2^k\right)\right) \forall k, n \in \mathbb{Z} && (2.2.4) \end{aligned}$$

where k, n are integers that scale and respectively dilate the mother $\Phi(t)$ wavelet function.

The Haar basis vectors for $N = 2^n$ are constructed using:

$$h_k(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \text{if } \frac{q-1}{2^p} \leq x < \frac{q-\frac{1}{2}}{2^p} \\ -2^{p/2} & \text{if } \frac{q-1}{2^p} \leq x < \frac{q}{2^p} \\ 0 & otherwise \end{cases} \tag{2.2.5}$$

The 2 × 2 Haar transform matrix is:

$$H_{(2)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.2.6}$$

If X is the input signal, the output of the separable transform is computed as:

$$Y = HXH^T \tag{2.2.7}$$

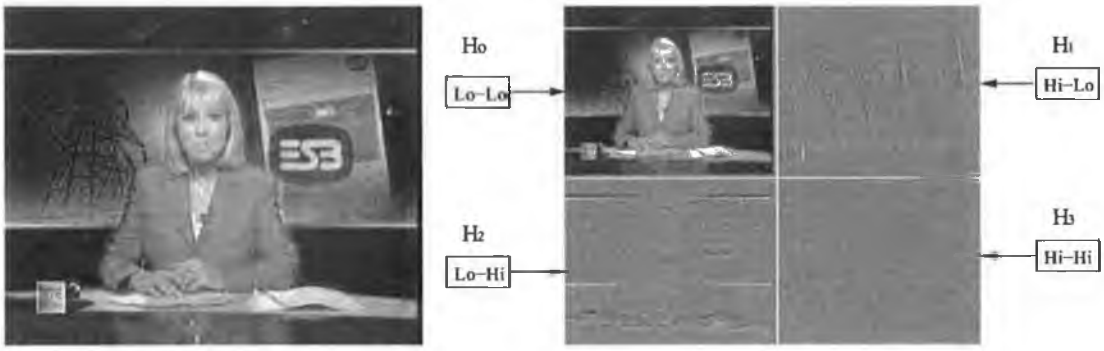


Figure 2.3: An image and its decomposition into frequency bands using a one-level Haar Transform

The Haar transform operates on adjacent horizontal elements and then on vertical elements. Most of the energy of the data is concentrated in the upper left (H_0) corner (Figure 2.3), whereas H_1, H_2 and H_3 provide different spectral characteristics. For example the H_2 area in Figure 2.3 can be seen as a low-pass horizontal filtering of the image followed by high-pass vertical filtering thus emphasizing vertical frequencies. A popular multiresolution image processing technique is Mallat's pyramidal decomposition [Mallat, 1989]. An example of an anchorperson image decomposed into frequency bands using a one-level Haar Transform is presented in Figure 2.3. From the computational point of view, the transform is an attractive technique for visual feature description.

To extract a feature vector, the image is tesselated into $M \times M$ non-overlapping blocks and for each block a feature vector is computed:

$$f = [f_{H_1}, f_{H_2}, f_{H_3}]$$

where, if for band H_x, $i = \overline{1,3}$, the coefficients corresponding to the block are:

$$\{C_{k,l}, C_{k+1,l}, \ldots, C_{k+\frac{M}{2},l+\frac{M}{2}}\}$$

then:

$$f_{H_x} = \left(\frac{1}{M/4} \sum_{i=0}^{M/2} \sum_{j=0}^{M/2} C_{k+i,l+j}^2 \right) \tag{2.2.8}$$

Then all feature values are standardized:

$$f_x = \frac{f_{H_x} - \mu_{H_x}}{\sigma_{H_x}} \tag{2.2.9}$$

which provides features with zero mean and unit standard deviation.

2.2.3.3 Gabor Functions

Gabor features have been used for texture segmentation, classification and pattern retrieval [Bovic et al., 1990].

A two dimensional Gabor function and its Fourier transform is given by:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jW_x\right] \tag{2.2.10}$$

$$G(u,v) = exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \tag{2.2.11}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$.

Using the $g(x,y)$ mother Gabor wavelet, similar to Equation 2.2.4 the family of Gabor wavelets can be obtained through scaling and dilations:

$$g_{mn}(x,y) = a^{-m}G(x',y'),\ a > 1,\ m,n = \text{ integer}$$

$$\begin{aligned} x' &= a^{-m}(xcos\theta + ysin\theta) \\ &\text{and} \\ y' &= a^{-m}(-xsin\theta + ycos\theta) \end{aligned} \tag{2.2.12}$$

where $\theta = n\pi/K$, K is the total number of orientations and a^{-m} is the scaling factor. Given the number of scales and orientations the filter parameters σ_u and σ_v are calculated. Usually the mean and standard deviation of the transform coefficients are used to form feature vectors and regions/images/patterns are compared using some form of similarity measure in the feature space.

2.2.4 Shape Features

The shape of detected objects can provide clues for image/video retrieval. Shape description [Jain, 1988] can be roughly classified into measurement-based and transform-based methods. A number of shape descriptors are presented next.

2.2.4.1 Area

Represents the number of pixels present in a region:

$$Area = \int_a^b f(x)dx$$

2.2.4.2 Perimeter

The number of pixels in the boundary of the shape or the curve length is given by:

$$Perimeter = \int_a^b \sqrt{1 + f'^2(x)}dx$$

2.2.4.3 Compactness

The compactness of a region is given by:

$$\frac{Perimeter^2}{Area}$$

The circle has the highest value of this ratio, being the most "compact" shape.

2.2.4.4 Roundness

Roundness of a shape can be computed as follows:

$$\frac{4 \cdot Area}{\pi \cdot \sqrt{\theta_M}}$$

θ_M being the major axis of the shape.

2.2.4.5 Aspect Ratio

The aspect ratio is computed as the ratio between the major axis and the minor axis of the shape: θ_M/θ_m where θ_M and θ_m represent the major axis, respective the minor axis.

2.2.4.6 Moment Invariants

Moments are used to describe the statistical properties of shapes [Jain, 1988]. The ij^{th} moment of a two-dimensional function is defined as:

$$m_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N} x_i y_j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y)} \tag{2.2.13}$$

We can see that m_{10} is the x component μ_x of the mean and m_{01} is the y component μ_y of the mean. Then the *central moments* are given as follows [Jain, 1989]:

$$\mu_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N} (x-\mu_x)^i (y-\mu_y)^j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y)} \tag{2.2.14}$$

Moments provide useful shape information. For instance, if μ_{20} and μ_{02} represent the variances of x and y and μ_{11} is the covariance between x and y we can compute the orientation of the closed curve by computing the eigenvalues and eigenvectors of the covariance matrix:

$$C = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$$

Then the orientation angle of the shape is calculated:

$$\theta = \frac{1}{2} arctan \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \tag{2.2.15}$$

2.2.4.7 Fourier Descriptors

With Fourier descriptors (FD) the shape of the object is represented in the frequency domain as coefficients of the Fourier series expansion of the region (object)'s shape signature [Kauppinen et al., 1995]. If $f(k)$ represents the shape signature (boundary coordinates or curvature), the Discrete Fourier transform of the shape is given by [Safar and Shahabi, 2003]:

$$F(u) = \frac{1}{\sqrt(N)} \sum_{k=0}^{N-1} f(k) exp\left(-\frac{j2\pi uk}{N}\right) \tag{2.2.16}$$

with $u = 0, 1, \ldots, N-1$ and N the number of samples or points. The general shape information of the curve is contained in the low-level frequencies, whereas the details of the shape are represented by the high-level frequency coefficients.

2.2.4.8 Curvature Scale Space

The Curvature Scale Space [Mokhtarian and Bober, 2003] is a multi-scale representation of the curvature points of the shape of an object as it evolves in time. Curvature points are inflection points in the shape. If we consider the contour f (as a signal), it's scale space is obtained by convolution with a Gaussian with increasing width or standard deviation:

$$\begin{aligned} L(t;0) &= f(t) \\ L(t;\sigma) &= \int_{\xi=-\infty}^{\infty} g(\xi;\sigma)f(t-\xi)d\xi \end{aligned} \tag{2.2.17}$$

where

$$g(\xi;\sigma) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{\xi^2}{2\sigma^2}\right) \tag{2.2.18}$$

is a symmetric, zero-mean Gaussian with standard deviation σ.

2.2.5 Motion Features

The motion of objects, background and the camera recording the scene can sometimes provide a significant insight into the action happening and provides another low-level feature characteristic for video (as all the above presented features can be extracted for still images). This operation is usually termed as *shot type characterization.* Although there are practically no restrictions on the type of movement objects and/or background may have, for the recording camera there are a number of types of movements (degrees of freedom). Based on these a shot type characterization can be as follows [Akutsu et al., 1992]:

1. panning - horizontal rotation
2. tilting - vertical rotation
3. zooming - focal length change
4. tracking - horizontal traverse movement
5. booming - vertical traverse movement

6. dollying - horizontal lateral movement

The motion vectors existing in the MPEG bitstream provide motion information for blocks, regions and video frames and can be easily extracted by parsing the compressed video stream. Camera operations cause global and dominant motion in the video sequences. Using motion vector information camera motion can be estimated generally using the least square error (LSE) method to minimize the error between the estimated and compensated motion. More complex methods make use of a 6-parameter affine motion model for this task.

Besides estimating the camera motion, the dynamics of a shot can be characterized by a number of descriptors.

2.2.5.1 Motion smoothness

This characteristic is defined as the ratio between the number of macroblocks with significant motion vectors and the number of macroblocks whose motion vectors has changed significantly.

2.2.5.2 Motion histogram

The motion histogram gives a representation of global motion in video [Kobla et al., 1996]. Using a pixel change ratio map to generate the motion histogram, the authors in [Haoran et al., 2004] use this feature for video clip retrieval and classification.

2.2.5.3 Average motion vector magnitude

[Divakaran and Sun, 2000] threshold motion vectors and count the number of short, medium and long runs of zeros of the thresholded motion vectors. This characteristic also gives indication about the location, size and shape of the moving objects in the video frame.

2.2.5.4 Motion activity

The motion activity defines the intensity, direction and spatial distribution of motion [Sun et al., 2002]. Through summation of the motion vectors in consecutive P- and

B-pictures, the cumulative motion will represent the overall motion in each location in time.

2.2.6 Audio Features

The audio track accompanying the video is an important source of information which coexists with the video signal in any TV programme. Sometimes the visual features alone do not hold enough information for the indexing/retrieval task, therefore video features in conjunction with audio information can be used to meet the proposed goal. Low-level audio features can be grouped into two categories: *time-domain* and *frequency-domain*. A summary of the most frequently used audio descriptors is given below and can be found in [Manjunath et al., 2001].

2.2.6.1 Fundamental Frequency

This low-level audio feature describes the fundamental frequency of the signal and can be used as a preliminary tool for higher level audio descriptors such as speech and music.

2.2.6.2 Short-Time Energy

Short-time energy is a convenient way of representing the amplitude of the audio signal over time. This value is given by:

$$E_n = \frac{1}{N}\sum_m [x(m)w(n-m)]^2 \tag{2.2.19}$$

where

$$w(x) = \begin{cases} 1, & 0 \leq x \leq N-1 \\ 0, & \text{otherwise} \end{cases}$$

where x(m) is the discrete time audio signal, n is time index of the short-time energy, and w(m) is a rectangular window.

2.2.6.3 Harmonicity

The harmonicity of an audio signal is given by the harmonic ratio. This quantity is given by the ratio between the harmonic power and the total power. For a sinusoidal signal this value reaches 1 and 0 for pure white noise.

2.2.6.4 Short-Time Average Zero-Crossing Rate

This feature describes the number of times the audio signal crosses the x-axis over time where N denotes the number of audio samples:

$$Z_n = \frac{1}{2} \sum_m |sgn[x(m)] - sgn[x(m-1)]| \; w(n-m), \tag{2.2.20}$$

where

$$sgn[x(n)] = \begin{cases} 1, & x(n) \geq 0, \\ -1, & x(n) < 0 \end{cases}$$

This audio feature is difficult to extract from the compressed audio bitstream but it is a helpful tool in discriminating between speech and musical signals.

2.2.6.5 Spectral Centroid

Spectral centroid is the center of gravity of the log-frequency power spectrum and gives an indication whether the power spectrum has predominantly low or high frequencies. In other terms measures the average frequency, weighted by amplitude, of a spectrum. It is formally defined as:

$$\textit{Spectral Centroid} = \frac{\sum_{k=1}^{N} kF[k]}{\sum_{k=1}^{N} F[k]} \tag{2.2.21}$$

where $F[k]$ is the amplitude corresponding to the k^{th} bin in the DFT spectrum.

2.2.6.6 Spectral Spread

This quantity is defined as the root mean square (RMS) of the log-frequency power spectrum with respect to its centre of gravity (centroid).

2.3 Summary

In this chapter a review of low-level audio-visual features useful for image and video indexing and retrieval was presented. Low-level features describe characteristics of the underlying audio-visual signal and can be extracted automatically. These features accompanied by metadata are used to populate current video databases and as the basis for retrieval. We can regard them as seeds for features with more appropriate meaning to human perception and understanding of the information contained. However, low-level feature extraction is a completely automatic process, the only concern is the speed of computation (or algorithm complexity/implementation). The more subtle problem is their usefulness in real image/video indexing and retrieval applications and that depends on the type of end-user application where they are used for. We have chosen in our applications mostly visual features: for instance audio features are not specifically relevant in news story segmentation (Chapter 4) because the audio track mostly consist of speech, occasionally interrupted by music during commercial breaks, but these segments are already excluded from analysis. Thus, using the algorithms developed, audio-related features could be incorporated but their relevance is questionable. However, that could be further investigated in the future. For our experiments presented in the following chapters we use the image colour characteristics, colour histograms and Haar wavelet-based features along with shot motion characterization.

Chapter 3

Face Detection For Video Indexing

3.1 Introduction

In the previous chapter an overview of low-level features for image and video indexing has been presented. The aim of this chapter is to give an overview of mid-level visual features which could be used in applications targeting video indexing. The main feature presented here deals with the presence or absence of human face in video content. It is quite easy to draw a line where low-level features end and mid-level ones begin. But it is sometimes subjective to define the boundary between mid-level and high level features. First of all, categorization into these layers should be seen from the perspective of the end-application to be developed. For instance, in the case of automatic face recognition systems, face is the only significant high-level feature - because that is the target. In our case, we aim to automatically index a broadcast video programme into segments defined as semantically meaningful. Existence or absence of face throughout the programme gives only a small indication about the higher units desired to be identified (for instance wildlife documentaries can be distinguished from chat shows, but no other evidence can be extracted). Moreover, it is universally true that video broadcasts have human activities as their subject, therefore we expect a high concentration of human faces present within the show. Face presence evidence is integrated into frameworks which are built to detect higher level semantic entities which are presented in the following chapters.

3.2 Face Detection

3.2.1 Motivation

The face is the primary focus in social life and plays a major role in identifying each individual and communicating emotions. The ability to recognize and distinguish between persons is a truly amazing characteristic of the human brain. The skill of recognizing faces which we learn through life is very robust despite major changes in hairstyle and other varying elements.

From the computer vision and pattern recognition point of view the face is regarded as a naturally structured but highly deformable object. The task which deals with detection is termed *face detection*. The problem of face detection can be stated as: given an arbitrary image determine if there are any faces present and if so register the spatial coordinates of them. Face detection and face localization are slightly different problems. Whilst detection accounts for identifying the presence of a face in a given image, face localization is a more subtle problem that can be stated: to what degree was the detection successful. This is quite an important step in the usual processing chain which follows face detection in most applications.

Face detection is the first step in the face processing and analysis chain. Face detection is directly relevant to face recognition which deals with classification of a sample individual by comparing it to a database of stored face images in order to authenticate the same individual. Other applications are: face tracking, face localization, gesture recognition, advanced video surveillance, security and access control. The area in which we are investigating the application of face detection in this thesis is image indexing, search and retrieval. A comprehensive review of face detection applications is reported in [Brady, 1982].

In our case, given a video programme, the task is to recognize whether or not a face (or faces) occur within each shot. Many approaches have been proposed to locate faces in generic scenes, which use shape, color, texture, motion, edge and statistical analysis [Yang et al., 2002]. When working with digital video, face detection algorithms have to deal with a significant constraint - computation time. In general, 30 minutes of MPEG-1 video contains approximately 55,000 individual video frames, thus the algorithm applied to an individual frame should be able to classify it as face/non-face in the shortest time possible. To reduce the computation time, typically only every N^{th} frame is selected for processing. Such sampling seems natural, considering the temporal redundancy between adjacent frames. In our approach to face detection, we use colour information as the primary tool for locating face regions. We make use of this source of information based on our test content and also considering computational constraints. The processing chain involves a sequential pruning of candidate regions until all criteria are satisfied. In general, skin colour based face detection algorithms are computationally modest, but are sensitive to noise, lighting conditions and the colour content of the objects present in the scene.

3.2.2 Literature Review

Face detection has been a focal point in the research community for a long time and recently attracted huge interest from industry as well, given the diversity of applications for which it has significance. This is quite normal given that information about a person's identity can be extracted using image evidence. Therefore it is not surprising the large number of methods identified in the literature which address the face detection task. Face detection is usually the preprocessing step for face recognition systems and the efficiency of the latter is strongly dependent on the accuracy of this step.

According to [Yang et al., 2002], techniques to detect faces in images can be categorized into:

- Knowledge-based methods
- Feature invariant methods
- Template matching methods
- Appearance-based methods

The first three categories are also named as feature-based approaches and the last category is also termed image-based methods.

Knowledge-based approaches rely on a-priori information about the characteristics of a human face. These are summarized into more-less complex rules and are embedded into the detection process. The second category of methods try to identify features and relationships between them irrespective to pose, illumination variation, general environment factors etc.

An assembly of templates are used for the third kind of face detection approaches as a database in order to compare new face samples based on correlation analysis. Sometimes facial features are used to build the template database. The last category of methods adopt a "holistic" approach to the problem, using machine learning techniques to construct models used for face detection.

Knowledge-based approaches try to classify faces based on human knowledge about what a face is and encode that information into the detection algorithm. Methods to detect the face boundary, symmetry and multiresolution rule-based techniques have been employed successfully to detect and locate faces in images [Kotropoulos and Pitas, 1997, Yang and Huang, 1998]. These can be classified as *top-down* methods. *Bottom-up* approaches try to localize facial features or primitives and infer a classification based on their properties. A large number of low-level features have been used to detect eyes, lips, eyebrows (colour, edge detection techniques, texture analysis, shape analysis, color segmentation etc.) [Burel and Carel, 1995, Graf et al., 1995, Yow and Cipolla, 1997].

In contrast to the above, appearance- or image-based methods use a probabilistic approach to learn face templates from examples: eigenfaces [Turk and Pentland, 1991], hidden-markov models [Samaria and Young, 1994], neural-networks [Rowley et al., 1998], bayesian classifiers [Schneiderman and Kanade, 1998] and distribution-based methods [Sung and Poggio, 1998] are techniques employed.

In the following we present feature-based and image-based approaches as their combination are the basis of our approach to face detection.

3.2.2.1 Feature-Based Approaches

Feature-based approaches are mostly applicable in face detection systems where features are available (colour, motion etc.) and a multiresolution scanning of the image is not preferred, as is the case in real-time detection systems. Color is one of the most widely used features because it provides cues to prune the search space in the image [Albiol et al., 2000, Cai and Goshtasby, 1999, Herodotu et al., 1998, Hsu et al., 2002, Kotropoulos and Pitas, 1997, Sobottka and Pittas, 1998]. [Terrillon and Akamatsu, 2000] presented a comparative evaluation of different colour spaces used in face detection.
Feature analysis uses feature search techniques based on relative positions of individual facial characteristics. Symmetry, head-and-shoulders hypothesis, location of the main face axis, existence of a pair of eyes have been used to increase the confidence of a successful detection [Craw et al., 1987, DeSilva et al., 1995]. [DeSilva et al., 1995] present a method for constructing a face template by searching for the eye plane using edge density variation and then detecting the mouth using the eye-plane as a reference. Others [Jeng et al., 1998] use anthropometric measures for feature detection. 2-D Gabor filters with different orientation and spatial frequencies are used to detect the eyes by searching in the sampling grid over the input image [Smeraldi et al., 2000]. [Hsu et al., 2002] and [Cooray and O'Connor, 2004] use color information in constructing eye- and mouth-maps to detect facial features. *Constellation analysis* allows a higher degree of flexibility in detecting facial features due to complex backgrounds and other disturbing elements in the image [Yow and Cipolla, 1997].

3.2.2.2 Image-Based Approaches

Due to the unpredictability of the appearance of faces in images and environmental conditions, using feature searching for face detection sometimes proves to be too restrictive. Image-based approaches use machine-learning techniques to find face and non-face classes within a training set and use these classes as a basis for discrimination. These methods usually apply a window-based scanning of the image at different resolutions with varying scale and scanning step, thus they are computationally expensive. Image-based methods for face detection can be divided into a number of subcategories: linear subspace methods, neural networks and statistical approaches.

Linear subspace methods consider that the human face forms a subspace or a cluster in the image space. Principal component analysis (PCA), factor analysis (FA) are techniques for efficient representation of the face images in the image space. PCA employs eigenvectors as basis vectors for face representation; the eigenvalues yield the principal components of the distribution of faces; each face then can be expressed in the form of a linear combination of the largest eigenvectors (corresponding to the largest eigenvalues). This method is described in Section 3.3.3.

Neural networks have been used for face detection [Rowley et al., 1998]. An enhanced version of this system uses a router neural network to perform detection of rotated faces irrespective of the angle in the image.

In the case of video, a helpful clue for the face detection task can be found in motion-based features as means of identifying moving objects. Motion segmentation through frame difference analysis can distinguish foreground objects against background. [Low and Ibrahim, 1997] compute frame difference to extract facial features. In [Crowley and Berard, 1997] the eye-pair is deducted from the horizontal and vertical displacement between moving regions in consecutive images.

Perhaps the most impressive approach to face detection is the one reported by [Viola and Jones, 2004]. It is a system capable to perform face detection almost in "real-time". An "integral image" is introduced to quickly compute features then in the second stage a cascade of classifiers eliminate background regions while focusing more on promising "face-like" regions.

3.2.3 Face Detection Using Colour Information

Colour information is now generally available for most video content. This brings a helpful clue for automatic face detection systems which deal with speed requirements and a large volume of data to be processed. The main idea behind colour-based face detection lies in using skin-tone filtering. It has been shown that the distribution of skin colour across different ethnic groups, under good lighting conditions is compact. There are advantages and disadvantages to using colour for face detection applications. A strong point which makes this method suitable for the case of video programs is that colour filtering is easy to implement and the results are effective in controlled environments. Robustness is achieved if a suitable colourspace is used which separates luminance from chrominance. Such a space is also suitable for skin colour distribution and thresholding. This also overcomes sensitiveness to pose, rotation, shading, facial expression and cluttered background. Problems using colour as a primary cue in face detection arise when the image/video content is captured in unconstrained environments. Noisy detection results and false skin-coloured objects/regions (body parts) are another inconvenience associated with this category of approaches.

3.2.3.1 Skin Structure

The human skin has a layered structure and its particular colour is given by the combination of blood (red) and melanin (yellow) (Figure 3.1 [1]). The absorbtion of light by the melanin and haemoglobin gives the skin spectra. It has higher reflectance for long wavelengths (red, orange) than for short ones (blue) as reported early in [Edwards and Duntley, 1939].

Broadly speaking, skin pixel detection methods rely on:

- non-parametric modelling (e.g. direct threshold, histogram-based)
- parametric modelling (Gaussian or mixture of Gaussians)

One of the easiest ways of discriminating skin from non-skin pixels is to use decision rules on the pixel values. A successfully used colour model for skin-tone detection

[1] Image available at http://www.homestead.com/doctorderm/skinanatomy.html

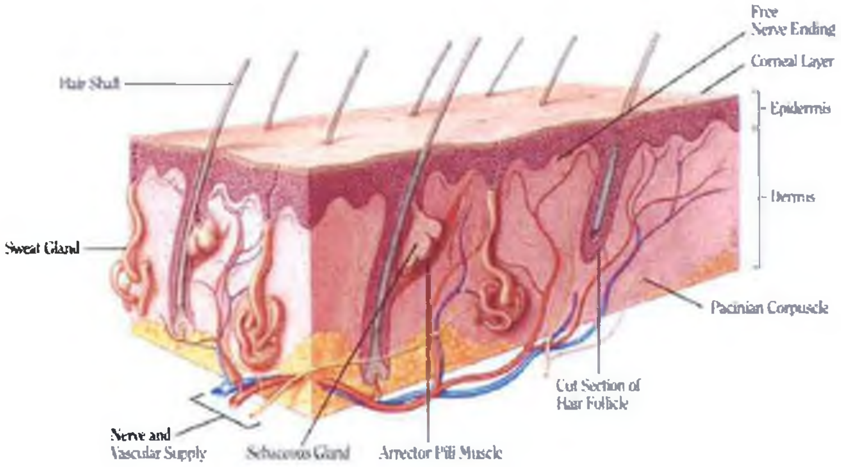


Figure 3.1: Skin structure

is the RGB representation. Given that skin coloured samples don't cluster properly in the direct RGB representation and because skin appearance is influenced mainly by chrominance, a normalized RGB representation is typically used [H.P. Graf and Petajan, 1996, J.Yang and A.Waibel, 1996]. The normalized *r, g, b* are derived from the *R, G, B* components as:

$$r = \frac{R}{R+G+B}$$
$$g = \frac{G}{R+G+B}$$
$$b = \frac{B}{R+G+B}$$

Usually the r and g components are used for discrimination of skin-like pixels in an image. Approaches using *HSV (HSI)* have been widely reported [C.H. Lee and Park, 1996, D.Saxe and R.Foulds, 1996, R.Kjeldsen and J.Kender, 1996]. Using HSV colorspace representation has also advantages in detecting facial features such as lips, eyes and eyebrows if this is required.

3.2.3.2 Parametric Skin Modeling

A. Single Gaussian Model

Skin color can be modeled by an elliptical Gaussian Probability Distribution Function (PDF) [Terrillon and Akamatsu, 2000]:

$$f(x|skin) = \frac{1}{2\pi|\Sigma_s|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_s)^T\Sigma_s^{-1}(x-\mu_s)\right\} \tag{3.2.1}$$

The model parameters can be estimated using a training skin database:

$$\begin{aligned}\mu_s &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ \Sigma_s &= \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_s)^T(x_i-\mu_s)\end{aligned}$$

To estimate how close a pixel is to the skin model the Mahalanobis distance is usually computed:

$$d^2 = (x-\mu_s)^T\Sigma_s^{-1}(x-\mu_s) \tag{3.2.2}$$

B. Mixture of Gaussians

Alternatively, skin colour can be modeled using a mixture of Gaussians. The mixture density function is expressed as a sum of Gaussian kernels:

$$P(x) = \sum_{i=1}^{N}\left(w_i \frac{1}{2\pi|\Sigma|_i^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right\}\right) \tag{3.2.3}$$

[Phung et al., 2002]. The number of components in the Gaussian mixture can be supplied or it can be determined using an optimization criteria (e.g. the Minimum Description Length). After the densities are determined, an unknown skin pixel is labeled as skin or non-skin using a Bayesian classifier.

3.2.3.3 White Balance Technique for Color Correction

A technique which performs colour compensation for different lighting conditions in an unconstrained environment is known as white balance colour correction. This technique uses the reference white as the brightest luminance value in an image where N percent of those pixels have the reference grayscale value [Hsu et al., 2002]. If the above criteria is met the colour components are then updated such that the average grayscale value of the reference-white pixels in the image are rescaled to 255.

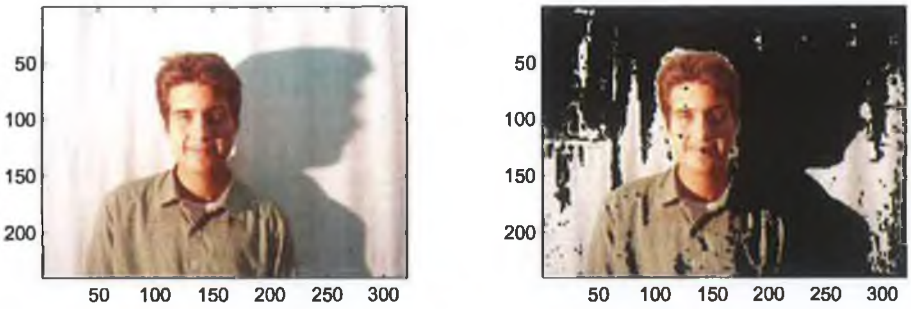


Figure 3.2: Sample skin-colour filtering without colour correction

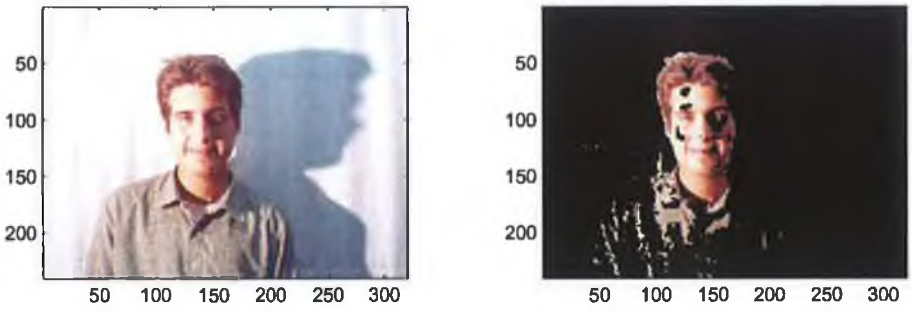


Figure 3.3: Sample skin-colour filtering with colour correction

The technique is described in full detail in [Hsu et al., 2002]. The gray values of the top N percent of the brightest pixels are averaged and each colour component of the remaining pixels are computed as:

$$\begin{aligned} R_{trans} &= R + (255 - mean_R) && (3.2.4) \\ G_{trans} &= G + (255 - mean_G) && (3.2.5) \\ B_{trans} &= B + (255 - mean_B) && (3.2.6) \end{aligned}$$

Figure 3.2 and 3.4 show sample images from the MPEG-7 test set skin-tone filtered without colour correction whilst in Figure 3.3 and 3.5 are the skin detection results which underwent colour-correction apriori are presented.

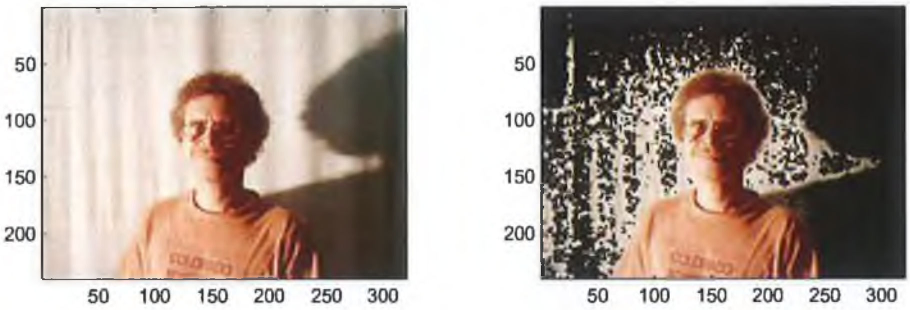


Figure 3.4: Sample skin-colour filtering without colour correction

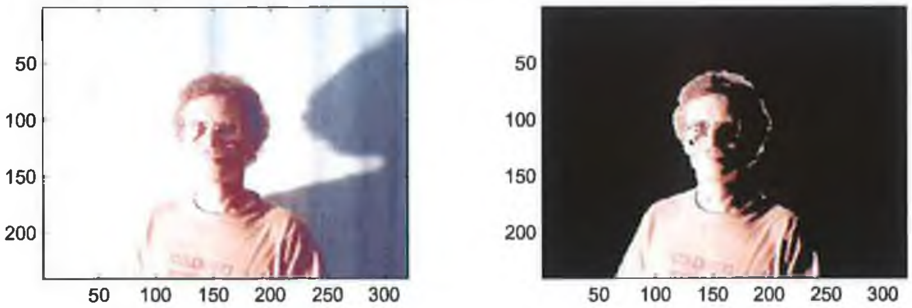


Figure 3.5: Sample skin-colour filtering with colour correction

3.3 Proposed Face Detection System

In our approach to face detection, we use colour information as the primary tool for locating face regions. The processing chain involves a sequential pruning of candidate regions until all criteria are satisfied. In general, skin colour based face detection algorithms are computationally modest, but are sensitive to noise, lighting conditions and the colour content of the objects present in the scene. The face detection algorithm [Czirjek et al., 2003] is illustrated in Figure 3.6 and described in the followings.

Given an image, the algorithm operates at region level. After skin-colour filtering (with optional colour correction techniques applied), connected components are filtered in size and shape and are grouped into regions. Skin-tone detection is only a preprocessing step and is aimed at reducing the search space for possible face-like regions. Then the main algorithm examines these regions and performs splitting and verification of the connected components until certain criteria are met. A region which does not satisfy the compactness constraint is split into a number of disjoint regions

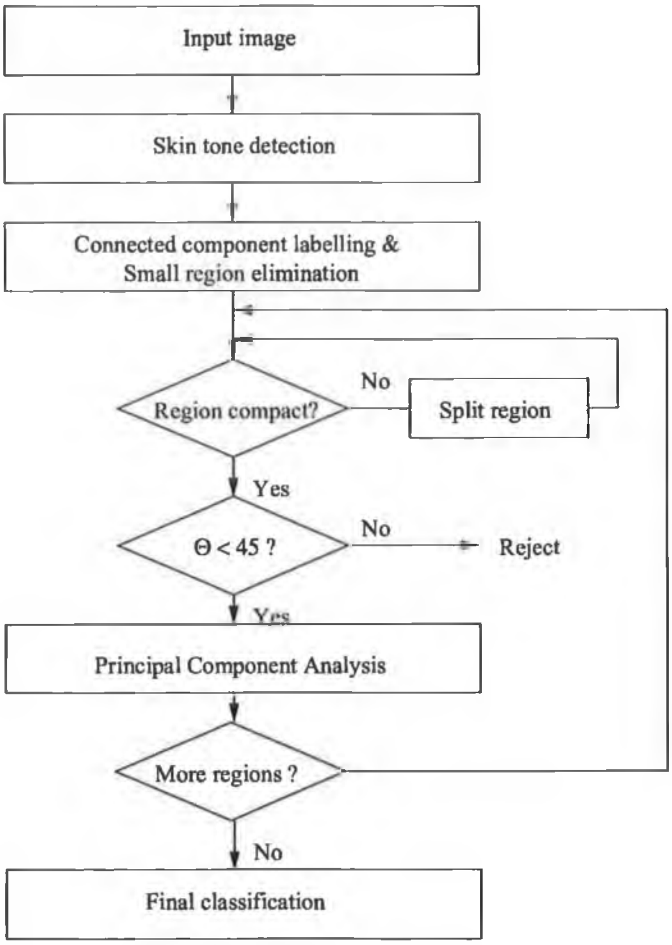


Figure 3.6: The Face detection algorithm

which in turn will become new candidates to be verified. We introduced a relaxed condition on the orientation angle (Θ) the region should have; this is motivated by the fact that we are dealing with candidate, perhaps split regions and we do not want to rely solely on a strict splitting procedure before classifying the candidate region. These candidates are then classified into face- and non-face regions according to the minimum error to the face space constructed using Principal Component Analysis.

3.3.1 Morphological Filtering

Because of the nature of the classification used, the output of skin tone filtering - a skin-mask - will be populated with many isolated pixels. To eliminate this undesirable effect, we apply a morphological opening and closing with a square kernel of $N \times N$ (experimental results have indicated a suitable value of $N = 5$). After filtering, we obtain smoothed homogeneous areas of connected pixels. Connected component labeling is then performed, which gives the number of regions used in the next stages of the algorithm.

3.3.2 Skin Region Processing

Even after applying morphological filtering to the skin-map, regions with a small number of pixels may be present. In order to reduce the number of false candidate regions, areas less than 625 pixels are ignored. We have chosen this threshold based on the assumption that a face with size smaller than 25 × 25 pixels should not be detected by our approach. Horizontal and vertical strips, which are less likely to contain a human face, are also ignored. These regions are detected by identifying a significant difference between the width and height of the region's bounding box, with the condition that the smaller dimension does not exceed 25 pixels.

It is possible that other objects in a visual scene have similar colour characteristics to human skin, or that other objects are merged with the face (e.g. hands, back-ground, etc). In general, when dealing with skin-colour based face detection, the following scenarios can occur:

1. face candidate region is separated (not connected to other objects)
2. a face region is merged with other objects due to their similar colour characteristics
3. a candidate region is a false alarm
4. a candidate region is split into unconnected regions which are not valid for detection by themselves, as they should be part of the same entity. This situation usually occurs when there is a close-up of a face in the scene
5. a candidate region is not detected due to insufficient colour information (black/white movies or poor lighting conditions)

In our approach candidates belonging to scenarios 1 and 3 are handled by the principal component analysis module outlined below. We do not address scenarios 4 or 5 in this work. In order to address scenario 2, where the face is merged with other parts of the scene, we perform an iterative splitting procedure on the connected component. To this end, we introduce a measure of region compactness as the ratio between the area of the connected component and its bounding box. This value indicates the degree of convexity of a region.

The compactness of a candidate region S signals if it should be partitioned into sub-regions or not. If the ratio falls below a threshold, k disjoint sub-regions are formed by maximizing the compactness of each subsequent sub-region:

$$S_i i = 1, \ldots k, \bigcup_{i}^{K} S_k = S \tag{3.3.1}$$

It is known that the aspect of a "perfect" face is close to the "golden ratio" [Frakas and Munro, 1987]. Therefore, we divide regions which deviate from this value into sub-regions so that the ratio between height and width approaches the ideal whilst the maximum compactness constraint is obeyed. If the width of the bounding box is greater than the height the splitting procedure operates from top to bottom, otherwise it propagates horizontally. An illustrative example is a head-and-shoulder image, commonly found in news anchorperson shots, where the head is merged with the clothes due to similar color (e.g. a yellow jacket). In this case, region partitioning will segregate the head area of the body. Assuming that the human face has an elliptical shape, for each compact region the best-fit ellipse is calculated based on moments [Jain, 1989]. The orientation angle of the ellipse is given by:

$$\theta = \frac{1}{2} arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) \tag{3.3.2}$$

where $\mu_{i,j}$ denote the central moments of the connected component. If the orientation of the major axis falls in the range $\theta \in [-45^o, 45^o]$ the region is selected for the classification stage, otherwise it is rejected. Selected regions are extracted, reverse tilted if $\theta < 0$, rescaled to 32×32 pixels, histogram equalized and passed to the principal component analysis module for final classification.

3.3.3 Principal Component Analysis

Using a collection of test images, a face space is constructed for discriminating the candidate face regions. Given a set of vectors x, where each vector is an image with rows in lexicographic ordering, the basis vectors can be computed by solving the eigenvalue problem:

$$\Lambda = P^T \Sigma P \tag{3.3.3}$$

where Σ is the covariance matrix of x and P the eigenvectors matrix. The extracted regions are normalized, rescaled and then arranged into a vector x. The principal component vector is obtained by projecting x into the face space spanned by the eigenfaces:

$$y = P_M^T (x - \overline{x}) \tag{3.3.4}$$

where is the mean image of the training set x. The measure of "faceness" of the input sample relies on the reconstruction error, expressed as the difference between the input image and its reconstruction using only N eigenvectors corresponding to the highest eigenvalues [Moghaddam and Pentland, 1997, Turk and Pentland, 1991]. This is termed the distance from face space (DFFS):

$$\epsilon^2 = ||x - \overline{x}||^2 \tag{3.3.5}$$

The distance between the projected input image and the mean face image in the feature space is given by the norm of the principal component vector. Since the variance of a principal component vector is given by its associated eigenvalue , the squared Mahalanobis distance measure gives a measure of the difference between the projection of the test image and the mean face image (see Figure 3.7) of the training set x:

$$d^2 = \sum_{i=1}^{N} \frac{y_i}{\lambda_i} \tag{3.3.6}$$

where are the projection coefficients and are the associated eigenvalues. Therefore can be expressed as the distance in face space (DIFS). Given these two distances a combined error criterion is calculated:

$$e = d^2 + c \cdot \epsilon^2 \tag{3.3.7}$$

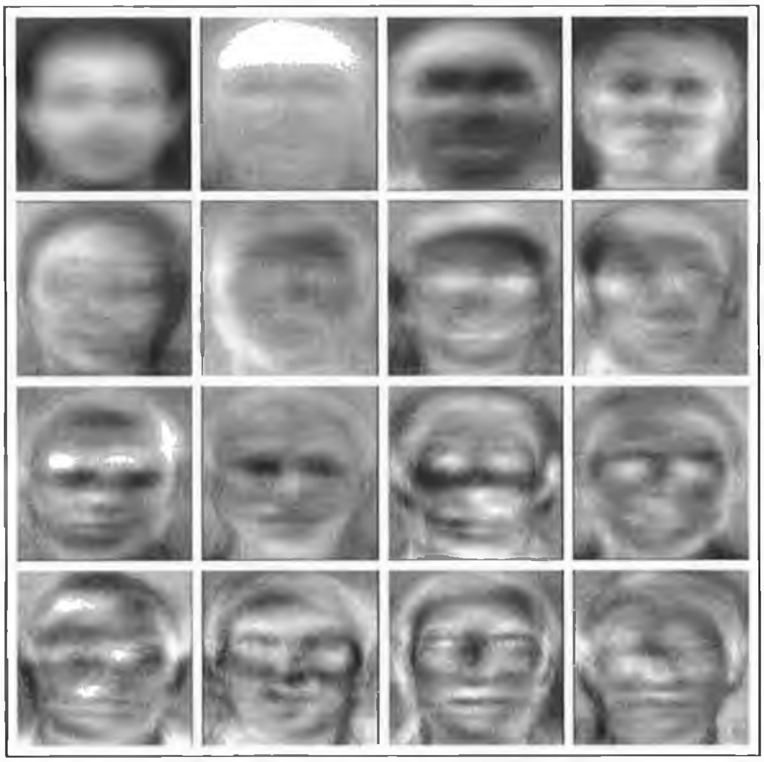


Figure 3.7: Mean face and Eigenfaces (0-14) computed for the ORL face database

with c chosen to be a suitable constant value. Because PCA is sensitive to scale and rotation, a Gaussian multiresolution pyramid could be used to reduce the number of missed positive candidates, to the detriment of computation time.

3.4 Face Image Clustering in Video Sequences

Given a set of faces extracted from video frames, we address the problem of combining these individual face samples into a number of disjoint clusters. The topic of incremental face sequence matching has been addressed in the literature [Fitzgibbon and Zisserman, 2002, Raytchev and Murase, 2003, Satoh, 1999, S.Satoh and Katayama, 2000]. In [Satoh, 1999] a semi-automatic annotation system is first employed for training. For this, face sequences are extracted and aggregated using the underlying structure of drama video sequences. The face sets are then clustered using an eigenface-based method. The performances of different face sequence matching techniques are evaluated on actual drama content in [S.Satoh and Katayama, 2000]. A novel method for incremental learning is described in [Raytchev and Murase, 2003] where unsupervised recognition is accomplished by constructing a graph in which similar views are chained in the image space depending on local similarity measures.

Our clustering process is based on the *individual* PCA approach. In this approach, an eigenspace is constructed for each subject in the training set. In contrast to *universal* PCA where the space represents inter-variations of subjects and also intra-variations across different views of the same subject, in individual PCA, each subject has a characteristic eigenspace. Using the dissimilarity measures to each individual eigenspace correlated with spatio-temporal information, we attempt to identify new face sequences.

The processing chain starts by constructing N individual eigenspaces for extracted and manually classified faces corresponding to specific characters commonly observed in the test corpus programmes. These face spaces will play the role of reference or primary data-bases when comparing new acquired candidates. Within the programmes in our test corpus, only a small number of main persons usually appear so we considered it useful to form databases of these characters. Each extracted face candidate is a 32×32 grayscale image thus each data sample maps to a point in a 32×32 dimensional space. For each considered character the PCA algorithm described in section 3.3.3 is applied resulting in N eigenspaces.

3.4.1 Face Sequence Extraction

For each continuous video shot a sequence of faces corresponding to the same person are extracted. In this context, an important factor is the location of the extracted candidate in the frame. Generally in news video which is our primary test corpus, the person's position doesn't change drastically during an anchorperson shot or interview. We use a simple tracking method between adjacent I-frames, which examines the overlapping of the bounding boxes of the extracted faces. If the faces occur almost at the same position and more than 30% of the area overlaps they are considered to belong to the same person.

Potential face sequences are likely to exhibit similar dissimilarity values to the reference databases. If the number of candidates in a shot is higher than 15 we analyze the DFFS dissimilarity curve in order to establish the occurrence of a new face sequence. We have chosen this lower limit based on the assumption that an interview or any other significant part of a news program should not be shorter than 6 seconds.

3.4.2 New Face Sequence Classification

If the variance of the dissimilarity measures across a shot satisfies our constraints, these samples are regarded as a new sequence and they form a new face (character) database upon which PCA is performed. Since determining the eigenvectors is computationally intensive, the power method [Horn and Johnson, 1985] is used to efficiently calculate the dominant eigenvectors. Typical dissimilarity curves are illustrated in Figure 3.8. We can see that for the face sequence in shots #1 and #3 the dissimilarity values to each face database remain relatively the same and present a strong correlation.

It can be observed that between shot boundary changes the dissimilarity to each database remains relatively constant (discounting noise). For an anchorperson shot from the reference databases the DFFS to the correct face class exhibits an accentuated variation, whereas this is not the case for the other eigenspaces. This reflects the changes in expression and pose of the individual in that class.

If the dissimilarity measures fall within suitable thresholds for the duration of the shot and the number of samples exceeds a minimal value, then a face cluster is established. Mathematically, the condition is:

$$\frac{1}{M-1}\sum_{i=1}^{M-1}|\delta_{i,j}-\delta_{i+1,j}| < T_j \tag{3.4.1}$$

where $\delta_{i,j}$ represents the distance of sample i to eigenspace j, M denotes the number of samples across the shot and T_j a threshold for database j.

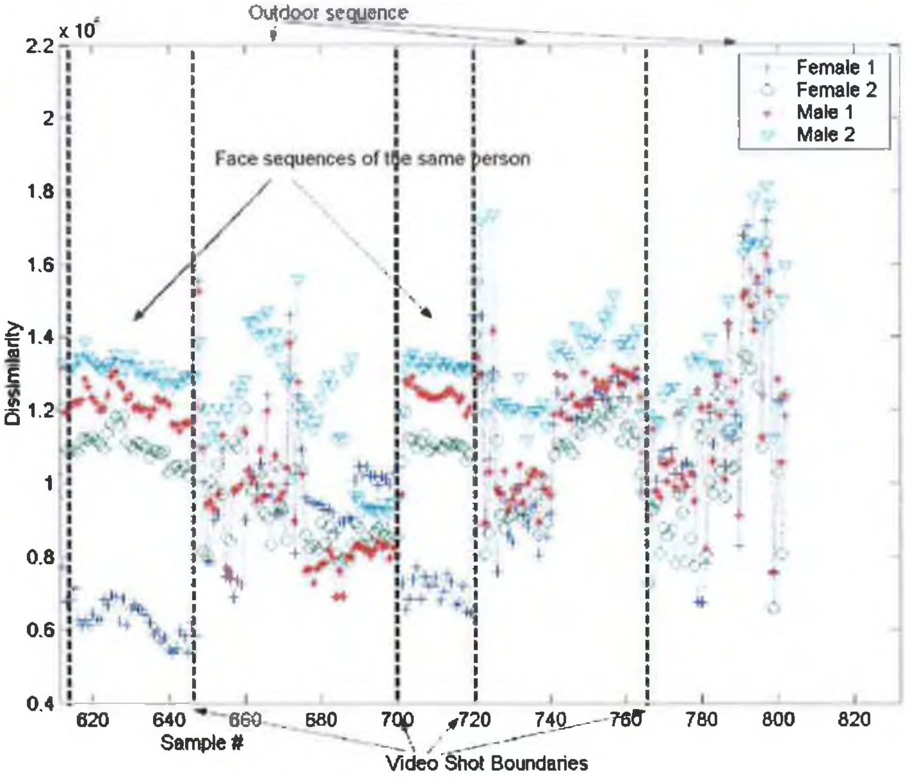


Figure 3.8: Face dissimilarity curves relative to four face databases across a number of shots

3.5 Experimental Validation

The results of the face detection algorithm on a number of selected Irish news broadcasts from the *Fischlár* [Dublin City University Ireland] news archive are summarized in Table 3.1, whereas the face clustering results are presented in Table 3.2. In calculating the ground truth used to evaluate the results obtained for our face detection algorithm we visually inspected each shot in the test corpus and recorded the number of faces present. The results presented in Table 3.1 are very encouraging with an average precision value of 71.50%

News ID	Extracted candidates	False alarms	Precision
06/09/02	2254	386	82 %
12/09/02	2501	523	79 %
15/09/02	1611	473	70 %
17/09/02	1574	510	67 %
19/09/02	1192	385	67 %
22/09/02	1303	489	62 %
23/09/02	1408	412	70 %
24/09/02	2715	677	75 %

Table 3.1: Face detection results

News ID	No faces	Real seq.	Detected seq.	Precision	No. non-faces in seq.
06/09/02	2254	18	27	0.66 %	17
12/09/02	2501	21	35	0.60 %	21
15/09/02	1161	20	28	0.71 %	15
17/09/02	1574	17	32	0.53 %	14
19/09/02	1192	26	30	0.86 %	19
22/09/02	1303	23	25	0.92 %	11
23/09/02	1408	15	28	0.53 %	23
24/09/02	2715	28	37	0.75 %	12

Table 3.2: Face clustering evaluation

In calculating the ground truth used to evaluate the results obtained in our face clustering experiments, we considered a ”real” face sequence as a shot where a person appears for at least 6 seconds without significant occlusions. This scenario is commonly found in outdoor reporting/studio interviews. From Table 3.2, it can be seen that the number of face sequences identified is higher than the actual sequences appearing during the broadcast. The reason for this is that consistently false sequences are also detected as real face classes because of successful tracking and low PCA error residual. It should be noted however, that the misclassification percentage is quite low. The number of newly formed face classes is not high considering that new sporadic faces are common in news broadcasts which typically contain a mixture of indoor (clean) and outdoor (noisy) segments.

From a computational point of view, for a 30 minutes news programme the face detection algorithm takes approx. 20 minutes (60% real time processing) on a Dual

Pentium III 700 MHz running Red Hat Linux 7.3, whereas the face clustering algorithm is much less computationally expensive, requiring only a few minutes execution time. Screenshots of the application GUI used for detection are presented in Figure 3.9 and the face clustering application is presented in Figure 3.10.

3.6 Summary

This chapter a review on face detection techniques was presented along with an iterative colour-based face detection and clustering method for video sequences. The clustering process facilitates automatically detecting new characters not in the reference face databases, but also enables the detection of other entities (e.g. coverage of parliamentary proceedings or major cast occurrence). This provides higher level information to multimodal indexing tools by characterizing the content in terms of human presence and reoccurrence. The face detection technique could be significantly improved using facial feature extraction for face verification, and outliers could be eliminated exploiting the temporal redundancy within consequent video frames using techniques such as Kalman filtering. The presented approach cannot deal with significant changes in ambient settings due to the fact that the underlying PCA is sensitive to scale, rotation and changing lighting conditions.

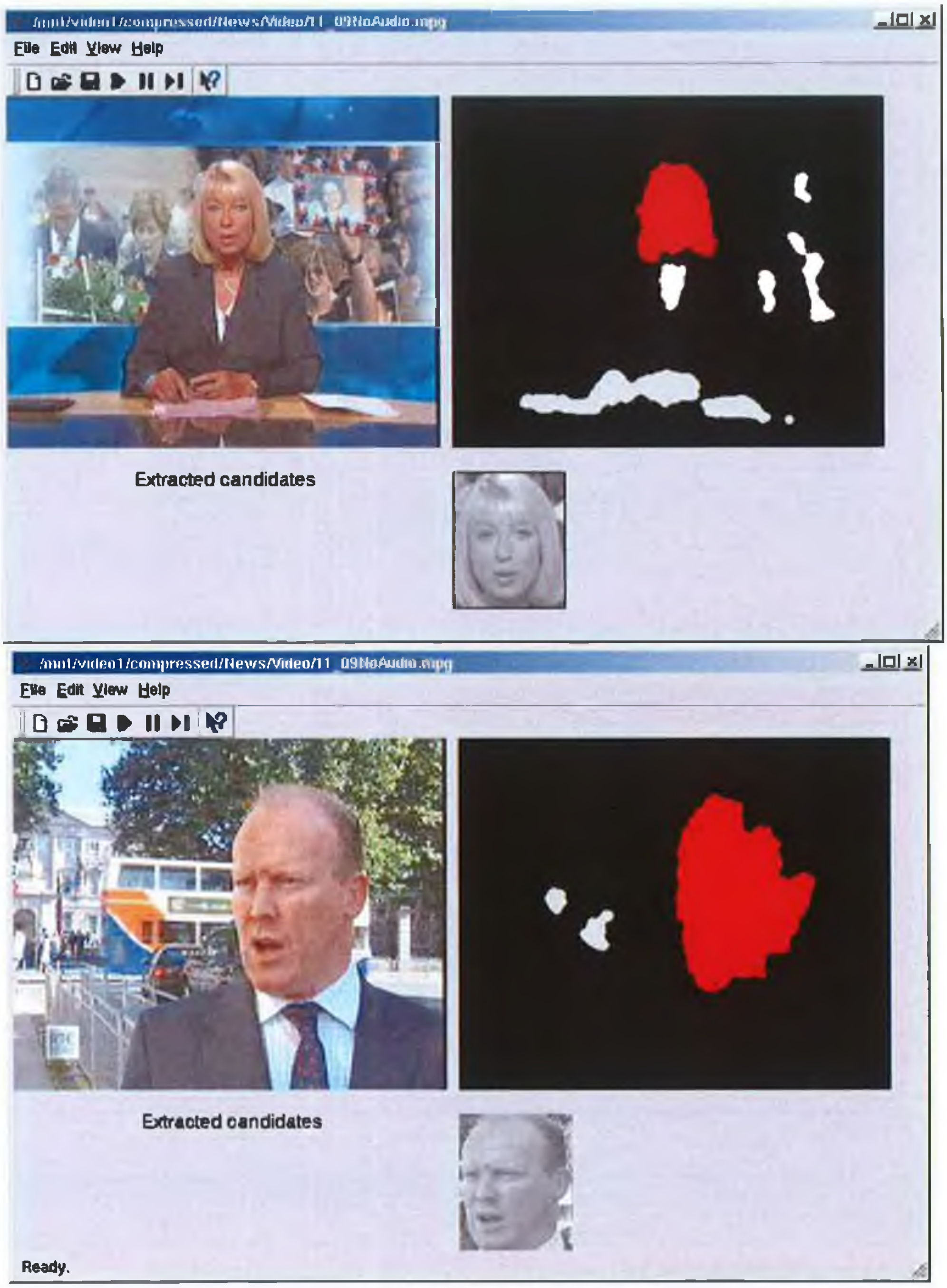

/mnt/video1/compressed/News/Video/11_09NoAudio.mpg
File Edit View Help
Extracted candidates
/mnt/video1/compressed/News/Video/11_09NoAudio.mpg
File Edit View Help
Extracted candidates
Ready.

Figure 3.9: Face detection application GUI

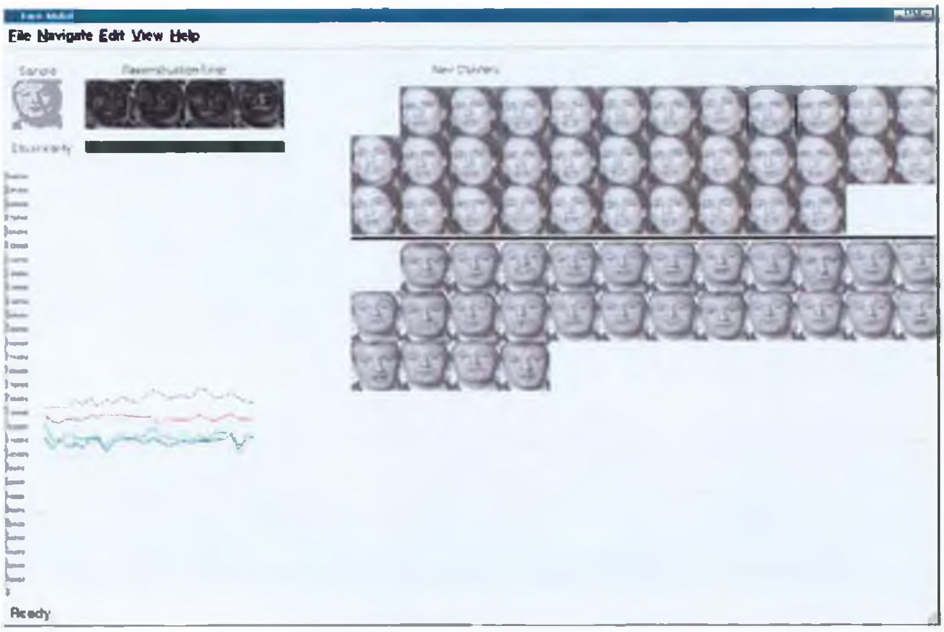


Figure 3.10: Face clustering application GUI

Chapter 4

Anchorperson Detection in News

In this chapter we present a framework which aims to segment raw news video broadcasts into semantically meaningful segments. This is achieved by detecting the main/major cast members which in this case correspond to the anchorperson. The low-level visual analysis which constitutes the main process in the proposed video segmentation scheme can be carried out for any given genre; however in order to extract higher level semantic entities, we consider genre specific production rules specific to news programmes. After presenting the related approaches to news broadcast indexing and the general structure of news programmes, we present our approach to the news story segmentation task.

4.1 Existing News Video Libraries

There has been much work in the research community addressing the topic of News Story Segmentation. In this section we present existing news video libraries followed by approaches to story segmentation.

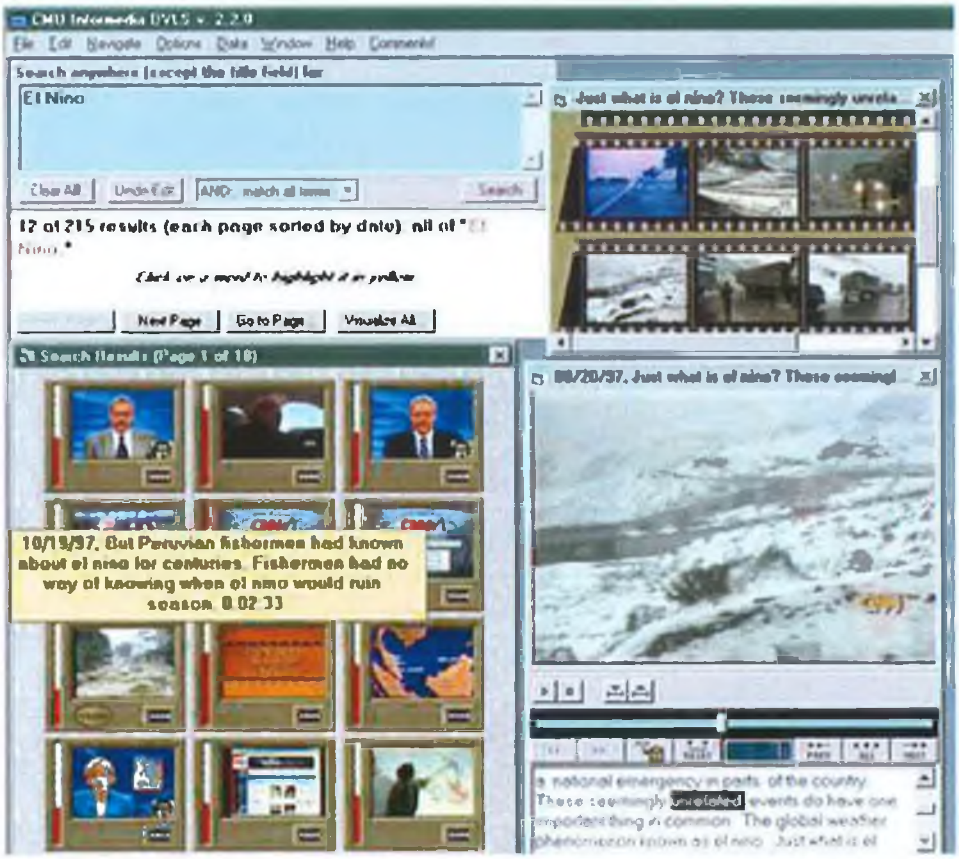
Perhaps one of the most successful initiatives comes from Carnegie Mellon University who developed the *Informedia* Digital Library System [Christel et al., 2002, Christel, 1999, Wactlar et al., 1999, 2000, 1996]. Started in 1994, the Informedia project aimed to explore novel dimensions in video indexing, search and retrieval. It makes use of a rich source of information to achieve the desired goal: a mixture of speech, image and language processing techniques are deployed to segment and classify news stories by topic. Up to October 1999 the system had over 1,600 indexed news broadcasts containing 40,000 individual news stories [Wactlar et al., 2000]. Once the broadcast segmentation is done, each news story is summarized at visual and textual level. Thumbnail key-frames and video skims accompany the textual description consisting of a topic and title.

At the visual level, image processing techniques determine shot/scene boundaries, camera motion detection (still, zoom, pan). Higher level analysis employs face detection, tracking and recognition in conjunction with open caption detection and recognition (OCR) and intelligent name extraction (using dictionary, thesaurus and parser) in order to associate names to faces [Satoh et al., 1999] detected at various points in the video. Transcripts generated by applying speech recognition to the audio track and closed-captions available from the broadcaster (CNN) that accompany each news story give a textual description, used as an information source to answer text-based queries. Figure 4.1 depicts the Informedia II search interface[1]. There is also an interesting and very helpful tool used especially for story search: *Name-It* [Satoh et al., 1999]. The goal of Name-It is to associate faces with names in news video. This process has three stages: face tracking, face identification and face-name association.

Face tracking also consists of three steps: face detection [Rowley et al., 1998] is employed first to extract candidate faces at a sampling interval of 10 frames. For each extracted candidate, skin regions are extracted based on skin colour modelling which are used for tracking the face(s) in time through the shot. Tracking is done by evaluating the degree of overlap of the extracted regions in successive frames. To perform the identification task, the most frontal face determined from the previous stages is used as input to an eigenface-based recognition method [Turk and Pentland, 1991]. Then Name-It locates and associates the corresponding character name from the open- and closed-captions.

The initial story segmentation algorithm is described in [Hauptman and Witbrok,

[1] Image available at www.informedia.cs.cmu.edu


CMU Informedia DVLS v. 2.2.0
File Edit Navigate Options Data Window Help Comments!
Search anywhere (except the title field) for
Clear All
Undo Edit
AND: match all terms
Search
Next Page
Go to Page
Visualize All
Search Results (Page 1 of 10)

El Nino

17 of 215 results (each page sorted by date) all of "El Nino"

Click on a word to highlight it in yellow

10/19/97. But Peruvian fishermen had known about el nino for centuries. Fishermen had no way of knowing when el nino would ruin season. 0:02:33

Just what is el nino? These seemingly unrela...

08/20/97. Just what is el nino? These seemingl

a national emergency in parts of the country. These seemingly unrelated events do have one important thing in common. The global weather phenomenon known as el nino. Just what is el

Figure 4.1: The Informedia II search interface

1998]. After shot boundary detection using colour histogram analysis, optical flow analysis on the MPEG bitstream and undesired commercial segment removal by detecting silent segments, black frames and analyzing the shot-cut-rate, the closed-captions and the text resulting from the speech recognition engine are aligned. This is a necessary step given that the closed captions hold important cues for determining the story boundaries. However, due to delays in typing or missing words from the transcript, the ASR is used as reference point, given it's precise timing information associated to spoken words. The *Dynamic Time Warping algorithm* is used to synchronize the two streams and then the content of ASR is inserted according to the time-reference points. The closed captions hold important special characters (">>>") which signals the beginning of a new news story and speaker turns as well

(">>"). If there are three or more lines missing from the teletext transcript a potential story boundary is also signalled, unless it belongs to a commercial segment.

However, recently Informedia underwent a major change in the way it segmented news broadcasts [Hauptmann et al., 2003]. The previously presented, mainly-text segmentation approach has been developed into a multimodal Support Vector Machine-based framework. In [Lin and Hauptmann, 2002] an SVM has been used to identify the weather reports. The TREC[2] Video Track in 2003 included a story segmentation task. The complete description of the system which makes use of two separate classifiers for anchorperson- and commercial shots can be found in the TREC proceedings [Hauptmann et al., 2003]. Two separate support vector machines are trained for each separate task, using colour (HSV histogram, texture and edge histogram), face information (size, position and confidence) and audio information (speaker ID, Short Time Fourier Transform) on a development corpus. It is worth mentioning that Informedia accepts text- along with image-based queries. Based on language processing techniques to classify topics within a broadcast, named entity extraction and georeferencing, the system provides searching of the video archive but also has interfaces that make use of geographical information to display news items on a map and allows exploration of news items based on geographical location.

Another interesting system is *ANSES*[3]: Automatic News Summarization Extraction System developed at Imperial College, London, UK [Pickering et al., 2003]. ANSES provides users with a daily digest of news stories gathered from BBC evening news. Whilst capturing the teletext information available from the content provider, it carries out text based story summarization using lexical chains similar to [Barzilay and Elhadad, 1997]. The processing chain is typical for this type of video analysis: (histogram-based) shot cut detection, keyframe extraction for shot representation and summarization, followed by the presentation interface designed for navigation, querying and playback.

[2] TREC stands for Text REtrieval Conference, sponsored by NIST (National Institute for Standards and Technologies) and is an annually held workshop to benchmark information retrieval systems.

[3] Demonstration system available at www.doc.ic.ac.uk/~mjp3/anses/

The *VISION* project developed at the University of Kansas [Gauch et al., 2000] addresses a broader type of content than just news broadcasts, however the goal is to provide a prototype digital video library for access over the internet to automatically indexed programmes. It uses a video shot segmentation method combined with an audio merging algorithm (based on endpoints detection and speaker identification) to determine logical stories. This proves to be a successful technique for scene construction given that (e.g. in news programmes) whilst the anchorperson narrates the storyline the video is composed of clips about the same subject.

One of the leading commercial companies providing media content management for corporations, media and entertainment companies is *Virage*®Inc [4]. From their suite of products and solutions Virage VS News Monitoring offers the advantages of automatic analysis, segmentation, personalization and hyper-linking of news items. Key issues are real-time monitoring and information access, alerting capabilities and real-time video segmentation and clip classification into story or commercial categories. The Intelligent Data Operating Layer (IDOL) server capabilities include retrieval, hyperlinking, categorization, alerting, profiling, clustering and personalization. Other technologies integrated include Dremedia™and SoftSound™for scene change detection, transcript alignment, and advanced audio and speech analysis.

VideoLogger [5] is a software product for automatic indexing of video broadcasts. In addition to automatic analysis of incoming video, Video Logger also performs automatic indexing from external timecoded sources such as sports feeds, transcripts, GPS data, existing database logs and allows manual annotations to be added, in real time or as a post process. A snapshot of Virage VS News Monitoring can be seen in Figure 4.2.

In the Físchlár [6] video indexing system shot boundaries are determined on the basis of colour-histogram comparison [O'Toole et al., 1999]. A 64 bin histogram is computed for the Y, U and V components of each decoded frame. These histograms are

[4] More information available at www.virage.com.

[5] www.virage.com.

[6] www.cdvp.dcu.ie

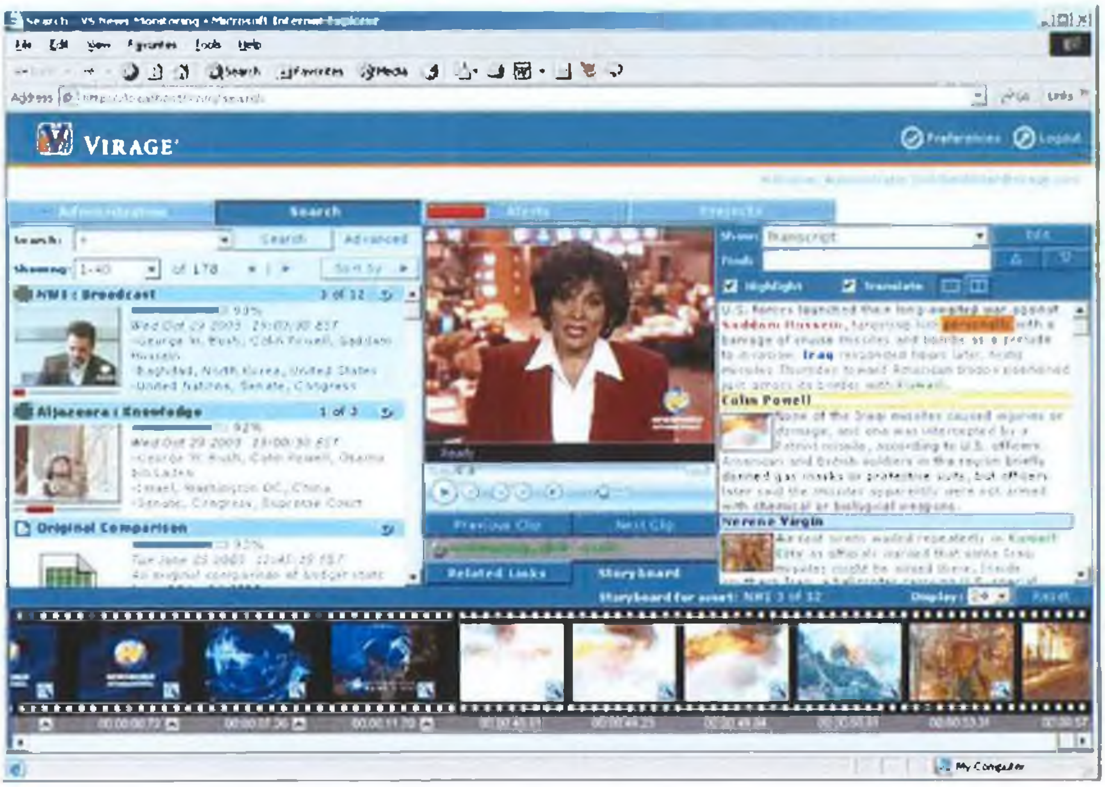


Figure 4.2: Virage's News Monitoring user interface

then concatenated into a single 192 point signature vector for the frame. The difference between successive frames is calculated using the cosine similarity measure. A dynamic thresholding technique is applied to these similarities in order to detect hard-cuts and gradual transitions.

The Físchlár News browsing system is an example of a platform designed for real-time, real-audience as described in [O'Hare et al., 2004]. Using the Físchlár News system, the user has the choice to select the day of broadcast through a calendar in the left panel which will bring up a summarized version of the news recording consisting of the anchorperson with a representative image in the background and also the first sentence of the associated teletext transcript for that story. When selecting a story the user is presented with a detailed view of the story represented as key-frame thumbnails. Clicking on any image the user will be presented with the video of the news item. Using ratings, a personalized profile will be constructed for

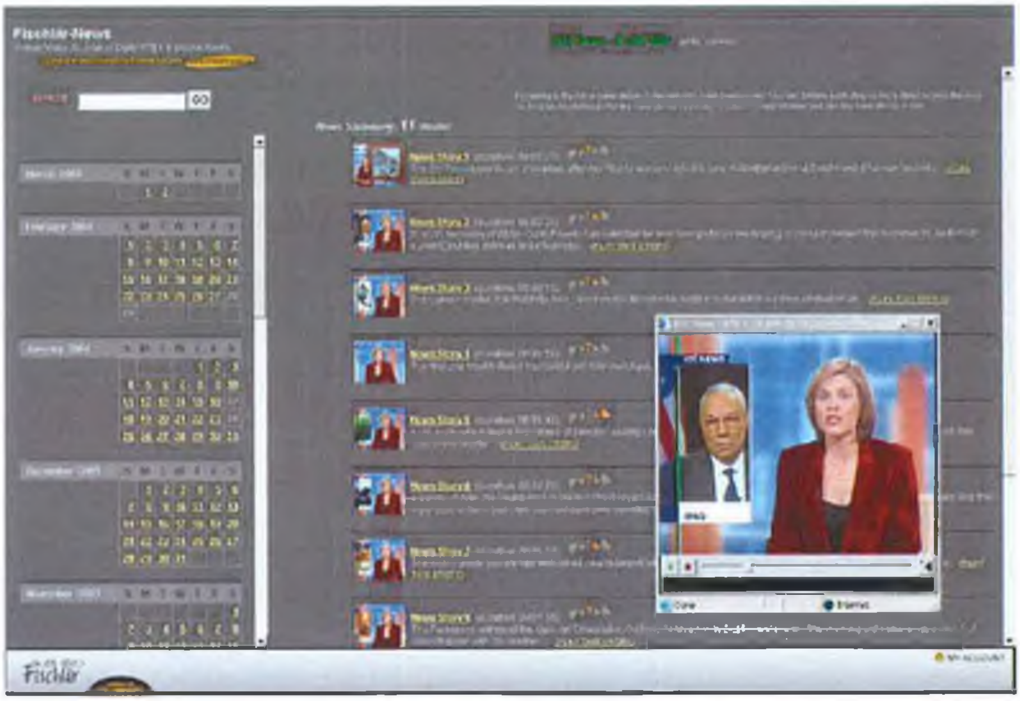


Figure 4.3: The Físchlár News Environment

each user which also provide links to favorite seen or missed broadcasts or stories. The personalisation engine is powered by the ClixSmart system [Smyth and Cotter, 2000].

A snapshot of the Físchlár News browsing environment is depicted in Figure 4.3.

4.2 Structure of a News Broadcast

News broadcasts are media productions which adhere to a certain structure. Unlike other genres, news are easier to analyze due to their underlying composition [Fischer et al., 1995]. A news programme is a collage of stories, each story being presented by a news reader or anchorperson and may be followed by reporting, interview, outdoor footage etc. with more details on the story. However, in most cases stories are not related within the same programme but they share the common fact that they are introduced by a person. There may be more anchorpersons present within the same news programme and they might present alternate stories or both present the same story. It is usual that the programme is interrupted by commercial segments during the broadcast. The news broadcast will end after the sport summaries (or financial reports) with the weather forecast which is usually the last segment of the programme. As presented in Section 4.3 the structure of news programmes has been modeled by different researchers [Eickeler and Müller, 1999, Merlino et al., 1997, Mittal et al., 2002]. Without being too specific, the structure of a typical news programme is illustrated in figure 4.4.

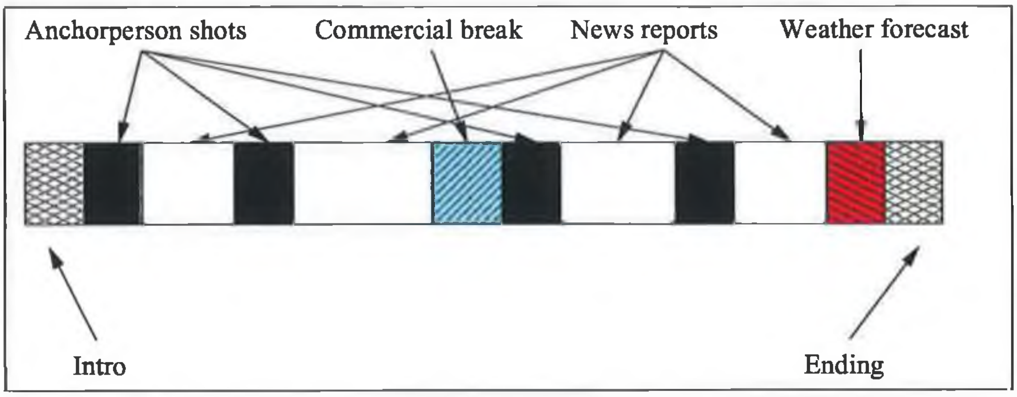


Figure 4.4: Structure of a News broadcast

Anchorperson shots are usually filmed in the same studio location. In terms of scene composition, there might be a representative graphic image in the background with a story "title" which could be used for story threading. Typical anchorperson shots from the RTE[7] evening news programme are illustrated in figure 4.5.

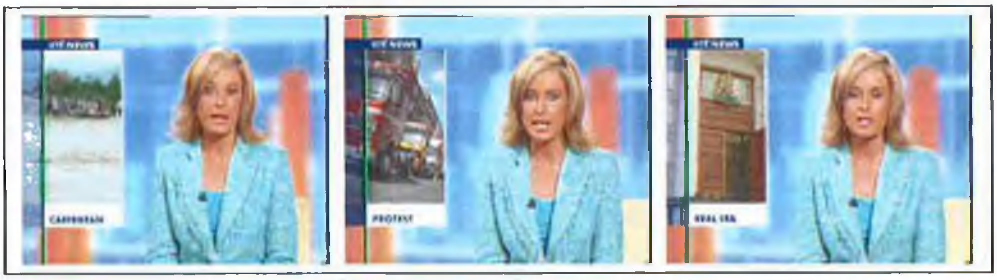


Figure 4.5: Examples of anchorperson shots

[7] Radio Telefís Éireann (RTÉ) is the Irish Public Service Broadcasting Organization.

4.3 Related Research

Most of the research on news story segmentation has focused on detecting anchorperson shots and segmenting the program based on the temporal location of these anchorperson shots. Further analyzes tackle the issue of classifying news stories into categories like: interviews, reports, sport summaries etc. [Hauptmann et al., 2003, Lin and Hauptmann, 2002, Wang et al., 2003b], thus presenting news stories at a detailed level.

Researchers in this area have exploited the fact that news programmes are a structured genre. One of the most used cues in story segmentation is the reoccurrence of anchorperson shots throughout the programme. Furthermore anchorperson shots are relatively easy to model and subsequently detect.

[Zhang et al., 1994] present an approach to anchorperson detection based on spatial and regional models of these shots. News video parsing usually requires an a-priori model based on domain knowledge. Parsing is a two step process: temporal video segmentation and shot classification. Temporal video segmentation is responsible for partitioning the video stream into individual shots. Shot classification tries to identify which shots are close to an anchorperson shot model. An *anchorperson shot model* is a sequence of *frame models* which in turn are composed of *regional models.* An anchorperson shot model incorporates temporal and spatial structure characteristics. To model the anchorframe four scenarios are considered corresponding to the physical location of the newsreader(s) and a frame is divided into regions corresponding to anchoperson, news-icon, program title. Because there is a large variety of anchorperson types to be considered as *model images*, the authors construct these models incrementally for the programme. Basically an anchorframe satisfies low motion activity and high colour similarity in the regions of interest over subsequent frames. To determine if a shot contains an anchorperson shot all frames are compared with the models, where here a frame model consists of the mean anchor-image over the shot. The limitation of this approach lies in the fact that if the spatial arrangement changes, the whole model construction must be updated.

In [Ariki and Saito, 1996] a method for news article extraction using DCT clustering and main studio setting estimation is presented. With each frame divided into an $N \times M$ grid, the DCT is applied on each block and three DCT components are extracted to form a $N \times M \times 3$ dimensional feature vector for each frame. Clustering takes place to extract cut points in the video. To estimate the studio setting, loop points are identified as clusters with high similarity values between feature vector signatures. A typical news story has the following structure: main studio, field report returning to the studio at the end, therefore a loop point is a cluster of similar cut points and represents an anchorshot. In order to distinguish between main anchor and sports anchor the ratio between the number of frames in a loop point and the number of loops with the same loop point (cluster) is calculated; the loop point with the highest value is chosen as main anchor.

[Hanjalic et al., 1998a,b] present a template-based semi-automatic anchorperson detection algorithm. It is assumed that an anchorperson shot should appear in the first K shots from the beginning of the programme. Based on this assumption, the anchorperson template is located as the shot from the K time-window from the beginning of the programme with the lowest overall dissimilarity measure to the rest of the shots comprising the news. To quantify the dissimilarity measure between shots a *shot image* is constructed by merging the near-start and near-end key-frames within the shot and diving this into non-overlapping blocks. The reason behind this tesselation is that the shot comparison technique should be able to identify shots with similar visual composition but allow for small differences. In a two-step row-wise scan, each block from a shot image is assigned to a block in the corresponding shot image which is compared to with the minimal distance in colour signature. Anchor template matching also takes into account the temporal factor in terms of distance to the last detected anchorshot.

An automatic news indexing and retrieval system is described in [Bertini et al., 2001]. Following histogram-based cut detection, a refinement is carried out considering the specific structure of news programmes. In this way, short shots are not taken into account for story segmentation and the assumption that two consecutive shots should not be both anchorshots therefore they should not be visually similar. Anchorshot detection is performed by a two-step classification. The fact that anchorshots reoccur throughout the video leads the authors to consider introducing the *shot lifetime* which measures the shortest temporal distance that includes all occurences of shots with similar visual characteristics. To identify anchorshots the values of the shot lifetime are fitted to a bimodal distribution. Those shots whose lifetime satisfy the threshold determined from a training database become anchorshots. To refine this classification, motion features are also considered to discard shots with high visual activity.

The approaches presented above take into account the statistical visual characteristics and temporal patterns of anchorperson shots during a news program and don't explicitly consider the actual scene composition in terms of the appearing objects. It is generally accepted that a typical scenario for presenting a news story consists of initial footage of the anchorman reading the story. In story segmentation anchorperson detection is essential for shot classification. As a consequence, by applying face detection algorithms the presence of reporters/anchorman/interviewees can be located. News programmes address quotidian events therefore it is very likely that a large number of human faces will be detected. However, anchorperson shots are recorded in a controlled environment (studio). Proper lighting, relative small motion and a non-occluded newsreader are factors which greatly facilitate the face detection task. These factors themselves are of importance in distinguishing between anchor- and reporting shots.

Approaches which employ face detection techniques for anchorperson detection are presented in [Avrithis et al., 2000, Gunsel et al., 1996, Ide et al., 1999, 2000, Tsekeridou and Pitas, 2001]. For this purpose, [Avrithis et al., 2000] employ a multiresolution RSST colour segmentation followed by skin colour matching and shape processing. Skin-tone colour matching is responsible for merging the regions obtained after RSST which correspond to the same facial area. After shape processing a face probability is computed to be used in news shot classification. Using the face probability score which also considers the size of the facial area in conjunction with temporal properties of these regions and the background motion, shots are assigned to four classes: single or double anchor, report/interview, static images and outdoor shots. Anchorshots are further filtered using a histogram-based clustering of possible anchorshots. The class with the largest size is selected.

The authors in [Ide et al., 1999] use face and lip detection to classify report and anchorshots. Temporal patterns of a moving person are also examined to detect walking and gathering. [Tsekeridou and Pitas, 2001] perform an exhaustive audio analysis to classify speaker changes. On the visual side, face regions are located by performing skin colour segmentation and coarse facial feature extraction. Once a candidate is identified, the aim of mouth template tracking is to gather evidence of a speaking face in the shot. Therefore it can be identified which speaker is talking during a story, or the anchorperson as the person which possess the highest speech frame ratio in a shot.

[Gunsel et al., 1996] establish anchorperson layouts by colour classification and template matching for skin-tone detection. To cover the scenarios which the authors consider frequent, five different anchor templates are constructed using positional cues (close-up, medium close-up left, two presenters left and right etc.). Histogram intersection is also considered given that studio settings don't change significantly during a news broadcast.

A family of approaches towards story segmentation are those based on modelling the news broadcast as a sequence of states. Parsing and classification is done using Hidden Markov Models. It is an approach which adheres to the general idea that temporal patterns can be modeled by a finite sequence of states and transitions between them. Once identified, those patterns considered to be "on-their-own" contain a higher degree of semantic information than video segments which are separated only by camera switching (shot change). In this way a larger temporal segment establishes a meaningful segment for content-based video indexing.

Early applications of Hidden Markov Models for parsing video programs have been carried out [Wolf, 1997]. In [Eickeler and Müller, 1999], for instance, six content classes are considered which model a news story (begin/end of newscast, anchorperson, report, interview, weather forecast) and four classes associated to editing effects (cut, wipe, dissolve, window change). The latter classes are used because the system works on a frame-by-frame basis, thus parsing and classification are executed in the same stage, as opposed to the approach where shot segmentation takes place prior to classification. The authors motivate their choice by the fact that editing effects are often part of the video model (e.g. not all scenes are separated by hard-cuts). Then with these ten content classes the video model is constructed (i.e. the topology of the Hidden Markov Model and possible transitions between states) considering a-priori knowledge about the structure of the news. Having the video frame as the elementary item, feature vectors are extracted for each frame consisting of inter-frame motion difference, motion intensity, grayscale histogram difference and the logarithmic energy of the audio signal for each processed frame. The way the system is designed also allows other logical units to be identified: interview, report segment, weather forecast.

[Wermer et al., 2002] present a system similar to [Eickeler and Müller, 1999], with the difference that they make use of the audio track, by applying a speech recognition tool, towards topic detection. Visual and audio features are combined in a similar manner to [Eickeler and Müller, 1999], and an adapted video model is created which reflects the topic structure in a news programme. The disadvantage of this model lies in its rigidity and it needs to be adapted or extended when applied to news programmes from different broadcasters. The automatic topic identification algorithm uses words generated by the automatic speech recognizer engine and assigns each word to an unique index in a vocabulary. Each topic is modelled with a single state HMM having index numbers from the dictionary corresponding to the extracted words as observations. A second approach investigates sub-word features (character based) considering the existence of ASR errors.

A two-level multi-modal framework is presented in [Chaisorn et al., 2002]. In the first stage or at shot level, low-level visual features, temporal features such as motion activity, audio type (speech, music, silence) etc. and high-level features (face, text overlay) are extracted to index each shot. A decision tree classifies each shot into 13 categories. At the next level a four state Hidden Markov Model (HMM) is used to detect story boundaries based on the results of the previous stage, but adding extra features such as speaker change and location change.

Finite State Automata (FSA) have been used to parse news broadcasts into news story segments [Merlino et al., 1997, Mittal et al., 2002]. A text-based story segmentation using FSA is described in [Merlino et al., 1997], while in [Mittal et al., 2002] a parsing tree construced by FSA is employed to divide the news programme into it's constituent semantic units.

A model-free anchorperson identification system is described in [Gao and Tang, 2002]. To distinguish between anchorperson and news footage a graph-theoretical cluster analysis is performed. With the keyframes as vertices, the minimum spanning tree of the graph is constructed and all edges whose weight exceeds a threshold (dissimilarity value) are cut. In the next step the task is to find all trees in the forest. Considering that an anchorman appears at least two times during the broadcast, the trees with more than two nodes are considered as potential anchor-clusters. These are determined by imposing story length and spatial similarity constraints.

[Hsu and Chang, 2003] developed a statistical framework for news video indexing. The framework is a maximum entropy model which fuses mid-level audio-visual features to approximate the probability of a story boundary at a point in the programme. Audio and video raw features (pitch jump, significant pause, speech, anchor face) and mid-level features (commercial, ASR-based text segmentation) are wrapped using differentiation, binarization and fed into the maximum entropy model.

4.4 News Story Indexing Based on Video Shot Clustering

4.4.1 Shot Clustering

Anchorperson shots, in general, are filmed in a studio with controlled lighting, common background, the same camera settings etc., hence they tend to be visually similar to each other. The following two algorithms present techniques for clustering visually similar shots into groups and identifying clusters which correspond to anchorpersons by applying genre-specific knowledge. However, this knowledge is limited to the typical structure of a news programme and could not be applied easily to other types of programmes; when indexing related genre shows, a new set of rules applied to clusters of similar shots should be considered (e.g. interviews, talk shows, etc.) [Li and Kuo, 2003].

The shot clustering is the core component in our approach to segmentation of news programmes. The algorithm implemented is based on the temporally constrained clustering approach of [Rui et al., 1999]. In the following, two different algorithms are described.

4.4.1.1 Algorithm I

Prior to clustering, the video is segmented into shots using a histogram-comparison shot cut detection algorithm and features are extracted as described in Section 2.3. For each shot a key frame is selected based on the similarity between its colour signature and the average colour signature over the entire shot. Then a colour histogram quantized into 64 bins for each band (Y, C_b and C_r) represents the colour feature for a shot.

For this algorithm we use only the histogram feature associated with each shot. The algorithm groups shots based on the similarity of their colour composition and the temporal distance between the shots [O'Connor et al., 2001]. The processing chain is depicted in Figure 4.6. Each shot is represented by the colour feature vector of the keyframe. Thus, each shot is represented as a point in a multidimensional feature

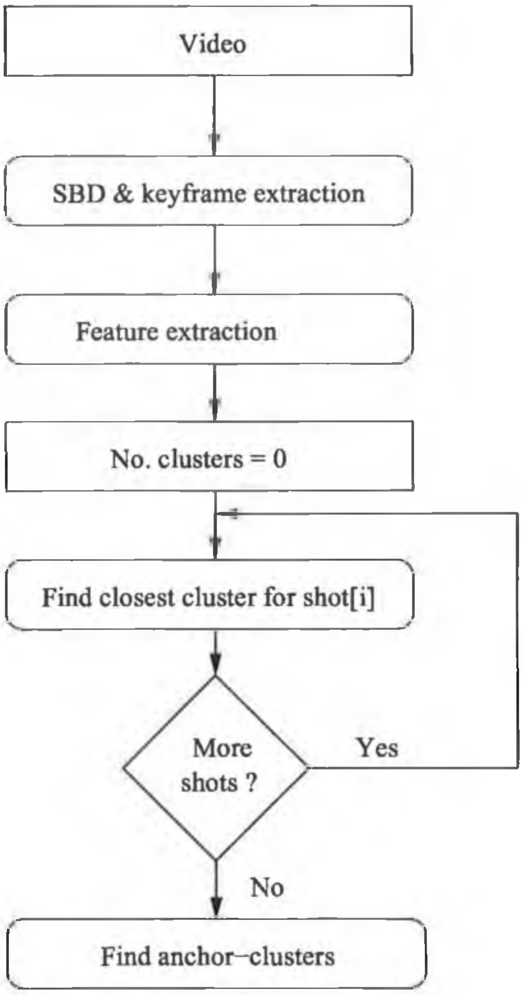


Figure 4.6: News story segmentation workflow

space. To quantify the distance between points in this space we have chosen the cosine distance measure which provides the similarity between shots and is given by:

$$ColSim(S_A, S_B) = CMD(S_A, S_B) = \frac{\sum_i x_i \cdot y_i}{\|X\| \, \|Y\|} \tag{4.4.1}$$

Commercial content is omnipresent in almost any broadcast video. The content and character of advertisement breaks embedded into a program are significantly distinctive from the topic of the broadcast in progress. These intermissions are not of any interest from our point of view and these segments are discarded from news content analysis. Automatic detection of advertisements has been an objective of video analysis in the past [Lienhart et al., 1997]. In order to mark the segments in video corresponding to advertisement we employ the method described in [Sadlier et al., 2002]. This approach is highly efficient due to the fact that analysis takes place on the audio- and video track directly from the MPEG-1 bitstream.

The shot clustering algorithm is listed in Figure 4.7 where N denotes the number of shots in the video, M the number of clusters formed, c_i and s_i correspond to shots and clusters respective, θ_k represents the k^{th} group's threshold, and τ is a constant ($= 3$) needed to compute the dynamic threshold.

Similar to Rui's approach, the algorithm clusters similar shots into groups. However, the decision to place two shots in the same group depends not only on the colour similarity between them, but also how close two shots are to each other temporally. In this way, two shots that are very similar in terms of their colour composition but far apart in time will not be placed in the same group. The colour similarity is weighted by a temporal closeness factor defined to be linearly decreasing:

$$TW(S_A, S_B) = \max\left(0, 1 - \frac{S_A - S_B}{T_L}\right) \tag{4.4.2}$$

where $S_A - S_B$ is the temporal distance between shots expressed in the number of shots which separate them and T_L is the desired length of the temporal weighting function. While [Rui et al., 1999] use clusters of spatially and temporally similar shots as intermediary structures towards scene detection in motion pictures, in news programmes anchorperson shots are separated by a longer distance. A *scene* in news is quite a different concept, at this level meaning a news story. This motivates us to set the temporal penalty T_L to a sufficiently large value to avoid micro-segmentation, in other words to avoid the situation where anchorperson shots are more likely to be dispersed amongst a number of smaller clusters.

The overall shot similarity becomes:

$$Sim(S_A, S_B) = TW(S_A, S_B) * ColSim(S_A, S_B) \tag{4.4.3}$$

If there are currently M groups in the clustering process, then the candidate grouping

C to which to assign the current shot, S_{curr} is calculated as:

$$C = max\{Sim(S_{curr}, C_i)\}\ \forall C_i \in C_1 \dots C_M \tag{4.4.4}$$

Once the candidate group C is found, a decision must be made as to whether or not S_{curr} can be added to that group or not. This decision is based on a threshold. For groups with less than three members a static threshold is used. For groups that have more than three members (necessary to compute the similarity variance) we use a *dynamic threshold.* The advantage of using a dynamic threshold which is based on the group statistics, is that the threshold is continuously updated as shots are added to that particular group.

ShotClustering I(N, shots)

```
$M \leftarrow 1$
$c_1 \leftarrow s_1$
FOR $i = 2$ TO N DO
  $maxSim \leftarrow \max_{j=1,\dots,M}\{Sim(s_i, c_j)\}$
  $closestCluster \leftarrow \{j | maxSim = Sim(s_i, c_j), \forall j = 1, \dots, M\}$
  $k \leftarrow closestCluster$
  IF $maxSim < \theta_k$ and $\#\{c_k\} > \tau$
  THEN
    $c_k = c_k \cup \{s_i\}$
    UPDATE $\theta_k$
  ELSE
    $M \leftarrow M + 1$
    $c_M = c_M \cup \{s_i\}$
  ENDIF
ENDFOR
RETURN(clusters)
```

Figure 4.7: Shot clustering algorithm (I)

For each group with three or more members, the shot similarities between adjacent shots are determined. The *the shot similarity mean*, μ, and the *shot similarity standard deviation*, σ, are calculated. The condition which must be satisfied in order for a shot to join a particular group is: $| \mu - sim | < k \cdot \sigma$, where sim is the overall similarity between the current shot S_{curr} and the closest group C and the constant $k \simeq 1.25$

has been experimentally determined. When a shot is assigned to a group, the group statistics are updated and the threshold modified accordingly. If the decision is made *not* to assign S_{curr} to C, then this shot forms a new group C_{M+1}. This occurs quite frequently at the start of the clustering process until a representative set of clusters is obtained.

The result of the clustering process is a set of groups of shots upon a set of rules derived from news programme analysis is applied to identify anchorperson groups. The set of rules is common to both clustering-based news segmentation algorithms presented here and are described in Section 4.4.2. Once identified, the selected anchorgroups are re-arranged chronologically and become entry points in the news programme's table of contents.

4.4.1.2 Algorithm II

The second algorithm presented here is a variation of the previously described technique. The algorithm above is a one-pass process with no feedback on the results of clustering, therefore a more robust approach was investigated; however this procedure has increased computational costs.

Similar to the algorithm described in Figure 4.7 shots are clustered based on their likeness. This time however, the temporal attraction factor is omitted in computing the overall similarity between two shots. The colour similarity between two shots are calculated according to Equation 4.4.1 with their colour signatures a-priori normalized. The feature vector describing a shot is extended with the Haar features computed as in Equation 2.2.8, normalized and placed in a vector.

Sequential clustering algorithms are fast and straightforward, a fact which makes them computationally attractive for our task. But it is well known that the clustering results are dependent on the order the input vectors are presented to the algorithm [Theodoridis and Koutroumbas, 1999]. To overcome this handicap, given that poor clustering implies disappointing news segmentation results, the sequential shot clustering algorithm is instantiated R times with the input vector corresponding to the order of shots randomized, the overall best cluster being selected as candidate

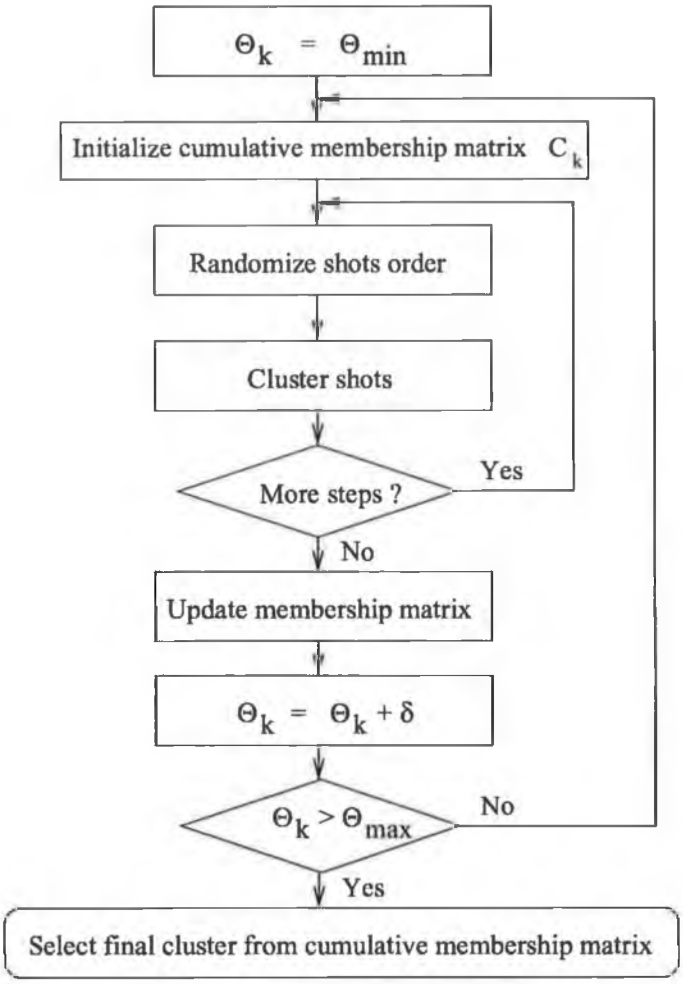

$\Theta_k = \Theta_{min}$
Initialize cumulative membership matrix C_k
Randomize shots order
Cluster shots
More steps ?
Yes
No
Update membership matrix
$\Theta_k = \Theta_k + \delta$
$\Theta_k > \Theta_{max}$
No
Yes
Select final cluster from cumulative membership matrix

Figure 4.8: Algorithm II

anchorperson group per R runs. Starting with a minimum shot similarity allowed, the above process is repeated until a maximum allowed similarity threshold value is reached. For each θ_k threshold the suitable clusters are selected. The reasons behind this are two-fold: first, by running R times the clustering algorithm, we expect that anchorperson shots will appear together in the same groups; secondly, the potential anchorgroup selection is embedded in the clustering process. When selecting the best fit cluster, the anchorgroup detection rules are applied (Section 4.4.2). As we might anticipate, using a linearly increasing global threshold (as opposed to the previous approach) will lead to more and more homogeneous clusters of shots, with their number increasing as the threshold becomes higher and cluster members decreasing. The goal is to find the best arrangement of shots c_j^{θ} in the threshold interval $[\theta_{min}, \theta_{max}]$ which statistically are the most frequent successful groupings, satisfying the set of imposed rules Ψ (Section 4.4.2). Let the membership matrix at step θ_k and run $r, r = 1, \ldots, R$, $\mathcal{C}^{\theta_k,r}$ be:

$$\begin{array}{c|cccccccc}
\mathcal{C}^{\theta_k,r} & s_1 & s_2 & \cdots & s_p & \cdots & s_q & \cdots & s_N \\
\hline
c_1 & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
c_2 & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
c_m & 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
c_n & 0 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
c_M & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0
\end{array}$$

Assuming that clusters c_m and c_n above satisfy the set of rules Ψ, for each shot $s_\xi \in c_i^{\theta_k,r}$ with $i = \{m, n\}$ a *cumulative membership matrix* $\mathfrak{C}^{\theta_k}$ is updated based on the information contained from the previous steps $\mathcal{C}^{\theta_k,r-1}$. If $s_\xi \in c_i^{\theta_k,r}$ and $s_\xi \in c_j^{\theta_k,r-1}$ then:

$$\mathfrak{C}^{\theta_k}_{i,\xi} = c^{\theta_k,r}_{i,\xi} + c^{\theta_k,r-1}_{j,\xi} \tag{4.4.5}$$

In other words, by using $\mathfrak{C}^{\theta_k}$ we keep track of co-occurrences of shots in candidate anchorperson shots. Obviously, the membership matrixes in consecutive steps might be different. To update $\mathfrak{C}^{\theta_k}$, the groups $c_m^{\theta_k,r}$ and $c_n^{\theta_k,r}$ are matched in the cumulative matrix by intersection. For example, if c_m is matched in the cumulative matrix, after a number of steps, $\mathfrak{C}^{\theta_k}$ could be, for example:

$\mathfrak{C}_r^{\theta_k}$	s_1	s_2	$\cdots$	s_p	$\cdots$	s_q	$\cdots$	s_N
c_1	0	0	$\cdots$	0	$\cdots$	0	$\cdots$	0
c_2	1	0	$\cdots$	0	$\cdots$	0	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
c_m	7	0	$\cdots$	0	$\cdots$	9	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
c_n	0	15	$\cdots$	12	$\cdots$	0	$\cdots$	10
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
c_P	0	0	$\cdots$	0	$\cdots$	0	$\cdots$	0

ShotClustering II(N, s, r, θ_{min}, θ_{max}, δ)

```
FOR $\theta = \theta_{min}$ TO $\theta_{max}$
DO
  WHILE $r \geqslant 0$ DO
    randomize shots order $\{s_i\}$
    FOREACH $s_{i,i=1,\ldots,N}$ DO
      $maxSim \leftarrow \max_{j=1,\ldots,M} \{Sim(s_i, c_j)\}$
      $closestCluster \leftarrow \{j|maxSim = Sim(s_i, c_j), \forall j = 1, \ldots, M\}$
      $k \leftarrow closestCluster$
      IF $maxSim < \theta$ THEN
        $c_k = c_k \cup \{s_i\}$
      ELSE
        $c_{M+1} = \{s_i\}$
        $M \leftarrow M + 1$
      ENDIF
    ENDFOR
    select best cluster $w_r$ from $\{c_j\}_{j=1,\ldots,M}$
    $r \leftarrow r - 1$
  ENDWHILE
  select best cluster $w_\theta$ for all runs $r$
  $\theta \leftarrow \theta + \delta$
ENDFOR
select best cluster $W$ from $\{w_\theta\}_{\theta=\theta_{min},\ldots,\theta_{max}}$
RETURN($W$)
```

Figure 4.9: Shot clustering algorithm (II)

A diagram depicting the approach is illustrated in figure 4.8. For a certain threshold θ_k the number of clusters varies around a mean. Using the cumulative membership matrix at θ_k updated after each individual run, shots which are selected as anchorperson shots belong to the clusters which satisfy the anchorperson selection criteria and have the highest overall value.

After all runs, the winning anchor-cluster is chosen and the items in this collection represent news story entries in a video browsing/summarization system.

The algorithm is listed in Figure 4.9.

4.4.1.3 Computational Complexity

We estimate the computational complexity, for the first algorithm as follows:
$\mathcal{C} = \mathcal{O}\left(\sum_{i=1}^{N} m_i\right)$, where m_i represents the number of clusters at step i which, in worst case, leads to $\mathcal{O}(\frac{N(N+1)}{2})$ or $\mathcal{O}(N^2)$. For the second algorithm:
$\mathcal{C} = \mathcal{O}\left(N \times R \times N \times \frac{\theta_{max}-\theta_{min}}{\delta}\right)$, thus $\mathcal{C} = \mathcal{O}(k \times N^2)$. The constant k depends on the number of runs R, the minimum and maximum threshold and the threshold step δ. Experimentally, for $\delta = 0.05$ threshold step, $\theta_{min} = 0.65, \theta_{max} = 0.97$ threshold limits and $R = 10$ runs per $\theta_i \Rightarrow k = 320$, which means $k \approx N$ (typical for news programmes $N \simeq 350$).

4.4.2 Anchorperson Identification

The result of the shot clustering algorithms are a set of groups of shots which satisfy the similarity constraints outlined above. Based on observations from RTE, CNN and ABC news, anchorperson shots in our corpus consist of a quasi-static background and the foreground person who reads the news. This feature ensures high similarity between subsequent anchorperson shots throughout the news programme. When detecting anchorperson shots there are a number of possible scenarios:

- only one anchorperson exists in the whole programme;
- there are two anchorpersons, each taking every second story;
- there are two anchorpersons, one main anchor and one for sport stories and/or weather.

In order to decide which groups constitute anchorperson groups, we examine all groups and attempt to successively discount groups on the basis of criteria designed to distinguish anchorperson groups.

The anchorperson identification consists of a set of rules which are applied upon all groups and select those which satisfy the following conditions:

1. *Anchorshot scattering rule* - the range of shots (the distance between the first and the last shot in the group) should be higher than a predefined value, since anchorperson shots are typically spread throughout the programme. This ensures that potential anchorgroups should cover a significant portion of the news.

2. *Colour similarity rule* - the group similarity should be higher than a predetermined value. This criterion is used because anchorperson shots tend to be very similar to each other.

3. *Temporal length* - the mean anchorperson shot length should be longer than 6 seconds approx., due to the fact that at least one sentence should be spoken when presenting a news story.

4.5 Neural Network Based Anchorperson Shot Identification

4.5.1 Network Architecture

Neural Networks are analytic techniques modeled after the (hypothesized) processes of learning in the cognitive system and the neurological functions of the brain. They are capable of predicting new observations (on specific variables) from other observations (on the same or other variables) after executing a process of so-called learning from existing data.

Recurrent neural networks are known for their capabilities of learning time-dependent data sequences and performing nonlinear prediction on non-stationary time series. Recurrent neural networks possess feedback connections which allows them to maintain an internal state and also provides features relevant to the entire time-dependent series. Multi-layer feedforward (MLF) networks are suitable for learning *static* input-output mappings. However, in case of time-dependent processes, typically the history or current system state helps guiding the learning process [Haykin, 1994]. As any system with feedback, the stability of recurrent neural networks during the learning phase suffers and they are often are difficult to train.

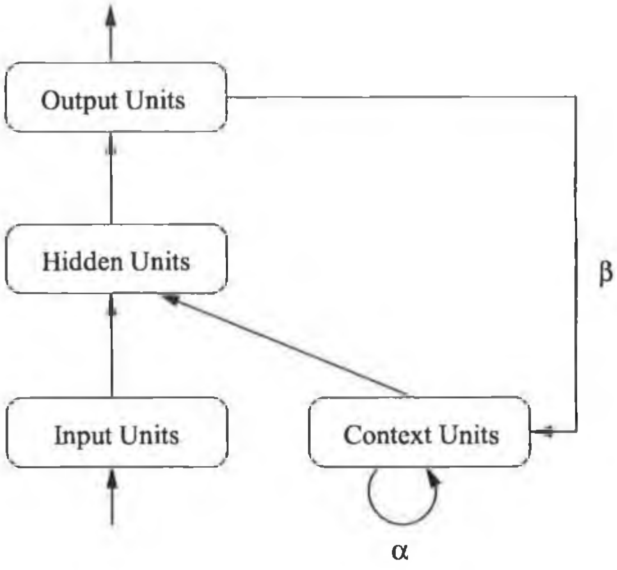


Figure 4.10: Jordan network

Recurrent neural networks (RNN) can be grouped into partial- and fully RNNs depending on the characteristics of the feedback connections. Fully RNNs use unconstrained fully interconnected architectures. The most popular partial RNN architectures used were proposed by Elman [Elman, 1990] and Jordan [Jordan, 1989]. Elman networks have connections from the hidden layer to the context units. In Jordan RNNs the context units receive input from the output layer and also from themselves.

We adopt recurrent neural networks for story segmentation based on the fact that news programs have a structure with repetitive patterns. Instead of only static input-output mapping we incorporate also history for story segmentation by using Jordan RNNs. The structure of such a network is illustrated in Figure 4.10.

4.5.2 Feature Learning

To adequately learn the input-output mapping or more precisely the transfer function, the neural network is presented with a large set of training patters and a set of patterns which constitute the validation set. For our experiment, we consider the input vector as a concatenation of several feature vectors, each describing particular image and temporal characteristics. The input vector receives data from four feature extraction modules:

1. keyframe's YC_bC_r colour histogram
2. Haar coefficients
3. video shot length
4. forward- and backward- inter-shot similarities within a temporal window

The first type of information represents the colour composition of the keyframe associated to the shot under investigation in the form of the YC_bC_r colour histogram. To avoid huge and consequently impractical sizes for the input vector, the histogram is quantized into 16 bins for each colour band, thus the colour-related section of the input vector consist of a $16 \times 3 = 48$ size string.

Haar features provide information about the textural composition of images in a hierarchical fashion. Haar descriptors [Mallat, 1989] are simple and computationally efficient wavelet decomposition techniques in image processing. For our purpose, the feature vector which composes this part of the input feature vector is constructed from the H_1, H_2 and H_3 coefficients computed for each block in the image as described in Section 2.2.3.2. Each key-frame is tesselated into $M \times M$ non-overlapping blocks for which the Haar coefficients are extracted. The H_0 coefficient has been excluded because it contains most of the energy of the signal which has been partly already included in the input feature vector. With image blocks of size $M \times M, M = 64$ pixels, the part of the feature vector corresponding to Haar coefficients has the length of $20 \times 3 = 60$.

Forward and backward similarities between video shots describe the homogeneity of a temporal window. For this, two values are computed as the ratio between the number of shots with high similarity to the one under investigation and half the window size, centered on the current shot and are weighted by the similarity value.

As mentioned in previous sections, the temporal length of anchorperson shots should meet some constraints. As opposed to the previous approaches, there is no threshold applied to this feature, instead we expect the neural network to learn the characteristics of this feature and it's importance as well. For this descriptor one position is allocated in the feature vector and has the value of the shot length expressed in number of video frames normalized to the $[0, 1]$ interval.

Concatenating the features from the four modules presented above leads to a 111-length input vector. When designing the neural network there are no strict rules regarding the number and size of hidden layers. However in deciding the size of hidden layers, the length of the input layer is a significant factor; if N is the length of the input vector, the size of the hidden layer is often chosen as $\sim \sqrt{N}$ [Haykin, 1994]. We considered 15 neurons (with sigmoid activation) for the hidden layer (following the above condition). With one unit in the output layer the context unit also consist of a single unit with inputs from the output layer and itself. There is only one output element because there is only one binary output: news story start or other type of shot. For training and testing of this approach the *Stuttgart Neural Network Simulator (SNNS)* software package has been used [Zell et al., 2000].

Back-propagation is one of the most widely used supervised learning algorithms. Backpropagation is a gradient descent iterative optimization algorithm of the mean square error (MSE) expressed as a function of the weights. The algorithm consists of a forward- and a backward-stage; first the input pattern is applied to the network and the activations of the hidden and output layers are computed. The difference between the real output and the desired output represents the learning error. In the second stage the derivative of this error is back-propagated and the weights are adjusted accordingly to reduce this. Mathematically the weight update for a variable

s can be written as:

$$\Delta w_s = -\eta \frac{\partial E}{\partial w_s} \tag{4.5.1}$$

where E represents the error, w_s the weight and η, called the learning rate determines how big the step is in the direction opposite to the gradient of E. This process is repeated until the network stabilizes and converges.

The backpropagation algorithm, in it's simplest form, is known to be quite slow. Improved versions of the algorithm found in the literature use the *momentum* technique and adaptive learning rate. The idea behind the momentum technique is to keep the minimization process in the same general direction and escape from local minima. This involves adding a weight adjustment proportional to the previous change in time. In this case Equation 4.5.1 becomes:

$$\Delta w_s(t+1) = -\eta \frac{\partial E}{\partial w_s} + \mu \Delta w_s(t) \tag{4.5.2}$$

where μ is the momentum coefficient; this process clearly requires that the previous weight adjustments need to be stored. The main rationale behind using adaptive learning rates for each unit is to increase the rate if the error descrease is in the same direction as in the previous step t (and opposite otherwise).

Perhaps the most difficult tasks in training neural networks are the choice of architecture and setting the learning parameters. For training the Jordan network the backpropagation algorithm with momentum and flat spot elimination from the SNNS (Stuttgart Neural Network Simulator) package has been used. A portion of the learning curve (error graph) is illustrated in Figure 4.11. It can be observed that the network error graph presents an abrupt decrease in the first 250 steps of learning the input samples. This is explained by the fact that there is only one intermediate layer this and (perhaps) the samples are separable by the network plus the momentum factor in the algorithm which ensures that the weights are updated in the direction according to the error gradient decrease. However, Jordan networks - and usually recurrent systems - are hard to stabilize and we can see that in the fluctuations of the learning error curve. However, the flat spot elimination factor also contributes to this behavior of the network.

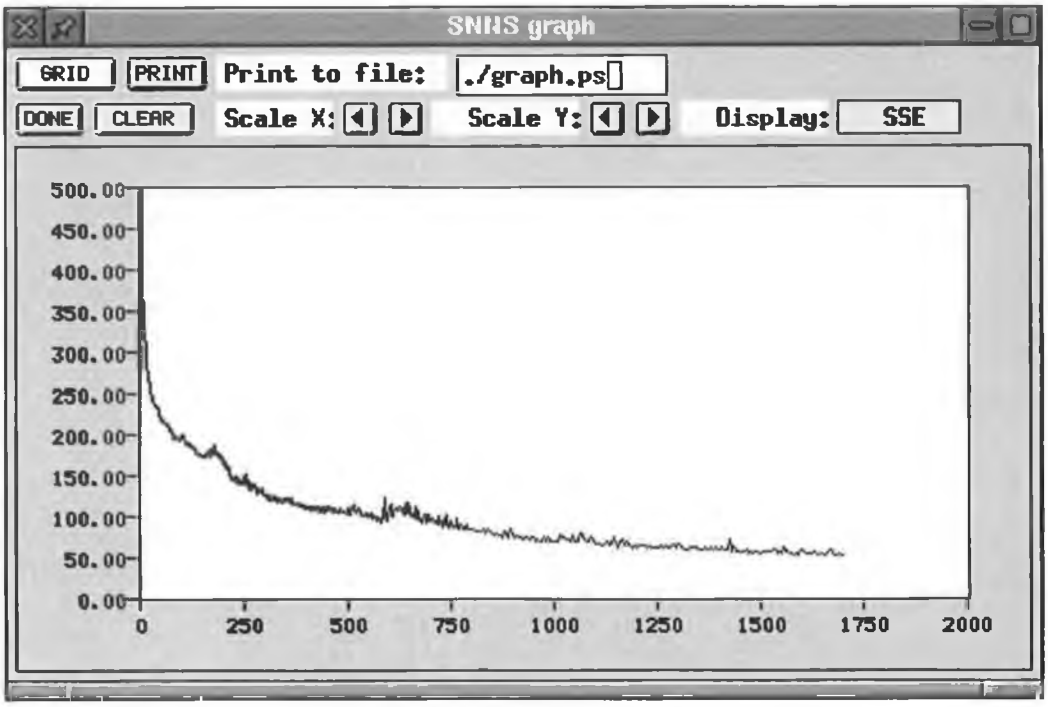


Figure 4.11: Network learning error graph

4.6 Experimental Validation

In this section the experimental results of the above three algorithms are presented and their performances are compared.

In order to test the efficiency of the above presented news story segmentation algorithms, we use a large annotated news corpus available from three public broadcasters. This set of programmes consist of 102 RTÉ news recorded over a period of four months and annotated at story level. In addition to this, the development and test set available from the TREC 2003 video track [8] were also used in evaluating the story segmentation task. This comprises 120 hours of ABC and CNN evening news dating back to 1998. In total are 4,437 news items. This set is split into a development and test set respectively. For the two clustering-based algorithms described in Section 4.4 this has no effective meaning, given that the segmentation is driven by rules. The effectiveness of the presented story segmentation algorithms are measured using the

[8] http://www-nlpir.nist.gov/projects/trecvid/

standard precision and recall:

$$\text{Precision} = \frac{\text{Number of correctly detected stories}}{\text{Number of correctly detected stories} + \text{Number of false stories}}$$

$$\text{Recall} = \frac{\text{Number of correctly detected stories}}{\text{Number of correctly detected stories} + \text{Number of missed stories}}$$

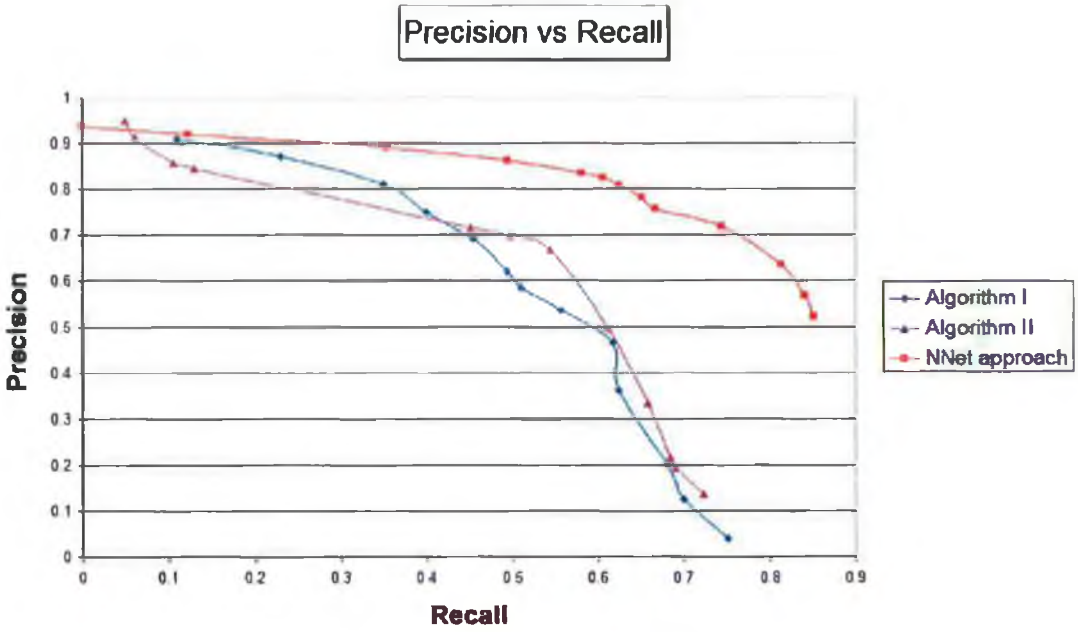


Figure 4.12: Precision vs Recall

It Figure 4.12 can be observed that the neural network-based story segmentation outperforms the two clustering-based approaches. It is quite understandable given the large corpus the training has been conducted (RTE, ABC and CNN evening news). Using the ensemble of rules for determining the anchor-clusters, Algorithm I performs well, but it's highest peak is much lower as Algorithm II and the precision falls quite rapidly. Algorithm II is more robust in general, but I introduces more false alarms, therefore oversegmentting the news programme.

4.7 Summary

This chapter presented an overview of existing news video libraries and described a generic news story segmentation system based on three different algorithms. Two of these algorithms use rules in order to facilitate the anchorperson shot classification process. A novel recurrent neural network-based anchoperson shot classification has been proposed which uses visual features extracted for the representative keyframe in conjunction with inter-shot colour similarity. These tests were carried out using material from three stations and they performed quite well despite the variations in production style specific to each broadcaster. However, incorporating more low- and/or mid-level audio-visual features could prove to the overall performance of the news story segmentation schemes presented.

Chapter 5

Major Cast Detection in Movies

This chapters presents an approach towards major cast detection in movies. In contrast with other types of programmes, motion pictures have a high degree of freedom in many ways. This is very natural, considering their goal of entertaining an audience. Each movie reflects the film director's taste in the genre and subject chosen but also it's own style in composition, rhythm etc. in production. Using image evidence such as presence of face and face cluster, we try to identify the temporal location of the appearences of major characters within the movie.

5.1 Introduction

A fundamental problem in video databases is the management of content to make the system useful for the end-user. The major cause of the above deficiency is the large amount of audio-visual data and the relatively small amount of description at various levels. There are still a limited number of tools to describe, organize and manage video. We can only assume the types of information or interaction a user will require from the system. The domain of these interests is practically unlimited, but from our point of view we try to offer the user the ability to search, retrieve videos and ease interaction. An interesting aspect is related to the user type. There are different requirements for a technical, professional user who uses the system regularly, possible for business purposes, and non-technical consumers. The requirements are different, so the technologies addressing their needs are somehow different.

Multimedia content is defined at perceptual and conceptual level. At perceptual level, it is characterized by the perception properties: colour, motion, acoustic features etc. Conceptual level analysis is more application driven, and the approaches are determined by the interested semantics of different applications. It is important to consider what type of content or genre we are addressing. From a practical point of view, searching through a movie for a specific segment makes sense if the user had already seen it, otherwise fragments of a movie don't follow a story line and don't present the underlying idea. Automatic video summarization techniques help overcome this. A summary of the movie is generated to give a quick view of the program, therefore the user will consider watching it or not.
Major cast detection in movies helps identifying the time slots where the principal characters appear throughout the motion picture. This process gives:

- Character information
- Time reference to their appearance (when)
- In conjunction with textual sources (movie script) gives semantic information (who, what, where)

The major cast is domain specific: talk shows, interviews, news, documentaries, movies have a slightly different way of denoting the principal character. For instance, in a talk show the interviewee, in my opinion, should be considered more important than the moderator. Of course, the target application is the driving force for the approach subsequently adopted. In movies, a starting point towards major cast detection is to identify the segments which include dialogs between players. Usually, throughout a movie principal characters are engaged in dialogues, therefore we considered this element as an indication of their presence.

5.2 Dialogue Detection

Dialogue between players is a common mid-high level semantic element in movies. It involves and carries a lot of information like the plot, action unfolding etc. An important element is how/when a dialog sequence begins. From the film-making literature, the editing patterns provide meaningful information for deciding when a dialog scene actually begins [Li and Kuo, 2003]. In movies, actors are not placed in their allocated positions ready to read their script lines; instead a natural sequence of events is presented (e.g. they meet, walk, talk and finally depart etc.). Dialogue scenes can be classified according to the number of players involved: 2 player dialogue scene, 3 player, 4+ players. For a two player dialog scene, a film director has a number of base-setups: face to face, side-by-side, players behind one another. Dialogue patterns also incorporate master-shots (clip presenting both players in the scene) and re-establishing shots . These shots play the role of "remainders" to the viewer, such as an indication where the action takes place, or to emphasise the spatial relationship between the players. Often these shots interrupt the dialog for a short period. When a conversation takes place between characters, a pause is necessary because dialogs require the viewer to focus on the story flow. These types of shots are typically used halfway throughout the scene. Other types of intermediate shots depict actors moving from one place to another or "interplays" when an actor is temporarily excluded.

Given the large palette of scenarios a film director can use for a dialogue scene, it is necessary to adopt an approach towards automatic dialogue detection which takes this into account. Dialogue detection techniques make use of a grammar-like description of the scene. Alternation of similar shots can be used for parsing the conversational scene. Other methods use more or less complicated finite state automata to model a dialog [Chen and Ozsu, 2002, Lehane et al., 2002]. [Alatan, 2001] uses face detection and audio track analysis combined into a Hidden Markov Model to detect dialogue-like scenes in motion pictures. Our adopted approach tries not to categorize a dialog into a predefined number of scenarios; instead a learning technique is employed. Given a training corpus consisting of positive/negative annotated examples, by training a recurrent neural network classifier we aim to identify segments which resemble dialogues.

The classifier needs audio-visual features extracted from video data. In the following we present the features chosen and explain their.

1. Colour distribution - in itself it's not characteristic for dialogue scenes. It becomes relevant by the consistence of the same setting throughout the dialog scene. We measure how similar the shot is to the preceding and following adjacent shots.

2. Actor (face) - is an important visual feature signalling human presence (talking or silent). The framing of the actors is also a good indicative for the repeated appearance of the same character (detailed informations on character framing - region of interest - and camera angles in [Arijon, 1981]. We use a confidence measure of a face being present in the shot as a weighted sum of the face confidence and skin percentage ratio - used as an auxiliary measure to help in case of missed faces.

3. Scene motion - static dialogue shots possess certain characteristic signature. We compute the average motion vector values in a shot.

4. Audio features - for this the mean audio level per shot is calculated.

5. Shot length - is a factor normalized to the [0-1] interval and represents the length of a shot in frames

Given this ensemble of features for each video shot, an input vector is constructed by concatenating the audio-visual (above) signatures for the shot. A dialog shot can comprise a variable number of shots, therefore a "memory" is more likely to perform better classifying a shot as being part of a dialog if the adjacent - previous and next shots are also considered in the input vector expressed by the colour similarity values. The special (changing) editing rules used by the film director - master shots, establishing-shots - are part of the dialog and the aim is that the classifier will be able to learn these "special" cases (wider learning space).

To identify dialog-scenes we use a recurrent neural network with $N + K$ input units, corresponding to the N low- and mid-level features extracted for each video shot in

Movie	Extracted dialogues	Correct	Missed
Enigma	35	29	3
Fathers Day	56	43	5
Jakob the liar	32	29	4
Legal eagles	25	26	2
Simpatico	39	37	5
The parent trap	46	41	7

Table 5.1: Dialogue detection results

conjunction with the K shot-similarity features in the neighborhood of the current shot. We use a two unit output layer, one signalling the dialog presence and the second being the time-delayed output - or - last shots dialog status. This way we train the network to avoid glitches of short dialog-like segments (below 3 shots long) which could not be considered as conversational scenes. In fact it is generally accepted that a dialog scene should consist of at least 5 consecutive shots in a two-person conversation [Arijon, 1981].

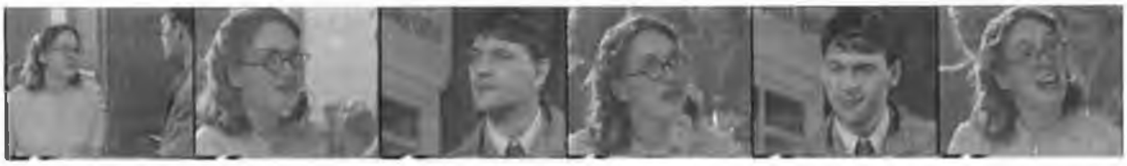


Figure 5.1: Typical dialogue scene

The Elman recurrent neural network used is depicted in Figure 5.2. It consists of four layers: and input layer (9 neurons), two hidden layers (with associated context layers) and and output layer with two neurons - one being the output for the shot under investigation and the second one represents the output of the network for the previous shot. This is necessary to ensure detecting a continuous dialog and to prevent arbitrary shots to be classified as being dialogues. The SNNS [Zell et al., 2000] neural network simulator has been used to carry out the experiments.

The results are encouraging, however in order to train the neural network a large training set is required. To have a training set, a manual annotation on each movie has been carried out and each shot part of a dialog scene has been labelled. In

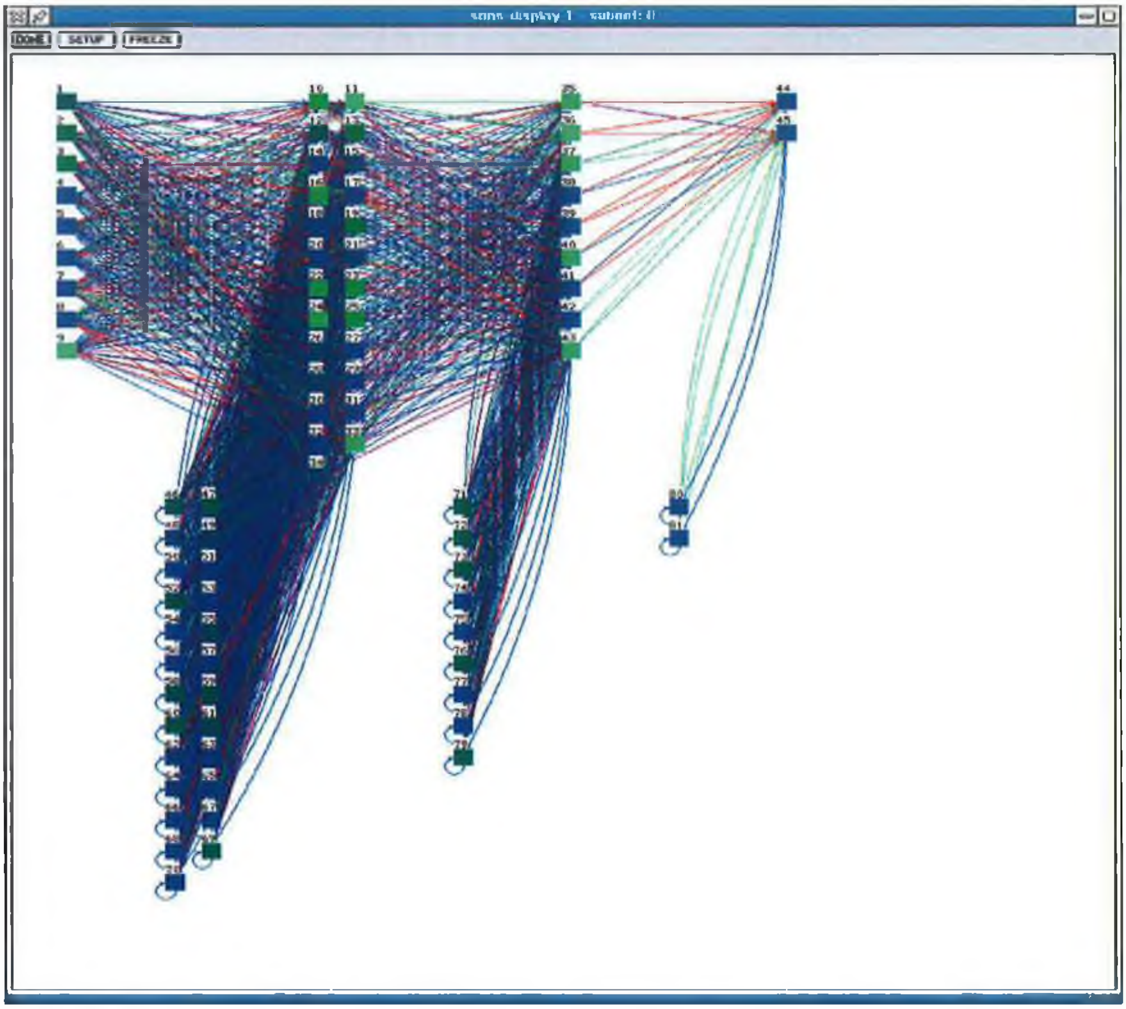


Figure 5.2: Elman Neural Network

the testing or classification phase of new test material the 2^nd output of the neural network does indicate accurately shots being part of a dialog segment but also tends to merge subsequent dialogs which explains the missed dialogue scenes in Table 5.1

5.3 Cast Listing in Motion Pictures

There are a number of approaches towards major cast detection reported in the literature. [Satoh et al., 1999] use visual and text information to associate faces with names. [Liu and Wang, 2001] use speaker-face correlation to identify anchorpersons, but the target applications were primarily news broadcasts. To generate a list of major players, it should be noted that principal characters take up considerable screen time so that they are remembered. We can measure the degree of importance of a character (from the output of dialog detection module) as the ratio between the length of its occurrence and the duration of the dialog. Principal actors also take up more space, generally in focus in the center of the screen. This is especially true of films. Dialogue scenes with over-the-shoulder shots, close-ups and point-of-view shots mean that the character's face occupies a large portion of the screen space and thus ensures that the viewer is more likely to remember him. Given this, a cast listing could be generated by clustering features extracted from dialogs - including face information (degree of confidence, most frontal face selection from sequence, face framing, degree of importance/frame/shot/scene), and selecting the set which corresponds to the most relevant ones. Presented with the results of detected major characters, users may easily digest the main scheme by skimming the list of major casts and sampling related video clips.

With the identified dialogue scenes as semantic items, we could use the face clustering technique presented in Section 3.4 to generate major cast clusters and to give the user the temporal location of the appearing characters within the movie.

5.4 Summary

In this chapter a method for major character detection in movies is presented. The posed method starts from locating the major characters in a movie based on identifying the dialogue scenes. A general framework for character detection, for combining multiple cues, for different applications is necessary and useful.It this framework, it is worth considering the techniques used by film producers. In this way, the cinematic methods used by film directors give extra useful information. Nonetheless, there are unlimited variations, especially because directors tend to improvise, thus making this kind of analysis very challenging. The above framework could be especially enhanced for indexing, search, retrieval by associating textual data to the extracted "pseudo-semantic" information based on low-level feature analysis.

Chapter 6

Conclusions

Our reseach work dealt with video indexing adapted to genre. To do this, automatic low- and mid-level features have been extracted and a number of methods have been described to perform news story segmentation and major character detection in movies.

Chapter 2 describes the existing techniques to segment video streams into basic units called shots which are used as building blocks in video indexing and retrieval. Keyframes extracted from a shot characterize or "summarize" a shot into one representational unit. To represent a shot unit a number of low-level audio-visual features can be extracted automatically for any given video programme using colour analysis, edge detection, motion characterisation, audio analysis etc. With these features we built the framework to segment a broadcast into semantically meaningful units, more understandable from the end-user's point of view. Usually low-level features are the signature representing shots in video databases (often wrapped into MPEG-7 XML format) and are used for retrieval.

Face detection is a challenging problem in computer vision. Chapter 3 gives an overview of the existing methods in the literature which address this problem and proposes an approach mostly oriented for video. Using video as our material has some advantages in performing face detection. A combination of skin-colour based filtering and sking mask processing followed by Principal Component Analysis using a face database is described with experimental validation.

News broadcast are an important segment of the media, therefore we addressed the problem of news story segmentation to create a table-of-contents in terms of individual news stories. A news story very often begins with a shot of the anchorperson reading the news and we use this clue in our approach. Three methods are described here: two shot-clustering based techniques with constraints and a more general, neural network based framework. It is expected that a learning-based approach will outperform a rule-based approach and we compare these methods in terms of their efficiency.

The last chapter deals with a more general task of identifying major characters in movie content. A recurrent neural network using low- and mid-level audio-visual features is trained to first identify dialog scenes, upon which face clustering could be applied to present the user with a list of main actors which appear in the movie together with the associated temporal points.

To increase face detection robustness and efficiency, future work should address facial feature extraction for face verification and constellation analysis on top of the method proposed in this thesis. Facial feature analysis in conjunction with skin filtering could be a powerful tool in automatic face detection. In case of face clustering, we should investigate other techniques like unsupervised clustering using Kohonen maps and identify representative face clusters without using preliminary face databases.

Segmenting news broadcasts into individual news stories is an important step. However, future work should be concerned with identifying higher level structures which appear during a usual news programme. Examples are: financial briefings, weather, political interviews etc. but this task would require advanced learning techniques using multi-modal features.

Movie analysis is a relatively new area of research and end-applications are quite diverse. With broadband technology becoming available to home users movie characterization, scene detection and scene characterization could prove to be very helpful in the users' choice and a quick search tool for specific scenes in large online movie databases.

References

A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba. Video Indexing Using Motion Vectors. In *Proceedings of SPIE Visual Communications and Image Processing*, volume 1818, pages 1522–1530, 1992.

A.A. Alatan. Automatic Multi-Modal Dialogue Scene Indexing. In *Proceedings of IEEE International Conference on Image Processing*, pages 374–377, 2001.

A. Albiol, L. Torres, C.A. Bouman, and E.J. Delp. Face Detection for Pseud-Semantic Labelling in Video Databases. In *Proceedings of IEEE International Conference on Image Processing*, 2000.

D. Arijon. *Grammar of the Film Language.* Focal Press Ltd., London, 1981.

Y. Ariki and Y. Saito. Extraction of TV News Articles Based on Scene Cut Detection Using DCT Clustering. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 847–850, 1996.

F. Arman, A. Hsu, and M. Chiu. Feature Management for Large Video Databases. In *Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases*, volume 1908, pages 2–12, 1993.

Y. Avrithis, N. Tsapatsoulis, and S. Kollias. Broadcast News Parsing Using Visual Cues: a Robust Face Detection Approach. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 3, pages 1469–1472, 2000.

R. Barzilay and M. Elhadad. Using Lexical Chains for Text Summarization. In *Proceedings of Intelligent Scalable Text Summarization Workshop, Madrid, Spain*, 1997.

M. Bertini, D. Bimbo, and P. Pala. Context-based Indexing and Retrieval of TV News. *Pattern Recognition Letters*, 22(5):503–516, April 2001.

J. S. Boreczky and L. Rowe. Comparison of Video Shot Boundary Detection Techniques. In *Proceedings of IS&T/SPIE Storage and Retrieval for Still Image and Video Databases IV*, volume 2670, pages 170–179, February 1996.

A.C. Bovic, M. Clark, and W.S. Geisler. Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

M. Brady. Computational Approaches to Images Understanding. *Computing Surveys*, 14:3–71, 1982.

G. Burel and D. Carel. Face Localization via Shape Statistics. In *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, pages 154–159, 1995.

J. Cai and A. Goshtasby. Detecting Human Faces in Color Images. *Image and Vision Computing*, 18:63–75, 1999.

J. Calic, S. Sav, E. Izquierdo, S. Marlow, N. Murphy, and N. O'Connor. Temporal Video Segmentation for Real-Time Key Frame Extraction. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 3632–3635, 2002.

J.S. Kim C.H. Lee and K.H. Park. Automatic Human Face Location in a Complex Background using Motion and Colour Information. *Pattern Recognition*, (11):1877–1889, 1996.

L. Chaisorn, T-S. Chua, and C-H. Lee. The Segmentation of News Video Into Story Units. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 1, pages 73–76, 2002.

D. Chen, J.-M. Odobez, and H. Bourlard. Text Detection and Recognition in Images and Video Frames. *Pattern Recognition*, 37(3):595–608, March 2004.

L. Chen and M.T. Ozsu. Rule-Based Scene Extraction from Video. In *IEEE International Conference on Image Processing*, volume 2, pages 737–740, 2002.

M. Christel, A. Hauptmann, H. Wactlar, and T. Ng. Collages as Dynamic Summaries for News Video. In *Proceedings of ACM Conference on Multimedia*, 2002.

M.G. Christel. Visual Digests for News Video Libraries. In *Proceedings of ACM Conference on Multimedia*, pages 303–311, 1999.

S. Cooray and N. O'Connor. Facial Feature Extraction and Principal Component Analysis for Face Detection in Color Images. In *Proceedings of International Conference on Image Analysis and Recognition*, 2004.

I. Craw, H. Ellis, and J.R. Lisman. Automatic Extraction of Face-Features. *Pattern Recognition Letters*, pages 183–187, 1987.

J.L Crowley and F. Berard. Multi-Model Tracking of Faces in Video Communications. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.

C. Czirjek, N. O'Connor, S. Marlow, and N. Murphy. Face Detection and Clustering for Video Indexing Applications. In *Proceedings of Advanced Concepts for Intelligent Vision Systems*, pages 215–221, 2003.

D. DeMenthon, V. Kobla, and D. Doermann. Video Summarization by Curve Simplification. In *Proceedings of ACM Conference on Multimedia*, pages 211–218, 1998.

L.C. DeSilva, K. Aizawa, and M. Hatori. Detection and Tracking of Facial Features by Using a Facial Feature Model and Deformable Circular Template. *IEICE Transactions on Information Systems*, E78-D:1195–1207, 1995.

A. Divakaran and H. Sun. A Descriptor for Spatial Distribution of Motion Activity for Compressed Video. In *Proceedings of SPIE Conference on Storage and Retrieval for Media Databases*, pages 392–398, 2000.

D.Saxe and R.Foulds. Towards Robust Skin Identification in Video Images. In *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 379–384, 1996.

Centre for Digital Video Processing Dublin City University Ireland. http://www.cdvp.dcu.ie.

E. Edwards and S. Duntley. Skin Colors. In *TIME Magazine, Aug. 28*, 1939.

S. Eickeler and S. Müller. Content Based Video Indexing of TV Broadcast News Using Hidden Markov Models. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2997–3000, 1999.

J.L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

S. Fischer, R. Lienhart, and W. Effelsberg. Automatic Recognition of Film Genres. In *Proceedings of ACM Conference on Multimedia*, pages 295–304, 1995.

A. Fitzgibbon and A. Zisserman. On Affine Invariant Clustering and Automatic Cast Listing in Movies. In *Proceedings of European Conference on Computer Vision*, volume 3, pages 304–320, 2002.

L.G. Frakas and I.R. Munro. *Anthropometric Facial Proportions in Medicine.* Thomas Books, Springfield, IL, 1987.

X. Gao and X. Tang. Unsupervised Video-Shot Segmentation and Model-Free Anchorperson Detection for News Video Story Parsing. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(9):765–776, September 2002.

S. Gauch, J.M. Gauch, and K.M. Pua. The VISION Digital Video Library Project. *The Encyclopedia of Library and Information Science*, 68(31), August 2000.

A. Girgensohn and J.S. Boreczky. Time-Constrained Keyframe Selection Technique. *Multimedia Tools and Applications*, 11(3):347–358, 2000.

H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating Faces and Facial Parts. In *Proceedings of International Conference on Computer Vision*, pages 41–46, 1995.

P.O. Gresle and T.S. Huang. Gisting of Video Documents: A Key Frame Selection Algorithm Using Relative Activity Measure. In *Proceedings of* 2^{nd} *International Conference on Visual Information Systems*, 1997.

B. Gunsel, A.M. Ferman, and A.M. Tekalp. Video Indexing Through Integration of Syntactic and Semantic Features. In *Proceedings 3rd IEEE Workshop on Applications of Computer Vision*, pages 90–95, 1996.

A. Hanjalic, R.L. Lagendijk, and J. Biemond. Template-based Detection of Anchorperson Shots in News Programs. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 148–152, 1998a.

A. Hanjalic, R.L. Lagendijk, and J. Biemond. Semi-Automatic News Analysis, Indexing and Classification System Based on Topics Preselection. In *Proceedings of SPIE Storage and Retrieval of Image and Video Databases VII*, volume 3656, pages 86–97, 1998b.

A. Hanjalic, G.C. Langelaar, and P.M.B. van Roosmalen. *Image and Video Databases: Restoration, Watermaking and Retrieval.* Elsevier Science, Amsterdam, The Netherlands, 2000.

Y. Haoran, D. Rajan, and C.-L. Tien. Motion Histogram: A New Motion Feature to Index Motion Content in Video Segment. In *International Conference on Knowledge and Information Engineering*, 2004.

A.G. Hauptman and M.J. Witbrok. Story Segmentation and Detection of Commercials in Broadcast News Video. In *Advances in Digital Libraries, Santa Barbara, California, April 22-24*, 1998.

A. Hauptmann, R.V. Baro, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, R. Yan, and H.D. Wactlar. Informedia at TRECVID 2003: Analyzing and Searching Broadcast News Video. In *Proceedings of (VIDEO) TREC 2003 (Twelft Text Retrieval Conference), Gaithersburg, Maryland, USA, November*, 2003.

S. Haykin. *Neural Networks.* Macmillan College Publishing Company, 1994.

N. Herodotu, K.N. Plantaniotis, and A.N. Venetsanopoulos. Automatic Location and Tracking of the Facial Region in Colour Video Sequences. *Signal Processing: Image Communication*, 14:359–388, 1998.

R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

D. Gibson M. Kocheisen H.P. Graf, E. Cosatto and E. Petajan. Multi-modal System for Locating Heads and Faces. In *Proceedings of IEEE Internaltional Conference on Automatic Face and Gesture Recognition*, pages 277–282, 1996.

R. Hsu, M. Abdel-Mottaleb, and A.K. Jain. Face Detection in Colour Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:696–706, 2002.

W. Hsu and S-F. Chang. A Statistical Framework for Fusing Mid-Level Perceptual Features in News Story Segmentation. In *Proceedings of IEEE International Conference on Multimedia and Expo*, 2003.

I. Ide, K. Yamamoto, and H. Tanaka. Automatic Video Indexing Based on Shot Classification. In *Proceedings of First International Conference on Advanced Multimedia Content Processing, Osaka, Japan*, volume 1554. Lecture Notes in Computer Science, Springer-Verlag, 1999.

I. Ide, R. Hamada, S. Sakai, and H. Tanaka. Scene Identification in News Video by Character Region Segmentation. In *ACM Multimedia Workshops*, pages 195–200, 2000.

A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1988.

A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, NJ, 1989.

S.H. Jeng, H.Y.M Liao, C.C. Han, M.Y. Chern, and Y.T Liu. Facial Feature Detection Using Geometrical Face Model: An Efficient Approach. *Pattern Recognition*, 31: 273–282, 1998.

M. Jordan. Generic Constraints on Underspecified Target Trajectories. In *Proceedings of the International Joint Conference on Neural Networks*, volume I, pages 217–225, 1989.

J.Yang and A.Waibel. A Real-Time Face Tracker. In *Proc of the IEEE 3rd Workshop on Applications of Computer Vision*, 1996.

H. Kauppinen, T. Seppanen, and M. Pietikainen. An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:201–206, February 1995.

T. Kikukawa and S. Kawafuchi. Development of an Automatic Summary Editing System for the Audio Visual Resources. *Transactions of the Institute of Electronics, Information and Communication Engineers*, J75-A(2), 1992.

V. Kobla, D. Doermann, and K.-I. Lin. Archiving, Indexing and Retrieval of Video in the Compressed Domain. In *Proceedings of SPIE on Multimedia, Storage and Archiving Systems*, volume 2916, pages 78–89, 1996.

C. Kotropoulos and I. Pitas. Rule-Based Face Detection in Frontal Views. In *Proceedings if IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2537–2540, 1997.

B. Lehane, N. O'Connor, and N. Murphy. Dialog Scene Detection Using Low and Mid-Level Features. In *International Workshop on Image, Video and Audio Retrieval and Mining*, 2002.

Y. Li and C.-C. J. Kuo. *Video Content Analysis using Multimodal Information for Movie Content Extraction, Indexing and Representation.* Kluwer Academic Publishers, 2003.

R. Lienhart, C. Kuhmunch, and W. Effelsberg. On the Detection and Recognition of Television Commercials. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 509–516, 1997.

W-H Lin and A. Hauptmann. News Video Classification Using SVM-based Multimodal Classifiers and Combination Strategies. In *Proceedings of ACM Conference on Multimedia*, pages 323–326, 2002.

Z. Liu and Y. Wang. Major Cast Detection in Video Using Both Audio and Visual Information. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001.

B.K. Low and M.K. Ibrahim. A Fast and Accurate Algorithm for Facial Feature Segmentation. In *Proceedings of IEEE International Conference on Image Processing*, 1997.

S.G. Mallat. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11: 674–693, July 1989.

B.S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Standard.* New York, Wiley, 2001.

J. Meng, Y. Juan, and S.-F. Chang. Scene Change Detection in an MPEG Compressed Video Sequence. In *Proceedings of IS&T/SPIE Digital Video Compression: Algorithms and Technologies*, volume 2419, pages 14–25, 1995.

A. Merlino, D. Morey, and M. Maybury. Broadcast News Naviagation using Story Segmentation. In *Proceedings of ACM Conference on Multimedia*, 1997.

A. Mittal, L-F. Cheong, and A. Nair. Parsing Video Programs into Individual Segments Using FSA Modeling. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 429–432, 2002.

M. Moghaddam and A. Pentland. probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19: 696–710, 1997.

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

A. Nagasaka and Y. Tanaka. Automatic Video Indexing and Full-Video Search for Object Appearances. *Visual Database Systems*, 2:113–127, 1992.

N. O'Connor, C. Czirjek, S. Deasy, S. Marlow, N. Murphy, and A. Smeaton. News Story Segmentation in the Físchlár Video Indexing System. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 418–421, 2001.

N. O'Hare, A. Smeaton, C. Czirjek, N. O'Connor, and N. Murphy. A Generic News Story Segmentation System and its Evaluation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1028–1031, 2004.

C. O'Toole, A. Smeaton, N. Murphy, and S. Marlow. Evaluation of Automatic Shot Boundary Detection on a Large Video Test Suite. In *Proceedings of The Challenge of Image Retrieval - 2^{nd} UK Conference on Image Retrieval*, 1999.

S.-C. Pei and Y.-Z. Chou. Efficient MPEG Compressed Video Analysis Using Macroblock Type Information. *IEEE Transactions on Multimedia*, 1(4):321–333, December 1999.

S. L. Phung, A. Bouzerdoum, and D. Chai. A Novel Skin Color Model in YCbCr Colorspace and its Application to Human Face Detection. In *Proceedings of IEEE International Conference on Image Processing*, volume I, pages 289–292, 2002.

M.J. Pickering, L. Wong, and S.M. Rüger. ANSES: Summarization of News Video. In *Proceedings of International Conference on Image and Video Retrieval*, pages 425–434. Springer-Verlag, 2003.

W.K. Pratt. *Digital Image Processing.* John Wiley, New York, 1991.

K.R. Rao and J.J. Hwang. *Techniques and Standards for Image, Video and Audio Coding.* Prentice-Hall PTR, 1996.

B. Raytchev and H. Murase. Unsupervised Face Recognition by Associative Chaining. *Pattern Recognition*, 36:245–257, 2003.

R.Kjeldsen and J.Kender. Finding Skin in Color Images. In *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 312–317, 1996.

H.A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

Y. Rui and T.S. Huang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communications and Image Representation*, 10(4), April 1999.

Y. Rui, T.S. Huang, and S. Mehrotra. Constructing Table-of-Content for Videos. *ACM Journal of Multimedia Systems*, 7(5):359–368, 1999.

D. Sadlier, S. Marlow, N. O'Connor, and N. Murphy. Automatic TV Advertisement Detection from MPEG Bitstream. *Pattern Recognition*, 35(12):2719–2726, December 2002.

M.H. Safar and C. Shahabi. *Shape Analysis and Retrieval of Multimedia Objects.* Kluwer Academic Publishers, 2003.

F. Samaria and S. Young. HMM Based Architecture for Face Identification. *Image and Vision Computing*, 12:537–583, 1994.

S. Satoh. Towards Actor/Actress Identification in Drama Videos. In *Proceedings of ACM International Conference on Multimedia*, 1999.

S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and Detecting Faces in News Videos. *IEEE MultiMedia*, 6(1):22–35, 1999.

H. Schneiderman and T. Kanade. Probabilistic Modeling of Local Appearence and Spatial Relationships for Object Recognition. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 45–51, 1998.

B. Shen and I.K. Sethi. Direct Feature Extraction from Compressed Images. In *Storage & Retrieval for Image and Video Databases*, volume 2670, 1996.

F. Smeraldi, O. Carmona, and J. Bigun. Saccadic Search with Gabor Features Applied to Eye Detection and Real-Time Head Tracking. *Image and Vision Computing*, 18: 323–329, 2000.

B. Smyth and P. Cotter. A Personalised TV Listings Service for the Digital TV Age. *Journal of Knowledge-Based Systems*, 13(2-3):53–59, 2000.

K. Sobottka and I. Pittas. A Novel Method for Automatic Face Segmentation. *Signal Processing: Image Communication*, 12:263–281, 1998.

S.Satoh and N. Katayama. Comparative Evaluation of Face Sequence Matching for Content-Based Video Access. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 163–168, 2000.

X. Sun, B.S. Manjunath, and A. Divakaran. Representation of Motion Activity in Hierarchical Levels for Video Indexing and Filtering. In *Proceedings if IEEE International Conference on Image Processing*, 2002.

K.-K. Sung and T. Poggio. Example-Basd Learning for View-Based Human Face Detection. 20:39–51, 1998.

J. Terrillon and S. Akamatsu. Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene

Images. In *proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 54–61, 2000.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, 1999.

S. Tsekeridou and I. Pitas. Content-Based Video Parsing and Indexing Based on Audio-Visual Interaction. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(4):522–535, April 2001.

M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

P. Viola and M. Jones. Robust Real-Time Face Detection. *International Journal on Computer Vision*, 57:137–154, 2004.

H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons Learned from Building a Terabyte Digital Video Library. *IEEE Computer*, 32(2):66–73, 1999.

H. Wactlar, A. Hauptmann, M. Christel, R.A. Houghton, and A.M. Olligschlaeger. Complementaty Video and Audio Analysis for Broadcast News Archives. *Communications of the ACM*, 43(2):42–47, February 2000.

H.D. Wactlar, T. Kanade, M.A. Smith, and S.M. Stevens. Intelligent Access to Digital Video: Informedia Project. *IEEE Computer*, 29(5):46–52, May 1996.

H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, and H. Sun. Survey of Compressed-Domain Features used in Audio-Visual Indexing and Analysis. *Journal of Visual Communication and Image Processing*, 14:150–183, 2003a.

P. Wang, R. Cai, and S.-Q. Yang. A Hybrid Approach to News Video Classification with Multi-Modal Features. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia*, volume 2, pages 787–791, 2003b.

S. Wermer, U. Iurgel, A. Kosmala, and G. Rigoll. Automatic Topic Identification in Multimedia Broadcast Data. In *Proceedings of IEEE Conference on Multimedia and Expo*, pages 41–44, 2002.

W. Wolf. Hidden Markov Model Parsing of Video Programs. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2609–2611, 1997.

W. Wolf. Key Frame Selection by Motion Analysis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 1228–1231, 1996.

G. Yang and T.S. Huang. Human Face Detection in Complex Background. *Pattern Recognition*, 27:53–63, 1998.

M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analyisis and Machine Intelligence*, 24:34–58, 2002.

B.-L. Yeo and B. Liu. Rapid Scene Analysis on Compressed Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533–544, December 1995.

K.C. Yow and R. Cipolla. A Probabilistic Framework for Perceptual Grouping of Features for Human Face Detection. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 16–21, 1997.

R. Zabih, J. Miller, and K. Mai. A Feature-based Algorithmg for Detecting and Classifying Production Effects. *Multimedia Systems*, 7:119–128, 1999.

A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hübner, S. Döring, K.-U. Herrmann, T. Soyez, M. Mschmalzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner, B. Kett, G. Clemente, and J. Wieland. SNNS: Stuttgart Neural Network Simulator User Manual, version 4.2. Insitute for Parallel and Distributed High performance Systems, University of Stuttgart, http://www-ra.informatik.uni-tuebingen.de/SNNS/, 2000.

H. Zhang, Y. Gong, S. Smoliar, and S. Tan. Automatic Parsing of News Video. In *IEEE International Conference on Multimedia Computing and Systems*, pages 45–54, 1994.

H. Zhang, J. Wu, D. Zhong, and S.W. Smoliar. An Integrated System for Content-Based Video Retrieval and Browsing. *Pattern Recognition*, 30(4):643–658, April 1997.

H. J. Zhang, A. Kankanhalli, and W. Smoliar. Automatic Partition of Full-Motion Video. *Multimedia Systems*, 1(1):10–28, 1993.

Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrotra. Adaptive Key Frame Extraction Using Unsupervised Clustering. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages 866–870, 1998.

D. Ziou and S. Tabbone. Edge Detection Techniques - An Overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559, 1998.