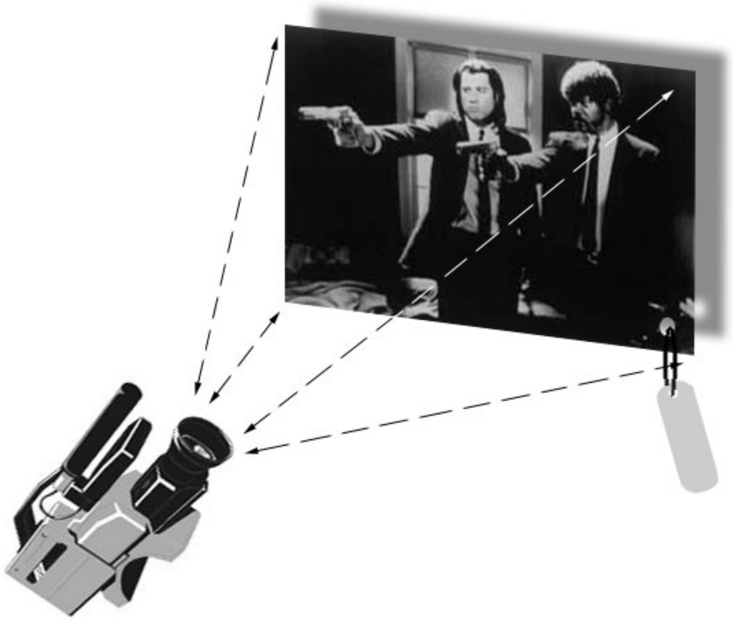
Camera Distance Classification

Indexing Video Shots based on Visual Features



Cees G.M. Snoek

October 2000

Universiteit van Amsterdam

Camera Distance Classification

Indexing Video Shots based on Visual Features

M.Sc. Thesis of

Cees G.M. Snoek

*for the completion of the study Business Information Systems,
specialization Technical Information Systems.*

Under supervision of Drs. Jeroen Vendrig.

October 2000.

Intelligent Sensory Information Systems
Informatics Institute
Faculty of Science &
Faculty of Economics and Econometrical Science
University of Amsterdam

Abstract

In this thesis we describe a method that automatically indexes shots from cinematographic video data based on the camera distance used. The proposed method can be used for automatic analysis and interpretation of the meaning of the shot within a video stream, as an assistance tool for video librarians, and as indexing mechanism to be used within a video database system. Three types of camera distance, originating from the art of filming, are distinguished. Based on extracted and evaluated visual features an integrated classification method is proposed and evaluated. It was found that, although discriminative power of some features was limited, classification of cinematographic video based on visual features is possible in the majority of shots.

Keywords: Video Indexing, Video Databases, Digital Libraries, Visual Features, Face Detection, Camera Distance, Video Classification.

Contents

Chapter 1

Introduction

Video applications are used in different fields varying from education, broadcasting, publishing and military intelligence. However, the effective usage of video is limited by a lack of viable systems that enable easy and effective organization and retrieval of information from those sources. Also, the time-dependent nature of video makes it a very difficult medium to represent and manage as was observed by Zhang et al in [28]. New techniques for the indexing and searching of digital video sources are an interesting field of research covering visual as well audio and textual data.

An ideal video information system should automatically classify the content of a given video package and provide the user with the ability to search for data of interest. Therefore content analysis of individual segments is necessary to identify appropriate index terms. To analyze the distinctive components of digital video, video sequences have to be divided into workable segments. The most natural candidates for this segmentation are video shots. Shots are unanimous regarded as the building blocks of videos, for example by Bolle and Davenport [1, 5]. Shots consist of one or more related frames that represent a continuous (camera) action in time and space.

1.1 Objectives

In this thesis we will look at shots from the perspective of a film-director. From a film-directors point of view shots can be made from different distances with respect to the *mise-en-scene*, or what appears in the film frame. This camera distance forms the basis of editing in movies and on television. Editing is the combining of shots with different camera distances to create an effective visual presentation. So camera distance can be used for the automatic analysis and interpretation of the meaning of a shot, for example to emphasize an emotional break point. Furthermore it offers a classification for the shot that can be used for indexing, which makes it suitable to use within a video database system and as a tool for video librarians. This project aims at classification of a given shot, based on the camera distance used, by evaluating features of visual information. We therefore pose the following thesis:

Based on visual features, automatic determination of the camera distance used is feasible in a vast majority of shots from cinematographic video data.

To achieve this goal we will present a series of visual features and evaluate if they are discriminatory, in itself or combined, for camera distance based classification. Since the classification originally stems from cinematography, our domain will include just video that obeys its rules. Therefore we focus on feature films like popular Hollywood productions and American TV sitcoms. The video data used has to meet one requirement: it has to feature human beings as main actors, since one of our features heavily depends on the presence of human faces. The final classification is done at the shot level as defined by the shot segmentation that accompanies our video data. We will only focus on shots that have one distance, and thus exclude shots where camera distance changes, for example when someone walks towards the camera or the camera zooms in.

Note that camera distance is an abstract term rather than technical. Because of this inexactness, distinction between distance classes is rather fuzzy. In the remainder of this report we will refer to this abstract camera distance as *the* camera distance.

1.2 Organization

In the next chapter we will first give an overview of some research on video indexing. We start the chapter with the structure of video and how it can be parsed into shots, following this we will focus on video analysis and how our research relates to this field. Then we discuss some existing video classification methods. We end chapter 2 with a camera distance classification technique on images. In chapter 3 we will define our distance domain and discuss how other types of distance relate. In chapter 4 we will focus on visual features that can be helpful to classify the used camera distance in video shots. In chapter 5 all features will be evaluated, based on which we also propose our integrated final classification method that also will be evaluated. In chapter 6 an overview of the proposed system is presented together with some implementation issues. Finally, we will end our thesis in chapter 7 by presenting our conclusions together with some directions for future research regarding this subject.

Chapter 2

Video Indexing

Several researchers, e.g. Davis and Jain in [6, 12], believe that we are currently in the crucial phase of a second *Gutenberg shift*, because after the invention of printing, digital video is causing a new revolution in human communication. It is obvious that the ever growing processing power and increasing data storage combined with new compression techniques and the advent of the Internet will lead to an exponential increase in the usage of digital video.

To cope with this increasing usage, algorithms have to be developed that enable us to parse, index, browse, search, retrieve, manipulate, and (re)sequence video according to representations of its content, as was noted by Davis in [6]. In this chapter we will give an overview of some video indexing research. We will start with highlighting the structure of video and how video streams can be parsed into shots. Following this we will focus on video analysis, and how our research relates to this field. Then we will discuss some existing methods for video classification on scene and shot level. To our knowledge no prior research is done on classification of shots based on the camera distance used. We did encounter a method that classified images based on the used camera distance. This method will be highlighted in the final section from this chapter together with an interesting technique that might be useful for camera distance based classification.

2.1 Video structure

Video indexing, or annotation, is the process of attaching content-based labels to video. This indexing forms the crux for future database-based video applications. In [5] Davenport et al stated that this indexing of video data could best be done at the shot level, because shots can be regarded as the building blocks of video. The hierarchical

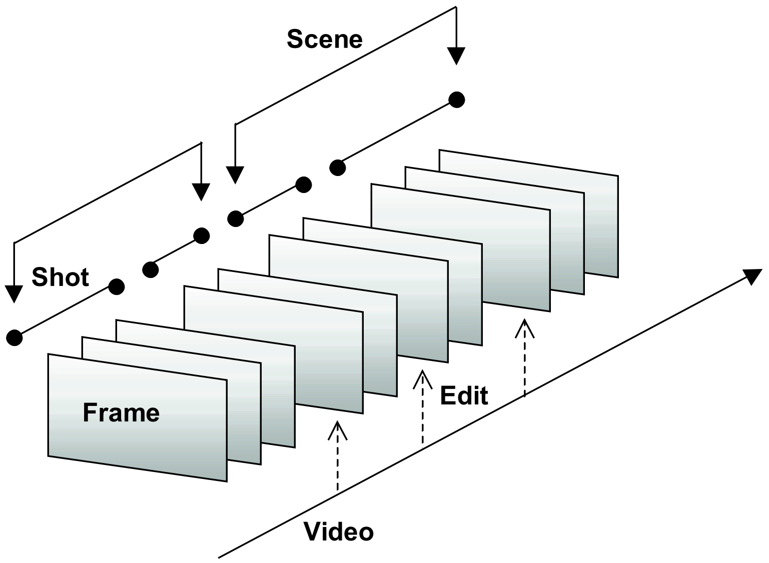


Figure 2.1: *Video structure*

structure of video is graphically represented in figure 2.1.

Basically video is ordered as a logically related collection of scenes that are composed of different interrelated shots that are separated by some sort of edit. Furthermore related scenes can be grouped into episodes or acts. To analyze and index video on the shot level, raw video data has to be parsed into shots. Different techniques already exist to segment video on the shot level, ironically called *scene-change detection*[1] techniques. These techniques differentiate edits between cuts, or camera breaks, and gradual transitions such as fade in/out, wipes and dissolves.

Cut detection techniques can be grouped into two categories: cut detection on uncompressed or compressed video. Different algorithms are proposed to detect hard cuts in uncompressed video, all of which rely on comparison of successive frames with some sort of threshold on either pixel, region, or frame level. The second category exploits the internal structure of the MPEG file standard to segment video streams. For an extensive overview of different methods we refer to the survey of Brunelli in [3] and references therein.

Detection of gradual effects can be done in several ways. Since the transition is gradual,

[1] This phrase is inherited from early research on video parsing and is nowadays also, more commonly, referred to as shot boundary detection.

comparison of successive frames is insufficient. Therefore the so called plateau detection uses every k-th frame to detect dissimilarities between frames. Another approach is based on effect modeling, where video production based mathematical models are used to spot edit effects. A third approach uses the changing of intensity edges in successive frames to detect gradual transitions but also camera breaks. Finally the twin-comparison approach introduced by Zhang et al in [28] uses a dual threshold to detect cuts and gradual transitions. We again refer to the survey of Brunelli et al in [3] and references therein for an extensive coverage of above mentioned concepts.

2.2 Video analysis

To index video segments, researchers generally agree to differentiate between content independent and content dependent features. Content independent features are also referred to as meta features, information about the data. When considering feature films this type of data typically refers to producer information or date of production. In [12] Jain et al separate content dependent features into *Q-* and *R-Features* on video and image data. Where *Q-Features* are domain model based, *qualitative* labels of video and *R-Features* are low level, domain independent *raw* models of video. The shot distance is a typical example of a video *Q-Feature*, since it labels a temporal piece of video depending on its content. A lighting level expressed in average Lux is a typical example of a video *R-Feature*, since it is a raw data value indexed by time that has to be further processed when used for labeling. In [13] Kashyap et al use a different terminology in which they speak about metadata instead of features. Moreover they differentiate between content-dependent metadata, content-independent metadata and content-descriptive metadata. They state that content-descriptive metadata, of which Q-features are an instance, are a special case of content-dependent metadata.

In [10] Hampapur et al group existing work on content based video access and video indexing into three main categories, see [10] for examples:

- *High level indexing*;
- *Low level indexing*;
- *Domain specific indexing*;

The high level indexing techniques are primarily designed from the perspective of manual indexing or annotation. The second category, low level indexing, provide access to video based on properties like color, texture, shape, etc. Primary limitation of these techniques is the lack of semantics attached to the features, which makes the utility limited from a user perspective. To overcome the limitations of both mentioned categories, domain specific indexing techniques use the high level structure of video to constrain the low level video feature extraction and processing. These techniques are effective in their intended domain of application. Primary limitation is their narrow range of applicability, and the fact that they do not consider video as a component of a video database system but in isolation, which makes these techniques not suited for a generic indexing methodology.

Hampapur et al propose a new methodology for designing video indexing schemes, which uses low level machine derivable indices to map into the set of application specific desired video indices, where the mapping is created based on the domain constraints. They view video indexing from a video database perspective while utilizing the structure inherent in video to derive the indices. Our work can be regarded as an instance of this methodology, since we will use a set of features that are mapped with respect to the camera distance used, to index video on the shot level.

2.3 Video classification

Classification of video can be done at the different hierarchical components of the video stream. In this section we will highlight some known classification methods from literature to introduce the reader to classification methodologies available, and discuss if a camera distance feature would be useful for these methods.

2.3.1 Scene classification

Some interesting research on the topic of video classification is conducted at Mannheim University, where the Movie Content Analysis (MoCA) project aims at automatically extracting structural and semantic content of video. They booked some promising results in the area of genre and scene classification. In [15] Lienhart et al developed a scheme for reliably identifying scenes which clusters shots according to detected dialogs, resembling settings and similar audio. The method they adhere starts with recovering shots from the video. After this segmentation, values for each semantic feature are calculated. Currently audio, color, and orientation features are supported as well as face detection. Next an Euclidean metric is used to determine the distance between shots with respect to each feature, resulting in a distance table. Based on the distance tables the authors are able to merge shots into scenes. The authors claim that the distance tables can be used to construct a hierarchical video representation, which ultimately would lead to an intuitive video table of contents (VToC) by finding acts, scenes and shots. Obviously this is an interesting property for video annotating systems.

The concept of VToC's was also considered by Rui et al, in [20] they present a video structure analysis tool which can assist in constructing video scene structures. Their approach to scene structure construction consists of four modules: shot boundary detection and key-frame[2] extraction, spatio-tem-poral feature extraction, time-adaptive grouping, and scene structure construction. Similar shots are grouped in a group, and semantically related groups are grouped in scenes. They base similarity of shots on *visual similarity* and *time locality*, where the former considers similar spatial and temporal features and the latter states that similar shots should be close to each other temporally. They are currently extending their work with more reliable and semantic-rich features based on audio content, close-caption content and object based content.

A camera distance feature might prove to be a beneficial feature for scene grouping. New scenes, especially new story lines, are likely to start with a long shot since this is often used to introduce the spectator to the location where the story is situated. So combined with other detected features camera distance might be a helpful clue in the detection of scene changes, and thus in scene grouping.

2.3.2 Shot classification

The research on shot classification is mainly geared towards domain dependent applications, like the classification of news-broadcasts or sport events. In [11] for example, an automatic indexing method for television news video is proposed by Ide et al, which indexes shots on the correspondence of image contents and semantic attributes of keywords (captions). They exploit a priori knowledge of the semantic structure of television news videos, and define five shot classes:

- *Speech/Report*;
- *Anchor*;
- *Walking*;
- *Gathering*;
- *Computer Graphics*;

To classify a speech/report shot the authors use face and lip movement detection. To distinguish anchor shots, the before mentioned classification is extended with the knowledge that anchor shots are graphically extremely similar and occur frequently in a news broadcast. Moreover the detection of a title caption, by using edge intensity

[2] A key-frame is a representative frame that can be regarded as a summary of a shot.

transitions, in an anchor shot is used to detect boundaries of news topics. A walking shot is classified by detecting the up and down oscillation of the bottom of a facial region. When more than two similar sized facial regions are detected in a frame, a shot is classified as gathering shot. Finally computer graphics shots are classified by the total duration of motionless frames. Their presented results for shot classification were well in terms of precision, but recall for some classes was relatively low. This was mostly due to the fact that faces were not (completely) recognized.

A comparable, but more generic, method was proposed by Fischer et al in [7] who presented an algorithm that automatically detects film genres in digital video. The authors propose a three-step approach where in the first phase syntactic properties such as color statistics, motion vectors and audio statistics are extracted. Secondly, style attributes are derived from the syntactic properties, e.g. shot lengths, camera motion, speech vs. music. Finally a style profile is composed and an *educational guess* is made as to the genre in which a shot belongs. They report promising results by combining different style attributes of video for content analysis, but unfortunately applicability is, yet, limited to only five (sub)genres, i.e. news casts, car racing, tennis, commercial and animated cartoon.

Besides domain dependent applications also more general applications are proposed in literature. In [4] for example Chan et al propose a video shot classification scheme using human faces, regardless of scale and background, which automatically detects the repeated occurrences of the same people and enables fast people related searching. First a face detector and eye localizer are applied to every shot to locate faces and corresponding eyes. Video frames that contain faces can then be clustered by utilizing one of the three following methods: a face template method, principal component analysis or clothes and hair color statistics. To perform clustering they used K-mean with Euclidean distance. It was found that shot classification performance using color statistics is relatively superior with respect to the other two methods. The scheme was tested using a short four-minute news video sequence containing three different persons. They believe that the fusion of color statistics with other personal features could prove beneficial in obtaining better classification of shots.

Though our research is limited to the domain of feature films and sitcoms, it is easy to imagine that a camera distance feature can be used for classification of other video footage also, e.g. documentaries and commercials. Ultimately this should result in a generic classification method for video shots.

2.4 Image classification

A classification method utilizing camera distance was presented by Ronfard et al in [18]. They have build some useful classes for describing television images based on extracted faces and captions. They use a shot representation based on key-frames, and extract faces and captions from it. For detected faces they define five distance classes and associate a face to one, or more, of these classes based on a quantity defined as $\frac{frame\ width}{face\ width}$. Together with the position of detected captions they claim that more specialized shot classes can be defined, an interview shot for example. We will use their proposed quantity together with two other quantities, that will be discussed in chapter 4, and will evaluate their applicability in chapter 5.

The size of detected faces can thus be used to classify an image within a distance class. Unfortunately the object filmed isn't always human, for example in movies featuring animals. To detect these *actors* a method proposed by Li et al in [14] might be useful. They propose a novel multi-resolution image segmentation algorithm for separating sharply focused objects-of-interest (OOI) from other foreground or background objects in so called low depth of field images. With these images only the OOI is in sharp focus whereas background objects are typically blurred to out-of-focus. They divide an image into blocks and classify each block as background or OOI. Features they use are the average intensity and the variance of wavelet coefficients in the high frequency bands. Their approach might be useful when used for camera distance classification because not only humans are segmented but also dogs for example. Obviously this is a necessary property to come to a more generic camera distance classification, since shots or images may contain objects other than human beings. Unfortunately the proposed algorithm can, yet, only be used to segment images with low depth of field and a sharp focussed OOI. This works fine with professional photographs, but will ultimately fail in the majority of shots from video data.

Chapter 3

Distance Domain

The frame in a video sequence not only implies space outside itself but also a position from which the material in the image is viewed. The framing used in a shot supplies the viewer with a sense of being far away from, or close to, the *mise-en-scene* of the shot as was noted by Bordwell in [2]. This aspect of framing is usually called camera distance. The standard measure that is used in judging this distance is the scale of the human body. Because there is no universal measure of camera distance, different types of distance exist. In what follows we will try to place some of those types into the three basic types we distinguish. All types are provided with examples.

3.1 Types of distance

In [2] Bordwell and Thompson distinguish eight different types of shots:

- *Extreme long shot*,

In the first one, the extreme long shot also known as establishing shot, human figures are barely visible, see figure 3.1a. This is the typical framing for landscapes, skyline views of cities, and other vistas. This type of shot is typically used to give the viewer an indication of where the story is situated.

- *Long shot*,

In the long shot, figures are more prominent, but the background still dominates and details are not evident as can be seen in figure 3.1b. This framing is often used to introduce the room or space where the scene is situated.

- *American shot, or medium long shot*,

A shot that is frequently used in Hollywood cinema is the so-called *plan américain* or American shot. Here, as in figure 3.1c, the human figure is framed from about the knees up. This shot permits a nice balance of figure and surroundings. When non-human subjects are framed from the same distance Bordwell et al don't speak of American shots but they call it a medium long shot.

- *Medium shot*,

Bordwell and Thompson define a medium shot as one that frames the human body from the waist up as in figure 3.1d. Gesture and expression become more visible in these types of shots.

- *Medium close-up*,

Besides the medium shot they also define the medium close-up, which frames the body from the chest up (figure 3.1e). Here facial expression becomes more visible.

- *Close-up*,

The close-up is traditionally the shot showing just the head, hands, feet, or a small object. It emphasizes facial expression, the details of a gesture, or a significant object, see figure 3.1f for an example.

- *Extreme close-up*,

Finally, the last shot that is distinguished by Bordwell et al is the extreme close-up, which singles out a portion of the face (e.g. eyes or lips), isolates a detail, or magnifies the minute (figure 3.1g).

3.2 Chosen representation

Unfortunately film-directors very often don't follow the strict *spectrum* of shot distances as defined by Bordwell et al. Because the framing used may vary between their distinguished boundaries, we chose to narrow their spectrum and (re)define three basic types (see table 3.1):

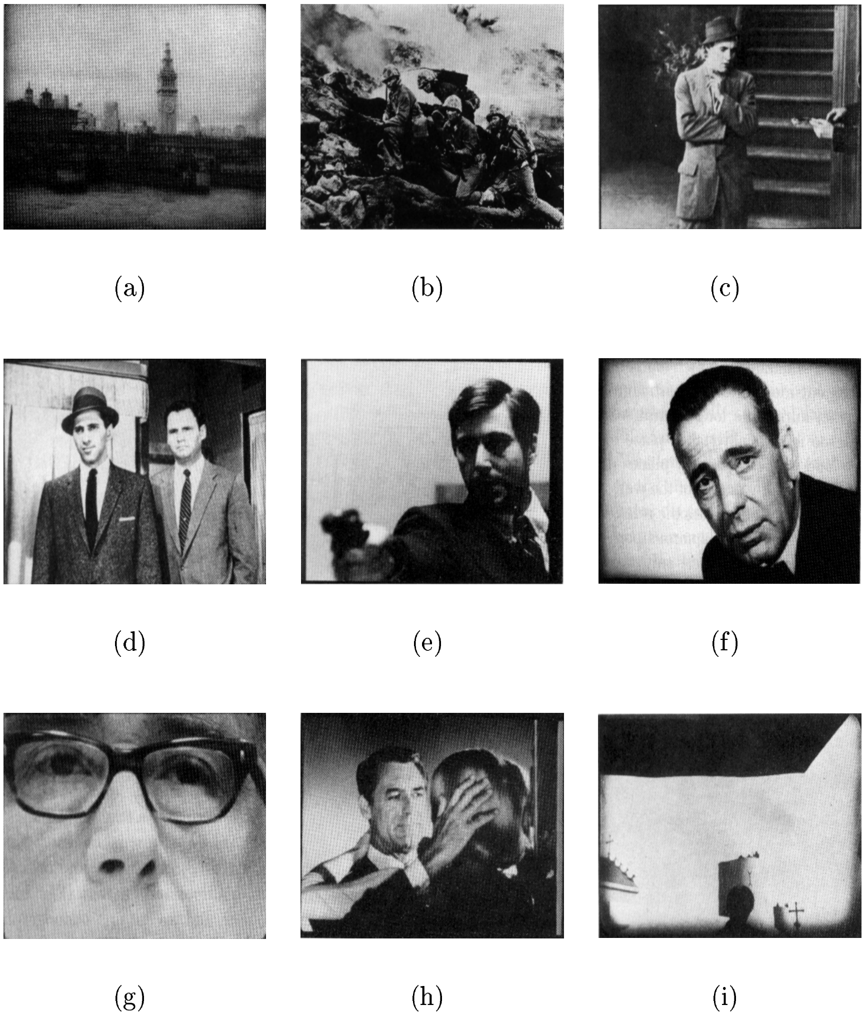


Figure 3.1: *Shot examples, as given in* [2]

- *Long shot*, this shot includes extreme long, long and American shot;
- *Medium shot*, this shot includes all shots that vary between American and medium shot until medium close-up;
- *Close-up*, this shot includes close-up and extreme close-up;

A special kind of shot is the so called *over-the-shoulder shot*. These kind of shots are taken over the shoulder of another actor, as in figure 3.1h, and are often used when two characters are interacting face-to-face. Filming over an actor's shoulder focuses the audience's attention on one actor at a time in a conversation, rather than on both.

When classifying these shots we will use the distance from camera to the actor in focus. In case of figure 3.1h classification should result in that of a medium shot.

An important observation that has to be made with respect to camera distance is that besides the distance, the size of the material framed is of great importance. This can be acknowledged as follows; from the same camera distance you could film a long shot of a person or a close-up of Kink Kong's elbow, and we would not call the shot in figure 3.1i a close-up just because only a head appears in the frame; the framing is that of a long shot because in scale the head is relatively small. Thus, in judging camera distance, the relative proportion of the material framed provides the basic determinant. Our main classification feature will be geared towards this observation, and shall be discussed in the next chapter.

Long Shot	**Medium Shot**	**Close-up**
Extreme long shot	Medium shot	Close-up
Long shot	Medium close-up	Extreme close-up
American shot		

Table 3.1: *Different shot types grouped in our three categories*

Chapter 4

Visual Features

In this chapter we will elaborate on the visual features we use to classify a shot as either long, medium or close-up. Features are distinguished by means of their measurement scope and class as defined by Vendrig and Worring in [25]. The scope refers to the extent to which a video stream component is used for a measurement, and can be local, in our case part of the frame, or global, the entire frame. The measurement class is defined as the type of video stream component on which a measurement operation is performed, and can be spatial, when measurement is performed on one frame, or spatiotemporal, in our case when measurement is performed on a whole shot. We start this chapter with discussion of local spatiotemporal features in section 4.1. Following this we will focus on global features. Therefore, in section 4.2 the global spatial features are discussed, and in section 4.3 we will elaborate on global spatiotemporal features. Finally we end this chapter with a discussion of the proposed visual features in section 4.4.

4.1 Local spatiotemporal features

The detection, or the abundance, of a human face in a frame provides us with information about the presence of human bodies. Thus the frame/face ratio ϕ can be used as a distinguishing feature for camera distance based classification, where ϕ is defined as:

$$\phi = \frac{frame\ size}{face\ size} \tag{4.1}$$

Because frame size and face size can be expressed in width, height and area we split ϕ into three different ratios:

$$\phi_{width} = \frac{frame\ width}{face\ width} \tag{4.2}$$

$$\phi_{height} = \frac{frame\ height}{face\ height} \tag{4.3}$$

$$\phi_{area} = \frac{frame\ area}{face\ area} \tag{4.4}$$

In contrast to the method proposed by Ronfard in [18] which only uses frame and face width, we also define ratios based on face height and area. Moreover we extend their local spatial approach by adding to the local detected ϕ a spatiotemporal component in subsection 4.1.4. The ϕ feature requires the successful detection of faces in video frames, therefore we need a face detector. In subsection 4.1.1 we will mention different face detection methods and stress the requirements needed within our problem statement. Following this we will discuss two different methods, one introduced by Rowley in [19] and one that combines color and shape. Finally we end this section with the classification method used to classify shots based on detected faces in all its frames.

4.1.1 Face detection methods and requirements

The goal of face detection is to identify all image regions which contain a face, regardless of its three-dimensional position and orientation and the lighting conditions used, and if present return their image location and extents [27]. This detection is by no means trivial because of variability in location, orientation, scale and pose. Furthermore, facial expressions, facial hair, glasses, make-up, occlusion, and lightning conditions are known to make detection error prone.

Over the years various methods for the detection of faces in images and image sequences are reported. In [27] Yang et al give a comprehensive and critical survey of current face detection methods. They classify still image detection methods into the following categories:

- *Knowledge-based methods.*
- *Feature invariant approaches.*
- *Template matching methods.*
- *View-based methods.*

Within our problem statement faces have to be detected invariant under change of scale, position, pose, orientation, occlusion and illumination intensity and direction, i.e. a variation in these circumstances should not result in failure of the face detection method. Therefore the chosen method has to meet these invariance requirements, or at

Invariance	**Knowledge**	**Feature**	**Template**	**View**
Scale	+/-	+	+/-	+
Occlusion	+/-	-	-	-
Illumination	+/-	+	+/-	+
Pose, frontal	+/-	+	+	++
Pose, non-frontal	+/-	+/-	-	-
Orientation, frontal	+/-	+	+/-	+/-
Orientation, non-frontal	+/-	+/-	-	-
Applicability	**-**	**+**	**-**	**+**

Table 4.1: *Face detection methods and their invariance*

least give the best approximation. When we use the observations made by Yang in [27] to compare the different methods, see table 4.1, it becomes immediately evident that the knowledge and template based methods are not suitable. In theory a knowledge-based method could be implemented to solve all the known problems inherent in face detection, in practice however it is impossible to enumerate all the possible cases. Template based methods are also not suited within our problem statement since most methods cannot deal with variation in scale and pose. Moreover templates have to be pre-defined by experts. The other two methods fit to our needs and shall be discussed in the following subsections. To detect frontal upright faces we will use a view-based method introduced by Rowley. To detect faces regardless of pose and orientation we will implement a face detector that combines multiple features, i.e. color and shape.

4.1.2 Rowley's algorithm

The face detection method proposed by Rowley in [19] is known to be one of the most reliable in digital image research, based on significant better results achieved on a similar test set when compared with other methods. The proposed system is able to detect about 90% of all upright and frontal faces, and more important the system only sporadically mistakes non-face areas as faces.

In short the method operates in two stages: the first component uses a multilayer neural network to learn face and non-face patterns from face/non-face images, the second component uses an arbitrator to combine the outputs. As shown in figure 4.1, the first component is a neural network that receives as input a 20 × 20 pixel region of an image and generates an output from 1 to -1, signifying the presence or absence of a face. To detect faces anywhere in the image, the neural network is applied at all image locations. Moreover subsampling of the input image is used to detect faces that are larger than 20 × 20 pixels. The second component merges overlapping detections and arbitrates between the outputs of multiple networks, by using simple arbitration

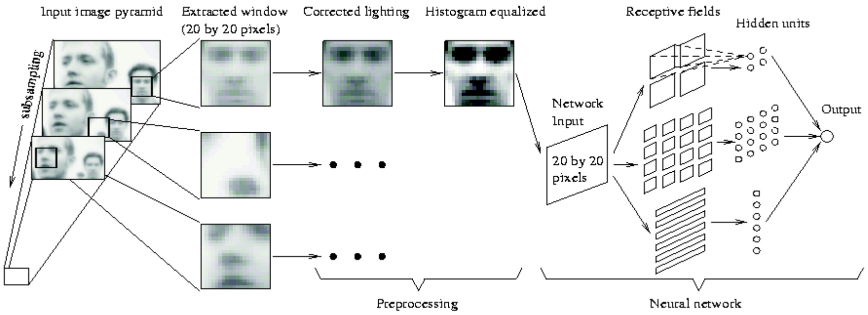


Figure 4.1: *The basic algorithm for face detection as proposed by Rowley in* [19]

schemes such as logic operators (AND/OR) and voting. For an in depth coverage of this method we refer to [19]. Note that we did not train the algorithm on our dataset but instead used the default as trained by Rowley.

Since the method is limited to detection of upright frontal faces, we need another method that is also able to detect faces invariant to changes in orientation and viewpoint.

4.1.3 Combining color and shape

The approach that combines color and shape begins with the detection of skin-like regions based on color information. Next, skin-like pixels are grouped together using connected component analysis. If the shape of the connected region is elliptic or oval, it becomes a face candidate. In case of color images, this processing order allows a very robust analysis, because faces differ significantly from the background by their color and shape. We will first motivate the choice for the color model used and how we segmented skin-like regions. Following this we will explain how we obtain faces from skin-like regions by incorporating shape features of the human face.

Skin color segmentation

Skin color forms an excellent cue for the detection of faces in complex scene images, because it is computationally fast and relatively robust to changes in illumination, orientation, viewpoint, scale and shading. Though, some problems exist when using color as a feature for the detection of faces, as was noted by Yang and Waibel in [26]. First, the color representation of a face obtained by a camera is influenced by many factors, such as ambient light, object movement, etc. Second, different cameras produce

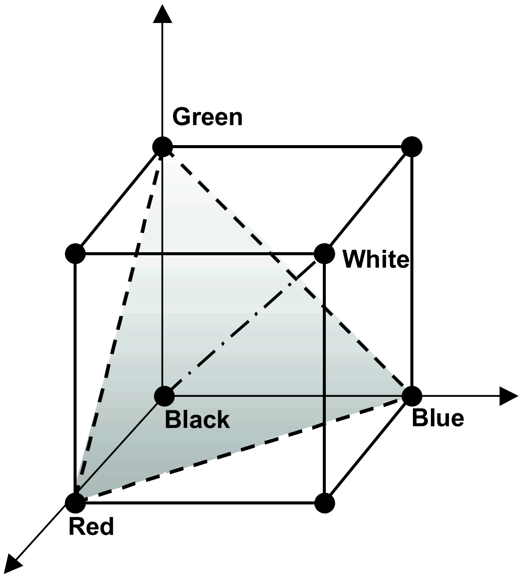


Figure 4.2: *RGB color model, with inside the rgb chromaticity triangle*

significantly different color values, even for the same person under the same lightning condition. Finally, human skin color varies between different persons. When we want to use color as a feature for the detection of faces, those problems have to be solved.

Colors are usually represented by one of the many color models. The red, green, and blue (RGB) color model is the *de facto* standard for image representations because it is widely used in color CRT monitors and color raster graphics. The model can be represented by a cube in the Cartesian coordinate system, as in figure 4.2. In this model al colors are represented by a triplet of R, G and B values. The main diagonal of the cube, connecting the black and white corners, represents the gray levels and defines the intensity I of the perceived color:

$$I(R, G, B) = R + G + B \tag{4.5}$$

Because intensity is part of the RGB color model it is not applicable for skin color-based face detection. This is acknowledged by the observation that skin color varies between different people with respect to race, gender and age and that several studies have shown that the major difference between skin colors lies in their intensity rather than their chrominance, as noted by Yang in [26]. Since intensity is not important for the characterizing of skin colors, under the normal lighting condition, we will use the normalized rgb color space to represent skin color. In [23] Terrillon and Akamatsu showed that this color space achieves an acceptable face detection performance. Furthermore they proved that normalization of RGB or CIE-XYZ values, yields chrominance spaces that are more efficient for skin color segmentation than CIE-SH and H-S spaces where such a normalization is not performed.

The rgb model is obtained by a normalization process, that defines a mapping from 3D to 2D. This mapping can be explained as follows: all points in the plane perpendicular to the main diagonal of the RGB color cube have the same intensity, thus the plane through the cube where $R + G + B = 1$, assuming white is in point (1, 1, 1), cuts out an equilateral triangle which is the standard rgb chromaticity triangle. The projection from RGB points to this chromaticity triangle is defined as:

$$r(R, G, B) = \frac{R}{R + G + B} \tag{4.6}$$

$$g(R, G, B) = \frac{G}{R + G + B} \tag{4.7}$$

$$b(R, G, B) = \frac{B}{R + G + B} \tag{4.8}$$

and graphically represented in figure 4.2.

The intensity axis I in RGB color space is projected onto $r = g = b = \frac{1}{3}$ in the chromaticity plane. Since $r + g + b = 1$ one of the features is redundant and can be discarded. Because of the sensitivity of feature b to noise, we will discard this feature. Gevers showed in [8] that, in contrast to color features RGB and I, rgb is invariant to shadows, surface orientation, illumination direction and illumination intensity for dull, matte objects. Which makes this color space suitable for color based object recognition, and thus for skin color based face detection. Drawback of the model is that, as a result of discretization of color values, the system becomes unstable when intensity is low. Furthermore, rgb is undefined for achromatic color ($R = G = B = 0$).

As was observed in [26] by Yang and Waibel, skin colors of different people are very close and differ mainly with respect to intensity. Moreover, they observed that skin colors of different people tend to cluster in rg-space. Under the assumption of a white illumination source we chose to model this cluster by a bounding rectangle. The extra noise resulting from this modeling step will be handled in the next step. After experiments on a representative collection of key-frames from different videos from our experimental data, see subsection 5.1.1, the following parameters for the r and g values were determined: $r_{min} = 0.40$, $r_{max} = 0.55$, $g_{min} = 0.23$ and $g_{max} = 0.33$. These parameters work well on images that contain white skin as well as the yellow and dark skin of human beings. For each pixel π in the frame we compute its r and g values and classify π as either skin or non-skin based on the following formula:

$$\pi_{skin}(x, y) = \begin{cases} \textit{skin}, & \text{if } r_{min} < r(\pi) < r_{max} \ \wedge \ g_{min} < g(\pi) < g_{max}; \\ \textit{non-skin}, & \text{otherwise}; \end{cases} \tag{4.9}$$

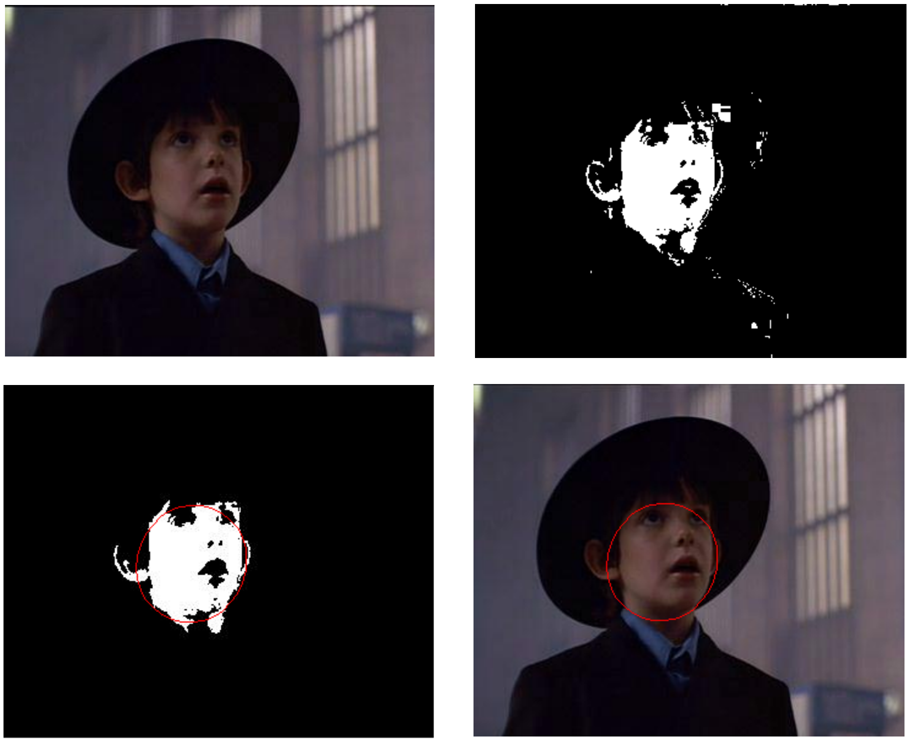


Figure 4.3: *Top left: the original video frame; Top right: the frame after skin color segmentation; Bottom left: the frame after connected component and shape analysis; Bottom right: frame with detected face;*

In figure 4.3 some results of the skin segmentation are visible, skin regions are detected correctly. Together with this successfully detected pixels some noise is detected in the background.

Recovering faces from shape

As becomes visible in figure 4.3 skin color helps in detecting human faces, but skin color alone cannot detect human faces correctly, because it also generates some noise. Therefore, the segmented frame is further processed by doing a shape analysis. Since the human face can be regarded as a more or less connected skin colored region, we first perform a connected component analysis on the segmented frame as was suggested by Sobottka and Pitas in [22]. The connected components are determined by applying a region growing algorithm at a coarse resolution of the segmented image, where connectivity is determined on the base of the 4-pixel neighborhood. The region growing algorithm removes falsely detected isolated pixels in the background as can be seen in figure 4.3. We are now left with a large connected area for the face, and several small

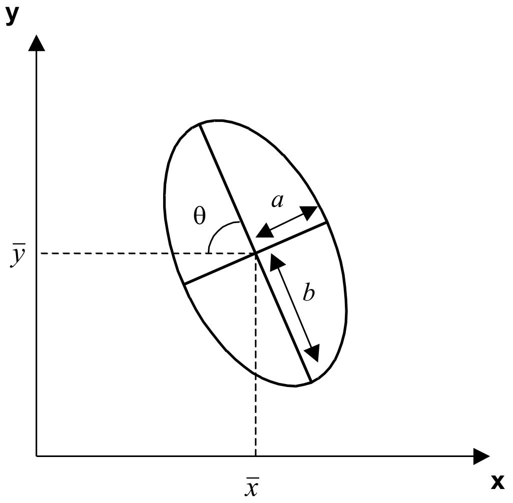


Figure 4.4: *An ellipse and its parameters*

connected components in the background. Based on shape information we are now able to make a further reduction of candidate regions.

An ellipse can approximate the oval shape of a face. Therefore looking for faces in frames could be performed by detecting objects with nearly elliptical shape. In [22] Sobottka et al stated that this can be done based either on edges or on regions. Because regions have the advantage of being more robust against noise and changes in illumination, we opt for this method. Therefore we compute for each connected component C the best-fit ellipse E on the base of moments. An ellipse is exactly defined by its center $(\overline{x}, \overline{y})$, its orientation θ and length a and b of its minor and major axis, see figure 4.4.

On the base of the computed elliptical parameters (see appendix A), we reduce the number of face candidates by applying to each ellipse two decision criteria concerning the aspect ratio and the number of skin pixels inside the ellipse. The aspect ratio defined as $\frac{2b}{2a}$ should vary between 0.4 and 1.6, face candidates with aspect ratios outside this domain are unlikely to be true faces, and are thus discarded. Besides the aspect ratio we also count the number of skin pixels inside the ellipse. When the number of skin pixels is less that 40% of the face area, defined as $2a \times 2b$ (the bounding box around the ellipse), a face candidate is also discarded. Ultimately this should result in the successful detection of faces in video frames, as in figure 4.3.

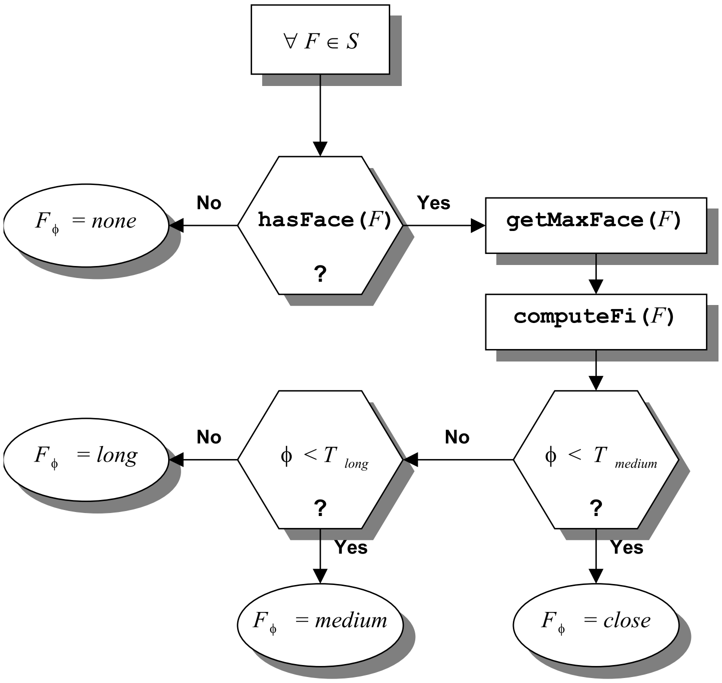


Figure 4.5: *Face based frame classification scheme*

4.1.4 Face based classification

Now that we are able to detect faces in a single video frame, it is possible to make a face based camera distance classification on video shots using the different frame/face ratios ϕ. In chapter 5 we will evaluate which ϕ gives best results, we will now first explain the face based classification algorithm. Note that faces detected with Rowley's algorithm are identified by squares opposed to the method that uses color and shape which models faces by means of ellipses. The area of the ellipse is represented by the bounding box around it $2a \times 2b$, width by $2a$ and height by $2b$. Because of this difference in face identification each ϕ has to be determined twice.

To make a face-based camera distance classification on the shot level, for each shot S we extract each frame F, $F \in S$, and check whether or not F contains a face. If F contains one or more faces the largest is used to compute the concerning ϕ. Based on the values of thresholds T_{medium} and T_{long} the frame with ratio ϕ, F_ϕ, is labeled as *close, medium* or *long*, see figure 4.5. When F contains no faces F_ϕ is labeled as *none*. In chapter 5 we will discuss which values for thresholds T_{long} and T_{medium} give best results for each ϕ.

After each frame, F_ϕ, is labeled as either close, medium, long or none, we are able to classify the whole shot S_ϕ, based on the percentages of F_ϕ from each class, i.e.

perClose, perMedium, perLong or perNone, within a shot. We have found it is not sufficient to assign a label to each shot based on the maximum percentage from a certain class because this resulted in too much shots classified as *none*, i.e. no face detected. Therefore we developed a sequential set of rules, like the ones given below:

```
IF perClose > 20 AND perMedium < 45 AND perNone < 75
   THEN close;
IF perMedium > 20 AND perNone < 75
   THEN medium;
IF perMedium > 20 AND perLong > 20 AND perLong < 30
   THEN medium;
...
```

Primary aim of this set of rules is to favor the detection of close-ups with respect to medium shots, long shots and none. The detection of medium shots with respect to long shots and none, and finally long shots with respect to none. This sequential rule-based favoring mechanism induced much better face based shot classification results than the one based on maximum.

4.2 Global spatial features

In this section we will describe three global spatial features. The features are global spatial since they are operated on the key-frame of a single shot.,We will first explain how an edge detector can help in discriminating between close-ups and long shots. Following this we will explain how the size of a detected homogeneous colored region relates to the unlikeness of a close-up.

4.2.1 Number of edge pixels

The first global spatial feature we will use are the number of edge pixels in a key-frame. To find edges in an image, generally a differentiating filter is used to extract local information. In theory this is accomplished by taking the derivatives in x and y direction resulting in the gradient vector $\nabla f(x, y)$. In practice however, taking the derivative is sensitive to noise and therefore seldom used in practical applications as noted by Smeulders in [21]. To accommodate for the noise, differentiation is often combined with some kind of smoothing filter. The *Sobel* operator combines both, the horizontal and vertical differential values are calculated by a convolution with the

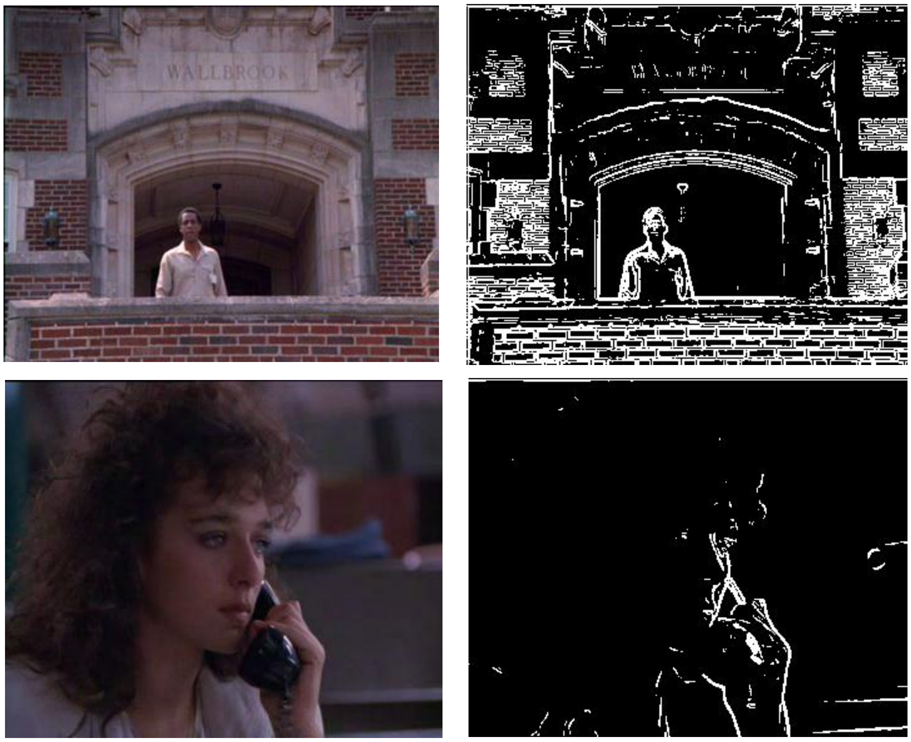


Figure 4.6: *Top: a long shot with complex patterned background resulting in 22915 detected edge pixels out of 101376; Bottom: a close-up with smooth background resulting in 2490 detected edge pixels. Note that the foreground object also generates few edge pixels*

masks (horizontal respectively vertical):

$$\left\{ \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \right\} \qquad \left\{ \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \right\} \tag{4.10}$$

We expect that this feature gives us the opportunity to distinguish long shots from close-ups. The rational behind this can be derived from the shot definitions. Recall from section 3.1 that in the long shot background dominates, contradicted to close-ups where focus is on a foreground object. A camera that focuses on the foreground will likely contain a somewhat blurred background resulting in less detected edge pixels when compared to long shots, which are likely to contain a complex patterned background, e.g. a building. This complex background will thus, in general, contain more edge pixels when compared to close-ups, see figure 4.6 for an example.

4.2.2 Region size

We define a region as a part of an image that shares a specific homogeneity criterion. In short the method we adhere to uses a color segmentation that identifies all homogeneous colored regions present in an image. Regions are segmented by means of multivalued morphological filters and a watershed algorithm, based on the method proposed by Gu in [9]. Based on the size of detected regions one of 64 histogram bins is incremented. Finally a threshold B_{large} is performed on the measurement interval, i.e. the histogram, resulting in a classification of the image/shot as one that contains one or more large regions or none.

This global spatial feature has some overlap with the number of edge pixels feature as mentioned in subsection 4.2.1. But is sufficiently different to use within our problem statement. The rational behind it is as follows: in general a key-frame of a close-up will show a detailed composition that consists of different colored regions. Moreover, a percentage of long shots will contain a large connected region, e.g. a clear blue sky or a green field. Unfortunately dark frames, i.e. frames filmed in darkness, also generate large connected (dark) regions so the method is not applicable to identify long-shots, but we expect that a key-frame with a region larger than a certain threshold is unlikely to be a close-up.

4.3 Global spatiotemporal features

In contrast to the global spatial features that were treated in section 4.2, we will describe in this section three features that also exploit the temporal properties of video shots. We start with the detection of camera work. Following this we will focus on shot duration and we will end with the average number of colors in a shot. Since all features depend on the temporal aspect and the whole frame, all are instances of a global spatiotemporal feature.

4.3.1 Camera work

Camera work is a collective noun for operations that can be performed by a camera during a shot. This is not limited to camera movement only, but also includes camera specific properties like zooming. There are several kinds of camera work, each creating its own specific effect, see also figure 4.7:

- *Pan*: rotates the camera on a vertical axis;

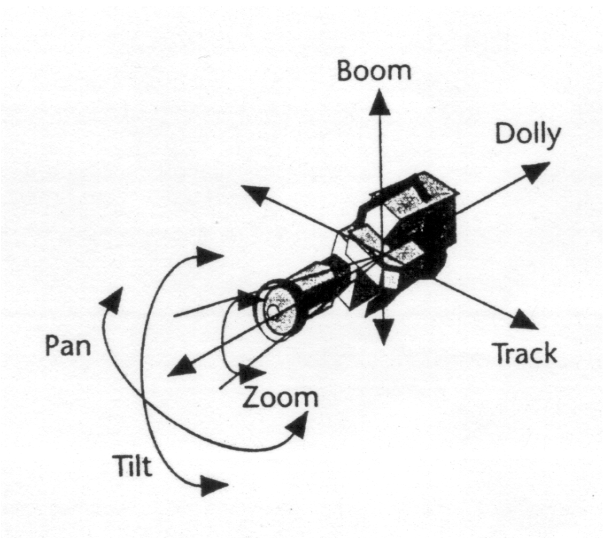


Figure 4.7: *Camera work* [24]

- *Tilt*: rotates the camera on a horizontal axis;
- *Dolly*: physically moving close to or away from subject;
- *Boom*: vertical camera movement to keep object in specific position;
- *Track*: horizontal camera movement to keep object in specific position;
- *Zoom*: magnifies or demagnifies the objects filmed;
- *Static*: no camera movement, nor effect;

Extracting the camera work in an arbitrary shot is a complicated task, we will use the method proposed by Nguyen in [17]. The method is limited in its applicability since it is only able to detect panning, zooming and static. These are the most popular kinds of camera work however, so the method will be sufficient in our case.

It is easy to imagine that when a zoom occurs in a shot the camera distance is likely to change. Therefore we will not classify shots that contain a zoom (in or out). Moreover we observed that shots that contain a pan are also likely to contain a change of camera distance, for example when the camera pans from one person to another person within a shot and distance changes from long to medium. So shots that contain a pan will also not be classified. Since the other types of camera movement are less likely to appear, we will neglect them and only focus on detected static shots.

The method for computation of instantaneous work of the camera between two consecutive frames F_{i+1} and F_i, first checks the presence of zooming. For completeness we repeat the algorithm as described by Nguyen in [17]. For a zoom the instantaneous motion vectors of pixels are described by the following model with parameters x_0, y_0 and Ω:

$$\begin{cases} v_x = \Omega(x - x_0) \\ v_y = \Omega(y - y_0) \end{cases} \tag{4.11}$$

where x, y represent pixel coordinates (counting from the image center), x_0, y_0 represent the zoom centroid, i.e. the point where the motion vectors converge or diverge, and Ω represents the magnitude of the zoom. x_0, y_0 and Ω are estimated via least-squares minimization as follows:

$$\min_{x_0, y_0, \Omega} \sum_{x,y} [F_{i+1}(x,y) - F_i(x - v_x, y - v_y)]^2 \tag{4.12}$$

If indeed there was a zoom, (x_0, y_0) should fall within the frame and $|\Omega|$ should be large enough, say: $|\Omega| > 2/s$, where s is the size of the frame. Zoom-in or zoom-out is determined according to whether $\Omega > 0$ or $\Omega < 0$. If the algorithm fails to detect a zoom, the minimization in (4.12) is repeated but with the translation model with two parameters T_x and T_y:

$$v_x = T_x \quad \text{and} \quad v_y = T_y$$

The obtained values of T_x and T_y allow to determine whether the camera is static or panning and if so in which direction.

To classify the camera work for the whole shot, the above algorithm is applied for six points equally spaced over the shot sequence. If two or more of the four detectable camera works (zoom-in, zoom-out, pan left or pan right) are detected the answer with majority is selected. If pan and zoom are both two or more times detected in the shot, panning is favored with respect to zooming. If no zoom or panning is detected, camera work defaults to static.

4.3.2 Shot duration

Every shot has a measurable duration, δ, defined as:

$$\delta = S_{end} - S_{start} + 1 \tag{4.13}$$

Where S_{end} is the end frame of the shot and S_{start} the start frame. δ can be a clue for discriminating between close-up and long shots, since in the latter the time necessary for the spectator to scan the shot for particular points of interest should be longer. This thus excludes long takes from being classified as close-up and short takes from being classified as long shot. To evaluate this feature, δ will be compared with the average shot duration, δ_{avg}, used in the particular video stream.

4.3.3 Average number of colors

The average number of colors in a shot, C_{shot}, is defined as:

$$C_{shot} = \frac{\sum_{S_{start}}^{S_{end}} C_{frame}}{\delta} \tag{4.14}$$

Where C_{frame} denotes the total number of different RGB-triplets in a frame, and δ is the duration of the shot as defined in equation 4.13.

This feature can be used as a detector of (extreme) close-up shots of objects, for example a lock on a door or a letter. Obviously such shots contain little variation in color, since objects are colored uniform. To test the usability of this feature, C_{shot} will be compared with an experimentally defined threshold.

4.4 Discussion

In this chapter we introduced a series of visual features that might be suitable for camera distance based classification of video shots. The features can be organized by means of their scope and class as defined by Vendrig and Worring in [25]. Also a type classification of features can be made based on observations from Jain [12], as mentioned in section 2.2. In this chapter we added to the features a matter of applicability with respect to camera distance based indexing. In what follows we would like to generalize this applicability.

It was found that not all features are suited to serve as a *classifier*, where classifier is defined as a feature that is capable to label a given shot into each of its three distance classes. Some were used as a *separator* and were only capable to distinguish between two distance classes, e.g. close and long. We also identified features that were able to *confirm* that a shot belongs to one certain class. Of course the opposite is also possible, we would like to call such a feature a *negator* because of its ability to ensure that a given shot does not belong to a certain distance class. And finally we also distinguish a *domain* feature that is capable of classification of a given shot within, or outside the application domain used. In our case the camera work feature is used to ensure that only static shots are processed. In table 4.2 an overview of the different proposed features is given together with their properties.

Feature	**Scope**	**Class**	**Type**	**Applicability**
ϕ *Rowley*	local	spatiotemporal	R	classifier
ϕ *Color/Shape*	local	spatiotemporal	R	classifier
Edge pixels	global	spatial	R	separator
Region size	global	spatial	R	negator
Camera work	global	spatiotemporal	Q	domain
Duration	global	spatiotemporal	meta	negator, separator
Colors	global	spatiotemporal	R	confirmer

Table 4.2: *Properties of proposed features*

Chapter 5

Distance Classification and Evaluation

In this chapter we will evaluate the visual features which we proposed in chapter 4, and present our integrated camera distance classification method that also will be evaluated. We will first start in section 5.1 with explaining the procedure we used for evaluation. In section 5.2 all features will be mapped to a distance value, evaluated and judged on their discriminating power with respect to camera distance. Based on results of this evaluation we propose a classification method in section 5.3, that also will be evaluated.

5.1 Procedure

In this section we will briefly present the experimental procedure we followed for evaluation of the features mentioned in chapter 4. We will start with the definition of the experimental video data set used. Following this we will elucidate the rules used to construct the groundtruth. Finally we will explain how we visualize classification results, and how to interpret them.

5.1.1 Experimental data

The experimental data we used was taken from the *ISIS* video database. This database currently consists of a collection of feature films and TV sitcoms. The feature films vary in genre: action, drama, and comedy are all present. The data is digitized in MPEG-1 format. We experimented with a total of near 2000 presegmented shots. To evaluate

Video data	*Close*	*Medium*	*Long*	*Total*	*undefined*
Friends 417	0	234	98	332	11
Life of Brian cd1	4	50	67	121	30
Rainman cd1	15	123	54	192	29
Witness cd1	72	110	39	221	28
Friends 418	0	225	90	315	14
A View to a Kill cd1	68	117	205	390	28
Witness cd2	63	96	76	235	8
Total	222	955	629	1806	148

Table 5.1: *Video data set, divided in training (top) and test set (bottom)*

the total amount of shots we grouped them in a *training set* and a *test set*. See table 5.1 for an overview. The training set was used to train our algorithms, where training is defined as the process of choosing the appropriate thresholds based on example data. For all shots we calculated the different features and stored them in a feature database. To evaluate the features we compared the results of the mapping from feature to camera distance, with a predefined *groundtruth*. The rules for construction of this groundtruth are described in the next subsection.

5.1.2 Building the groundtruth

Camera distance is not an exact defined distance but merely used in film industry as a convention. In chapter 3 we already defined the boundaries between the distance classes we distinguish, i.e. close-up, medium shot and long shot. The rules and procedures we used to manually classify each shot were as follows. We defined four label classes: *close, medium, long,* and *undefined*. After viewing each shot individually we used the predefined boundaries to label a shot as either close, medium or long. A shot was labeled undefined when camera distance changed during a shot or when the film director used a *trick shot*, for example when we see the reflection of a face in some kind of object. After the correct labeling of shots, we map the different features to a camera distance classification based on a certain threshold. The threshold values that give best classification results are chosen.

5.1.3 Visualization of results

We visualize the classification results by means of a *confusion matrix* as introduced by Hampapur in [10]. The reliability of the feature-based classification is measured in terms of correctness. The derived labels, acquired by means of mapping algorithms, are compared to a manually assigned label set. The confusion matrix is a $n \times n$ matrix, where n refers to the number of labels, three in case of close, medium and long. The diagonal entries in the matrix indicate the percentage of shots that are correctly classified. The off-diagonal entries indicate the percentage of misclassification performed by the feature based indexing algorithms. An ideal confusion matrix would be an identity matrix, since this indicates a perfect labeling.

5.2 Feature mapping and usability

In this section the extracted visual features, described in chapter 4, will be interpreted by tuning the different mapping thresholds, see figure 5.1. The labeled shots will be evaluated with respect to their usability. Where usability is merely defined as the discriminating power of a feature. We start this section with the evaluation of the local spatiotemporal features, i.e. the two different face detection methods. Following this we will discuss the global spatial features and we end with the global spatiotemporal features.

5.2.1 Local spatiotemporal features

To evaluate the usability of the local spatiotemporal features we compared the results of three different frame/face ratios, as mentioned in 4.1.4. Moreover we compared the results of a face detector as proposed by Rowley in [19] with one that combines color and shape features.

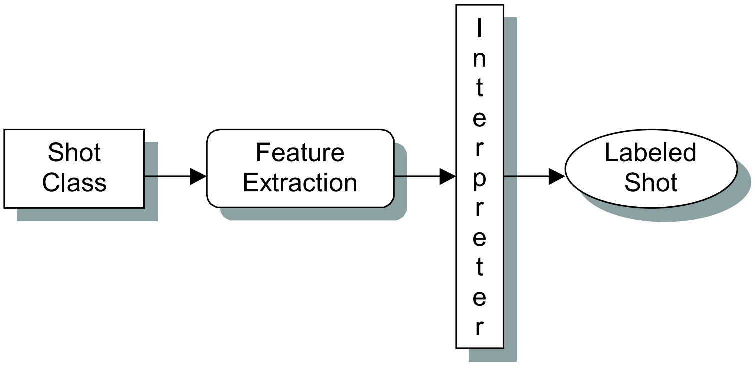


Figure 5.1: *Feature mapping*

Video data	*Close*	*Medium*	*Long*	*None*	*Total*	*undefined*
Rainman	5	119	30	56	210	11
Witness	53	107	10	57	227	22
Life of Brian	2	41	64	35	142	9
Friends 417	0	233	91	11	335	8
Total	60	500	195	159	914	50

Table 5.2: *Facetruth*

Facetruth

We first manually defined a facetruth where each shot was scanned for the presence of a face, we then related the face size to the camera distance used. This resulted in a label as either *close, medium, long, none* or *undefined.* We used the same rules for constructing the groundtruth as stated in 5.1.2. Note that the training set of the facetruth contains 50 shots that are undefined, contrasted to the original training set summarized in table 5.1 that contains 98 shots that are undefined. This is caused by the fact that many shots that are undefined in the original data set are labeled as none in the facetruth. The shots in the opening credits of *Life of Brian* for example, which contain no human faces, are labeled undefined in the original training set, but in the facetruth these shots are labeled as *none.* The manual labeling finally resulted in the *facetruth* as presented in table 5.2.

Rowley's algorithm

After initialization of the facetruth we tested the two face detectors with the different ratios as presented in 4.1.4. We first tested the method proposed by Rowley in [19], this method is known to detect about 90% of all upright and frontal faces, moreover the system only sporadically mistakes non-face areas as faces. Detected faces are identified in his system by means of a square. When we compared the different ratios we found that for all ratios similar results were obtained, see table 5.3. For ϕ_{width} thresholds were 2.0 for T_{medium} and 5.5 for T_{long}. For ϕ_{height} best thresholds were 1.6 for T_{medium} and 4.9 for T_{long}. Finally for ϕ_{area} best thresholds were 3.3 for T_{medium} and 30 for T_{long}.

When we analyze the confusion matrix, we observe that this facedetector indeed generates little noise, except for the misclassification of 10% of close-ups. This was due to a personage in *Witness*, a child. Close-ups of this personage sometimes resulted in medium classification because the detected face was smaller than that of an adult. Also detected animated faces in the opening credits of *Life of Brian* caused some misclassification, because these shots were labeled as *none* in the facetruth. This was also the

Labels	*Close*	*Medium*	*Long*	*None*	*Total*
Close	51.67%	10.00%	0.00%	38.33%	60
Medium	0.40%	71.60%	4.00%	24.00%	500
Long	0.51%	3.59%	32.82%	63.08%	195
None	0.63%	3.14%	2.52%	93.71%	159

Table 5.3: *Confusion matrix resulting from face detection method from Rowley, using* ϕ_{area}, ϕ_{width} or ϕ_{height}

case with a close-up of a newspaper article in *Witness* showing a picture of a face. Furthermore, sometimes the face of the actor in focus was missed but instead a frontal face from the background was detected, resulting in a wrong distance classification. A very small percentage of misclassification was due to shots that just fell between the wrong threshold boundaries.

Color and shape

Since the method that combines color and shape uses an ellipse to model detected faces, ratio thresholds found with Rowley's face detector were not usable. It was found that the best results were obtained by applying ϕ_{area}, though only slightly better than ϕ_{width} and ϕ_{height}. The results were obtained with 7.2 for T_{medium} and 38 for T_{long}, see table 5.4. With ϕ_{width} best results were obtained with thresholds 2.6 for T_{medium} and 6.0 for T_{long}, and with ϕ_{height} best thresholds were 2.8 for T_{medium} and 5.8 for T_{long}.

When we look at the confusion matrix we observe that, despite the correct classification in the majority of shots, this face detector is very sensitive to noise. From the 159 shots that contain no face, only 20% is correctly classified. Moreover 30% is classified as either medium or close. Recall that this classification results from a shot that contains many frames with rather large skin colored and ellipse shaped regions. The color segmentation obviously generates too much noise.

Labels	*Close*	*Medium*	*Long*	*None*	*Total*
Close	73.33%	21.67%	5%	0.00%	60
Medium	14.40%	61.60%	24.00%	0.00%	500
Long	0.51%	45.13%	53.33%	1.03%	195
None	13.83%	20.13%	45.91%	20.13%	159

Table 5.4: *Confusion matrix resulting from face detection method that combines color and shape, using* ϕ_{area}

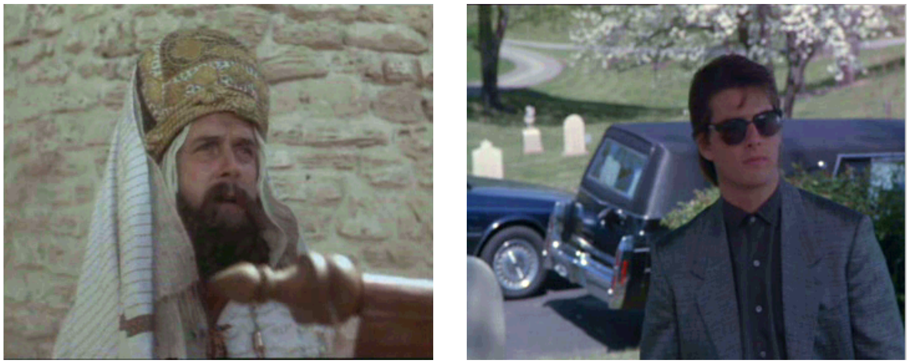


Figure 5.2: *Face based classification errors due to occlusion caused by garbs and beards (left) and sunglasses (right)*

Analysis of falsely classified shots shows that false classification especially occurs in dark parts of the video frames and also in frames that were shot in darkness, which can be explained by the instability of *rg*-space when intensity is low. Furthermore, face detection on characters in *Life of Brian* was difficult because of faces that were occluded by garbs and beards, this resulted in partial detection of faces and thus in misclassification. In *Rainman* this was also the case in shots where an actor wears dark sunglasses, see figure 5.2 for examples. A small percentage of misclassification was due to the fact that some shots fell within the limits of a wrong threshold.

5.2.2 Global spatial features

To evaluate the usability of the global spatial features, we compared the results of the number of edge pixels and region size within a key-frame with a predefined groundtruth that was constructed using the rules and procedures described in 5.1.2. Note that this groundtruth differs from the facetruth as defined in 5.2.1, since shots that contain no face also have a camera distance. For example a close-up of an object like a telephone or a medium shot showing the actor from behind.

Number of edge pixels

To evaluate the usability of our first global spatial feature, the number of edge pixels in a key-frame, we threshold a key-frame on the present number of edge pixels. We use two thresholds, E_{close} to classify close-ups, and E_{long} to classify long shots. We found that there exists a correlation between the number of edge pixels in a frame and the camera distance used. After testing with different thresholds the best trade-off between successful classification and misclassification was achieved with values of 4% for E_{close}

Labels	*Close*	*Medium*	*Long*	*None*	*Total*
Close	63.74%	0.00%	1.10%	35.16%	91
Medium	13.15%	0.00%	18.18%	68.67%	517
Long	9.30%	0.00%	44.19%	46.51%	258

Table 5.5: *Confusion matrix resulting from edge analysis*

and 13.5% for E_{long}, defined as:

$$E_{close} = 0.04 \times F_{pixels} \tag{5.1}$$

$$E_{long} = 0.135 \times F_{pixels} \tag{5.2}$$

where the percentage is taken from the total amount of pixels in a key-frame, F_{pixels}, see table 5.5 for results.

Classification results could be improved simply by enlarging the percentage value for E_{close} or reducing percentage value for E_{long}. Unfortunately this also resulted in worse performance because misclassification results of medium shots increased. Reasons for misclassification in case of falsely classified close-ups were due to:

- *Dark key-frames*, caused by filming in darkness or someone walking by in front of the camera;
- *Blurry key-frames*, caused by unsharp filming;
- *Uniform key-frames*, caused by uniform backgrounds like walls, lawns and sky;

Misclassification of shots as long was due to complex patterns in the key-frame caused by clear patterns in clothing, or background, e.g. bricked walls. In fact, the frames in figure 5.2 were not only falsely classified because of partial face detection but also because of complex patterns, which makes these frames/shots extremely difficult to classify.

Region size

Opposed to the former features, this feature is used in a different way, which makes visualization of results by means of a confusion matrix improper. Instead of using the feature as a classifier we will use this feature as a negator, i.e. ensure that a given shot is not a close-up. As stated in subsection 4.2.2 the rational behind this is that, in general, close-ups are unlikely to contain a very large uniform colored region.

After evaluation of this feature we found that it isn't applicable on many shots, since not all shots contain large regions. We found that classification was possible with only 10 percent of total shots (92 out of 866), though results were satisfying since from those 92 shots only 4 were close-ups (4.35%), which made us decide to use this feature anyway. The false classifications were due to extreme close-ups from uniform surfaces like a door or black clothing. As a value for threshold B_{large} we used that a region should be larger than 53% of the total amount of pixels in the image, F_{pixels}, so B_{large} is defined as:

$$B_{large} = 0.53 \times F_{pixels} \tag{5.3}$$

5.2.3 Global spatiotemporal features

In this subsection we will analyze usability of the global spatiotemporal features, we will first discuss the influence of the detection of camera work with respect to our data set. Following this we will compare results of classification based on shot duration with the predefined groundtruth.

Camera work

The camera work feature is used in a special way, we don't use it to classify a shot within a class. Instead it is used to narrow our application domain. Since camera work often changes the camera distance within a shot, only shots that have a *static* camera work label are processed. It is reasonable to assume that the results from the automatic camera work detection are right, so it will not be evaluated. Out of 866 defined shots from our training set 788 were left after applying the camera work detector.

Shot duration

Recall from section 4.4 that this feature is used as negator and separator. Therefore interpretation of the confusion matrix is somewhat different than those of preceding features. Instead of looking at the values of the diagonal, values of the cross-diagonal are of interest for this matrix. Moreover a separation is made only between close-ups and long shots. In table 5.6 an overview of the classification results from duration analysis are visualized.

We used two thresholds to map this feature. δ_{short} defined as:

$$\delta_{short} = 0.25 \times \delta_{avg} \tag{5.4}$$

Labels	*Close*	*Medium*	*Long*	*None*	*Total*
Close	24.18%	0.00%	5.49%	70.33%	91
Medium	10.83%	0.00%	16.83%	72.34%	517
Long	7.75%	0.00%	24.81%	67.44%	258

Table 5.6: *Confusion matrix resulting from duration analysis*

was used to ensure that a *short take* shot was not classified as long, and δ_{long} defined as:

$$\delta_{long} = 1.45 \times \delta_{avg} \tag{5.5}$$

to ensure that a *long take* shot was not classified as close-up. Parameters 0.25 and 1.45 were experimentally defined and gave lowest misclassification results. Failure of the δ feature was due to relatively long close-ups, mainly caused by long take shots showing a close-up face, for example when someone was making a phone call. Short long shots especially appeared in shots from action scenes.

Average number of colors

The average number of colors in a shot, used as a confirmer of close-up shots of objects or animals proved to be a feature with poor discriminative power. Results showed that it generated many false classification on our tests. Especially dark shots were frequently falsely classified. The results made us decide not to further evaluate this feature, so no confusion matrix is included.

5.3 Classification method

After evaluation of the different visual features, this section will be used to discuss how to integrate the different feature mappings to come to our final classification method. This method will be highlighted in subsection 5.3.1, In subsection 5.3.2 an evaluation of results of the final classification method will be presented. Classification results on the training set are compared with results on a test set, as defined in section 5.1.

5.3.1 Combining feature mappings

The final classification method is based on combined results from the feature mappings made in section 5.2, and is visually represented in figure 5.3. The actual classification algorithm is included as pseudo code in figure 5.4. Recall from section 5.2.3 that only static shots are processed. Based on the feature mappings made we conclude that most discriminating feature with respect to camera distance is ϕ_{area}, using Rowley's face detection method. Therefore we use mapping results from this feature to make a first segmentation between distance classes. Shots that are labeled as none based on this feature, i.e. no face is detected in the shot, are further processed.

When Rowley's algorithm detects no faces in the shot, we test the shot for the number of present edge pixels in a key-frame. Though its discriminating power is weaker when compared to ϕ_{area} based on color and shape, this feature was chosen because it generated less classification errors. Few detected edge pixels in a key-frame, indicating a close-up, was overruled when duration indicated a long take. Furthermore based on ϕ_{area}, using color and shape, and detected region size, classification was enhanced which resulted in better classification results, see figure 5.4 (line 5 till 17).

If a shot contains many edge pixels we label it as long, except for the case that in the same shot a ϕ_{area} using color and shape detected a close-up. In this case we label the shot as medium (lines 19-20 in figure 5.4). With this artifice we exploit the knowledge that ϕ_{area} using color and shape generates a lot of noise, often resulting in medium shots classified as close-ups.

Our last predicate is used when the edge feature cannot segment distance classes based on edge information. Classification is then based on ϕ_{area} using color and shape. The classification is enhanced by incorporating the values of the duration and region features (line 23 till 37 in figure 5.4).

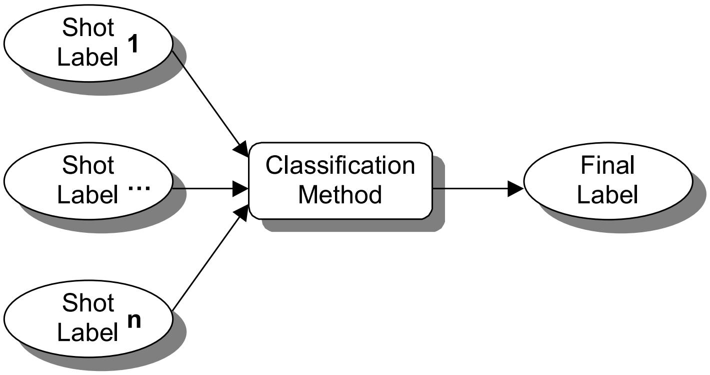


Figure 5.3: *Labeled shots, resulting from feature mappings, are combined to generate a final classification*

```
IF camWork == static
   IF phiRowley <> none
      THEN label = phiRowley;
   ELSE
      IF edges == few
         IF duration == longTake
            IF phiColorShape == medium
               THEN label = medium;
            ELSE
               THEN label = long;
         ELSE
            IF regionSize == small
               THEN label = close;
            ELSE IF phiColorShape == long
               THEN label = long;
            ELSE
               THEN label = close;
      ELSE IF edges == many
         IF phiColorShape == close
            THEN label = medium; // !!!
         ELSE
            THEN label = long;
      ELSE IF edges == normal
         IF regionSize == large
            IF phiColorShape == long OR none
               THEN label = long;
            ELSE
               THEN label = medium;
         ELSE IF duration == shortTake
            IF phiColorShape == medium
               THEN label = medium;
            ELSE
               THEN label = close;
         ELSE IF phiColorShape == long
            THEN label = long;
         ELSE
            THEN label = medium;
```

Figure 5.4: *Pseudo code description of the classification algorithm*

Labels	*Close*	*Medium*	*Long*	*Total*
Close	62.19%	28.05%	9.76%	82
Medium	4.42%	82.53%	13.05%	475
Long	3.46%	12.99%	83.55%	231

Table 5.7: *Confusion matrix resulting from final classification on training set*

5.3.2 Classification results

After finishing the classification method we are now ready to test whether the automatic determination of the camera distance used, based on visual features, resembles that of a manually assigned camera distance. To test the applicability of the proposed method we first used the training set of video data, results of this training set are visible in table 5.7. We compared the results with those from the test set in table 5.8. Note that the total amount of evaluated shots is less than the total as defined in table 5.1, since on that set camera work was not yet detected.

Overall results of the method shows that correct classification of cinematographic video data is possible in the majority of shots. Moreover comparable results of training and test set show that the method is sufficiently generic to be used as camera distance classification method. Only the detection of close-ups is less satisfying, but this can easily be explained by the fact that close-ups of an object or animal, e.g. TV-screen or horse, are not detected with current set of visual features. Of course the method used has some limitations resulting in shots that are not correctly classified. There are several reasons for this misclassification. Most important reason for failure is caused by the limitations of the face detectors used. They suffer from known limitations, especially darkness, occlusion, facial hair, partial detection, and false detection are reasons for failure. Very difficult to classify are shots where actors are filmed from the side, resulting in partial face detection or discarded faces because of an unrealistic facial aspect ratio. Of course visual features heavily suffer from shots filmed in darkness. Not only the face detectors are known to suffer from it, the edge and region feature also generate many false classification results caused by dark key-frames.

Labels	*Close*	*Medium*	*Long*	*Total*
Close	62.07%	19.83%	18.10%	116
Medium	3.93%	86.12%	9.95%	382
Long	6.73%	17.17%	76.10%	297

Table 5.8: *Confusion matrix resulting from final classification on test set*

Labels	*Close*	*Medium*	*Long*	*Total*
Close	0.00%	0.00%	0.00%	0
Medium	0.48%	92.86%	6.66%	210
Long	2.53%	6.33%	91.14%	79

Table 5.9: *Confusion matrix resulting from classification on Friends 418*

When we compare the results of the camera distance based classification on individual video data, it shows that classification results vary. We obtained best classification results on episodes of TV sitcom *Friends*, see table 5.9. Both processed episodes generated a correct classification in about 90 percent of all relevant shots. Results are better because faces are more prominent in this video data, almost every shot contains a (frontal) face. Also close-ups of humans or objects were not present in this video data. Moreover the lack of dark shots in the processed episodes made that visual features were extracted under more or less ideal circumstances.

Contrasted to the good classification results achieved on *Friends*, also somewhat less correct results were achieved on feature films *Life of Brian* and *A View to a Kill*. In table 5.10 classification results from *Life of Brian* are visualized. The extracted frame/face ratio from this feature film suffers from occluded faces caused by garbs and beards that were frequently present in the video data. Moreover a scene situated in a dark stable was responsible for many misclassified shots. In *A View to a Kill* the opening scene was situated on the North Pole. This caused that many long shots were not properly segmented based on the number of edge pixels in a key-frame, since the white surroundings generated very few edge pixels. This feature film also contained many action shots, these shots generally show a lot of motion in very short shots. Face detection was difficult in these shots, because often the actor is not filmed frontal. Moreover the duration feature failed because short action long shots were not correctly classified.

Labels	*Close*	*Medium*	*Long*	*Total*
Close	50.00%	25.00%	25.00%	4
Medium	8.00%	74.00%	18.00%	50
Long	6.35%	25.40%	68.25%	63

Table 5.10: *Confusion matrix resulting from classification on Life of Brian*

Chapter 6

System Overview

The proposed system combines figures 5.1 and 5.3 with database storage, leading to the conceptual architecture as presented in figure 6.1.

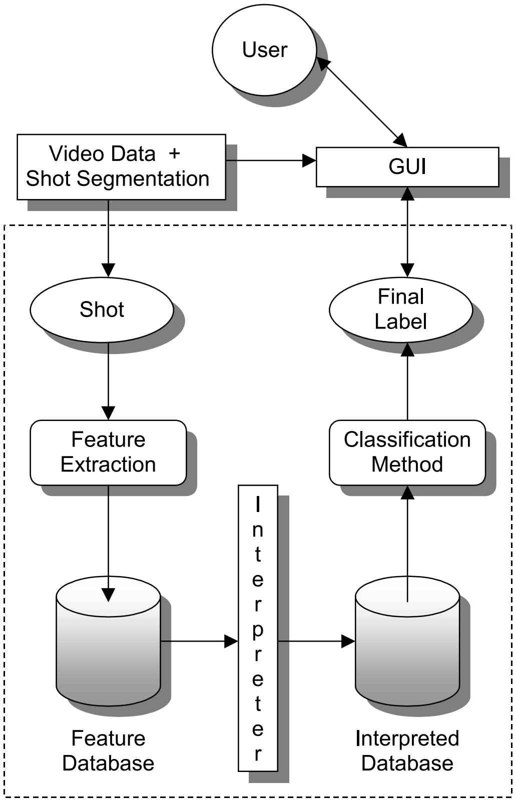
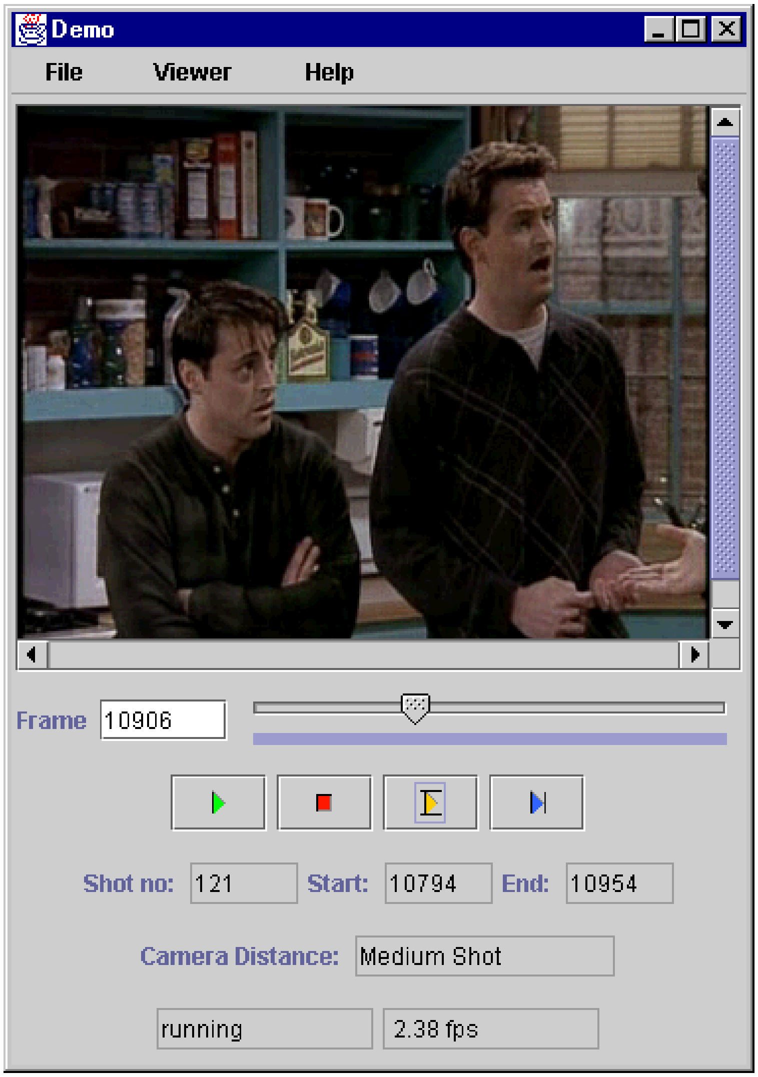


Figure 6.1: *Conceptual architecture of proposed system*

6.1 Implementation

The video data, accompanied with matching shot segmentation, forms the input of the system. The system was implemented in both C and C++ utilizing the Horus Image Library[1] for computation of feature values, storage, interpretation and final classification. Interaction with the user is realized by means of a graphical user interface (GUI) implemented in Java, see figure 6.2. Performance of the system is slow due to feature extraction. Since this has to be done only once for each stream of video, it is actually not an issue.



Frame 10906

Shot no: 121 Start: 10794 End: 10954

Camera Distance: Medium Shot

running 2.38 fps

Figure 6.2: *Graphical User Interface of proposed system*

[1] http://www.science.uva.nl/~horus

Chapter 7

Conclusion

In this section the generalized conclusions are presented, together with some directions for future research with respect to classification of video shots based on the camera distance used.

7.1 Generalized conclusions

In this thesis we showed that, based on visual features, automatic determination of the camera distance used, i.e. close-up, medium shot or long shot, is feasible in a vast majority of cinematographic video shots. It was also found that the fuzzy nature of camera distance is responsible for the reason that a perfect fit between visual features and camera distance is difficult to approximate. Features have no absolute truth with respect to the distance used. The usage of a frame/face ratio gives best classification results, but is dependent on the successful detection of an actors face. Unfortunately current *state-of-the-art* face detection methods still suffer from some limitations, especially darkness and occlusion are known to make face detection error prone. Another limitation of current face detection methods is the fact that non-frontal faces, e.g. faces filmed from the side, are difficult to detect.

The frame/face ratio was the only feature capable to distinguish between three defined distance classes. Besides this feature the number of edge pixels in a key-frame (representing a shot) was very well capable to distinguish between close-ups and long shots. These two features form the main pillars of the proposed classification method that is further enhanced with some other features whose discriminating power is less strong but still usable.

After evaluation of the proposed classification method we also conclude that results vary between different video streams. When the majority of shots are filmed in good lightning conditions, showing (frontal) faces in almost every shot, classification shows much better results. When those ideal circumstances are less present, especially in action shots and in video that frequently shows (partial) occluded faces, performance degrades. Since all visual features substantially suffer from shots filmed in darkness, performance of the final classification method also degrades for these types of shots. Since in dark shots the used camera distance is less relevant the proposed method is very well capable to be used for automatic analysis and interpretation of the meaning of the shot within a video stream, as an assistance tool for video librarians, and as indexing mechanism within a video database system. To leverage the indexing and searching of digital video to a higher level that leads to a true Gutenberg shift however, more research is still necessary.

7.2 Future research

Besides enhancement and extensions of the set of visual features used, especially with respect to face detection, future research should focus on techniques that exploit camera properties and possibilities. A restriction of the current system is that it is incapable of detection of close-ups showing objects or animals, since no features for its detection were included. Future research thus includes research on segmentation of focused objects for example by using the method described by Li in [14].

Another interesting question with respect to future research is whether similarity between shots that are close together can be used as an additional information source. The rational behind this is that when filming dialogs the camera often switches between the one actor and the other, often resulting in shots that use the same camera distance. So when in shot S_i a *medium* face is detected, and we know S_i resembles S_{i+2}, we might presume that distance in S_{i+2} is medium also, although, due to failure in face detection, no face is detected.

Since this thesis fully focused on visual features in video data to classify shots based on the camera distance used, no attention was given to the audio component of the video data. The usage of sound might not seem logical with respect to camera distance, but we believe that it might prove to be a valuable additional feature. The number of extracted voices from a shot for example, might be an indication for distance. A close-up is unlikely to contain more than one voice, and when three or more voices are identified within a shot, possibility of a long shot increases.

Bibliography

[1] R.M. Bolle, B.-L. Yeo, and M.M. Yeung. Video Query: Research directions. *IBM Journal of Research and Development*, Vol. 42, No. 2, pp. 233-252, March 1998.

[2] D. Bordwell and K. Thompson. *Film Art: An Introduction.* 5th ed. McGraw-Hill New York, 1997.

[3] R. Brunelli, O. Mich and C.M. Modena. A Survey on the Automatic Indexing of Video Data. *Journal of Visual Communication and Image Representation*, Vol. 10, No. 2, pp. 78-112, Jun 1999.

[4] Y. Chan, S.-H. Lin, Y.-P. Tan, and S.Y. Kung. Video Shot Classification Using Human Faces. In *Proceedings IEEE International Conference on Image Processing*, Lausanne, Switzerland (ICIP'96) Vol. 3 pp. 843-846, 1996.

[5] G. Davenport, T.A. Smith, and N. Pincever. Cinematic Primitives for Multimedia. *IEEE Computer Graphics & Applications*, pp. 67-74, July 1991

[6] M. Davis, Knowledge Representation for Video. In *Proceedings of the 12th National Conference on Artificial Intelligence*, Vol. 1, pp. 120-127, AAAI, Seattle, Washington, July 31 - August 4 1994.

[7] S. Fischer, R. Lienhart and W. Effelsberg. Automatic Recogntion of Film Genres. *Proceedings ACM Multimedia 95*, San Francisco, CA, pp. 295-304, Nov. 1995.

[8] T. Gevers. *Color Image Invariant Segmentation and Retrieval.* Phd Thesis, Universiteit van Amsterdam, May 1996.

[9] Ch. Gu. *Multivalued Morphology and Segmentation-Based Coding.* Phd Thesis, Ecole polytechnique federale de Lausanne, 1995.

[10] A. Hampapur, R. Jain, and T. Weymouth. Feature Based Digital Video Indexing. In *Proceedings of IFIP 2.6 Third Working Conference on Visual Database Systems VDB.3*, Lausanne, Switzerland, March 29-31 1995.

[11] I. Ide, K. Yamamoto, and H. Tanaka. Automatic Video Indexing Based on Shot Classification. *First International Conference on Advanced Multimedia Content*

Processing AMCP'98, Osaka, Japan, S. Nisho, F. Kishino eds., Lecture Notes in Computer Science Vol. 1554, Springer-Verlag, March 1999.

[12] R. Jain and A. Hampapur. Metadata in Video Databases. *ACM SIGMOD*, Vol. 23, No. 4, pp. 27-33, December 1994.

[13] V. Kashyap, K. Shah, and A. Sheth. Metadata for building the MultiMedia Patch Quilt. In S. Jajodia and V.S. Subrahmanium, editors, *Multimedia Database Systems: Issuses and Research*, 1995.

[14] J. Li, J.Z. Wang, R.M. Gray, and G. Wiederhold. Multiresolution Object-of-Interest Detection for Images with Low Depth of Field. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, Venice, Italy, 1999.

[15] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Scene Determination based on Video and Audio Features. In *Proceedings of the 6th IEEE International Conference on Multimedia Systems*. Vol. 1, pp. 685-690, 1999

[16] B. Lipkin. *LaTeX for Linux*, Springer-Verlag New York 1999.

[17] T.H. Nguyen. *DetectCUT Manual*, Internal publication, University of Amsterdam, December 1998.

[18] R. Ronfard, C. Garcia, and J. Carrive. Conceptual Indexing of Television Images based on Face and Caption Sizes and Locations. *Visual 2000*, November 1 - 3, Lyon, France, November 2000, accepted for publication.

[19] H.A. Rowley. *Neural Network-Based Face Detection.* Phd Thesis, Carnegie Mellon University, May 1999.

[20] Y. Rui, T.S. Huang, and S. Mehrotra. Exploring Video Structure Beyond the Shots. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, Austin, Texas USA, pp. 237-240, June 28-July 1, 1998.

[21] A.W.M. Smeulders and R. van den Boomgaard. *An Introduction to Image Processing and Computer Vision.* Faculty of Science, University of Amsterdam, August 1996.

[22] K. Sobottka and I. Pitas. Looking for Faces and Facial Features in Color Images. *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, Russian Academy of Sciences, Vol. 7, No. 1, 1997.

[23] J.-C. Terrillon and S. Akamatsu. Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images. *Accepted at Vision Interface'99 (VI'99)*, Trois-Rivieres, P.Q., Canada, 18-21 May 1999.

[24] Y. Tonomura, A. Akutsu, Y. Taniguchi and G. Suzuki. Structured Video Computing. *IEEE Multimedia*, Vol. 1, No. 3, pp. 34-43, 1994.

[25] J. Vendrig and M. Worring. Feature Driven Visualization of Video Content for Interactive Indexing. *Visual 2000*, November 1 - 3, Lyon, France, November 2000, accepted for publication.

[26] J. Yang and A. Waibel. A Real-time Face Tracker. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pp. 10-15, 1998.

[27] M.-H. Yang, D. Kriegman and N. Ahuja, Detecting Faces in Images: A Survey. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[28] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic Partitioning of Full-motion Video. *Multimedia Systems*, Vol. 1, No. 1, pp. 10-28, 1993.

Appendix A

Ellipse Analysis

The center $(\overline{x}, \overline{y})$ of an ellipse is given by the center of gravity of the connected component:

$$\overline{x} = \frac{1}{N} \sum_{(x,y) \, \in \, C} x \tag{A.1}$$

$$\overline{y} = \frac{1}{N} \sum_{(x,y) \, \in \, C} y \tag{A.2}$$

Where N denotes the number of pixels of the connected component C.

According to Sobottka [22] the orientation θ of an ellipse can be computed by using the central moments $\mu_{i,j}$ of the connected component:

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right) \tag{A.3}$$

By evaluating the moments of inertia, the length of major and minor axis of the best-fit ellipse E can be determined. With I_{min} the least and I_{max} the greatest moment of inertia of an ellipse with orientation θ:

$$I_{min} = \sum_{(x,y) \, \in \, C} (\ (x - \overline{x}) \cos\theta - (y - \overline{y}) \sin\theta\ )^2 \tag{A.4}$$

$$I_{max} = \sum_{(x,y) \, \in \, C} (\ (x - \overline{x}) \sin\theta - (y - \overline{y}) \cos\theta\ )^2 \tag{A.5}$$

the length a of the major axis and the length b of the minor axis can now be determined by:

$$a = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left(\frac{(I_{max})^3}{I_{min}}\right)^{\frac{1}{8}} \tag{A.6}$$

$$b = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left(\frac{(I_{min})^3}{I_{max}}\right)^{\frac{1}{8}} \tag{A.7}$$

Acknowledgements

The satisfaction you get from finishing a thesis makes you instantly forget al the trouble you had to go through before finishing such a task. Of course there were some problems in the period I worked on this thesis and finishing it would not have been possible without the help of many and some in particular. This section is devoted to those who were of great help for me during the last nine months.

In the first place I would like to thank the people at ISIS for their hospitality and willingness to help with all sorts of minor and major problems. Special thanks go to Marcel Worring for proofreading this thesis and making some useful comments. Above all I would like to thank supervisor Jeroen, whose insights, comments and positivism certainly steered me in the right direction and made this thesis to what it is now.

Besides the more physical help, the mental help was also of great value for me. Therefore I would like to thank my friends, colleague students, and of course my parents, little brothers and girlfriend.

Cees Snoek