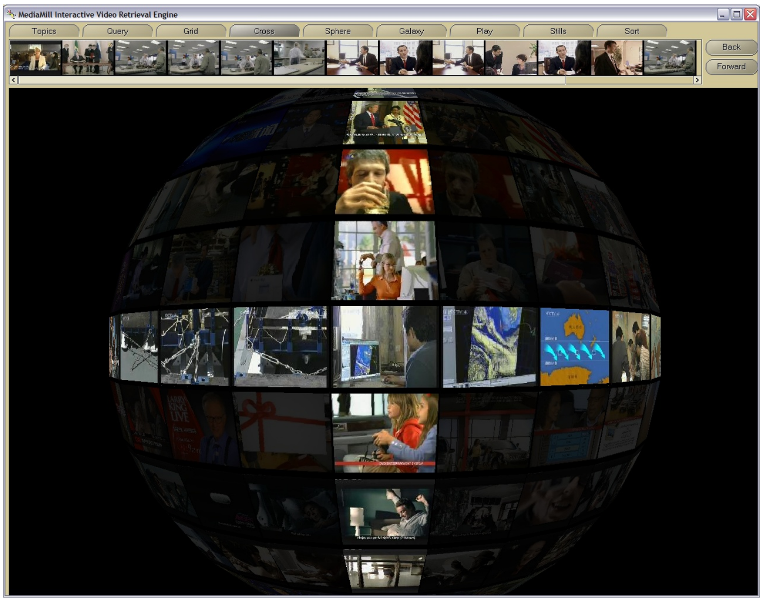
Semantic Indexing and Retrieval of Video

Tutorial SAMT 2006



Marcel Worring, Cees Snoek
Intelligent Systems Lab Amsterdam, University of Amsterdam
{worring,cgmsnoek}@science.uva.nl
http://www.mediamill.nl
tel: +31 20 5257521

MediaMill

The work described in this tutorial has been carried out at the Intelligent Systems Lab of the University of Amsterdam. The work was supported by the BSIK MultimediaN project.

Contents

Chapter 1

Introduction

The semantic gap between the low level information that can be derived from the visual data and the conceptual view the user has of the same data is a major bottleneck in video retrieval systems. It has dictated that solutions to image and video indexing could only be applied in narrow domains using specific concept detectors, e.g., *sunset* or *face*. This leads to lexica of at most 10-20 concepts. The use of multimodal indexing, advances in machine learning, and the availability of some large, annotated information sources, e.g., the TRECVID benchmark, has paved the way to increase lexicon size by orders of magnitude (now 100 concepts, in a few years 1,000). This brings it within reach of research in ontology engineering, i.e. creating and maintaining large, typically 10,000+ structured sets of shared concepts. When this goal is reached we could search for videos in our home collection or on the web based on their semantic content, we could develop semantic video editing tools, or develop tools that monitor various video sources and trigger alerts based on semantic events. This tutorial lays the foundation for these exciting new horizons.

Semantic video indexing requires a multi-disciplinary approach. To analyze the sensory and textual data, techniques from signal processing, speech recognition, computer vision, and natural language processing are needed. To understand the information, knowledge engineering and machine learning play an important role. For storing the tremendous amounts of data, database systems are required with high performance, the same holds for the network technology involved. Apart from technology, the role of the user in multimedia is even more important than in traditional systems, hence visualization and human computer interfacing form another fundamental basis in multimedia.

In this tutorial we do not focus on one of the underlying fields, but aim to provide the necessary insight in these fields to be able to use them in building solutions.

We begin our notes with the characteristics of video collections one can encounter in various domains in chapter 2 and end with the user interacting with the system to retrieve those stored video fragments. In between we discuss the methodologies required to make this possible. The first step is representing the raw image and video data in a way that it can be used in further processing in chapter 3. Tools for this are often based on machine learning so a short description of the underlying methods is given in chapter 4. Then we proceed to basic video analysis in chapter 5 and finally the core of the system: generic techniques for deriving concepts from a large lexicon in chapter 6. These concepts from the basis for the interactive retrieval system in chapter 7. The different chapters and their relations are depicted in figure 1.1.

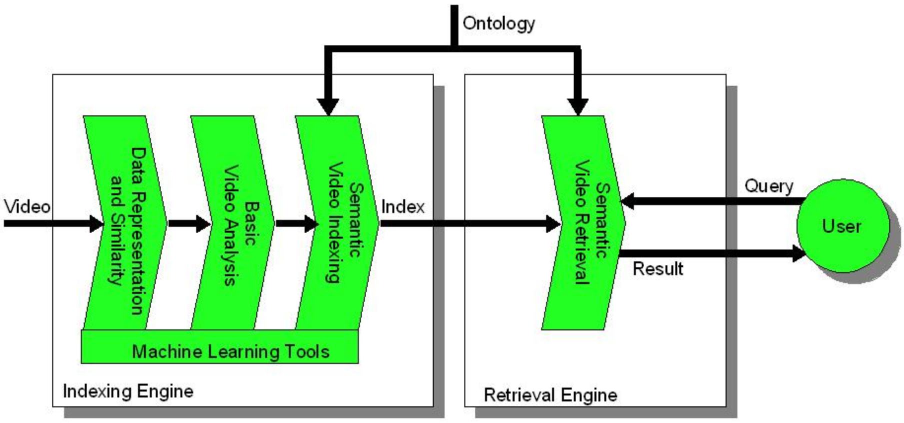


Figure 1.1: Overall architecture of a video indexing and retrieval system, giving a blueprint for the lecture notes.

Chapter 2

Data, Domains, and Applications

As indicated in the introduction multimedia information is appearing in all different kinds of application in various domains. Here we focus on video documents as their richest form they contain visual, auditory, and textual information. In this chapter we will consider how to analyze these domains and how to prepare the data for insertion into the database. To that end we first describe, in section 2.1 different domains and the way video data is produced and used. From there we categorize the data from the various applications in order to be able to select the right class of tools later (section 2.2). Then we proceed to the way the data is actually acquired in section 2.3. The role of external knowledge is considered in section 2.4. We then consider in detail how a video document is created, as this forms the basis for later indexing (section 2.5). Finally, we consider how this leads to a general framework which can be applied in different domains.

2.1 Data and applications

In the broadcasting industry the use of video documents is of course obvious. Large amounts of data are generated in the studios creating news, films, soaps, sport programs, and documentaries. In addition, their sponsors create significant amount of material in the form of commercials. Storing the material produced in a multimedia information systems allows to reuse the material later. Currently, one of the most important applications for the broadcasting industry is to do multi-channelling i.e. distributing essentially the same information via the television, internet and mobile devices. In addition interactive television is slowly, but steadily, growing e.g. allowing to vote on your different idol. More and more of the video documents in broadcasting are created digitally, however the related information still is not distributed alongside the programs and hence not always available for storing it in the information system.

Whereas the above video material is professionally produced, we see lot of unprofessional videos being shot by people with their private video camera, their webcam, or more recently with their PDA or telephone equipped with a camera. The range of videos one can encounter here is very large, as cameras can be used for virtually everything. However, in practice, many of the videos will mostly contain people or holiday scenes. The major challenge in consumer applications is organizing the data in such a way that you can later find all the material you shot e.g. by location, time, persons, or event.

In education the use of video material is somewhat related to the creation of documentaries in broadcasting, but has added interactive possibilities. Furthermore, you see and more and more lectures and scientific presentations being recorded with a camera and made accessible via the web. They can form the basis for new teaching material.

For businesses the use of electronic courses is an effective way of reducing the timeload for instruction. Next to this, videos are used in many cases for recording business meetings. Rather than scribing the meeting, action lists are sufficient as the videos can replace the extensive written notes. Furthermore, it allows people who were not attending the meeting to understand the atmosphere in which certain decisions were made. Another business application field is the observation of customers. This is of great importance for marketing applications.

Let's now move to the public sector where among other reasons, but for a large part due to september 11, there has been an increased interest in surveillance applications guarding public areas and detecting specific people, riots and the like. What is characteristic for these kind of applications is that the interest is only in a very limited part of the video, but clearly one does not know beforehand which part contains the events of interest. Not only surveillance, but also cameras on cash machines and the like provide the police with large amounts of material. In forensic analysis, video is therefore also becoming of great importance. Another application in the forensic field is the detection and identification of various videos found on PC's or the Internet containing material like child pornography or racism. Finally, video would be a good way of recording a crime scene for later analysis.

Somewhat related to surveillance is the use of video observation in health care. Think for example of a video camera in an old peoples home automatically identifying if someone falls down or has some other problem.

2.2 Categorization of data

Although all of the above applications use video, the nature of the videos is quite different for the different applications. We now give some categorizations to make it easier to understand the characteristics of the videos in the domain. A major distinction is between produced and observed video.

Definition 1 (Produced video data) *videos that are created by an author who is actively selecting content and where the author has control over the appearance of the video.*

Typical situations where the above is found is in the broadcasting industry. Most of the programs are made according to a given format. The people and objects in the video are known and planned.

For analysis purposes it is important to further subdivide this category into three levels depending in which stage of the process we receive the data:

- *raw data*: the material as it is shot.
- *edited data*: the material that is shown in the final program
- *recorded data*: the data as we receive it in our system

Edited data is the richest form of video as it has both content and a layout. When appropriate actions are taken directly when the video is produced, many indices can directly be stored with the data. In practice, however, this production info is often not stored. In that case recorded data becomes difficult to analyze as things like layout information have to be reconstructed from the data.

The other major category is formed by:

Definition 2 (Observed video data) *videos where a camera is recording some scene and where the author does not have means to manipulate or plan the content.*

This kind of video found is found in most of the applications, and the most typical examples are surveillance videos and meetings. However, also in broadcasting this is found. E.g. in soccer videos the program as a whole is planned, but the content itself cannot be manipulated by the author of the document.

Two other factors concerning the data are important:

- *quality of the data*: what's the resolution and signal-to-noise ratio of the data.

The quality of the video can vary significantly for the different applications, ranging from high-resolution, high quality videos in the broadcasting industry, to very low quality data from small cameras incorporated in mobile phones.

A final important factor is the

- *Application control*: how much control does one have on the circumstances under which the data is recorded.

Again the broadcasting industry goes to extreme cases here. E.g. in films the content is completely described in a script and lighting conditions are almost completely controlled, if needed enhanced using filters. In security applications the camera can be put at a fixed position which we now. For mobile phones the recording conditions are almost arbitrary.

Finally, a video is always shot for some reason following [77] we define:

- *Purpose* the reason for which the video document is made being entertainment, information, communication, or data analysis.

2.3 Data acquisition

A lot of video information is already recorded in digital form and hence can easily be transported to the computer. If the video is shot in analog way a capture device has to be used in the computer to digitize the sensory data. In both cases the result is a stream of frames where each frame is typically an RGB-image. A similar thing holds for audio. Next to the audiovisual data there can be a lot of accompanying textual information like the teletext channel containing text in broadcasting, the script of a movie, scientific documents related to a presentation given, or the documents which are subject of discussion at a business meeting.

Now let us make the result of acquisition of multimedia data more precise. For each modality the digital is a temporal sequence of *fundamental units*, which in itself do not have a temporal dimension. The nature of these units is the main factor discriminating the different modalities. The visual modality of a video document is a set of ordered images, or frames. So the fundamental units are the single image frames. Similarly, the auditory modality is a set of samples taken within a certain time span, resulting in audio samples as fundamental units. Individual characters form the fundamental units for the textual modality.

As multimedia datastreams are very large data compression is usually applied, except when the data can be processed directly and there is no need for storing the video. For video the most common compression standards are MPEG-1, MPEG-2 and MPEG-4. For audio mp3 is the best known compression standard.

Finally, apart from the multimedia data, there is lot of factual information related to the video sources, like e.g. the date of a meeting, the value at the stock market of the company discussed in a video, or viewing statistics for broadcast.

2.4 Ontology creation

As indicated above, many information sources are used for a video. Furthermore, videos rarely occur in isolation. Hence, there is always a context in which the video has to be considered. To that end for indexing and analysis purposes it is important to make use of external knowledge sources. Examples are plenty. In news broadcast news sites on the internet provide info on current issues, the CIA factbook contains information on current countries and presidents. Indexing film is greatly helped by considering the Internet Movie Database and so on. Furthermore, within a company or institute local knowledge sources might be present that can help in interpreting the data. For security applications a number of images of suspect people might be available.

All of the above information and knowledge sources have their own format and style of use. It is therefore important to structure the different knowledge sources into ontologies and make the information elements instantiations of concepts in the ontology.

Examples of ontologies are SnoMed, MeSH and the Gene Ontology for health care, AAT and Iconclass for art, and the generic ontologies WordNet and Cyc. Ontologies have various uses in the indexing and retrieval process. If existing, well-established ontologies are used: they provide a shared vocabulary, meaning that the terms themselves are agreed upon as well as their meaning, since meaning are partially captured in the (hierarchical) structure of the ontology. Ambiguous terms can be disambiguated, and relations between concepts in the ontology can be used to support the annotation and search process [69,71]. Ontologies are currently being used for manual annotation [74,143], and where manual annotations are not feasible or available, they have been used to aid retrieval based on captions or other text associated with the visual data [150].

2.5 Produced video documents

[*] As a baseline for video we consider how video documents are created in a production environment. In chapter 5 we will then consider the indexing of recorded videos in such an environment as in this manner all different aspects of a video will be covered.

An author uses visual, auditory, and textual channels to express his or her ideas. Hence, the content of a video is intrinsically multimodal. Let us make this more precise. In [114] multimodality is viewed from the system domain and is defined as “the capacity of a system to communicate with a user along different types of communication channels and to extract and convey meaning automatically”. We extend this definition from the system domain to the video domain, by using an authors perspective as:

Definition 3 (Multimodality) *The capacity of an author of the video document to express a predefined semantic idea, by combining a layout with a specific content, using at least two information channels.*

We consider the following three information channels or modalities, within a video document:

- *Visual modality*: contains the *mise-en-scène*, i.e. everything, either naturally or artificially created, that can be seen in the video document;
- *Auditory modality*: contains the speech, music and environmental sounds that can be heard in the video document;

[*] This section is adapted from [156].

- *Textual modality*: contains textual resources that describe the content of the video document;

For each of those modalities, definition 3 naturally leads to a semantic perspective i.e. which ideas did the author have, a content perspective indicating which content is used by the author and how it is recorded, and a layout perspective indicating how the author has organized the content to optimally convey the message. We will now discuss each of the three perspectives involved.

2.5.1 Semantic index

The first perspective expresses the intended semantic meaning of the author. Defined segments can have a different granularity, where granularity is defined as the descriptive coarseness of a meaningful unit of multimodal information [34]. To model this granularity, we define segments on five different levels within a semantic index hierarchy. The first three levels are related to the video document as a whole. The top level is based on the observation that an author creates a video with a certain purpose. We define:

- *Purpose*: set of video documents sharing similar intention;

The next two levels define segments based on consistent appearance of layout or content elements. We define:

- *Genre*: set of video documents sharing similar style;
- *Sub-genre*: a subset of a genre where the video documents share similar content;

The next level of our semantic index hierarchy is related to parts of the content, and is defined as:

- *Logical units*: a continuous part of a video document's content consisting of a set of named events or other logical units which together have a meaning;

Where named event is defined as:

- *Named events*: short segments which can be assigned a meaning that doesn't change in time;

Note that named events must have a non-zero temporal duration. A single image extracted from the video can have meaning, but this meaning will never be perceived by the viewer when this meaning is not consistent over a set of images.

At the first level of the semantic index hierarchy we use purpose. As we only consider video documents that are made within a production environment the purpose of data analysis is excluded. Genre examples range from feature films, news broadcasts, to commercials. This forms the second level. On the third level are the different sub-genres, which can be e.g. horror movie or ice hockey match. Examples of logical units, at the fourth level, are a dialogue in a drama movie, a first quarter in a basketball game, or a weather report in a news broadcast. Finally, at the lowest level, consisting of named events, examples can range from explosions in action movies, goals in soccer games, to a visualization of stock quotes in a financial news broadcast.

2.5.2 Content

The content perspective relates segments to elements that an author uses to create a video document. The following elements can be distinguished [19]:

- *Setting*: time and place in which the video's story takes place, can also emphasize atmosphere or mood;
- *Objects*: noticeable static or dynamic entities in the video document;
- *People*: human beings appearing in the video document;

Typically, setting is related to logical units as one logical unit is taken in the same location. Objects and people are the main elements in named events.

2.5.3 Recording the scene

The appearance of the different content elements can be influenced by an author of the video document by using modality specific style elements. For the visual modality an author can apply different style elements. She can use specific colors and lighting which combined with the natural lighting conditions defines the appearance of the scene. Camera angles and camera distance can be used to define the scale and observed pose of objects and people in the scene. Finally, camera movement in conjunction with the movement of objects and people in the scene defines the dynamic appearance. Auditory style elements are the loudness, rhythmic, and musical properties. The textual appearance is determined by the style of writing and the phraseology i.e. the choice of words, and the manner in which something is expressed in words. All these style elements contribute to expressing an author's intention.

2.5.4 Layout

The layout perspective considers the syntactic structure an author uses for the video document.

Upon the fundamental units an aggregation is imposed, which is an artifact from creation. We refer to this aggregated fundamental units as *sensor shots*, defined as a continuous sequence of fundamental units resulting from an uninterrupted sensor recording. For the visual and auditory modality this leads to:

- *Camera shots*: result of an uninterrupted recording of a camera;
- *Microphone shots*: result of an uninterrupted recording of a microphone;

For text, sensor recordings do not exist. In writing, uninterrupted textual expressions can be exposed on different granularity levels, e.g. word level or sentence level, therefore we define:

- *Text shots*: an uninterrupted textual expression;

Note that sensor shots are not necessarily aligned. Speech for example can continue while the camera switches to show the reaction of one of the actors. There are however situations where camera and microphone shots are recorded simultaneously, for example in live news broadcasts.

An author of the video document is also responsible for concatenating the different sensor shots into a coherent structured document by using *transition edits.* "He or she aims to guide our thoughts and emotional responses from one shot to another, so that the interrelationships of separate shots are clear, and the transitions between sensor shots are smooth" [19]. For the visual modality abrupt cuts, or gradual transitions[†], like wipes, fades or dissolves can be selected. This is important for visual continuity, but sound is also a valuable transitional device in

[†] A gradual transition actually contains pieces of two camera shots, for simplicity we regard it as a separate entity.

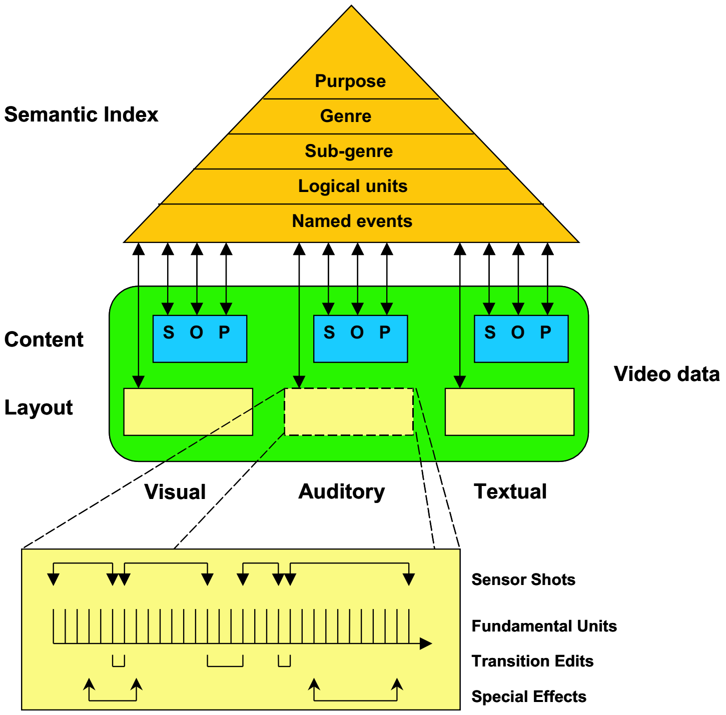


Figure 2.1: A unifying framework for multimodal video indexing based on an author's perspective. The letters S, O, P stand for setting, objects and people. An example layout of the auditory modality is highlighted, the same holds for the others.

video documents. Not only to relate shots, but also to make changes more fluid or natural. For the auditory transitions an author can have a smooth transition using music, or an abrupt change by using silence [19]. To indicate a transition in the textual modality, e.g. closed captions, an author typically uses ">>>", or different colors. They can be viewed as corresponding to abrupt cuts as their use is only to separate shots, not to connect them smoothly.

The final component of the layout are the optional visual or auditory *special effects*, used to enhance the impact of the modality, or to add meaning. Overlayed text, which is text that is added to video frames at production time, is also considered a special effect. It provides the viewer of the document with descriptive information about the content. Moreover, the size and spatial position of the text in the video frame indicate its importance to the viewer. "Whereas visual effects add descriptive information or stretch the viewer's imagination, audio effects add level of meaning and provide sensual and emotional stimuli that increase the range, depth, and intensity of our experience far beyond what can be achieved through visual means alone" [19]. Note that we don't consider artificially created content elements as special effects, as these are meant to mimic true settings, objects, or people.

2.6 Discussion

Based on the discussion in the previous sections we come to a unifying multimodal video indexing framework based on the perspective of an author. This framework is visualized in figure 2.1. It forms the basis for the discussion of methods for video indexing in chapter 5.

Although the framework is meant for produced video it is in fact a blueprint for other videos also. Basically, all of the above categories of videos can be mapped to the framework. However, not in all cases, all modalities will be used, and maybe there is no layout as no editing has taken place and so on.

Keyterms in this chapter

Ontologies, sensory data, digital multimedia data, data compression, purpose, genre, sub-genre, logical unit, named events, setting, objects, people, sensor shots, camera shots, microphone shots, text shots, transition edits, special effects

Chapter 3

Data Representation and Similarity

When a person is capturing data with a camera, microphone, or any other sensor a lot of data is collected. Example of applications that take this to the extreme for personal information can be found in [48] [72].

In this chapter we consider what the different data types are that play a role in any multimedia environment. We will put a considerable emphasis on ways of representing the multimedia content as this forms the basis for all later analysis.

3.1 Basic data types

Before multimedia entered the world all information systems just contained factual data. For factual data the common categorization is as follows [183]:

- *nominal data*: these values are taken from a selected set of symbols, where no relation is supposed between the different symbols.
- *ordinal data*: these values are taken from a selected set of symbols, where an ordering relation is supposed between the different symbols.
- *interval data*: quantities measured in fixed and equal units.
- *ratio data*: quantities for which the measurement scheme inherently defines a zero point.

An example of nominal data is the set of names of museums in Amsterdam {Rijksmuseum, Stedelijk Museum, Van Gogh Museum, ...}. For ordinal data a typical example is a scale composed of {low, medium, high}. Temperature measured in degrees Celcius is an example of interval data. Although a zero point is defined 20 degrees Celcius is not viewed as twice as hot as 10 degrees. Finally, time in seconds is a clear example of ratio data.

Geographic information systems have brought us the additional datatypes position in 2D $p = (x, y)$, and position in 3D, $p = (x, y, z)$ when the position of a person of object is measured over time these become $p(t) = (x(t), y(t))$ and $p(t) = (x(t), y(t), z(t))$ respectively. Now when considering the use of a camera we have in addition the notion of orientation of the camera as function of time $\theta(t) = (\theta_1(t), \theta_2(t), \theta_3(t))$. Other important time dependent functions are acceleration $A(t)$ and speed $S(t)$. A less frequently occurring time-dependent function is brain activity $F(t)$. This can be of great importance when measuring the emotional state of a person.

Another very important data type is the *graph*. A graph $G = \{V, E\}$ is composed of a set of vertices V and a set of edges E. The edges connect pairs of elements in E. A path in the graph is a set of subsequent edges where the vertex at the end of an edge is connected to the starting vertex on the next edge (except for the last element of the path). Several types of graphs exist. A *directed graph* is a graph where the edges have a direction associated with it. A *directed acyclic graph* is a graph where the edges have a direction, any path in the graph visits a vertex at most once. A *tree* is a special case of a directed acyclic graph.

3.2 Categorization of multimedia descriptions

To be able to retrieve multimedia objects with the same ease as we are used to when accessing text or factual data as well as being able to filter out irrelevant items in the large streams of multimedia reaching us requires appropriate descriptions of the multimedia data. Although it might seem that multimedia retrieval is the trivial extension of text retrieval it is in fact far more difficult. Most of the data is of sensory origin (image, sound, video) and hence techniques from digital signal processing and computer vision are required to extract relevant descriptions. Such techniques in general yield features which do not relate directly to the user's perception of the data, the so called semantic gap. More precisely defined as [151]:

> *The semantic gap is the lack of coincidence between the information that one can extract from the sensory[*] data and the interpretation that the same data have for a user in a given situation.*

Consequently, there are two levels of descriptions of multimedia content one on either side of the semantic gap. In addition to the content description there are also descriptions which are mostly concerned with the carrier of the information. Examples are the pixel size of an image, or the sampling rate of an audio fragment. It can also describe information like the owner of the data or the time the data was generated. It leads to the following three descriptive levels [70]:

- *perceptual descriptions*: descriptions that can be derived from the data
- *conceptual descriptions*: descriptions that cannot be derived directly from the data as an interpretation is necessary.
- *non-visual/auditory descriptions*: descriptions that cannot be derived from the data at all.

The non-visual/auditory descriptions are typically related to the carrier of the multimedia data, like the resolution of an image or the data of creation of an audio file.

Standards for the exchange of multimedia information like MPEG-7 [104] [105] [99] give explicit formats for descriptions attached to the multimedia object. However, these standards do not indicate how to find the values/descriptions to be used for a specific multimedia data object. Especially not how to do this automatically.

Clearly the semantic gap is exactly what separates the perceptual and conceptual level. In this chapter we will consider the representation of the different modalities at the perceptual level. In later chapters we will consider indexing techniques which are deriving descriptions at the conceptual level. If such indexing techniques can be developed for a specific conceptual term, this term basically moves to the perceptual category as the system can generate such a description automatically without human intervention.

[*] In the reference this is restricted to visual data, but the definition is applicable to other sensory data like audio as well.

3.3 Basic notions

Multimedia data can be described at different levels. The first distinction to be made is between the complete data item like an entire video, a song, or a text document and subparts of the data which can attain different forms. A second distinction is between the content of the multimedia data and the layout. The layout is closely related to the way the content is presented to the viewer, listener, or reader.

Now let us take a closer look at what kind of subparts one can consider. This is directly related to the methods available for deriving those parts, and it has a close resemblance to the different classes of descriptions considered.

Four different objects are considered:

- *perceptual object*: a subpart in the data which can be derived by weak segmentation, i.e. segmentation of the datafield based on perceptual characteristics of the data.
- *conceptual object*: a part of the data which can be related to conceptual descriptions and which cannot be derived from the data without an interpretation, the process of finding these objects is called strong segmentation.
- *partition*: the result of a partitioning of the datafield, which is not dependent on the data itself.
- *layout object*: the basic data elements which are structured and related by the layout.

Examples of the above categories for the different modalities will be considered in the subsequent sections. In the rest of the notes we will use :

- *multimedia item* : a full multimedia data item, a layout element, or an element in the partitioning of a full multimedia item.
- *multimedia object* is used it can be either a conceptual or perceptual object.

Hence, a multimedia item can be for example a complete video, an image, or an audio CD, but also a shot in a video, or the upper left quadrant of an image. A multimedia object can e.g. be a paragraph in a text, or a house in an image.

3.4 Audio representation

The info in this section is mostly based on [96].

3.4.1 Audio features

An audio signal is a digital signal in which amplitude is given as function of time. When analyzing audio one can consider the time-domain signal directly, but it can be advantageous to consider the signal also on the basis of the frequencies in the signal. The Fourier transform is a well known technique to compute the contribution of the different frequencies in the signal. The result is called the audio spectrum of the signal. An example of a signal and its spectrum are shown in figure 3.1.

We now consider audio features which can be derived directly from the signal.

- Average Energy: the average squared amplitude of the signal, an indication of the loudness.

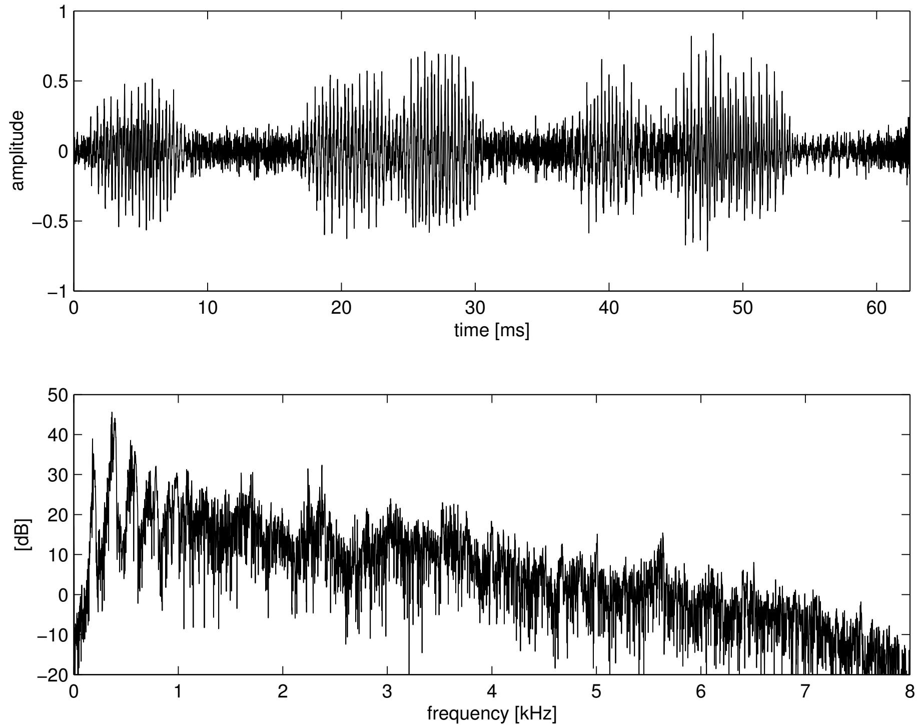


Figure 3.1: *An example of a signal in the time-domain (top) and its spectrum (bottom).*

- Zero Crossing Rate: a measure indicating how often the amplitude switches from positive to negative and vice versa.
- Rhythm: measures based on the pattern produced by emphasis and duration of notes in music or by stressed and unstressed syllables in spoken words.
- Linear Prediction Coefficients (LPC): measure of how well a sample in the signal can be predicted based on previous samples.

In the frequency domain there are is also an abundance of features to compute. Often encountered features are:

- Bandwidth: the frequency range of the sound, in its simplest form the difference between the largest and smallest non-zero frequency in the spectrum.
- Fundamental frequency: the dominant frequency in the spectrum.
- Brightness (or spectral centroid): the is normalized sum of all frequencies times how much this frequency is present in the signal.
- Mel-Frequency Cepstral Coefficients (MFCC): a set of coefficients designed such that they correspond to how humans hear sound.
- Pitch: degree of highness or lowness of a musical note or voice.
- Harmonicity: degree to which the signal is built out of multiples of the fundamental frequency.
- Timbre: characteristic quality of sound produced by a particular voice or instrument (subjective).

3.4.2 Audio segmentation

An audio signal can be long and over time the characteristics of the audio signal and hence its features will change. Therefore, in practice one partitions the signal into small segments of say a duration of 10 milliseconds. For each of those segments any of the features mentioned can then be calculated.

An audio signal can contain music, speech, silence, and other sounds (like cars etc.). The aim of weak segmentation of audio aims at decomposing the signal into these four components. An important step is decomposing the signal based on different ranges of frequency. A second important factor is harmonicity as this distinguishes music from other audio.

Strong segmentation of audio aims at detecting different conceptual objects like cars, or individual instruments in a musical piece. This will require to build models for each of the different concepts.

3.4.3 Temporal relations

When I have two different audio segments A and B , there is a selected set of temporal relations that can hold between A and B namely *precedes, meets, overlaps, starts, equals, finishes, during* and there inverses denoted by add _i at the end of the name (if B precedes A the relation precedes_i holds between B and A). These are known as the Allen's relations [8]. As equals is symmetric there are 13 such relations in total. The relations are such that for any two intervals A and B one and exactly one of the relations hold.

3.5 Image representation

3.5.1 Image features

A digital image is an array of pixels, where each pixel has a color. The basic representation for the color of the pixel is the triple R(ed), G(reen), B(lue). There are however many other color spaces which are more appropriate in certain tasks. We consider HSV and L*ab* here.

A first thing to realize is that the color of an object is actually a color spectrum, indicating how much a certain wavelength is present (white light contains an equal amount of all wavelengths). This is the basis for defining HSV. To be precise the three components of HSV are as follows: H(ue) is the dominant wavelength in the color spectrum. It is what you typically mean when you say the object is red, yellow, blue, green, purple etc. S(aturation) is a measure for the amount of white in the spectrum. It defines the purity of a color distinguishing for example signal-red from pink. Finally, the V(olume) is a measure for the brightness or intensity of the color. This make the difference between a dark and a light color if they have the same H and S values.

L*ab* is another color space that is used often. The L is similar to the V in HSV. The a and b are similar to H and V. The important difference is that in the L*ab* space the distance between colors in the color space is approximately equal to the perceived difference in the colors. This is important in defining similarity see section 3.8.

In the above, the color is assigned to individual pixels. All these colors will generate patterns in the image, which can be small, or large. These patterns are denoted with the general term *texture*. Texture is of great importance in classifying different materials like the line-like pattern in a brick wall, or the dot-like pattern of sand.

In an image there will be in general different colors and/or textures. This means there will be many positions in the image where there is a significant change in image data, in particular a change in color or texture. These changes form (partial) lines called *edges*.

The above measure give a local description of the data in the image. In many cases global measures, summarizing the information in the image are used. Most commonly these descriptions are in the form of color histograms counting how many pixels have a certain color. It can however, also, be a histogram on the directions of the different edge pixels in the image.

An image histogram looses all information on spatial configurations of the pixels. If I have a peak in the histogram at the color red, the pixels can be scattered all around the image, or it can be one big red region. Color coherence vectors are an alternative representation which considers how many pixels in the neighborhood have the same color. A similar representation can be used for edge direction.

The above histograms and coherence vectors can be considered as summaries of the data, they are non-reversible. I cannot find back the original image if I have the histogram. The following two descriptions do have that property.

The Discrete Cosine Transform (DCT) is a transform which takes an image and computes it frequency domain description. This is exactly the same as considered for audio earlier, but now in two dimensions. Coefficients of the low frequency components given a measure of the amount of large scale structure where high-frequency information gives information on local detailed information. The (Haar) wavelet transform is a similar representation, which also takes into account where in the image the specific structure is found.

3.5.2 Image segmentation

For images we can also consider the different ways of segmenting an image. A partition decomposes the image into fixed regions. Commonly this is either a fixed set of rectangles, or one fixed rectangle in the middle of the image, and a further partition of the remaining space in a fixed number of equal parts.

Weak segmentation boils down to grouping pixels in the image based on a homogeneity criterion on color or texture, or by connecting edges. It leads to a decomposition of the image where each region in the decomposition has a uniform color or texture.

For strong segmentation, finding specific conceptual objects in the image, we again have to rely on models for each specific object, or a large set of hand-annotated examples.

3.5.3 Spatial relations

If I have two rectangular regions we can consider the Allen's relations separately for the x- and y- coordinate. However, in general regions have arbitrary shape. There are various *2D spatial relations* that I can consider. Relations like left-of, above, surrounded-by, and nearest neighbor are an indication of the relative positions of regions in the image. Constraints like inside, enclosed-by are indications of *topological relations* between regions.

3.6 Video representation

3.6.1 Video features

As a video is a set of temporally ordered images its representation clearly shares many of the representations considered above for images. However, the addition of a time component also adds many new aspects.

In particular we can consider the observed movement of each pixel from one frame to another called the *optic flow*, or we can consider the motion of individual objects segmented from the video data.

3.6.2 Video segmentation

A partition of an image can be any combination of a temporal and spatial partition. For weak segmentation we have, in addition to color and texture based grouping of pixels, *motion based grouping* which groups pixels if the have the same optic flow i.e. move in the same direction with the same speed. Strong segmentation requires again object models. The result of either weak or strong segmentation is called a *spatio-temporal object*.

For video there is one special case of weak segmentation which is temporal segmentation. Thus, the points in time are detected where there is a significant change in the content of the frame. This will be considered in a more elaborate form later.

3.6.3 Spatio-temporal relations

Spatio-temporal relations are clearly a combination of spatial and temporal relations. One should note, however, that in a video spatial relations between two objects can vary over time. Two objects A and B can be in the relation A left-of B at some point in time, while the movement of A and B can yield the relation B left-of A at a later point in time.

3.7 Text representation

3.7.1 Text features

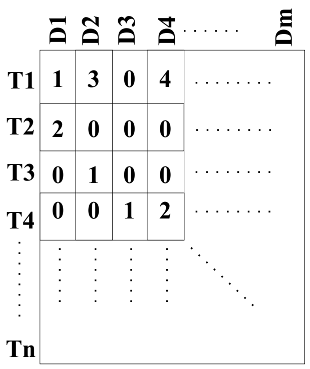
The basic representation for text is the so called bag-of-words approach. In this approach a kind of histogram is made indicating how often a certain word is present in the text. This histogram construction is preceded by a stop word elimination step in which words like the, in, etc. are removed. One also performs stemming on the words bringing each word back to its base. E.g. “biking” will be reduced to the verb “to bike”.

The bag-of-words model commonly used is the Vector Space Model [136]. The model is based on linear algebra. A document is modeled as a vector of words where each word is a dimension in Euclidean space. Let $T = \{t^1, t^2, \dots, t^n\}$ denote the set of terms in the collection. Then we can represent the terms d_j^T in document d_j as a vector $\vec{x} = (x_1, x_2, \dots, x_n)$ with:

$$x_i = \begin{cases} t_j^i & \text{if } t^i \in d_j^T \quad ; \\ 0 & \text{if } t^i \notin d_j^T \quad . \end{cases} \tag{3.1}$$

Where t_j^i represents the frequency of term t^i in document d_j. Combining all document vectors creates a term-document matrix. An example of such a matrix is shown in figure 3.2.

Depending on the context, a word has an amount of information. In an archive about information retrieval, the word ’`retrieval`’ does not add much information about this document, as the word ’`retrieval`’ is very likely to appear in many documents in the collection. The underlying rationale is that words that occur frequently in the complete collection have low information content. However, if a single document contains many occurrences of a word, the document is probably



	D1	D2	D3	D4	......	Dm
T1	1	3	0	4	....	
T2	2	0	0	0	....	
T3	0	1	0	0	....	
T4	0	0	1	2	....	
⋮	⋮	⋮	⋮	⋮	⋱	
Tn						

Figure 3.2: *A Term Document matrix.*

relevant. Therefore, a term can be given a weight, depending on its information content. A weighting scheme has two components: a global weight and a local weight. The global importance of a term is indicating its overall importance in the entire collection, weighting all occurrences of the term with the same value. Local term weighting measures the importance of the term in a document. Thus, the value for a term i in a document j is $L(i,j) * G(i)$, where $L(i,j)$ is the local weighting for term i in document j and $G(i)$ is the global weight. Several different term weighting schemes have been developed. We consider the following simple form here known as Inverse Term Frequency Weigthing. The weight w_j^i of a term t^i in a document j is given by

$$w_j^i = t_j^i * \quad \log(\frac{N}{t_*^i}) \tag{3.2}$$

where N is the number of documents in the collection and t_*^i denotes the total number of times word t^i occurs in the collection. The logarithm dampens the effect of very high term frequencies.

Going one step further one can also consider the co-occurrence of certain words in particular which words follow each other in the text. If one applies this to a large collection of documents to be used in analyzing other documents it is called a bi-gram language model. It gives the probability that a certain word is followed by another word. It is therefore also an instantiation of a Markov model. When three or more general n subsequent words are used we have a 3-gram or *n-gram* language model.

3.7.2 Text segmentation

Different parts of a document may deal with different topics and therefore it can be advantageous to partition a document into partitions of fixed size for which the word histogram is computed.

A useful technique, which can be considered an equivalent of weak segmentation, is *part-of-speech tagging* [98]. In this process each word is assigned the proper class e.g. verb, noun, etc. A simple way of doing this is by making use of a dictionary and a bi-gram language model. One can also take the tagged result and find larger chunks as aggregates of the individual words. This technique is known as *chunking* [2].

A more sophisticated, but less general approach, is to generate a grammar for the text and use a parser to do the part of speech tagging. It is, however, very difficult to create grammars which can parse an arbitrary text.

3.7.3 Textual relations

As text can be considered as a sequence of words, the order in which words are placed in the text yields directly a relation "precedes". This is similar to the Allen's relations for time, but as words cannot overlap, there is no need for the other relations.

In the case where also the layout of the text is taking into account i.e. how it is printed on paper, many other relations will appear as we can now consider the spatial relations between blocks in the documents.

3.8 Similarity

When considering collections of multimedia items, it is not sufficient to describe individual multimedia items. It is equally important to be able to compare different multimedia items. To that end we have to define a so called *(dis)similarity function* $\mathcal{S}$ to compare two multimedia items I_1 and I_2. The function $\mathcal{S}$ measures to what extent I_1 and I_2 look or sound similar, to what extent they share a common style, or to what extent they have the same interpretation. Thus, in general we distinguish three different levels of comparison:

- Perceptual similarity
- Layout similarity
- Semantic similarity

Each of these levels will now be described.

3.8.1 Perceptual similarity

Computing the dissimilarity of two multimedia items based on their data is mostly done by comparing their feature vectors. For an image this can for example be a HSV color histogram, but it can also be a HSV histogram followed by a set of values describing the texture in the image.

One dissimilarity function between the Euclidean distance. As not all elements in the vector might be equally important the distances between the individual elements can be weighted. To make the above more precise let $F^1 = \{f_i^1\}_{i=1,\dots,n}$ and $F^2 = \{f_i^2\}_{i=1,\dots,n}$ be the two vectors describing multimedia item I_1 and I_2 respectively. Then the dissimilarity S_E according to weighted Euclidean distance is given by:

$$S_E(F_1, F_2) = \sqrt{\sum_{i=1}^{n} w_i (f_i^2 - f_i^1)^2}$$

with $\vec{w} = \{w_i\}_{i=1,n}$ a weighting vector.

For color histograms (or any other histogram for that matter) the histogram intersection is also used often to denote dissimilarity. It does not measure Euclidean distance but takes the minimum of the two entries. Using the same notation as above we find:

$$S_{\cap}(F_1, F_2) = \sum_{i=1}^{n} \min{(f_i^2, f_i^1)}$$

In [164] it is shown that it is advantageous for computing the similarity that one computes the cumulative histogram $\hat{F}$ rather than the regular histogram. This is due to the fact that the use of cumulative histograms reduces the effect of having to limit the number of different bins in a histogram. The histograms entries are then computed as:

$$\hat{F}(m) = \sum_{k=0}^{m} F_k,$$

where F is the regular histogram. After computing the cumulative histogram, histogram intersection can be used to compute the similarity between the two histograms.

For comparing two texts it is common to compute the similarity rather than dissimilarity. Given the vector-space representation the similarity between two text vectors $\vec{q}$ and $\vec{d}$ both containing n elements is defined as:

$$\mathcal{S} = \vec{q} \cdot \vec{d} \quad , \tag{3.3}$$

where $\cdot$ denotes the inner product computed as

$$\vec{q} \cdot \vec{d} = \sum_{i=1}^{n} \vec{q_i} * \vec{d_i}$$

If you would consider the vector as a histogram indicating how many times a certain word occurs in the document (possibly weighted by the importance of the term) it boils down to histogram multiplication. Hence, when a terms occurs often in both texts the contribution of that term to the value of the inner product will be high.

A problem with the above formulation is the fact that larger documents will contain more terms and hence are on average more similar than short pieces of text. Therefore, in practice the vectors are normalized by their length

$$||\vec{q}|| = \sqrt{\sum_{i=1}^{n} \vec{q}_i^2}$$

Leading to the so called Cosine Similarity.

$$\mathcal{S} = \frac{\vec{q} \cdot \vec{d}}{||\vec{q}||||\vec{d}||}$$

This measure is called the cosine similarity as it can be shown that it equals the cosine of the angle between the two length normalized vectors.

3.8.2 Layout similarity

To measure the (dis)similarity of two multimedia items based on their layout a common approach is to transform the layout of the item to a string containing the essential layout structure. For simplicity we consider the layout of a video as this is easily transformed to a 1D-string, it can however be extended to 2 dimensions e.g. when comparing the layout of two printed documents. When the layouts of two multimedia items are described using strings they can be compared by making use of the edit-distance in [5] defined as follows:

Definition 4 (Edit Distance) *Given two strings $A : a_1, a_2, a_3, ..., a_n$ and $B = b_1, b_2, b_3, ..., b_m$ over some alphabet Σ, a set of allowed edit operations E and a unit cost for each operation, the edit distance of A and B is the minimum cost to transform A into B by making use of edit operations.*

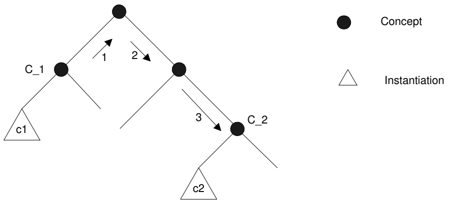


Figure 3.3: *Example of the semantic distance of two instantiations c_1, c_2 of concepts C_1 and C_2 organized in a tree structure. In this case the distance would be equal to 3.*

As an example let's take two video sequences and consider the transitions cut (c), wipe (w), and dissolve (d) and let a shot be denoted by symbol s. Hence, $\Sigma = \{c, w, d, s\}$. Two example sequences could now look like

$$A = scswsdscs$$

$$B = sdswscscscs$$

Now let $E = \{\text{insertion,deletion,substitution}\}$ and for simplicity let each operation have equal cost. To transform A into B we twice have to do a substitution to change the effects c and d used and two insert operations to add the final cs. Hence, in this case the edit distance would be equal to 4.

3.8.3 Semantic similarity

When multimedia items are described using concepts which are derived from an ontology, either annotated by the user or derived using pattern recognition, we can compare two multimedia items based on their semantics.

When the ontology is organized as a hierarchical tree an appropriate measure is the *semantic distance* i.e. the dissimilarity of two instantiations c_1, c_2 of the concepts C_1 and C_2 is to take the number of steps in the tree one has to follow to move from concept C_1 to C_2. If the ontology is organized as a set of hierarchical views this leads to a vector where each element in the vector is related to one of the views. A very simple example is shown in figure 3.3.

Keyterms in this chapter

Nominal data, ordinal data, interval data, ratio data, 2D-position, 3D-position, time-varying position, orientation, acceleration, speed, brain activity, graph, directed graph, directed acyclic graph, tree, perceptual metadata, conceptual metadata, non-visual/auditory metadata, multimedia item, multimedia object, partitioning, perceptual object, weak segmentation, conceptual object, strong segmentation, audio spectrum, Allen's relations, color space, texture, edges, spatial relations, topological relations, optic flow, spatio-temporal object, n-gram, part-of-speech tagging, chunking, (dis)similarity function, histogram intersection, Euclidean distance, cosine distance,edit distance,semantic distance

Chapter 4

Machine Learning Tools

4.1 Introduction

In the previous chapter we have made a first step in limiting the size of the semantic gap, by representing multimedia data in terms of various features and similarities. In particular, we have considered features which are of the perceptual class. In this chapter we will consider general techniques to use these features to find the conceptual label of a multimedia object by considering its perceptual features.

The techniques required to do the above task are commonly known as pattern recognition, where pattern is the generic term used for any set of data elements or features. The descriptions in this chapter are taken for the largest part from the excellent review of pattern recognition in [76].

4.2 Pattern recognition methods

Many methods for pattern recognition exist. Most of the methods fall into one of the four following categories:

- *Template matching*: the pattern to be recognized is compared with a learned template, allowing changes in scale and pose;

  This simple and intuitive method can work directly on the data. For images a template is a usually a small image (let's say 20x20 pixels), for audio is it a set of samples. Given a set of templates in the same class, one template representing the class is computed, e.g. by pixelwise averaging. In its simplest form any new pattern is compared pixelwise (for images) or samplewise (for audio) to the set of stored templates. The new pattern is then assigned to the class for which the correlation between the templates is highest.

  In practice template matching becomes more difficult as one cannot assume that two templates to be compared are near exact copies of one another. An image might have a different scale, the object in the image might have a different pose, or the audio template might have a different loudness. Hence, substantial preprocessing is required before template matching can take place. Invariant features can help in this problem (see section 4.5).

- *Statistical classification*: the pattern to be recognized is classified based on the distribution of patterns in the space spanned by pattern features;

- *Syntactic or structural matching*: the pattern to be recognized is compared to a small set of learned primitives and grammatical rules for combining primitives;

  For applications where the patterns have a apparent structure these methods are appealing. Thew allow the introduction of knowledge on how the patterns in the different classes are built from the basic primitives.

  As an example consider graphical symbols. Every symbol is built from lines, curves, and corners, which are combined into more complex shapes likes squares and polygons. Symbols can be distinguished by looking how the primitives are combined into creating the symbol.

  A major disadvantage of such methods is that it requires that all primitives in the pattern are detected correctly, which is not the case if the data is corrupted by noise.

- *Neural networks*: the pattern to be recognized is input to a network which has learned nonlinear input-output relationships;

Neural networks mimic the way humans recognize patterns. They can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. In the human brain those simple processors are called neurons, when simulated on a computer one calls them perceptrons. In both cases, the processors have a set of weighted inputs and “fire” if all inputs are above a certain threshold.

To train a neural network, input patterns are fed to the system and the expected output is defined. Then the weights for the different connections are automatically learned by the system. In most systems there are also a lot of perceptrons which are neither connected to the input or output, but are part of the so-called hidden layer. Such a neural network is called a multi-layer perceptron.

For specific cases neural networks and statistical classifiers coincide. Neural networks are appealing as they can be applied in many different situations. It is, however, difficult to understand why a neural network assigns a certain class to an input pattern.

Examples of the above methods are found throughout the lecture notes. The statistical methods are the most commonly encountered ones, they will be considered in more detail next.

4.3 Statistical methods

We consider methods here which assume that the patterns to be recognized are described as a feature vector, thus every pattern can be associated with a specific point in the feature space. In addition a distance function should be provided, which in our case would be a similarity function as defined in section 3.8. As an example consider the following very simple classification problem: a set of images of characters (i,o,a) for which only two features are calculated namely the width and the height of the bounding box of the character, and similarity is defined as the Euclidean distance.

Before we start describing different methods, let us first consider the general scheme for building classifiers. The whole process is divided into two stages:

- *Training*: in this stage the system builds a model of the data and yields a classifier based on a set of training patterns for which class information is provided by a ”supervisor”.

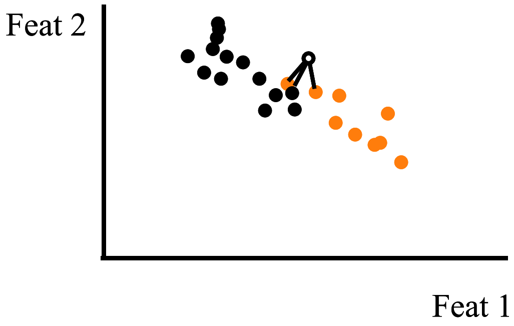


Figure 4.1: *Illustration of classification based on k-nearest neighbors.*

In the above a supervisor is defined. In literature those methods are therefore called supervised classification methods. The process is as follows. Every pattern is preprocessed to remove noise etc. Then a set of features are calculated, and a classifier is learned through a statistical analysis of the dataset. To improve the results, the system can adjust the preprocessing, select the best features, or try other features.

- *Testing*: patterns from a test set are given to the system and the classifier outputs the optimal label.

  In this process the system should employ exactly the same preprocessing steps and extract the same features as in the learning phase.

To evaluate the performance of the system the output of the classifier for the test patterns is compared to the label the supervisor has given to the test patterns. This leads to a confusion matrix, where one can see how often patterns are confused. In the above example it would for example indicate how many "i's" were classified as being part of the class "o".

After testing the performance of the system is known and the classifier is used for classifying unknown patterns into their class.

To make things more precise. Let the c categories be given by $\omega_1, \omega_2, ..., \omega_c$. Furthermore, let the vector consisting of n features be given as $\vec{x} = (x_1, x_2, ..., x_n)$.

A classifier is a system that takes a feature vector $\vec{x}$ and assigns to it the optimal class ω_i. The confusion matrix is the matrix C where the elements c_{ij} denote the number of elements which have true class ω_i, but are classified as being in class ω_j.

Conceptually the most simple classification scheme is

- *k-Nearest Neighbor*: assigns a pattern to the majority class among the k patterns with smallest distance in feature space;

For this method, after selection of the relevant features a distance measure d has to be defined. In principle, this is the similarity function described in chapter 3. Very often it is simply Euclidean distance in feature space. Having defined d, classification boils down to finding the nearest neighbor(s) in feature space. In 1-nearest neighbor classification the pattern $\vec{x}$ is assigned to the same class the nearest neighbor has. In k-nearest neighbors one uses majority voting on the class labels of the k points nearest in space. An example is shown in figure 4.1.

The major disadvantage of using k-nearest neighbors is that it is computationally expensive to find the k-nearest neighbors if the number of data objects is large.

In statistical pattern recognition a probabilistic model for each class is derived. Hence, one only has to compare a new pattern with one model for each class.

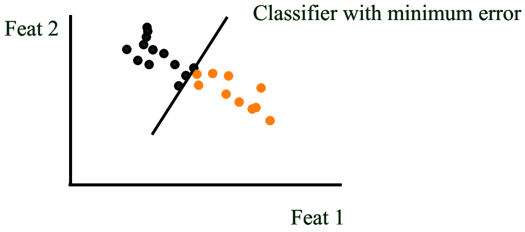


Figure 4.2: *Example of a 2D linear separator.*

The crux of statistical pattern recognition is then to model the distribution of feature values for the different classes i.e. giving a way to compute the conditional probability $P(\vec{x}|\omega_i)$.

- *Bayes Classifier*: assigns a pattern to the class which has the maximum estimated posterior probability;

Thus, assign input pattern $\vec{x}$ to class ω_i if

$$P(\omega_i|\vec{x}) > P(\omega_j|\vec{x}) \text{ for all } j \neq i \tag{4.1}$$

When feature values are be expected to be normally distributed (i.e. a Gaussian distribution) the above can be used to find optimal linear boundaries in the feature space. An example is shown in figure 4.2.

Note, that in general we do not know the parameters of the distribution (the mean μ and the standard deviation σ in the case of the normal distribution). These have to be estimated from the data. Most commonly, the mean and standard deviation of the training samples in each class are used.

If only a limited set of samples per class are available a Parzen estimator can be used. In this method a normal distribution is placed at every sample with fixed standard deviation. The probability of a sample to belong to the class is computed as a weighted linear combination of all these distributions. The weight is directly proportional to the distance of the new sample to the individual samples in the class.

In the Bayes classifier all features are considered at once, which is more accurate, but also more complicated than using features one by one. The latter leads to the

- *Decision Tree*: assigns a pattern to a class based on a hierarchical division of feature space;

To learn a decision tree a feature and a threshold are selected which give a decomposition of the training set into two parts such that the one part contains all elements for which the value is smaller than the threshold, and the other part the remaining patterns. Then for each of the parts the process is repeated till the patterns into a part are assumed to be all of the same class. All the decisions made can be stored in a tree, hence the name. The relation between a decision tree in feature space is illustrated in figure 4.3.

At classification time a pattern is taken and for the feature in the root node of the decision tree, the value of the corresponding feature is compared to the threshold. If the value is smaller the left branch of the tree is followed, the right branch is followed otherwise. This is repeated till a leaf of the tree is reached, which corresponds to a single class.

Finally, a very popular classification method is based on

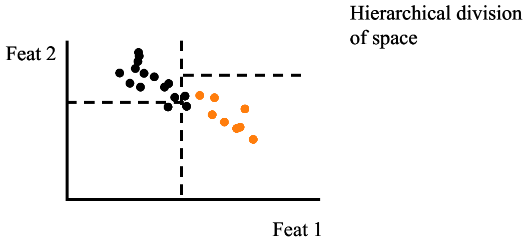


Figure 4.3: *Example of a hierarchical division of space based on a decision tree.*

- *Support Vector Machines*: the SVM tries to find an optimal hyperplane between two classes by maximizing the margin between these two classes ;

In the SVM framework each pattern x is represented in an n-dimensional space, spanned by extracted features. Within this feature space an optimal hyperplane is searched that separates it into two different categories, where the categories are represented by $+1$ and -1 respectively. The hyperplane has the following form: $\omega|(\mathbf{w} \cdot x + b)| \geq 1$, where $\mathbf{w}$ is a weight vector, and b is a threshold. A hyperplane is considered optimal when the distance to the closest training examples is maximum for both categories. This distance is called the margin, see the example in figure 4.4. The problem of finding the optimal hyperplane is a quadratic programming problem of the following form [176]:

$$\min_{\mathbf{w},\xi} \left\{ \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\Big(\sum_{i=1}^{l} \xi_i\Big) \right\} \tag{4.2}$$

Under the following constraints:

$$\omega|(\mathbf{w} \cdot x_i + b)| \geq 1 - \xi_i, \quad i = 1, 2, \ldots, l \tag{4.3}$$

Where C is a parameter that allows to balance training error and model complexity, l is the number of shots in the training set, and ξ_i are slack variables that are introduced when the data is not perfectly separable. These slack variables are useful when analyzing multimedia, since results of individual feature detectors typically include a number of false positives and negatives.

All of the above models assume that the features of an object remain fixed. For data which has a temporal dimension this is often not the case, when time progresses the features might change. In such cases it is needed to consider models which explicitly take the dynamic aspects into account.

The Hidden Markov Model is a suitable tool for describing such time-dependent patterns which can in turn be used for classification. It bears a close relation to the Markov model considered in chapter 3.

- *Hidden Markov Model (HMM)*: assigns a pattern to a class based on a sequential model of state and transition probabilities [98, 130];

Let us first describe the components which make up a Hidden Markov Model.

1. A set $S = s_1, ..., s_m$ of possible states in which the system can be.
2. A set of symbols V which can be output when the system is in a certain state
3. The state transition probabilities A indicating the conditional probability that at time $t+1$ it moves to state s_i if before it was in state s_j: $p(q_{t+1} = s_i | q_t = s_j)$

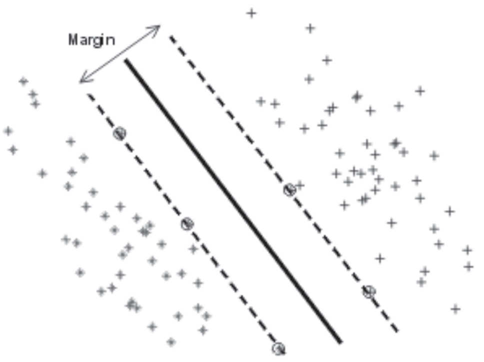


Figure 4.4: Visual representation of the Support Vector Machine framework. Here a two-dimensional feature space consisting of two categories is visualized. The solid bold line is chosen as optimal hyperplane because of the largest possible margin. The circled data points closest to the optimal hyperplane are called the support vectors

4. The probabilities that a certain symbol in V is output when the system is in a certain state.

5. Initial probabilities that the system starts in a certain state.

An example of a HMM is shown in figure 4.5.

There are two basic tasks related to the use of HMM for which efficient methods exist in literature:

1. Given a sequential pattern how likely is that it is generated by some given Hidden Markov Model?

2. Given a sequential pattern and a Hidden Markov Model what is the most likely sequence of states the system went through?

To use HMMs for classification we first have to find models for each class. This is often done in a supervised way. From there all the probabilities required are estimated by using the training set. Now to classify a pattern task 1 mentioned above is used to find the most likely model and thus the most likely class. Task 2 can be used to take a sequence and classify each part of the sequence with it most likely state.

Many variations of HMMs exists. For example, in the above description discrete variables were used. One can also use continuous variables leading to continuous observation density Markov models. Another extension are product HMM where first an HMM is trained for every individual feature, the results of which are then combined in a new HMM for integration.

4.4 Dimensionality reduction

In many practical pattern recognition problems the number of features is very high and hence the feature space is of a very high dimension. A 20-dimensional feature space is hard to imagine and visualize, but is not very large in pattern recognition. Luckily enough there is often redundant information in the space and one can reduce the number of dimensions to work with. One of the best known methods for this purpose is *Principal Component Analysis* (PCA).

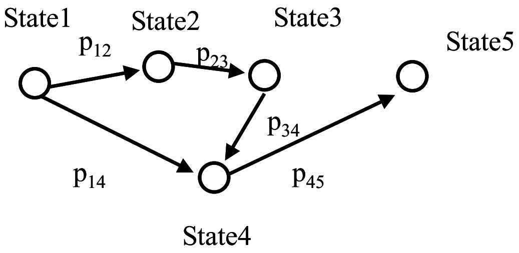


Figure 4.5: *Example of a Hidden Markov Model.*

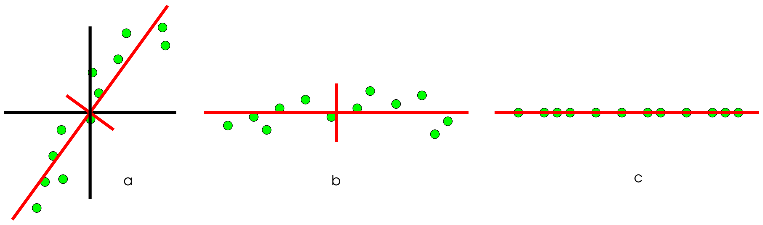


Figure 4.6: *An example of dimension reduction, from 2 dimensions to 1 dimension. (a) Two-dimensional datapoints. (b) Rotating the data so the vectors responsible for the most variation are perpendicular. (c) Reducing 2 dimensions to 1 dimension so that the dimension responsible for the most variation is preserved.*

Explaining the full details of the PCA is beyond the scope of the lecture notes, so we explain its simplest form, reducing a 2D feature space to a 1D feature space. See figure figure 4.6.

The first step is to find the line which best fits the data. Then every datapoint is projected onto this line. Now, to build a classifier we only consider how the different categories can be distinguished along this 1D line. Much simpler than the equivalent 2D problem. For higher dimensions the principle is the same, but instead of a line one can now also use the best fitting plane which corresponds to using 2 of the principle components. In fact, the number of components in component analysis is equal to the dimension of the original space. By leaving out one or more of the least important components, one gets the principal components.

4.5 Invariance

Selecting the proper classifier is important in obtaining good results, but finding good features is even more important. This is a hard task as it depends on both the data and the classification task. A key notion in finding the proper features is invariance defined as [151]:

A feature f of t is invariant under W if and only if f_t remains the same regardless the unwanted condition expressed by W,

$$t_1 \overset{W}{\sim} t_2 \Longrightarrow f_{t_1} = f_{t_2} \tag{4.4}$$

To illustrate, consider again the simple example of bounding boxes of characters. If I want to say something about the relation between the width w and height h of the bounding box I could consider computing the difference $w - h$, but if I would scale the picture by a factor 2 I would get a result which is twice as large. If, however, I would take the aspect ratio w/h it would be invariant under scaling.

In general, we observe that invariant features are related to the intrinsic properties of the element in the image or audio. An object in the image will not change if I use a different lamp. A feature invariant for such changes in color would be good for classifying the object. On the other hand, the variant properties do have a great influence on how I perceive the image or sound. A loud piece of music might be far more annoying than hearing the same piece of music at normal level.

Keyterms in this chapter

Classification, training, testing, confusion matrix, principal component analysis, k-nearest neighbor, template matching, Bayes classifier, decision tree, support vector machine, hidden Markov model, invariance

Chapter 5

Basic Video Analysis*

5.1 Introduction

For browsing, searching, and manipulating video documents, an index describing the video content is required. It forms the crux for applications like digital libraries storing multimedia data, or filtering systems [115] which automatically identify relevant video documents based on a user profile. To cater for these diverse applications, the indexes should be rich and as complete as possible.

Until now, construction of an index is mostly carried out by documentalists who manually assign a limited number of keywords to the video content. The specialist nature of the work makes manual indexing of video documents an expensive and time consuming task. Therefore, automatic classification of video content is necessary. This mechanism is referred to as video indexing and is defined as the process of automatically assigning content-based labels to video documents [58].

When assigning an index to a video document, three issues arise. The first is related to granularity and addresses the question: *what* to index, e.g. the entire document or single frames. The second issue is related to the modalities and their analysis and addresses the question: *how* to index, e.g. a statistical pattern classifier applied to the auditory content only. The third issue is related to the type of index one uses for labelling and addresses the question: *which* index, e.g. the names of the players in a soccer match, their time dependent position, or both.

Which element to index clearly depends on the task at hand, but is for a large part also dictated by the capabilities of the automatic indexing methods, as well as on the amount of information that is already stored with the data at production time. As discussed in chapter 2 one of the most complex tasks is the interpretation of a recording of a produced video as we have to reconstruct the layout and analyze the content. If, however, we are analyzing the edited video with all layout information as well as scripts are available in for example MPEG-7 format the layout reconstruction and a lot of indexing is not needed and one can continue to focus on the remaining indexing tasks.

Most solutions to video indexing address the *how* question with a unimodal approach, using the visual [32, 55, 120, 165, 170, 191, 195], auditory [39, 52, 100, 117, 118, 124, 184], or textual modality [23, 65, 196]. Good books [46, 61] and review papers [20, 24] on these techniques have appeared in literature. Instead of using one modality, multimodal video indexing strives to automatically classify (pieces of) a video document based on multimodal analysis. Only recently, approaches

*This chapter is adapted from [156].

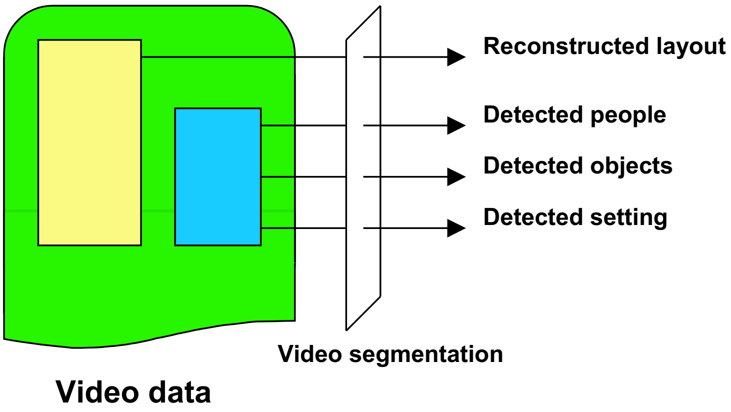


Figure 5.1: *Data flow in unimodal video document segmentation.*

using combined multimodal analysis were reported [7, 12, 38, 73, 109, 125, 139] or commercially exploited, e.g. [33,128,179].

One review of multimodal video indexing is presented in [181]. The authors focus on approaches and algorithms available for processing of auditory and visual information to answer the *how* and *what* question. We extend this by adding the textual modality, and by relating the *which* question to multimodal analysis. Moreover, we put forward a unifying and multimodal framework. Our work should therefore be seen as an extension to the work of [20, 24, 181]. Combined they form a complete overview of the field of multimodal video indexing.

The multimodal video indexing framework is defined in section section 2.5. This framework forms the basis for structuring the discussion on video document segmentation in section 5.2. In section 5.3 the role of conversion and integration in multimodal analysis is discussed. An overview of the index types that can be distinguished, together with some examples, will be given in section 5.4. Finally, in section 5.5 we end with a perspective on open research questions.

As indicated earlier we focus here on the indexing of a recording of produced and authored documents. Hence, we start off form the recorded datastream without making use of any descriptions that could have been added at production time. Given that this is the most elaborate task many of the methods are also applicable in the other domains that we have considered in chapter 2.

5.2 Video document segmentation

For analysis purposes the process of authoring should be reversed. To that end, first a segmentation should be made that decomposes a video document in its layout and content elements. Results can be used for indexing specific segments. In many cases segmentation can be viewed as a classification problem, and hence pattern recognition techniques are appropriate. However, in video indexing literature many heuristic methods are proposed. We will first discuss reconstruction of the layout for each of the modalities. Finally, we will focus on segmentation of the content. The data flow necessary for analysis is visualized in figure 5.1.

5.2.1 Layout reconstruction

Layout reconstruction is the task of detecting the sensor shots and transition edits in the video data. For analysis purposes layout reconstruction is indispensable. Since the layout guides the spectator in experiencing the video document, it should also steer analysis.

For reconstruction of the visual layout, several techniques already exist to segment a video document on the camera shot level, known as *shot boundary detection*[†]. Various algorithms are proposed in video indexing literature to detect cuts in video documents, all of which rely on computing the dissimilarity of successive frames. The computation of dissimilarity can be at the pixel, edge, block, or frame level. Which one is important is largely dependent on the kind of changes in content present in the video, whether the camera is moving etc. The resulting dissimilarities as function of time are compared with some fixed or dynamic threshold. If the dissimilarity is sufficiently high a cut is declared.

Block level features are popular as they can be derived from motion vectors, which can be computed directly from the visual channel, when coded in MPEG, saving decompression time.

For an extensive overview of different cut detection methods we refer to the survey of Brunelli in [24] and the references therein. An overview of the current performance of cut detection algorithms can be found at the TRECVID benchmark on-line proceedings [1].

Detection of transition edits in the visual modality can be done in several ways. Since the transition is gradual, comparison of successive frames is insufficient. The first researchers exploiting this observation where Zhang et al [190]. They introduced the twin-comparison approach, using a dual threshold that accumulates significant differences to detect gradual transitions. For an extensive coverage of other methods we again refer to [24], we just summarize the methods mentioned. First, so called plateau detection uses every k-th frame. Another approach is based on effect modelling, where video production-based mathematical models are used to spot different edit effects using statistical classification. Finally, a third approach models the effect of a transition on intensity edges in subsequent frames.

Detection of abrupt cuts in the auditory layout can be achieved by detection of silences and transition points, i.e. locations where the category of the underlying signal changes. In literature different methods are proposed for their detection.

In [117] it is shown that average energy, E_n, is a sufficient measure for detecting silence segments. E_n is computed for a window, i.e. a set of n samples. If the average for all the windows in a segment are found lower than a threshold, a silence is marked. Another approach is taken in [192]. Here E_n is combined with the zero-crossing rate (ZCR), where a zero-crossing is said to occur if successive samples have different signs. A segment is classified as silence if E_n is consistently lower than a set of thresholds, or if most ZCRs are below a threshold. This method also includes unnoticeable noise.

Li et al [90] use silence detection for separating the input audio segment into silence segments and signal segments. For the detection of silence periods they use a three-step procedure. First, raw boundaries between silence and signal are marked in the auditory data. In the succeeding two steps a fill-in process and a throwaway process are applied to the results. In the fill-in process short silence segments are relabelled signal and in the throwaway process low energy signal segments are relabelled silence.

Besides silence detection [90] also detects transition points in the signal segments by using break detection and break merging. They compute an onset break, when a

[†] As an ironic legacy from early research on video parsing, this is also referred to as scene-change detection.

clear increase of signal energy is detected, and an offset break, when a clear decrease is found, to indicate a potential change in category of the underlying signal, by moving a window over the signal segment and compare E_n of different halves of the window at each sliding position. In the second step, adjacent breaks of the same type are merged into a single break.

In [192] music is distinguished from speech, silence, and environmental sounds based on features of the ZCR and the fundamental frequency. To assign the probability of being music to an audio segment, four features are used: the degree of being harmonic (based on the fundamental frequency), the degree to which the audio spectrum exhibits a clear peak during a period of time an indication for the presence of a fundamental frequency , the variance of the ZCR, and the range of the amplitude of the ZCR.

The first step in reconstructing the textual layout is referred to as tokenization, in this phase the input text is divided into units called tokens or characters. Detection of text shots can be achieved in different ways, depending on the granularity used. If we are only interested in single words we can use the occurrence of white space as the main clue. However, this signal is not necessarily reliable, because of the occurrence of periods, single apostrophes and hyphenation [98]. When more context is taken into account one can reconstruct sentences from the textual layout. Detection of periods is a basic heuristic for the reconstruction of sentences, about 90% of periods are sentence boundary indicators [98]. Transitions are typically found by searching for predefined patterns.

Since layout is very modality dependent, a multimodal approach for its reconstruction won't be very effective. The task of layout reconstruction can currently be performed quite reliably. However, results might improve even further when more advanced techniques are used, for example methods exploiting the learning capabilities of statistical classifiers.

5.2.2 Content segmentation

In subsection 2.5.2 we introduced the elements of content. Here we will discuss how to detect them automatically, using different detection algorithms exploiting visual, auditory, and textual information sources.

People detection

Detection of people in video documents can be done in several ways. They can be detected in the visual modality by means of their faces or other body parts, in the auditory modality by the presence of speech, and in the textual modality by the appearance of names. In the following, those modality specific techniques will be discussed in more detail. For an in-depth coverage of the different techniques we refer to the cited references.

Most approaches using the visual modality simplify the problem of people detection to detection of a human face. Face detection techniques aim to identify all image regions which contain a face, regardless of its three-dimensional position, orientation, and lighting conditions used, and if present return their image location and extents [187]. 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, see [187] for a comprehensive and critical survey of current face detection methods. From all methods currently available the one proposed by Rowley in [131] performs the best [126]. The neural network-based system is able to detect about 90% of all upright and frontal faces, and more important the system only sporadically mistakes non-face areas for faces.

When a face is detected in a video, face recognition techniques aim to identify the person. A common used method for face recognition is matching by means of *Eigenfaces* [121]. In eigenface methods templates of size let's say 20x20 are used. For the example numbers this leads to a 20x20=400 dimensional space. Using Principal Component Analysis a subspace capturing the most relevant information is computed. Every component in itself is again a 20x20 template. It allows to identify which information is most important in the matching process. A drawback of applying face recognition for video indexing, is its limited generic applicability [139]. Reported results [14,121,139] show that face recognition works in constrained environments, preferably showing a frontal face close to the camera. When using face recognition techniques in a video indexing context one should account for this limited applicability.

In [102] people detection is taken one step further, detecting not only the head, but the whole human body. The algorithm presented first locates the constituent components of the human body by applying detectors for head, legs, left arm, and right arm. Each individual detector is based on the Haar wavelet transform using specific examples. After ensuring that these components are present in the proper geometric configuration, a second example-based classifier combines the results of the component detectors to classify a pattern as either a person or a non-person.

A similar part-based approach is followed in [44] to detect naked people. First, large skin-colored components are found in an image by applying a skin filter that combines color and texture. Based on geometrical constraints between detected components an image is labelled as containing naked people or not. Obviously this method is suited for specific genres only.

The auditory channel also provides strong clues for presence of people in video documents through speech in the segment. When layout segmentation has been performed, classification of the different signal segments as speech can be achieved based on the features computed. Again different approaches can be chosen.

In [192] five features are checked to distinguish speech from other auditory signals. First one is the relation between amplitudes of ZCR and energy curves. The second one is the shape of the ZCR curve. The third and fourth features are the variance and the range of the amplitude of the ZCR curve. The fifth feature is about the property of the fundamental frequency within a short time window. A decision value is defined for each feature. Based on these features, classification is performed using the weighted average of these decision values.

A more elaborated audio segmentation algorithm is proposed in [90]. The authors are able to segment not only speech but also speech together with noise, speech or music with an accuracy of about 90%. They compared different auditory feature sets, and conclude that temporal and spectral features perform bad, as opposed to Mel-frequency cepstral coefficients (MFCC) and linear prediction coefficients (LPC) which achieve a much better classification accuracy.

When a segment is labelled as speech, speaker recognition can be used to identify a person based on his or her speech utterance. Different techniques are proposed, e.g. [107, 118]. A generic speaker identification system consisting of three modules is presented in [118]. In the first module feature extraction is performed using a set of 14 MFCC from each window. In the second module those features are used to classify each moving window using a nearest neighbor classifier. The classification is performed using a ground truth. In the third module results of each moving window are combined to generate a single decision for each segment. The authors report encouraging performance using speech segments of a feature film.

A strong textual cue for the appearance of people in a video document are words which are names. In [139], for example, natural language processing techniques using a dictionary, thesaurus, and parser are used to locate names in transcripts. The system calculates four different scores. The first measure is a grammatical score based on the part-of-speech tagging to find candidate nouns. The second is a lexical score indicating whether a detected noun is related to persons. The situational score is the third score giving an indication whether the word is related to social activities involving people. Finally, the positional score for each word in the transcripts measures where in the text of the newsreader the word is mentioned. A net likelihood score is then calculated which together with the name candidate and segment information forms the system's output. Related to this problem is the task of named entity recognition, which is known from the field of computational linguistics. Here one seeks to classify every word in a document into one of eight categories: person, location, organization, date, time, percentage, monetary value, or none of the above [17]. In the reference name recognition is viewed as a classification problem, where every word is either part of some name, or not. The authors use a variant of an HMM for the name recognition task based on a bi-gram language model. Compared to any other reported learning algorithm, their name recognition results are consistently better.

In conclusion, people detection in video can be achieved using different approaches, all having limitations. Variance in orientation and pose, together with occlusion, make visual detection error prone. Speech detection and recognition is still sensitive to noise and environmental sounds. Also, more research on detection of names in text is needed to improve results. As the errors in different modalities are not necessarily correlated, a multimodal approach in detection of persons in video documents can be an improvement. Besides improved detection, fusion of different modalities is interesting with respect to recognition of specific persons.

Object detection

Object detection forms a generalization of the problem of people detection. Specific objects can be detected by means of specialized visual detectors, motion, sounds, and appearance in the textual modality. Object detection methods for the different modalities will be highlighted here.

Approaches for object detection based on visual appearance can range from detection of specific objects to detection approaches of more general objects. An example from the former is given in [141], where the presence of passenger cars in image frames is detected by using multiple histograms. Each histogram represents the joint statistics of a subset of wavelet coefficients and their position on the object. The authors use statistical modelling to account for variation, which enables them to reliably detect passenger cars over a wide range of points of view.

In the above, we know what we are looking for and the number of classes is small so one can perform strong segmentation. If not, grouping based on motion i.e. weak segmentation is the best in absence of other knowledge. Moreover, since the appearance of objects might vary widely, rigid object motion detection is often the most valuable feature. Thus, when considering the approach for general object detection, motion is a useful feature. A typical method to detect moving objects of interest starts with a segmentation of the image frame. Regions in the image frame sharing similar motion are merged in the second stage. Result is a motion-based segmentation of the video. In [113] a method is presented that segments a single video frame into independently moving visual objects. The method follows a bottom-up approach, starting with a color-based decomposition of the frame. Regions are then merged based on their motion parameters via a statistical test, resulting in superior performance over other methods, e.g. [9, 185].

Specific objects can also be detected by analyzing the auditory layout segmentation of the video document. Typically, segments in the layout segmentation first need to be classified as environmental sounds. Subsequently, the environmental sounds are further analyzed for the presence of specific object sound patterns. In [184, 192] for example, specific object sound patterns e.g. dog bark, ringing telephones, and different musical instruments are detected by selecting the appropriate auditory features.

Detecting objects in the textual modality also remains a challenging task. A logical intermediate step in detecting objects of interest in the textual modality is part-of-speech tagging. Though limited, the information we get from tagging is still quite useful. By extracting and analyzing the nouns in tagged text for example, and to apply chunking [2], one can make some assumptions about objects present. To our knowledge chunking has not yet been used in combination with detection of objects in video documents. Its application however, might prove to be a valuable extension to unimodal object detection.

Successful detection of objects is limited to specific examples. A generic object detector still forms the holy grail in video document analysis. Therefore, multimodal object detection seems interesting. It helps if objects of interest can be identified within different modalities. Then the specific visual appearance, the specific sound, and its mentioning in the accompanying textual data can yield the evidence for robust recognition.

Setting detection

For the detection of setting, motion is not so relevant, as the setting is usually static. Therefore, techniques from the field of content-based image retrieval can be used. See [151] for a complete overview of this field. By using for example key frames, those techniques can easily be used for video indexing. We focus here on methods that assign a setting label to the data, based on analysis of the visual, auditory, or textual modality.

In [167] images are classified as either indoor or outdoor, using three types of visual features: one for color, texture, and frequency information. Instead of computing features on the entire image, the authors use a multi-stage classification approach. First, sub-blocks are classified independently, and afterwards another classification is performed using the k-nearest neighbor classifier.

Outdoor images are further classified into city and landscape images in [175]. Features used are color histograms, color coherence vectors, Discrete Cosine Transform (DCT) coefficients, edge direction histograms, and edge direction coherence vectors. Classification is done with a weighted k-nearest neighbor classifier with leave-one out method. Reported results indicate that the edge direction coherence vector has good discriminatory power for city vs. landscape. Furthermore, it was found that color can be an important cue in classifying natural landscape images into forests, mountains, or sunset/sunrise classes. By analyzing sub-blocks, the authors detect the presence of sky and vegetation in outdoor image frames in another paper. Each sub-block is independently classified, using a Bayesian classification framework, as sky vs. non-sky or vegetation vs. non-vegetation based on color, texture, and position features [174].

Detecting setting based on auditory information, can be achieved by detecting specific environmental sound patterns. In [184] the authors reduce an auditory segment to a small set of parameters using various auditory features, namely loudness, pitch, brightness, bandwidth, and harmonicity. By using statistical techniques over the parameter space the authors accomplish classification and retrieval of several sound patterns including laughter, crowds, and water. In [192] classes of natural and synthetic sound patterns are distinguished by using an HMM, based on timbre and rhythm. The authors are capable of classifying different environmental setting sound patterns, including applause, explosions, rain, river flow, thunder, and

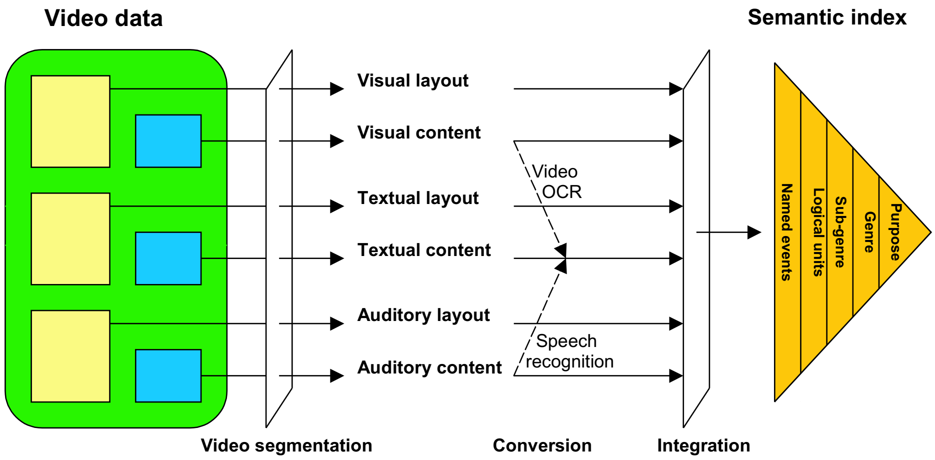


Figure 5.2: *Role of conversion and integration in multimodal video document analysis.*

windstorm.

The transcript is used in [31] to extract geographic reference information for the video document. The authors match named places to their spatial coordinates. The process begins by using the text metadata as the source material to be processed. A known set of places along with their spatial coordinates, i.e. a gazetteer, is created to resolve geographic references. The gazetteer used consists of approximately 300 countries, states and administrative entities, and 17000 major cities worldwide. After post processing steps, e.g. including related terms and removing stop words, the end result are segments in a video sequence indexed with latitude and longitude.

We conclude that the visual and auditory modality are well suited for recognition of the environment in which the video document is situated. By using the textual modality, a more precise (geographic) location can be extracted. Fusion of the different modalities may provide the video document with semantically interesting setting terms such as: outside vegetation in Brazil near a flowing river. Which can never be derived from one of the modalities in isolation.

5.3 Multimodal analysis

After reconstruction of the layout and content elements, the next step in the inverse analysis process is analysis of the layout and content to extract the semantic index. At this point the modalities should be integrated. However, before analysis, it might be useful to apply modality conversion of some elements into more appropriate form. The role of conversion and integration in multimodal video document analysis will be discussed in this section, and is illustrated in figure 5.2.

5.3.1 Conversion

For analysis, conversion of elements of visual and auditory modalities to text is most appropriate.

A typical component we want to convert from the visual modality is overlayed text. Video Optical Character Recognition (OCR) methods for detection of text in video frames can be divided into component-based, e.g. [146], or texture-based methods, e.g. [91]. A method utilizing the DCT coefficients of compressed video was proposed in [194]. By using Video OCR methods, the visual overlayed text object can be converted into a textual format. The quality of the results of Video OCR vary, depending on the kind of characters used, their color, their stability over time, and the quality of the video itself.

From the auditory modality one typically wants to convert the uttered speech into transcripts. Available speech recognition systems are known to be mature for applications with a single speaker and a limited vocabulary. However, their performance degrades when they are used in real world applications instead of a lab environment [24]. This is especially caused by the sensitivity of the acoustic model to different microphones and different environmental conditions. Since conversion of speech into transcripts still seems problematic, integration with other modalities might prove beneficial.

Note that other conversions are possible, e.g. computer animation can be viewed as converting text to video. However, these are relevant for presentation purposes only.

5.3.2 Integration

The purpose of integration of multimodal layout and content elements is to improve classification performance. To that end the addition of modalities may serve as a verification method, a method compensating for inaccuracies, or as an additional information source.

An important aspect, indispensable for integration, is synchronization and alignment of the different modalities, as all modalities must have a common timeline. Typically the time stamp is used. We observe that in literature modalities are converted to a format conforming to the researchers main expertise. When audio is the main expertise, image frames are converted to (milli)seconds, e.g. [73]. In [7,38] image processing is the main expertise, and audio samples are assigned to image frames or camera shots. When a time stamp isn't available, a more advanced alignment procedure is necessary. Such a procedure is proposed in [78]. The error prone output of a speech recognizer is compared and aligned with the accompanying closed captions of news broadcasts. The method first finds matching sequences of words in the transcript and closed caption by performing a dynamic-programming[‡] based alignment between the two text strings. Segments are then selected when sequences of three or more words are similar in both resources.

To achieve the goal of multimodal integration, several approaches can be followed. We categorize those approaches by their distinctive properties with respect to the processing cycle, the content segmentation, and the classification method used. The processing cycle of the integration method can be iterated, allowing for incremental use of context, or non-iterated. The content segmentation can be performed by using the different modalities in a symmetric, i.e. simultaneous, or asymmetric, i.e. ordered, fashion. Finally, for the classification one can choose between a statistical or knowledge-based approach. An overview of the different integration methods found in literature is in table 5.1.

Most integration methods reported are symmetric and non-iterated. Some follow a knowledge-based approach for classification of the data into classes of the semantic index hierarchy [43,106,125,137,171]. In [171] for example, the auditory and visual modality are integrated to detect speech, silence, speaker identities, no face shot / face shot / talking face shot using knowledge-based rules. First, talking people are detected by detecting faces in the camera shots, subsequently a knowledge-based measure is evaluated based on the amount of speech in the shot.

Many methods in literature follow a statistical approach [7, 26, 37, 38, 73, 78, 79, 109, 139, 182]. An example of a symmetric, non-iterated statistical integration

[‡] Dynamic programming is a programming technique in which intermediate results in some iterative process are stored so they can be reused in later iterations, rather than recomputed.

Table 5.1: *An overview of different integration methods.*

	Content Segmentation		**Classification Method**		**Processing Cycle**	
	Symmetric	*Asymmetric*	*Statistical*	*Knowledge*	*Iterated*	*Non-Iterated*
[7]	✓		✓			✓
[12]		✓		✓	✓	
[26]	✓		✓			✓
[37]	✓		✓			✓
[38]	✓		✓			✓
[43]	✓			✓		✓
[73]	✓		✓			✓
[73]		✓	✓			✓
[78]	✓		✓			✓
[79]	✓		✓			✓
[106]	✓			✓		✓
[109]	✓		✓		✓	
[125]	✓			✓		✓
[137]	✓			✓		✓
[139]	✓		✓			✓
[163]		✓		✓	✓	
[171]	✓			✓		✓
[182]	✓		✓			✓

method is the Name-It system presented in [139]. The system associates detected faces and names, by calculating a co-occurrence factor that combines the analysis results of face detection and recognition, name extraction, and caption recognition. A high-occurrence factor indicates that a certain visual face template is often associated with a certain name in either the caption in the image, or in the associated text hence a relation between face and name can be concluded.

Hidden Markov Models are frequently used as a statistical classification method for multimodal integration [7, 37, 38, 73]. A clear advantage of this framework is that it is not only capable to integrate multimodal features, but is also capable to include sequential features. Moreover, an HMM can also be used as a classifier combination method.

When modalities are independent, they can easily be included in a product HMM. In [73] such a classifier is used to train two modalities separately, which are then combined symmetrically, by computing the product of the observation probabilities. It is shown that this results in significant improvement over a unimodal approach.

In contrast to the product HMM method, a neural network-based approach doesn't assume features are independent. The approach presented in [73], trains an HMM for each modality and category. A three layer perceptron is then used to combine the outputs from each HMM in a symmetric and non-iterated fashion.

Another advanced statistical classifier for multimodal integration was recently proposed in [109]. A probabilistic framework for semantic indexing of video documents based on so called multijects and multinets is presented. The multijects model content elements which are integrated in the multinets to model the relations between objects, allowing for symmetric use of modalities. For the integration in the multinet the authors propose a Bayesian belief network [119], which is a probabilistic description of the relation between different variables. Significant improvements of detection performance is demonstrated. Moreover, the framework supports detection based on iteration. Viability of the Bayesian network as a symmetric integrating classifier was also demonstrated in [79], however that method doesn't support iteration.

In contrast to the above symmetric methods, an asymmetric approach is presented in [73]. A two-stage HMM is proposed which first separates the input video document into three broad categories based on the auditory modality, in the second stage another HMM is used to split those categories based on the visual modality. A drawback of this method is its application dependency, which may result in less effectiveness in other classification tasks.

An asymmetric knowledge-based integration method, supporting iteration, was proposed in [12]. First, the visual and textual modality are combined to generate semantic index results. Those form the input for a post-processing stage that uses those indexes to search the visual modality for the specific time of occurrence of the semantic event.

For exploration of other integration methods, we again take a look in the field of content-based image retrieval. From this field methods are known to integrate the visual and textual modality by combining images with associated captions or HTML tags. Early reported methods used a knowledge base for integration, e.g. the Piction system [163]. This system uses modalities asymmetrically, it first analyzes the caption to identify the expected number of faces and their expected relative positions. Then a face detector is applied to a restricted part of the image, if no faces are detected an iteration step is performed that relaxes the thresholds. More recently, Latent Semantic Indexing (LSI) [35] has become a popular means for integration [26, 182]. LSI is symmetric and non-iterated and works by statistically associating related words to the conceptual context of the given document. In effect it relates documents that use similar terms, which for images are related to features in the image. Thus, it has a strong relation to co-occurrence based methods. In [26] LSI is used to capture text statistics in vector form from an HTML document. Words with specific HTML tags are given higher weights. In addition, the position of the words with respect to the position of the image in the document is also accounted for. The image features, that is the color histogram and the dominant orientation histogram, are also captured in vector form and combined they form a unified vector that the authors use for content-based search of a WWW-based image database. Reported experiments show that maximum improvement was achieved when both visual and textual information are employed.

In conclusion, video indexing results improve when a multimodal approach is followed. Not only because of enhancement of content findings, but also because more information is available. Most methods integrate in a symmetric and non-iterated fashion. Usage of incremental context by means of iteration can be a valuable addition to the success of the integration process. Usage of combined statistical classifiers in multimodal video indexing literature is still scarce, though various successful statistical methods for classifier combinations are known, e.g. bagging, boosting, or stacking [76]. So, probably results can be improved even more substantially when advanced classification methods from the field of statistical pattern recognition, or other disciplines are used, preferably in an iterated fashion.

5.4 Semantic video indexes

The methodologies described in section 5.3 have been applied to extract a variety of the different video indexes described in subsection 2.5.1. In this section we systematically report on the different indexes and the information from which they are derived. As methods for extraction of purpose are not mentioned in literature, this level is excluded. Figure 5.3 presents an overview of all indexes and the methods in literature which can derive them.

5.4.1 Genre

“Editing is an important stylistic element because it affects the overall rhythm of the video document” [19]. Hence, layout related statistics are well suited for indexing a video document into a specific genre. Most obvious element of this editorial style is the average shot length. Generally, the longer the shots, the slower the rhythm of the video document.

The rate of shot changes together with the presence of black frames is used in [67] to detect commercials within news broadcast. The rationale behind detection of black frames is that they are often broadcasted for a fraction of a second before, after, and between commercials. However, black frames can also occur for other reasons. Therefore, the authors use the observation that advertisers try to make commercials more interesting by rapidly cutting between different shots, resulting in a higher shot change rate. A similar approach is followed in [92], for detecting commercials within broadcasted feature films. Besides the detection of monochrome frames and shot change rate, the authors use the edge change ratio and motion vector length to capture high action in commercials.

Average shot length, the percentage of different types of edit transitions, and six visual content features, are used in [170] to classify a video document into cartoons, commercials, music, news and sports video genres. As a classifier a specific decision tree called C4.5 is used [85] which can work both on real and symbolic values.

In [37] the observation is made that different genres exhibit different temporal patterns of face locations. They furthermore observe that the temporal behavior of overlaid text is genre dependent. In fact the following genre dependent functions can be identified:

- *News*: annotation of people, objects, setting, and named events;
- *Sports*: player identification, game related statistics;
- *Movies/TV series*: credits, captions, and language translations;
- *Commercials*: product name, claims, and disclaimers;

Based on results of face and text tracking, each frame is assigned one of 15 labels, describing variations on the number of appearing faces and/or text lines together with the distance of a face to the camera. These labels form the input for an HMM, which classifies an input video document into news, commercials, sitcoms, and soaps based on maximum likelihood.

Detection of generic sport video documents seems almost impossible due to the large variety in sports. In [84], however, a method is presented that is capable of identifying mainstream sports videos. Discriminating properties of sport videos are the presence of slow-motion replays, large amounts of overlayed text, and specific camera/object motion. The authors propose a set of eleven features to capture these properties, and obtain 93% accuracy using a decision tree classifier. Analysis showed that motion magnitude and direction of motion features yielded the best results.

Methods for indexing video documents into a specific genre using a multimodal approach are reported in [43, 73, 79]. In [73] news reports, weather forecasts, commercials, basketball, and football games are distinguished based on audio and visual information. The authors compare different integration methods and classifiers and conclude that a product HMM classifier is most suited for their task, see also 5.3.2.

The same modalities are used in [43]. The authors present a three-step approach. In the first phase, content features such as color statistics, motion vectors and audio statistics are extracted. Secondly, layout features are derived, e.g. shot lengths, camera motion, and 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 layout and content attributes of video for analysis, and can find five (sub)genres, namely news broadcasts, car racing, tennis, commercials, and animated cartoons.

Besides auditory and visual information, [79] also exploits the textual modality. The segmentation and indexing approach presented uses three layers to process low-, mid-, and high-level information. At the lowest level features such as color, shape, MFCC, ZCR, and the transcript are extracted. Those are used in the mid-level to detect faces, speech, keywords, etc. At the highest level the semantic index is extracted through the integration of mid-level features across the different modalities, using Bayesian networks, as noted in subsection 5.3.2. In its current implementation the presented system classifies segments as either part of a talk show, commercial or financial news.

5.4.2 Sub-genre

Research on indexing sub-genres, or specific instances of a genre, has been geared mainly towards sport videos [43, 73, 135] and commercials [32]. Obviously, future index techniques may also extract other sub-genres, for example westerns, comedies, or thrillers within the feature film genre.

Four sub-genres of sport video documents are identified in [135]: basketball, ice hockey, soccer, and volleyball. The full motion fields in consecutive frames are used as a feature. To reduce the feature space, Principal Component Analysis is used. For classification two different statistical classifiers were applied. It was found that a continuous observation density Markov model gave the best results. The sequences analyzed were post-edited to contain only the play of the sports, which is a drawback of the presented system. For instance, no crowd scenes or time outs were included. Some sub-genres of sport video documents are also detected in [43, 73], as noted in section 5.4.1.

An approach to index commercial videos based on semiotic and semantic properties is presented in [32]. The general field of semiotics is the study of signs and symbols, what they mean and how they are used. For indexing of commercials the semiotics approach classifies commercials into four different sub-genres that relate to the narrative of the commercial. The following four sub-genres are distinguished: practical, critical, utopic, and playful commercials. Perceptual features e.g. saturated colors, horizontal lines, and the presence or absence of recurring colors, are mapped onto the semiotic categories. Based on research in the marketing field, the authors also formalized a link between editing, color, and motion effects on the one hand, and feelings that the video arouses in the observer on the other. Characteristics of a commercial are related to those feelings and have been organized in a hierarchical fashion. A main classification is introduced between commercials that induce feelings of *action* and those that induce feelings of *quietness*. The authors subdivide action further into suspense and excitement. Quietness is further specified in relaxation and happiness.

5.4.3 Logical units

Detection of logical units in video documents is extensively researched with respect to the detection of scenes or Logical Story Units (LSU) in feature films and sitcoms. An overview and evaluation of such methods is presented in [177]. A summary of that paper follows. After that we consider how to give the LSU a proper label.

Logical story unit detection

In cinematography an LSU is defined as *a series of shots that communicate a unified action with a common locale and time* [19]. Viewers perceive the meaning of a video at the level of LSUs [20, 133].

A problem for LSU segmentation using visual similarity is that it seems to conflict with its definition based on the semantic notion of common locale and time. There is no one-to-one mapping between the semantic concepts and the data-driven visual similarity. In practice, however, most LSU boundaries coincide with a change of locale, causing a change in the visual content of the shots. Furthermore, usually the scenery in which an LSU takes place does not change significantly, or foreground objects will appear in several shots, e.g. talking heads in the case of a dialogue. Therefore, visual similarity provides a proper base for common locale.

There are two complicating factors regarding the use of visual similarity. Firstly, not all shots in an LSU need to be visually similar. For example, one can have a sudden close-up of a glass of wine in the middle of a dinner conversation showing talking heads. This problem is addressed by the *overlapping links* approach [60] which assigns visually dissimilar shots to an LSU based on temporal constraints. Secondly, at a later point in the video, time and locale from one LSU can be repeated in another, not immediate succeeding LSU.

The two complicating factors apply to the entire field of LSU segmentation based on visual similarity. Consequently, an LSU segmentation method using visual similarity depends on the following three assumptions:

Assumption 1 *The visual content in an LSU is dissimilar from the visual content in a succeeding LSU.*

Assumption 2 *Within an LSU, shots with similar visual content are repeated.*

Assumption 3 *If two shots σ_x and σ_y are visually similar and assigned to the same LSU, then all shots between σ_x and σ_y are part of this LSU.*

For parts of a video where the assumptions are not met, segmentation results will be unpredictable.

Given the assumptions, LSU segmentation methods using visual similarity can be characterized by two important components, viz. the shot distance measurement and the comparison method. The former determines the (dis)similarity mentioned in assumptions 1 and 2. The latter component determines which shots are compared in finding LSU boundaries. Both components are described in more detail.

Shot distance measurement. The shot distance δ represents the dissimilarity between two shots and is measured by combining (typically multiplying) measurements for the *visual distance* δ^v and the *temporal distance* δ^t. The two distances will now be explained in detail.

Visual distance measurement consists of dissimilarity function δ^v_f for a visual feature f measuring the distance between two shots. Usually a threshold τ^v_f is used to determine whether two shots are close or not. δ^v_f and τ^v_f have to be chosen such that the distance between shots in an LSU is small (assumption 2), while the distance between shots in different LSUs is large (assumption 1).

Segmentation methods in literature do not depend on specific features or dissimilarity functions, i.e. the features and dissimilarity functions are interchangeable amongst methods.

Temporal distance measurement consists of temporal distance function δ^t. As observed before, shots from not immediate succeeding LSUs can have similar content. Therefore, it is necessary to define a time window τ^t, determining what shots in a video are available for comparison. The value for τ^t, expressed in shots or frames, has to be chosen such that it resembles the length of an LSU. In practice, the value has to be estimated since LSUs vary in length.

Function δ^t is either binary or continuous. A binary δ^t results in 1 if two shots are less than τ^t shots or frames apart and ∞ otherwise [189]. A continuous δ^t reflects the distance between two shots more precisely. In [133], δ^t ranges from 0 to

Table 5.2: Classification of LSU segmentation methods.

	Temporal distance function	
Comparison method	*Binary*	*Continuous*
Sequential	Overlapping links [60], [86], [89], [6], [29]	Continuous video coherence [82], [166], [95]
Clustering	Time constrained clustering [189], [93], [135]	Time adaptive grouping [133], [82], [178]

1. As a consequence, the further two shots are apart in time, the closer the visual distance has to be to assigned them to the same LSU. Time window τ^t is still used to mark the point after which shots are considered dissimilar. Shot distance is then set to ∞ regardless of the visual distance.

The *comparison method* is the second important component of LSU segmentation methods. In *sequential iteration*, the distance between a shot and other shots is measured pair-wise. In *clustering*, shots are compared group-wise. Note that in the sequential approach still many comparisons can be made, but always of one pair of shots at the time.

Methods from literature can now be classified according to the framework. The visual distance function is not discriminatory, since it is interchangeable amongst methods. Therefore, the two discriminating dimensions for classification of methods are temporal distance function and comparison method. Their names in literature and references to methods are given in table 5.2. Note that in [82] 2 approaches are presented.

Labelling logical units

Detection of LSU boundaries alone is not enough. For indexing, we are especially interested in its accompanying label.

A method that is capable of detecting dialogue scenes in movies and sitcoms, is presented in [7]. Based on audio analysis, face detection, and face location analysis the authors generate output labels which form the input for an HMM. The HMM outputs a scene labeled as either, establishing scene, transitional scene, or dialogue scene. According to the results presented, combined audio and face information gives the most consistent performance of different observation sets and training data. However, in its current design, the method is incapable of differentiating between dialogue and monologue scenes.

A technique to characterize and index violent scenes in general TV drama and movies is presented in [106]. The authors integrate cues from both the visual and auditory modality symmetrically. First, a measure of activity for each video shot is computed as a measure of action. This is combined with detection of flames and blood using a predefined color table. The corresponding audio information provides supplemental evidence for the identification of violent scenes. The focus is on the abrupt change in energy level of the audio signal, computed using the energy entropy criterion. As a classifier the authors use a knowledge-based combination of feature values on scene level.

By utilizing a symmetric and non-iterated multimodal integration method four different types of scenes are identified in [137]. The audio signal is segmented into silence, speech, music, and miscellaneous sounds. This is combined with a visual similarity measure, computed within a temporal window. Dialogues are then detected based on the occurrence of speech and an alternated pattern of visual labels, indicating a change of speaker. When the visual pattern exhibits a repetition the scene is labeled as story. When the audio signal isn't labeled as speech, and the visual information exhibits a sequence of visually non-related shots, the scene is labeled as action. Finally, scenes that don't fit in the aforementioned categories are indexed as generic scenes.

In contrast to [137], a unimodal approach based on the visual information source is used in [188] to detect dialogues, actions, and story units. Shots that are visually similar and temporally close to each other are assigned the same (arbitrary) label. Based on the patterns of labels in a scene, it is indexed as either dialogue, action, or story unit.

A scheme for reliably identifying logical units which clusters sensor shots according to detected dialogues, similar settings, or similar audio is presented in [125]. The method starts by calculating specific features for each camera and microphone shot. Auditory, 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 the features, resulting in a so called distance table. Based on the distance tables, shots are merged into logical units using absolute and adaptive thresholds.

News broadcasts are far more structured than feature films. Researchers have exploited this to classify logical units in news video using a model-based approach. Especially anchor shots, i.e. shots in which the newsreader is present, are easy to model and therefore easy to detect. Since there is only minor body movement they can be detected by comparison of the average difference between (regions in) successive frames. This difference will be minimal. This observation is used in [55, 145, 191]. In [55, 145] also the restricted position and size of detected faces is used.

Another approach for the detection of anchor shots is taken in [16, 59, 75]. Repetition of visually similar anchor shots throughout the news broadcast is exploited. To refine the classification of the similarity measure used, [16] requires anchor shots candidates to have a motion quantity below a certain threshold. Each shot is classified as either anchor or report. Moreover, textual descriptors are added based on extracted captions and recognized speech. To classify report and anchor shots, the authors in [75] use face and lip movement detection. To distinguish anchor shots, the aforementioned classification is extended with the knowledge that anchor shots are graphically similar and occur frequently in a news broadcast. The largest cluster of similar shots is therefore assigned to the class of anchor shots. Moreover, the detection of a title caption is used to detect anchor shots that introduce a new topic. In [59] anchor shots are detected together with silence intervals to indicate report boundaries. Based on a topics database the presented system finds the most probable topic per report by analyzing the transcribed speech. Opposed to [16, 75], final descriptions are not added to shots, but to a sequence of shots that constitute a complete report on one topic. This is achieved by merging consecutive segments with the same topic in their list of most probable topics.

Besides the detection of anchor persons and reports, other logical units can be identified. In [38] six main logical units for TV broadcast news are distinguished, namely, begin, end, anchor, interview, report, and weather forecast. Each logical unit is represented by an HMM. For each frame of the video one feature vector is calculated consisting of 25 features, including motion and audio features. The resulting feature vector sequence is assigned to a logical unit based on the sequence of HMMs that maximizes the probability of having generated this feature vector sequence. By using this approach parsing and indexing of the video is performed in one pass through the video only.

Other examples of highly structured TV broadcasts are talk and game shows. In [80] a method is presented that detects guest and host shots in those video documents. The basic observation used is that in most talk shows the same person is host for the duration of the program but guests keep on changing. Also host shots are typically shorter since only the host asks questions. For a given show, the key frames of the N shortest shots containing one detected face are correlated in time to find the shot most often repeated. The key host frame is then compared against all key frames to detect all similar host shots, and guest shots.

In [186] a model for segmenting soccer video into the logical units break and play is given. A grass-color ratio is used to classify frames into three views according to video shooting scale, namely global, zoom-in, and close-up. Based on segmentation rules, the different views are mapped. Global views are classified as play and close-ups as breaks if they have a minimum length. Otherwise a neighborhood voting heuristic is used for classification.

5.4.4 Named events

Named events are at the lowest level in the semantic index hierarchy. For their detection different techniques have been used.

A three-level event detection algorithm is presented in [57]. The first level of the algorithm extracts generic color, texture, and motion features, and detects spatio-temporal object. The mid-level employs a domain dependent neural network to verify whether the detected objects belong to conceptual objects of interest. The generated shot descriptors are then used by a domain-specific inference process at the third level to detect the video segments that contain events of interest. To test the effectiveness of the algorithm the authors applied it to detect animal hunt events in wildlife documentaries.

Violent events and car chases in feature films are detected in [103], based on analysis of environmental sounds. First, low level sounds as engines, horns, explosions, or gunfire are detected, which constitute part of the high level sound events. Based on the dominance of those low level sounds in a segment it is labeled with a high level named event.

Walking shots, gathering shots, and computer graphics shots in broadcast news are the named events detected in [75]. A walking shot is classified by detecting the characteristic repetitive up and down movement 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 a gathering shot. Finally, computer graphics shots are classified by a total lack of motion in a series of frames.

The observation that authors use lightning techniques to intensify the drama of certain scenes in a video document is exploited in [169]. An algorithm is presented that detects flashlights, which is used as an identifier for dramatic events in feature films, based on features derived from the average frame luminance and the frame area influenced by the flashing light. Five types of dramatic events are identified that are related to the appearance of flashlights, i.e. supernatural power, crisis, terror, excitement, and generic events of great importance.

Whereas a flashlight can indicate a dramatic event in feature films, slow motion replays are likely to indicate semantically important events in sport video documents. In [116] a method is presented that localizes such events by detecting slow motion replays. The slow-motion segments are modelled and detected by an HMM.

One of the most important events in a sport video document is a score. In [12] a link between the visual and textual modalities is made to identify events that change the score in American football games. The authors investigate whether a chain of keywords, corresponding to an event, is found from the closed caption stream or not. In the time frames corresponding to those keywords, the visual stream is analyzed. Key frames of camera shots in the visual stream are compared with predefined templates using block matching based on the color distribution. Finally, the shot is indexed by the most likely score event, for example a touchdown.

Besides American football, methods for detecting events in tennis [101, 165, 193], soccer [21, 53], baseball [132, 193] and basketball [140, 195] are reported in

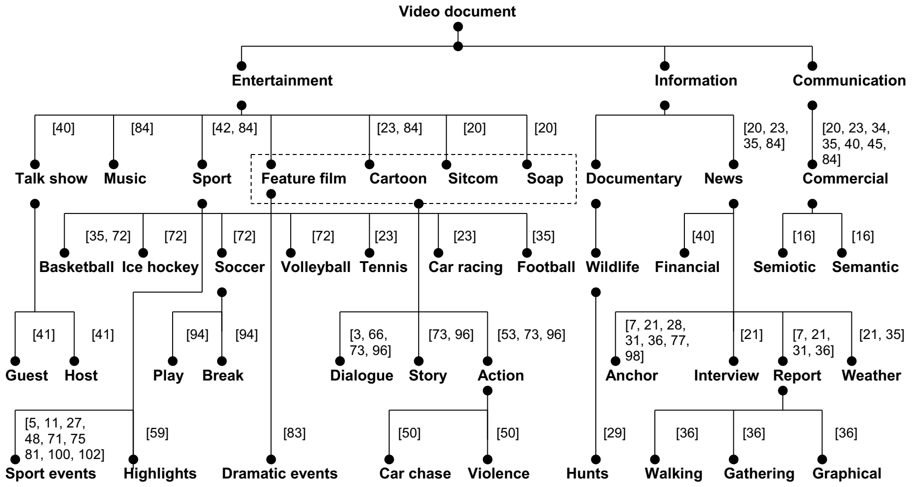


Figure 5.3: *Semantic index hierarchy with instances as found in literature. From top to bottom instances from genre, sub-genre, logical units, and named events. The dashed box is used to group similar nodes. Note, this picture is taken from [156], the numbers indicated as references are not correct.*

literature. Commonly, the methods presented exploit domain knowledge and simple (visual) features related to color, edges, and camera/object motion to classify typical sport specific events e.g. smashes, corner kicks, and dunks using a knowledge-based classifier. An exception to this common approach is [132], which presents an algorithm that identifies highlights in baseball video by analyzing the auditory modality only. Highlight events are identified by detecting excited speech of the commentators and the occurrence of a baseball pitch and hit.

Besides semantic indexing, detection of named events also forms a great resource for reuse of video documents. Specific information can be retrieved and reused in different contexts, or reused to automatically generate summaries of video documents. This seems especially interesting for, but is not limited to, video documents from the sport genre.

5.4.5 Discussion

Now that we have described the different semantic index techniques, as encountered in literature, we are able to distinguish the most prominent content and layout properties per genre. As variation in the textual modality is in general too diverse for differentiation of genres, and more suited to attach semantic meaning to logical units and named events, we focus here on properties derived from the visual and auditory modality only. Though, a large amount of genres can be distinguished, we limit ourselves to the ones mentioned in the semantic index hierarchy in figure 5.3, i.e. talk show, music, sport, feature film, cartoon, sitcom, soap, documentary, news, and commercial. For each of those genres we describe the characteristic properties.

Most prominent property of the first genre, i.e. talk shows, is their well-defined structure, uniform setting, and prominent presence of dialogues, featuring mostly non-moving frontal faces talking close to the camera. Besides closing credits, there is in general a limited use of overlayed text.

Whereas talk shows have a well-defined structure and limited setting, music clips show great diversity in setting and mostly have ill-defined structure. Moreover, music will have many short camera shots, showing lots of camera and object motion, separated by many gradual transition edits and long microphone shots containing music. The use of overlayed text is mostly limited to information about the performing artist and the name of the song on a fixed position.

Sport broadcasts come in many different flavors, not only because there exist a tremendous amount of sport sub-genres, but also because they can be broadcasted live or in summarized format. Despite this diversity, most authored sport broadcasts are characterized by a voice over reporting on named events in the game, a watching crowd, high frequency of long camera shots, and overlayed text showing game and player related information on a fixed frame position. Usually sport broadcasts contain a vast amount of camera motion, objects, and players within a limited uniform setting. Structure is sport-specific, but in general, a distinction between different logical units can be made easily. Moreover, a typical property of sport broadcasts is the use of replays showing events of interest, commonly introduced and ended by a gradual transition edit.

Feature film, cartoon, sitcom, and soap share similar layout and content properties. They are all dominated by people (or toons) talking to each other or taking part in action scenes. They are structured by means of scenes. The setting is mostly limited to a small amount of locales, sometimes separated by means of visual, e.g. gradual, or auditory, e.g. music, transition edits. Moreover, setting in cartoons is characterized by usage of saturated colors, also the audio in cartoons is almost noise-free due to studio recording of speech and special effects. For all mentioned genres the usage of overlayed text is limited to opening and/or closing credits. Feature film, cartoon, sitcom, and soap differ with respect to people appearance, usage of special effects, presence of object and camera motion, and shot rhythm. Appearing people are usually filmed frontal in sitcoms and soaps, whereas in feature films and cartoons there is more diversity in appearance of people or toons. Special effects are most prominent in feature films and cartoons, laughter of an imaginary public is sometimes added to sitcoms. In sitcoms and soaps there is limited camera and object motion. In general cartoons also have limited camera motion, though object motion appears more frequently. In feature films both camera and object motion are present. With respect to shot rhythm it seems legitimate to state that this has stronger variation in feature films and cartoons. The perceived rhythm will be slowest for soaps, resulting in more frequent use of camera shots with relative long duration.

Documentaries can also be characterized by their slow rhythm. Other properties that are typical for this genre are the dominant presence of a voice over narrating about the content in long microphone shots. Motion of camera and objects might be present in the documentary, the same holds for overlayed text. Mostly there is no well-defined structure. Special effects are seldom used in documentaries.

Most obvious property of news is its well-defined structure. Different news reports and interviews are alternated by anchor persons introducing, and narrating about, the different news topics. A news broadcast is commonly ended by a weather forecast. Those logical units are mostly dominated by monologues, e.g. people talking in front of a camera showing little motion. Overlayed text is frequently used on fixed positions for annotation of people, objects, setting, and named events. A report on an incident may contain camera and object motion. Similarity of studio setting is also a prominent property of news broadcasts, as is the abrupt nature of transitions between sensor shots.

Some prominent properties of the final genre, i.e. commercials, are similar to those of music. They have a great variety in setting, and share no common structure, although they are authored carefully, as the message of the commercial has to be conveyed in twenty seconds or so. Frequent usage of abrupt and gradual transition, in both visual and auditory modality, is responsible for the fast rhythm. Usually lots of object and camera motion, in combination with special effects, such as a loud volume, is used to attract the attention of the viewer. Difference with music is that black frames are used to separate commercials, the presence of speech, the superfluous and non-fixed use of overlayed text, a disappearing station logo, and the fact that commercials usually end with a static frame showing the product or brand of interest.

Due to the large variety in broadcasting formats, which is a consequence of guidance by different authors, it is very difficult to give a general description for the structure and characterizing properties of the different genres. When considering sub-genres this will only become more difficult. Is a sports program showing highlights of today's sport matches a sub-genre of sport or news? Reducing the prominent properties of broadcasts to instances of layout and content elements, and splitting of the broadcasts into logical units and named events seems a necessary intermediate step to arrive at a more consistent definition of genre and sub-genre. More research on this topic is still necessary.

5.5 Conclusion

Viewing a video document from the perspective of its author, enabled us to present a framework for multimodal video indexing. This framework formed the starting point for our review on different state-of-the-art video indexing techniques. Moreover, it allowed us to answer the three different questions that arise when assigning an index to a video document. The question *what to index* was answered by reviewing different techniques for layout reconstruction. We presented a discussion on reconstruction of content elements and integration methods to answer the *how to index* question. Finally, the *which index* question was answered by naming different present and future index types within the semantic index hierarchy of the proposed framework.

At the end of this review we stress that multimodal analysis is the future. However, more attention, in the form of research, needs to be given to the following factors:

1. *Content segmentation*

   Content segmentation forms the basis of multimodal video analysis. In contrast to layout reconstruction, which is largely solved, there is still a lot to be gained in improved segmentation for the three content elements, i.e. people, objects, and setting. Contemporary detectors are well suited for detection and recognition of content elements within certain constraints. Most methods for detection of content elements still adhere to a unimodal approach. A multimodal approach might prove to be a fruitful extension. It allows to take additional context into account. Bringing the semantic index on a higher level is the ultimate goal for multimodal analysis. This can be achieved by the integrated use of different robust content detectors or by choosing a constrained domain that ensures the best detection performance for a limited detector set.

2. *Modality usage*

   Within the research field of multimodal video indexing, focus is still too much geared towards the visual and auditory modality. The semantic rich textual modality is largely ignored in combination with the visual or auditory modality. Specific content segmentation methods for the textual modality will have their reflection on the semantic index derived. Ultimately this will result in semantic descriptions that make a video document as accessible as a text document.

3. *Multimodal integration*

   The integrated use of different information sources is an emerging trend in video indexing research. All reported integration methods indicate an improvement of performance. Most methods integrate in a symmetric and non-iterated fashion. Usage of incremental context by means of iteration can be a valuable addition to the success of the integration process. Most successful integration methods reported are based on the HMM and Bayesian network framework, which can be considered as the current state-of-the-art in multimodal integration. There seems to be a positive correlation between usage of advanced integration methods and multimodal video indexing results. This paves the road for the exploration of classifier combinations from the field of statistical pattern recognition, or other disciplines, within the context of multimodal video indexing.

4. *Technique taxonomy*

   We presented a semantic index hierarchy that grouped different index types as found in literature. Moreover we characterized the different genres in terms of their most prominent layout and content elements, and by splitting its structure into logical units and named events. What the field of video indexing still lacks is a taxonomy of different techniques that indicates why a specific technique is suited the best, or unsuited, for a specific group of semantic index types.

The impact of the above mentioned factors on automatic indexing of video documents will not only make the process more efficient and more effective than it is now, it will also yield richer semantic indexes. This will form the basis for a range of new innovative applications.

Keyterms in this chapter

Shot boundary detection, edit detection, layout reconstruction, people detection, object detection, setting detection, conversion, synchronization, multimodal integration, iterated/non-iterated integration, symmetric/a-symmetric integration, statistics or knowledge based integration, semantic video index, logical story unit detection, overlapping links, visual distance, temporal distance, logical unit labelling, named event detection

Chapter 6

Semantic Video Indexing*

6.1 Introduction

Query-by-keyword is the paradigm on which machine-based text search is still based. Elaborating on the success of text-based search engines, query-by-keyword also gains momentum in multimedia retrieval. For multimedia archives it is hard to achieve access, however, when based on text alone. Multimodal indexing is essential for effective access to video archives. For the automatic detection of specific concepts, the state-of-the-art has produced sophisticated and specialized indexing methods, see our previous work [156] and the work of Naphade and Huang [110] for an overview. Other than their textual counterparts, generic methods for semantic indexing in multimedia are neither generally available, nor scalable in their computational needs, nor robust in their performance. As a consequence, semantic access to multimedia archives is still limited. Therefore, there is a case to be made for a new approach to semantic video indexing.

The main problem for any semantic video indexing approach is the semantic gap between data representation and their interpretation by humans, as identified by Smeulders *et al.* [151]. In efforts to reduce the semantic gap, many video indexing approaches focus on specific semantic concepts with a small intra-class and large inter-class variability of content. Typical concepts and their detectors are *sunsets* by Smith and Chang [152] and the work by Zhang *et al.* on *news anchors* [191]. These concepts have become icons for video indexing. Although they have aided in achieving progress, this approach is limited when considering the plethora of concepts waiting to be detected. It is simply impossible to bridge the semantic gap by designing a tailor-made solution for each concept.

In this paper, we propose a novel approach for generic semantic indexing of multimedia archives. It builds on the observation that produced video is the result of an authoring process. When producing a video, an author departs from a conceptual idea. The semantic intention is then articulated in (sub) consciously selected conventions and techniques for the purpose of emphasizing aspects of the content. The intention is communicated in context to the audience by a set of commonly shared notions. We aim to link the knowledge of years of media science research to semantic video analysis, see for example Boggs and Petrie [19] and Bordwell and Thompson [22]. We use the authoring-driven process of video production as the leading principle for generic video indexing.

Viewing semantic video indexing from an authoring perspective has the advantage that the most successful existing video indexing methods may be combined

*This chapter is adapted from [158].

in one architecture. We first consider the vast amount of work performed in developing detection methods for specialized concepts [7, 12, 57, 63, 152, 180, 191]. If we measure the success of these methods in terms of benchmark detection performance, Informedia [63, 180] stands out. They focus on combining techniques from computer vision, speech recognition, natural language understanding, and artificial intelligence into a video indexing and retrieval environment. This has resulted in a large set of isolated and specialized concept detectors [63]. We build our generic indexing approach in part on the outputs of their detectors, but we do not use them in isolation.

In comparison to specialized detection methods, generic semantic indexing is rare. We discuss three successful examples of generic semantic indexing approaches [10, 40, 159]. In the first one, Fan *et al.* [40] propose the *ClassView* framework. The framework combines hierarchical semantic indexing with hierarchical retrieval. At the lowest level, the framework supports indexing of shots into concepts based on a large set of low-level visual features. At the second level a Bayes classifier maps concepts to semantic clusters. By assigning shots to a hierarchy of concepts, the framework supports queries based on semantic and visual similarity. As the authors indicate, the framework will provide more meaningful results if it would support multimodal content analysis. We aim for generic semantic indexing also, but we include multimodal analysis from the beginning. In the second generic method [10], Amir *et al.* propose a system for semantic indexing using a detection pipeline. The pipeline starts with multimodal feature extraction. Based on these features the pipeline then generates several unimodal statistical models for a lexicon of semantic concepts. For integration of modalities and models at the concept level, Ensemble Fusion, amongst others, is applied. This fusion scheme includes normalization of confidence scores, several combiner functions, and parameter optimization, see also [172]. All multimodal concepts then serve as the input for a so called Multinet [111] that uses the combination of concepts for final semantic classification. The pipeline optimizes the result by rule-based post filtering. We interpret the success of the system by the fact that all modules in the pipeline select the best of multiple hypotheses, and the exhaustive use of machine learning. Moreover, the authors were among the first to recognize that semantic indexing profits substantially from context. We adopt and extend their ideas related to hypothesis selection, machine learning, and the use of context for semantic indexing. All of the above generic methods ignore the important influence of the video production style in the analysis process. In addition to content and context, we identify layout and capture in [159] as important factors for semantic indexing of produced video. We propose in [159] a generic framework for produced video indexing combining four sets of style detectors in an iterative semantic classifier. Results indicate that the method obtains high accuracy for rich semantic concepts, rich meaning that concepts share many similarities in their video production process. The framework is less suited for concepts that are not stylized. In the current paper, we generalize the idea of using style for semantic indexing.

We propose a generic approach for semantic indexing, we call the *semantic pathfinder*. It combines the most successful methods for semantic video indexing [10, 63, 159, 180] into an integrated architecture. The design principle is derived from the video production process, covering notions of content, style, and context. The architecture is built on several detectors, multimodal analysis, hypothesis selection, and machine learning. The semantic pathfinder combines analysis steps at increasing levels of abstraction, corresponding to well-known facts from the study of film and television production [19, 22]. Its virtue is its ability to learn the best path, from all explored analysis steps, on a per-concept basis. To demonstrate the effectiveness of the semantic pathfinder, the semantic indexing experiments are evaluated within the 2004 NIST TRECVID video retrieval benchmark [148, 149].

The organization of this paper is as follows. First, we introduce the TRECVID benchmark in Section 6.2. Our system architecture for generic semantic indexing is presented in Section 6.3. We present results in Section 6.4.

6.2 TRECVID Benchmark

Evaluation of multimedia systems has always been a delicate issue. Due to copyrights and the sheer volume of data involved, multimedia archives are fragmented and mostly inaccessible. Therefore, comparison of systems has traditionally been difficult, often impossible even. To accommodate these hardships NIST started organizing the TRECVID video retrieval benchmark. The benchmark aims to promote progress in video retrieval via open, metrics-based evaluation [148,149]. Tasks include camera shot segmentation, story segmentation, semantic concept detection[†], and several search tasks. Because of its widespread acceptance in the field, resulting in large participation of teams from both academic and corporate research labs worldwide, the benchmark can be regarded as the *de facto* standard to evaluate performance of multimedia indexing and retrieval research. We have participated in the semantic concept detection task of the 2004 NIST TRECVID video retrieval benchmark.

6.2.1 Multimedia Archive

The video archive of the 2004 TRECVID benchmark extends the data set used in 2003. The archive is composed of 184 hours of ABC World News Tonight and CNN Headline News and is recorded in MPEG-1 format. The training data consists of the archive used in 2003. It contains approximately 120 hours covering the period of January until June 1998. The 2004 test data contains the remaining 64 hours, covering the period of October until December 1998. Together with the video archive, CLIPS-IMAG [129] provided a camera shot segmentation. We evaluate semantic indexing within the TRECVID benchmark, to demonstrate the effectiveness of the semantic pathfinder for semantic access to multimedia archives.

6.2.2 Evaluation Criteria

Participation in TRECVID is based on the submission of results for one or more of the concepts in the semantic concept detection task. Where a submission, or run, contains a ranked list of at most 2000 camera shots per semantic concept, and for each concept, participants are allowed to submit up to 10 runs.

To determine the accuracy of submissions we use *average precision* and *precision at 100*, following the standard in TRECVID evaluations. The average precision is a single-valued measure that is proportional to the area under a recall-precision curve. This value is the average of the precision over all relevant judged shots. Hence, it combines precision and recall into one performance value. Let $L^k = \{l_1, l_2, \dots, l_k\}$ be a ranked version of the answer set A. At any given rank k let $R \cap L^k$ be the number of relevant shots in the top k of L, where R is the total number of relevant shots. Then average precision is defined as:

$$average\ precision = \frac{1}{R} \sum_{k=1}^{A} \frac{R \cap L^k}{k} \psi(l_k) \ , \tag{6.1}$$

[†] TRECVID refers to this task as the feature extraction task, to prevent misunderstanding with feature extraction as defined in the semantic pathfinder we refer to it as the semantic concept detection task.

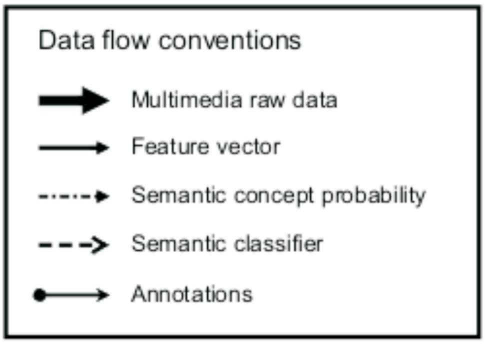


Figure 6.1: Data flow conventions as used in this paper. Different arrows indicate difference in data flows.

where indicator function $\psi(l_k) = 1$ if $l_k \in R$ and 0 otherwise. As the denominator k and the value of $\psi(l_k)$ are dominant in determining average precision, it can be understood that this metric favours highly ranked relevant shots.

TRECVID uses a pooled ground truth P, to reduce labor-intensive manual judgments of all submitted runs. They take from each submitted run a fixed number of ranked shots, which is combined into a list of unique shots. Every submission is then evaluated based on the results of assessing this merged subset, i.e. instead of using R in Eq. (6.1), P is used, where $P \subset R$. This is a fair comparison for submitted runs, since it assures that for each submitted run at least a fixed number of shots are evaluated at the more important top of the ranked list. However, using a pooled ground truth based on manual judgment comes with a price. In addition to mistakes by relevance assessors that may appear, using a pooling mechanism for evaluation means that the ground truth of the test data is incomplete.

Apart from average precision, we also report the precision at depth 100 in the result set. This value gives the fraction of correctly annotated shots within the first 100 retrieved results.

6.3 Semantic Pathfinder

Before we elaborate on the video indexing architecture, we first define a lexicon Λ_S of 32 semantic concepts. The lexicon is indicative for future efforts to detect as much as 1000 concepts [62]. At present, it serves as a non-trivial illustration of concept possibilities. In addition, the anticipated positive influence of the lexicon on the result of the 10 benchmark concepts is taken into account. The semantic concept lexicon consists of the following concepts:

- Λ_S = {*airplane take off, American football, animal, baseball, basket scored, beach, bicycle, Bill Clinton, boat, building, car, cartoon, financial news anchor, golf, graphics, ice hockey, Madeleine Albright, news anchor, news subject monologue, outdoor, overlayed text, people, people walking, physical violence, road, soccer, sporting event, stock quotes, studio setting, train, vegetation, weather news*};

The lexicon contains both general concepts, like *people*, *car*, and *beach*, as well as specific concepts such as *airplane take off* and *news subject monologue*. We aim to detect all 32 concepts with the proposed system architecture.

The semantic pathfinder is composed of three analysis steps. It follows the reverse authoring process. Each analysis step in the path detects semantic concepts.

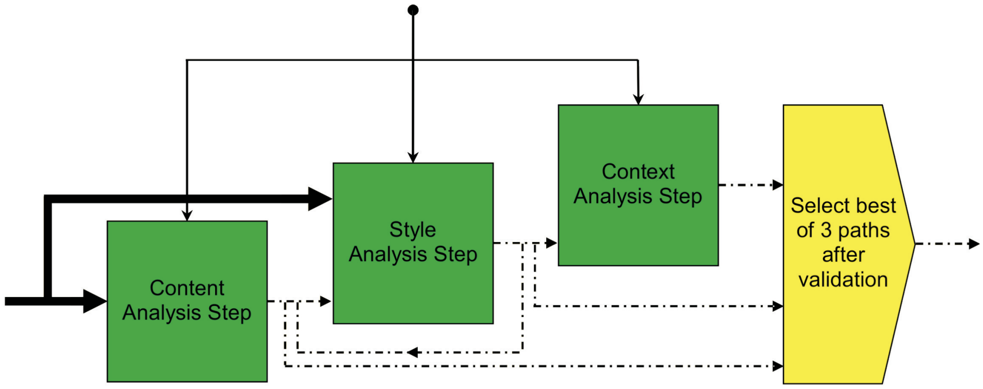


Figure 6.2: The semantic pathfinder for one concept, using the conventions of figure 6.1.

In addition, one can exploit the output of an analysis step in the path as the input for the next one. The semantic pathfinder starts in the *content analysis step*. In this analysis step, we follow a data-driven approach of indexing semantics. The *style analysis step* is the second analysis step. Here we tackle the indexing problem by viewing a video from the perspective of production. This analysis step aids especially in indexing of rich semantics. Finally, to enhance the indexes further, in the *context analysis step*, we view semantics in context. One would expect that some concepts, like *vegetation*, have their emphasis on content where the style (of the camera work that is) and context (of concepts like *graphics*) do not add much. In contrast, more complex events, like *people walking*, profit from incremental adaptation of the analysis to the intention of the author. The virtue of the semantic pathfinder is its ability to find the best path of analysis steps on a per-concept basis.

The analysis steps in the semantic pathfinder exploit a common architecture, with a standardized input-output model, to allow for semantic integration. The conventions to describe the system architecture are indicated in figure 6.1. An overview of the semantic pathfinder is given in figure 6.2.

6.3.1 Analysis Step General Architecture

We perceive semantic indexing in video as a pattern recognition problem. We first need to segment a video. We opt for camera shots, indicated by i, following the standard in TRECVID evaluations. Given pattern x, part of a shot, the aim is to detect a semantic concept ω from shot i using probability $p(\omega|x_i)$. Each analysis step in the semantic pathfinder extracts x_i from the data, and exploits a learning module to learn $p(\omega|x_i)$ for all ω in the semantic lexicon Λ_S. We exploit supervised learning to learn the relation between ω and x_i. The training data of the multimedia archive, together with labeled samples, are for learning classifiers. The other data, the test data, are set aside for testing. The general architecture for supervised learning in each analysis step is illustrated in figure 6.3.

Supervised learning requires labeled examples. In part, we rely on the ground truth provided in TRECVID 2003 [94]. We remove the many errors from this annotation effort. It is extended manually to arrive at an incomplete, but reliable ground truth[‡] for all concepts in lexicon Λ_S. We split the training data a priori into a non-overlapping training set and validation set to prevent overfitting of classifiers in the semantic pathfinder. It should be noted that a reliable validation set would

[‡] [Online]. Available: `http://www.science.uva.nl/~cgmsnoek/tv/`.

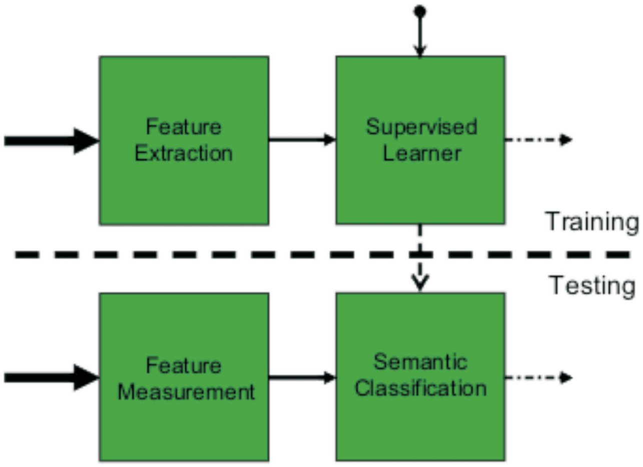


Figure 6.3: General architecture of an analysis step in the semantic pathfinder, using the conventions of figure 6.1.

ideally require an as large as possible percentage of positively labeled examples, which is comparable to the training set. In practice this may be hard to achieve, however, as some concepts are sparse. The training set we use contains 85% of the training data, the validation set contains the remaining 15%. We summarize the percentage of positively annotated examples for each concept in training and validation set in table 6.1.

We choose from a large variety of supervised machine learning approaches to obtain $p(\omega|x_i)$. For our purpose, the method of choice should be capable of handling video documents. To that end, ideally it must learn from a limited number of examples, it must handle unbalanced data, and it should account for unknown or erroneously detected data. In such heavy demands, the Support Vector Machine (SVM) framework [27, 176] has proven to be a solid choice [10, 155]. The usual SVM method provides a margin, $\gamma(x_i)$, in the result. We prefer Platt's conversion method [127] to achieve a posterior probability of the result. It is defined as:

$$p(\omega|x_i) = \frac{1}{1 + \exp(\alpha\gamma(x_i) + \beta)} \ , \tag{6.2}$$

where the parameters α and β are maximum likelihood estimates based on training data. SVM classifiers thus trained for ω, result in an estimate $p(\omega|x_i, \vec{q})$, where $\vec{q}$ are parameters of the SVM yet to be optimized.

The influence of the SVM parameters on concept detection is significant [108]. We obtain good parameter settings for a classifier, by using an iterative search on a large number of SVM parameter combinations. We measure average precision performance of all parameter combinations and select the combination that yields the best performance, $\vec{q}^*$. Here we use a 3-fold cross validation [76] to prevent overfitting of parameters. The result of the parameter search over $\vec{q}$ is the improved model $p(\omega|x_i, \vec{q}^*)$, contracted to $p^*(\omega|x_i)$.

This concludes the introduction of the general architecture of all analysis steps in the semantic pathfinder.

Table 6.1: Semantic concepts and the percentage of positively labeled examples used for the training set and the validation set.

Semantic Concept	**Training** (%)	**Validation** (%)	**Semantic Concept**	**Training (%)**	**Validation** (%)
Weather news	0.51	0.43	Golf	0.14	0.25
Stock quotes	0.26	0.30	People	3.89	3.99
News anchor	3.91	3.99	American football	0.05	0.10
Overlayed text	0.26	0.17	Outdoor	7.52	8.60
Basket scored	1.07	0.97	Car	1.57	2.10
Graphics	1.06	1.05	Bill Clinton	0.97	1.41
Baseball	0.74	0.66	News subject monologue	3.84	3.96
Sporting event	2.27	2.44	Animal	1.35	1.34
People walking	1.92	1.97	Road	1.44	1.98
Financial news anchor	0.35	0.35	Beach	0.42	0.61
Ice hockey	0.36	0.47	Train	0.21	0.36
Cartoon	0.60	0.73	Madeleine Albright	0.18	0.02
Studio setting	4.94	4.65	Building	4.95	4.81
Physical violence	2.73	3.14	Airplane take off	0.89	0.87
Vegetation	1.60	1.59	Bicycle	0.28	0.27
Boat	0.55	0.45	Soccer	0.06	0.09

6.3.2 Content Analysis Step

We view video in the content analysis step from the data perspective. In general, three data streams or modalities exist in video, namely the auditory modality, the textual modality, and the visual one. As speech is often the most informative part of the auditory source, we focus on visual features, and on textual features obtained from transcribed speech. After modality specific data processing, we combine features in a multimodal representation. The data flow in the content analysis step is illustrated in figure 6.4.

Visual Analysis

In the visual modality, we aim for segmentation of an image frame f into regional visual concepts. Ideally, a segmentation method should result in a precise partitioning of f according to the object boundaries, referred to as strong segmentation. However, weak segmentation, where f is partitioned into internally homogenous regions within the boundaries of the object, is often the best one can hope for [151]. We obtain a weak segmentation based on a set of visual feature detectors. Prior to segmentation we remove the border of each frame, including the space occupied by a possible ticker tape. The basis of feature extraction in the visual modality is weak segmentation.

Invariance was identified in [151] as a crucial aspect of a visual feature detector, e.g. to design features which limit the influence of accidental recording circumstances. We use color invariant visual features [49] to arrive at weak segmentation. The invariance covers the photometric variation due to shadow and shading, and geometrical variation due to scale and orientation. This invariance is needed as the conditions under which semantic concepts appear in large multimedia archives may vary greatly.

The feature extraction procedure we adhere to, computes per pixel a number of invariant features in vector $\vec{u}$. This vector then serves as the input for a multi-class SVM [27] that associates each pixel to one of the regional visual concepts defined in a visual concept lexicon Λ_V, using a labeled training set. Based on Λ_S, we define the following set of regional visual concepts:

- $\Lambda_V = \{$*colored clothing, concrete, fire, graphic blue, graphic purple, graphic*

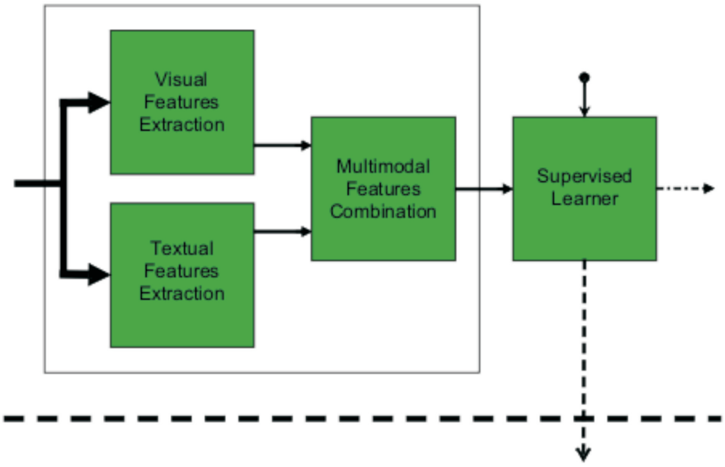

Visual Features Extraction
Textual Features Extraction
Multimodal Features Combination
Supervised Learner

Figure 6.4: Feature extraction and classification in the content analysis step, special case of figure 6.3.

yellow, grassland, greenery, indoor sport court, red carpet, sand, skin, sky, smoke, snow/ice, tuxedo, water body, wood};

As we use invariant features, only a few examples per visual concept class are needed; in practice less then 10 per class. This pixel-wise classification results in the image vector $\vec{w}_f$, where $\vec{w}_f$ contains one component per regional visual concept, indicating the percentage of pixels found for this class. Thus, $\vec{w}_f$ is a weak segmentation of frame f in terms of regional visual concepts from Λ_V, see figure 6.5 for an example segmentation.

We use Gaussian color measurements to obtain $\vec{u}$ for weak segmentation [49]. We decorrelate RGB color values by linear transformation to the opponent color system [49]:

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad . \tag{6.3}$$

Smoothing these values with a Gaussian filter, $G(\sigma)$, suppresses acquisition and compression noise. Moreover, we extract texture features by applying Gaussian derivative filters. We vary the size of the Gaussian filters, $\sigma = \{1, 2, 3.5\}$, to obtain a color representation that is compatible with variations in the target object size (leaving out pixel position parameters):

$$\hat{E}_j(\sigma) = G_j(\sigma) * E, \quad \hat{E}_{\lambda j}(\sigma) = G_j(\sigma) * E_\lambda, \quad \hat{E}_{\lambda\lambda j}(\sigma) = G_j(\sigma) * E_{\lambda\lambda} \ , \tag{6.4}$$

where $j \in \{\varnothing, x, y\}$ indicates either spatial smoothing or spatial differentiation and that from now on the hat symbol $(\hat{\cdot})$ implies a dependence on σ. Normalizing each opponent color value by its intensity suppresses global intensity variations. This results in two chromaticity values per color pixel:

$$\hat{C}_\lambda = \frac{\hat{E}_\lambda}{\hat{E}}, \quad \hat{C}_{\lambda\lambda} = \frac{\hat{E}_{\lambda\lambda}}{\hat{E}} \quad . \tag{6.5}$$

Furthermore, we obtain rotationally invariant features by taking Gaussian derivative filters and combining the responses into two chromatic gradients:

$$\hat{C}_{\lambda w} = \sqrt{\hat{C}^2_{\lambda x} + \hat{C}^2_{\lambda y}}, \quad \hat{C}_{\lambda\lambda w} = \sqrt{\hat{C}^2_{\lambda\lambda x} + \hat{C}^2_{\lambda\lambda y}} \ , \tag{6.6}$$

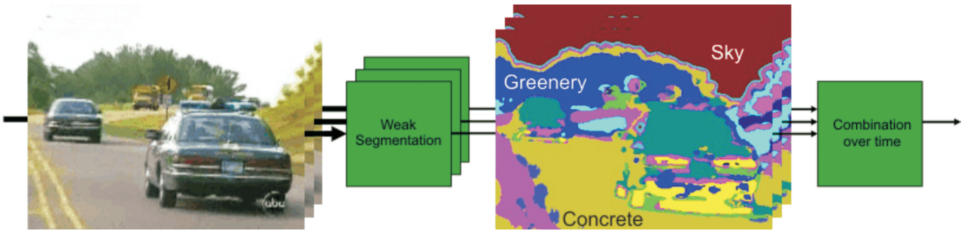


Figure 6.5: Computation of the visual features, see figure 6.4, is based on weak segmentation of an image frame into regional visual concepts. A combination over time is used to select one frame as representative for the shot.

where $\hat{C}_{\lambda x}$, $\hat{C}_{\lambda y}$, $\hat{C}_{\lambda\lambda x}$, and $\hat{C}_{\lambda\lambda y}$ are defined as:

$$\begin{aligned} \hat{C}_{\lambda x} &= \frac{\hat{E}_{\lambda x}\hat{E} - \hat{E}_{\lambda}\hat{E}_{x}}{\hat{E}^2}, \quad \hat{C}_{\lambda\lambda x} = \frac{\hat{E}_{\lambda\lambda x}\hat{E} - \hat{E}_{\lambda\lambda}\hat{E}_{x}}{\hat{E}^2}, \\ \hat{C}_{\lambda y} &= \frac{\hat{E}_{\lambda y}\hat{E} - \hat{E}_{\lambda}\hat{E}_{y}}{\hat{E}^2}, \quad \hat{C}_{\lambda\lambda y} = \frac{\hat{E}_{\lambda\lambda y}\hat{E} - \hat{E}_{\lambda\lambda}\hat{E}_{y}}{\hat{E}^2} \quad . \end{aligned} \tag{6.7}$$

The seven measurements computed in Eq. (6.4–6.6), and each calculated over three scales, yield a 21 dimensional invariant feature vector $\vec{u}$ per pixel.

Segmenting image frames into regional visual concepts at the granularity of a pixel is computationally intensive. We estimate that the processing of the entire TRECVID data set would have taken around 250 days on the fastest sequential machine available to us. As a first reduction of the analysis load, we analyze 1 out of 15 frames only. For the remaining image processing effort we apply the Parallel-Horus software architecture [144]. This architecture, consisting of a large collection of low-level image processing primitives, allows the programmer to write sequential applications with efficient parallel execution on commonly available commodity clusters. Application of Parallel-Horus, in combination with a distributed cluster consisting of 200 dual 1-Ghz Pentium-III CPUs [13], reduced the processing time to less than 60 hours [144].

The features over time are combined into one vector for the shot i. Averaging over individual frames is not a good choice, as the visual representation should remain intact. Instead, we opt for a selection of the most representative frame or visual vector. To decide which f is the most representative for i, weak segmented image $\vec{w}_f$ is the input for an SVM that computes a probability $p^*(\omega|\vec{w}_f)$. We select $\vec{w}_f$ that maximizes the probability for a concept from Λ_S within i, given as:

$$\vec{v}_i = \arg\max_{f \in f_i} p^*(\omega|\vec{w}_f) \quad . \tag{6.8}$$

The visual vector $\vec{v}_i$, containing the best weak segmentation, is the final result of the visual analysis.

Textual Analysis

In the textual modality, we aim to learn the association between uttered speech and semantic concepts. A detection system transcribes the speech into text. From the text we remove the frequently occurring stopwords. After stopword removal, we are ready to learn semantics.

To learn the relation between uttered speech and concepts, we connect words to shots. We make this connection within the temporal boundaries of a shot. We derive a lexicon of uttered words that co-occur with ω using the shot-based annotations of the training data. For each concept ω, we learn a separate lexicon, Λ_T^ω, as this uttered word lexicon is specific for that concept. We modify the procedure for Person X concepts, i.e. *Madeleine Albright* and *Bill Clinton*, to optimize results. In broadcast news, a news anchor or reporter mentions names or other indicative words just before or after a person is visible. To account for this observation, we stretch the shot boundaries with five seconds on each side for Person X concepts. For these concepts, this procedure assures that the textual feature analysis considers even more textual content. For feature extraction we compare the text associated with each shot with Λ_T^ω. This comparison yields a text vector $\vec{t_i}$ for shot i, which contains the histogram of the words in association with ω.

Multimodal Analysis and Classification

The result of the content analysis step is a multimodal vector $\vec{m}_i$ that integrates all unimodal results. We concatenate the visual vector $\vec{v}_i$ with the text vector $\vec{t_i}$, to obtain $\vec{m}_i$. After this modality fusion, $\vec{m}_i$ serves as the input for the supervised learning module. To optimize parameter settings, we use 3-fold cross validation on the training set. The content analysis step associates probability $p^*(\omega|\vec{m}_i)$ with a shot i, for all ω in Λ_S.

6.3.3 Style Analysis Step

In the style analysis step we conceive of a video from the production perspective. Based on the four roles involved in the video production process [153, 159], this step analyzes a video by four related style detectors. Layout detectors analyze the role of the editor. Content detectors analyze the role of production design. Capture detectors analyze the role of the production recording unit. Finally, context detectors analyze the role of the preproduction team, see figure 6.6. Note that in contrast to the content analysis step, where we learn specific content features from a data set, content features in the style analysis step are generic and independent of the data set.

Style Analysis

We develop detectors for all four production roles as feature extraction in the style analysis step. We refer to our previous work for specific implementation details of the detectors [153, 159, Appendix A]. We have chosen to convert the output of all style detectors to an ordinal scale, as this allows for easy fusion.

For the layout $\mathcal{L}$ the length of a camera shot is used as a feature, as this is known to be an informative descriptor for genre [156]. Overlayed text is another informative descriptor. Its presence is detected by a text localization algorithm [138]. To segment the auditory layout, periods of speech and silence are detected based on an automatic speech recognition system [47]. We obtain a voice-over detector by combining the speech segmentation with the camera shot segmentation [159]. The set of layout features is thus given by: $\mathcal{L} = \{$*shot length, overlayed text, silence, voice-over*$\}$.

As concerns the content $\mathcal{C}$, a frontal face detector [142] is applied to detect people. We count the number of faces, and for each face its location is derived [159]. Apart from faces, we also detect the presence of cars [142]. In addition, we measure the average amount of object motion in a camera shot [155]. Based on speaker identification [47] we identify each of the three most frequent speakers. The camera shot is checked for the presence on the basis of speech from one of the three [159]. The length of text strings recognized by Video Optical Character Recognition [138] is used as a feature [159]. In addition, the strings are used as input for a named

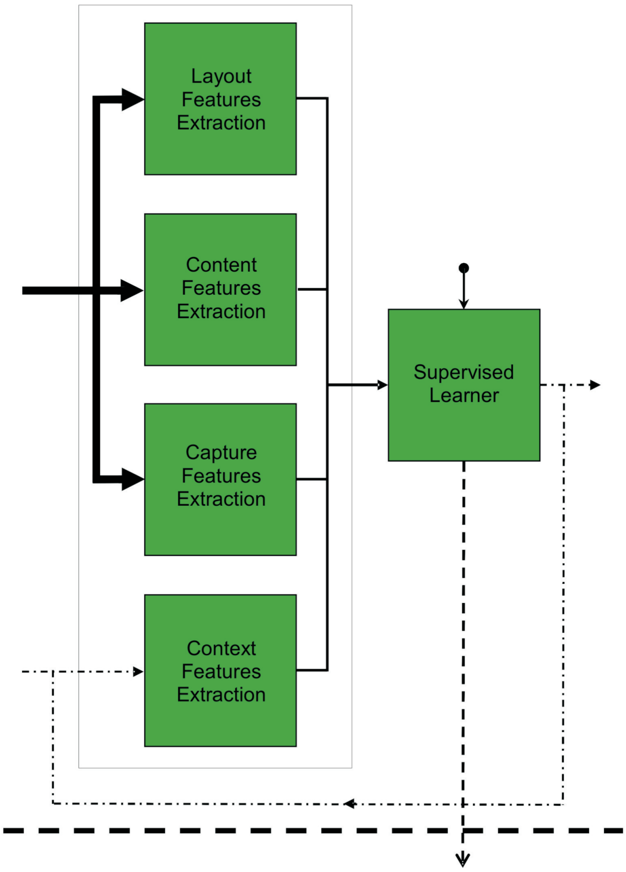

Layout Features Extraction
Content Features Extraction
Capture Features Extraction
Context Features Extraction
Supervised Learner

Figure 6.6: Feature extraction and classification in the style analysis step, special case of figure 6.3.

entity recognizer [180]. On the transcribed text obtained by the LIMSI automatic speech recognition system [47], we also apply named entity recognition. The set of content features is thus given by: $\mathcal{C}$ ={*faces, face location, cars, object motion, frequent speaker, overlayed text length, video text named entity, voice named entity*}.

For capture $\mathcal{T}$, we compute the camera distance from the size of detected faces [142, 159]. It is undefined when no face is detected. In addition to camera distance, several types of camera work are detected [11], e.g. pan, tilt, zoom, and so on. Finally, for capture we also estimate the amount of camera motion [11]. The set of capture features is thus given by: $\mathcal{T}$ = {*camera distance, camera work, camera motion*}.

The context $\mathcal{S}$ serves to enhance or reduce the correlation between semantic concepts. Detection of *vegetation* can aid in the detection of a *forest* for example. Likewise, the co-occurrence of a *space shuttle* and a *bicycle* in one shot is improbable. As the performance of semantic concept detectors is unknown and likely to vary between concepts, we exploit iteration to add them to the context. The rationale here is to add concepts that are relatively easy to detect first. They aid in detection performance by increasing the number of true positives or reducing the number of false positives. As initial concept we detect news reporters. We recognize news reporters by edit distance matching of strings, obtained from the transcript and video text, with a database of names of CNN and ABC affiliates [159]. The other

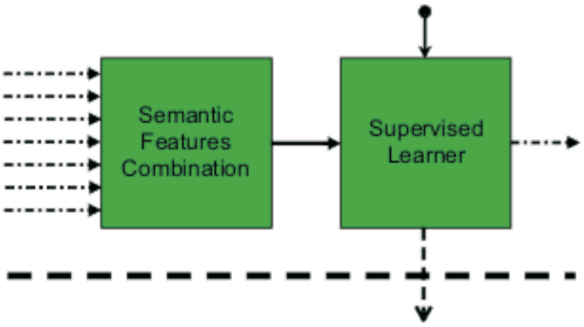


Figure 6.7: Feature extraction and classification in the context analysis step, special case of figure 6.3.

concepts that are added to the context stem from Λ_S. To prevent bias from domain knowledge, we use the performance on the validation set of all concepts from Λ_S in the content analysis step as the ordering for the context. For this ordering we again refer to table 6.1. To assign detection results for the first and least difficult concept, $\omega_1 =$ *weather news*, we rank all shot results on $p_i^*(\omega_1|\vec{m}_i)$. This ranking is then exploited to categorize results for ω_1 into one of five levels. The basic set of context features is thus given by: $\mathcal{S} = \{$*news reporter, content analysis step* $\omega_1\}$.

The concatenation of $\{\mathcal{L}, \mathcal{C}, \mathcal{T}, \mathcal{S}\}$ for shot i yields the style vector $\vec{s}_i$. This vector forms the input for an iterative classifier that trains a style model for each concept in lexicon Λ_S.

Iterative Style Classification

We start from an ordering of concepts in the context, as defined above. The iteration of the classifier begins with concept ω_1. After concatenation with the other style features this yields $\vec{s}_{i,1}$ the first style vector of the first iteration. $\vec{s}_{i,1}$ contains the combined results of the content analysis step and the style analysis step. We classify ω_1 again based on $\vec{s}_{i,1}$. This yields the a posterior probability $p^*(\omega_1|\vec{s}_{i,1})$. When $p^*(\omega|\vec{s}_i) \geq \delta$ the concept ω_1 is considered present in the style representation, else it is considered absent. The threshold δ is set a priori at a fixed value of 0.5. In this process the classifier replaces the feature for concept ω_1, from the content analysis step, by the new feature ω_1^+. The style analysis step adds more aspects of the author influence to the results obtained with the content analysis step. In the next iteration of the classification procedure, the classifier adds $\omega_2 =$ *stock quotes* from the content analysis step to the context. This yields $\vec{s}_{i,2}$. As explained above, the classifier replaces the ω_2 feature from the content analysis step by the styled version ω_2^+ based on $p^*(\omega_2|\vec{s}_{i,2})$. This iterative process is repeated for all ω in lexicon Λ_S.

We classify all ω in Λ_S again in the style analysis step. As the result of the content analysis step is only one of the many features in our style vector representation in the style analysis step, we also use 3-fold cross validation on the training set to optimize parameter settings in this analysis step. We use the resulting probability as output for concept detection in the style analysis step. In addition, it forms the input for the next analysis step in our semantic pathfinder.

6.3.4 Context Analysis Step

The context analysis step adds context to our interpretation of the video. Our ultimate aim is the reconstruction of the author's intent by considering detected concepts in context.

Table 6.2: Test set precision at 100 after the three steps, for a lexicon of 32 concepts. The best result is given in bold. The corresponding path is selected in the semantic pathfinder.

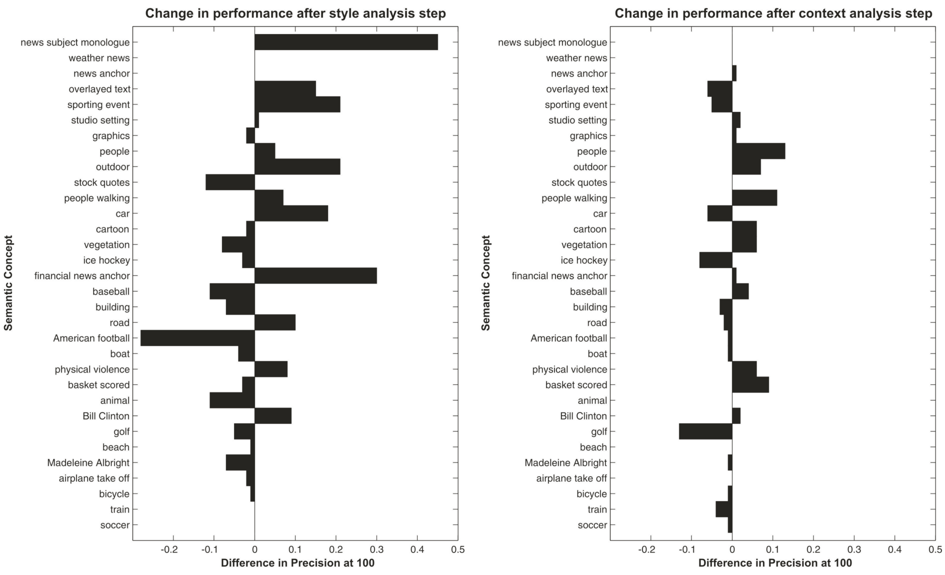
Semantic Concept	Content Analysis Step	Style Analysis Step	Context Analysis Step	Semantic Pathfinder
News subject monologue	0.55	**1.00**	1.00	1.00
Weather news	**1.00**	1.00	1.00	1.00
News anchor	0.98	0.98	**0.99**	0.99
Overlayed text	0.84	**0.99**	0.93	0.99
Sporting event	0.77	**0.98**	0.93	0.98
Studio setting	0.95	0.96	**0.98**	0.98
Graphics	0.92	0.90	**0.91**	0.91
People	0.73	0.78	**0.91**	0.91
Outdoor	0.62	0.83	**0.90**	0.90
Stock quotes	**0.89**	0.77	0.77	0.89
People walking	0.65	0.72	**0.83**	0.83
Car	0.63	0.81	**0.75**	0.75
Cartoon	0.71	0.69	**0.75**	0.75
Vegetation	**0.72**	0.64	0.70	0.72
Ice hockey	**0.71**	0.68	0.60	0.71
Financial news anchor	0.40	**0.70**	0.71	0.70
Baseball	**0.54**	0.43	0.47	0.54
Building	**0.53**	0.46	0.43	0.53
Road	0.43	0.53	**0.51**	0.51
American football	**0.46**	0.18	0.17	0.46
Boat	0.42	0.38	**0.37**	0.37
Physical violence	0.17	0.25	**0.31**	0.31
Basket scored	0.24	0.21	**0.30**	0.30
Animal	0.37	0.26	**0.26**	0.26
Bill Clinton	**0.26**	0.35	0.37	0.26
Golf	**0.24**	0.19	0.06	0.24
Beach	0.13	0.12	**0.12**	0.12
Madeleine Albright	**0.12**	0.05	0.04	0.12
Airplane take off	0.10	0.08	**0.08**	0.08
Bicycle	0.09	**0.08**	0.07	0.08
Train	**0.07**	0.07	0.03	0.07
Soccer	**0.01**	0.01	0.00	0.01
Mean	*0.51*	*0.53*	*0.54*	*0.57*

Semantic Analysis

The style analysis step yields a probability for each shot i and all concepts ω in Λ_S. The probability indicates whether a concept is present. We use the 32 concept scores as semantic features. We fuse them into context vector $\vec{c}_i$, see figure 6.7.

From $\vec{c}_i$ we learn relations between concepts automatically. To that end, $\vec{c}_i$ serves as the input for a supervised learning module, which associates a contextual probability $p^*(\omega|\vec{c}_i)$ to a shot i for all ω in Λ_S. To optimize parameter settings, we use 3-fold cross validation on the previously unused data from the validation set.

The output of the context analysis step is also the output of the entire semantic pathfinder on video documents. On the way we have included in the semantic pathfinder, the results of the analysis on raw data, facts derived from production by the use of style features, and a context perspective of the author's intent by using semantic features. For each concept we obtain a probability based on content, style, and context. We select from the three possibilities the one that maximizes average precision based on validation set performance. The semantic pathfinder provides us with the opportunity to decide whether a one-shot analysis step is best for the concept only concentrating on content, or a two-analysis step classifier increasing discriminatory power by adding production style to content, or that a concept profits most from a consecutive analysis path using content, style, and context.



(a) (b)

Figure 6.8: Influence of the style analysis step (a) and the context analysis step (b) on precision at 100 performance for a lexicon of 32 semantic concepts. Note a considerable decrease (American football) or increase (news subject monologue) in performance when adding production style information. The same phenomenon is repeated for context information in golf (decrease) and people (increase).

6.4 Results

6.4.1 Detection of 32 Semantic Concepts

We evaluated detection results for all 32 concepts in each analysis step. Given the already enormous size of the data sets and the large amounts of annotation – yet limited in terms of completeness – we have performed one pass for 32 concepts through the entire semantic pathfinder. We report the *precision at 100*, which indicates the number of correct shots within the first 100 results – assuming there are more than 100 relevant shots per concept – in table 6.2.

We observe from the results that the learned best path (printed in bold) indeed varies over the concepts. The virtue of the semantic pathfinder is demonstrated by the fact that for 12 concepts, the learning phase indicates it is best to concentrate on content only. For 5 concepts, the semantic pathfinder demonstrates that a two-step path is best (where in 15 cases addition of style features has a marginal positive or negative effect). For 15 concepts, the context analysis step obtains a better result. Context aids substantially in the performance for 5 concepts. As an aside we note that the precision at 100, when averaged over all concepts, steadily increases from 0.51 to 0.57 while traversing the different semantic analysis paths.

The results demonstrate the virtue of the semantic pathfinder. Concepts are divided by the analysis step after which they achieve best performance. Some concepts are just content, style does not affect them. In such cases as *American football* there is style-wise too much confusion with other sports to add new value in the path. Shots containing *stock quotes* suffer from a similar problem. Here false positives contain many stylistically similar results like graphical representations of survey and election results. For complex concepts, analysis based on content and style is not enough. They require the use of context. The context analysis step is especially good in detecting named events, like *people walking*, *physical violence*, and *basket scored*. The results offer us the possibility to categorize concepts according to the analysis step of the semantic pathfinder that yields the best performance.

The content analysis step seems to work particularly well for semantic concepts that have a small intra-class variability of content: *weather news* and *news anchor* for example. In addition, this analysis step aids in detection of accidental content like *building*, *vegetation*, *bicycle*, and *train*. However, for some of those concepts, e.g. *bicycle* and *train*, the performance is still disappointing. Another observation is that when one aims to distinguish sub-genres, e.g. *ice hockey*, *baseball*, and *American football*, the content analysis step is the best choice.

After the style analysis step, we obtain an increase in performance for 12 concepts, see figure 6.8a. Especially when the concepts are semantically rich: e.g. *news subject monologue*, *financial news anchor*, and *sporting event*, the style helps. As expected, index results in the style analysis step improve on the content analysis step when style is a distinguishing property of the concept and degrade the result when similarity in style exists between different concepts.

Results after the context analysis step in figure 6.8b show that performance increases for 13 concepts. The largest positive performance difference between the context analysis step and the style analysis step occurs for concept *people*. Concept *people* profits from sport-related concepts like *baseball*, *basket scored*, *American football*, *ice hockey*, and *sporting event*. In contrast, *golf* suffers from detection of *outdoor* and *vegetation*. When we detect *golf*, these concepts are also present frequently. The inverse, however, is not necessarily the case, i.e. when we detect *outdoor* it is not necessarily on a golf court. Based on these observations we conclude that, apart from named events, detection results of the context analysis step are similar to those of the style analysis step. Index results improve based on presence of semantically related concepts, but the context analysis step is unable to capture the semantic structure between concepts and for some concepts, this is leading to a drop in performance.

The above results show that the semantic pathfinder facilitates generic video indexing. In addition, the semantic pathfinder provides the foundation of a technique taxonomy for solving semantic concept detection tasks. The fact that sub-genres like *ice hockey*, *golf*, and *American football* behave similarly indicate the predictive value of the pathfinder for other sub-genres. The same holds for semantically rich concepts like *news subject monologue*, *financial news anchor*, and *sporting event*. We showed that for named events, such as *basket scored*, *physical violence*, and *people walking*, one should apply a detector that is based on the entire semantic pathfinder. The significance of the semantic pathfinder is its generalizing power combined with the fact that addition of new information in the analysis can be considered by concept type.

6.4.2 Benchmark Comparison

We performed an experiment within the TRECVID benchmark to show the effectiveness of the semantic pathfinder for detection of semantic concepts among 12 present-day video indexing systems. The TRECVID 2004 procedure prescribes that 10 pre-defined concepts are evaluated. Hence, we report the official benchmark results for 10 concepts in our lexicon only. The 10 benchmark concepts are, however, representative for the entire lexicon of 32. All evaluations are based on the semantic pathfinder.

We compare our work with the 11 other participants in TRECVID 2004. We select from each participant the system tuning with the best performance for a

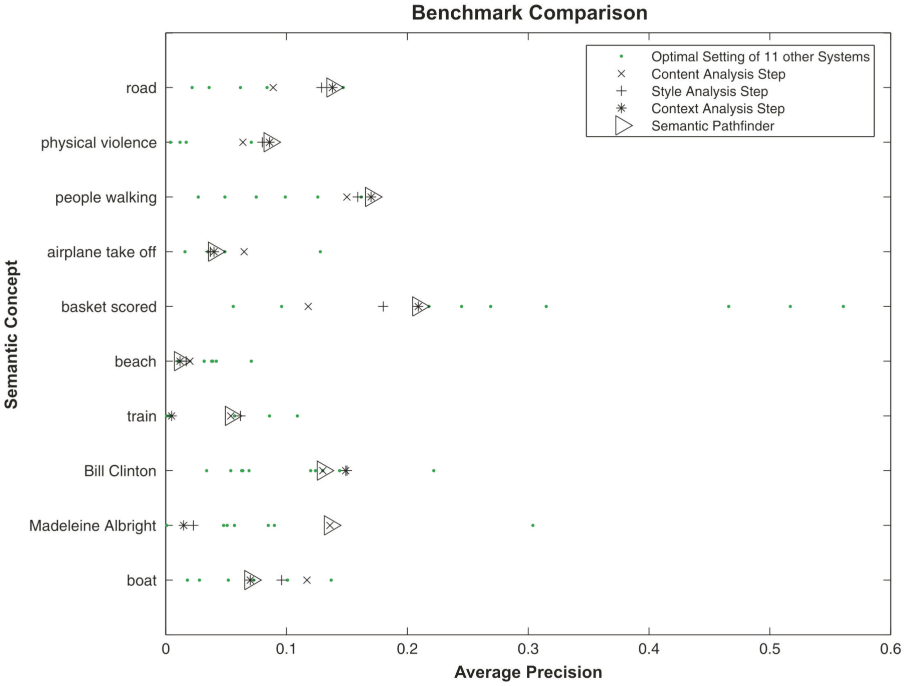


Figure 6.9: Comparison of semantic pathfinder results with 11 other present-day indexing systems in the TRECVID 2004 benchmark [148, 149].

concept out of a maximum of 10 tunings. For ease of explanation we do not take the optimal tunings of the semantic pathfinder, as reported in [157], into account. Instead, we use a similar parameter setting for all concepts. Hence, we favor other systems in this comparison. Results are visualized in figure 6.9 for each concept.

Relative to other video indexing systems the semantic pathfinder performs the best for two concepts, i.e. *people walking* and *physical violence*, and second for five concepts, i.e. *boat*, *Madeleine Albright*, *Bill Clinton*, *airplane take off*, and *road*. For two concepts we perform moderate, i.e. *basket scored* and *beach*. Here the best approaches are based on specialized concept detection methods that exploit domain knowledge. The big disadvantage of these methods is that they are specifically designed and implemented for one concept. They do not scale to other concepts. The benchmark results show that the semantic pathfinder allows for generic indexing with state-of-the-art performance.

6.4.3 Usage Scenarios

The results from the semantic pathfinder facilitate the development of various applications. The lexicon of 32 semantic concepts allows for querying a video archive by concept. In [161], we combined into a semantic video search engine query-by-concept, query-by-keyword, query-by-example, and interactive filtering. In addition to interactive search, the set of indexes is also applicable in a personalized retrieval setting. A feasible scenario is that users with a specific interest in sports are provided with personalized summaries when and where they need it. The sketched applications provide a semantic access to multimedia archives.

6.5 Conclusion

We propose the semantic pathfinder for semantic access to multimedia archives. The semantic pathfinder is a generic approach for video indexing. It is based on the observation that produced video is the result of an authoring process. The semantic pathfinder exploits the authoring metaphor in an effort to bridge the semantic gap. The architecture is built on a variety of detector types, multimodal analysis, hypothesis selection, and machine learning. The semantic pathfinder selects the best path through content analysis, style analysis, and context analysis. After machine learning it appears that the analysis is completed after content analysis only when concepts share many similarities in their multimodal content. It appears also that the semantic path runs up to style analysis when the professional habits of television are evident to the concept. Finally, it exploits a path based on content, style, and context for concepts that are primarily intentional, see table 6.2 and figure 6.8.

Experiments with a lexicon of 32 semantic concepts demonstrate that the semantic pathfinder allows for generic video indexing, while confirming the value of the authoring metaphor in indexing. In addition, the results over the various analysis steps indicate that a technique taxonomy exists for solving semantic concept detection tasks; depending on whether content, style, or context is most suited for indexing. Finally, the semantic pathfinder is successfully evaluated within the 2004 TRECVID benchmark. With one and the same set of system parameters two concepts, i.e. *people walking* and *physical violence*, came out best against 11 other present-day systems with average precision scores, remember that this measure indicates the average of the precision after every relevant item is retrieved, of 0.170 and 0.086 respectively. For five concepts our system scored second best, i.e. *boat* (0.117), *Madeleine Albright* (0.136), *Bill Clinton* (0.150), *airplane take off* (0.065), and *road* (0.138). Just two performed poorly in this comparison, i.e. *basket scored* (0.209) and *beach* (0.020). The results show that the semantic pathfinder allows for state-of-the-art performance without the need of implementing specialized detectors. We consider this the best indication of the validity of the approach.

A semantic pathfinder is as strong as its weakest analysis step. Introduction of feature selection and knowledge representations in the various analysis steps will improve results. In its current form the context analysis step takes the results of the style analysis step for granted; and results are only adapted when there is enough contextual evidence from the other concepts to do so. Improvement of the semantic pathfinder along these lines is topic of future research.

For the moment, the average precision resulting from completely automatic indexing ranges from 0.020 to 0.209. In absolute terms, these performance values are still quite low. In 64 hours of produced video only a small fraction of the relevant instances in the footage are retrieved within the first few ranked results. For selecting illustrative footage, this may already be sufficient. This is not yet so for tasks that require accurate retrieval. However, the trend in results over the past years indicates that automated search in video archives lures at the horizon.

Keyterms in this chapter

Authoring metaphor, generic video indexing, semantic pathfinder, content analysis, style analysis, context analysis, TRECVID benchmark, average precision, precision at 100

Chapter 7

Semantic Video Retrieval*

7.1 Introduction

The technology for searching through text has evolved to a mature level of performance. Browsers and search engines have found in the Internet a medium to prosper, opening new ways to do business, science, and to be social. All of this was realized in just 15 years. That success has whet the appetite for retrieval of multimedia sources, specifically of the medium video. Present-day commercial video search engines [18, 54] often rely on just a filename and text metadata in the form of closed captions [54] or transcribed speech [18]. This results in a disappointing performance, as quite often the visual content is not mentioned, or properly reflected in the associated text. The text often covers the emotion of the video, but this is highly specific for context and wears quickly. In addition, when videos originate from non-English speaking countries, such as China or the Netherlands, querying the content becomes more difficult because automatic speech recognition is so much harder to achieve. At any rate, visual analysis up to the standards of text will deliver robustness to the multimedia search.

In contrast to text-based video retrieval, the content-based image retrieval research community has emphasized a visual-only approach. It has resulted in a wide variety of image and video search systems [25, 28, 36, 45, 50, 56, 81, 97, 122, 134, 152]. A common denominator in these prototypes is their dependence on low-level visual information such as color, texture, shape, and spatiotemporal features. Users query an archive containing visual feature values rather than the images. They do so by sketches, or by providing example images using a browser interface. Query-by-example can be fruitful when users search for the same object under slightly varying circumstances and when the target images are available indeed. If proper example images are unavailable, content-based image retrieval techniques are not effective at all. Moreover, users often do not understand similarity of low-level visual features. They expect semantic similarity. In other words, when searching for cars, an input image of a red car should also trigger the retrieval of yellow colored cars. The current generation of video search engines offers low-level abstractions of the data, where users seek high-level semantics. Thus, query-by-example retrieval techniques are not that effective in fulfilling the users' needs. The main problem for any video retrieval methodology aiming for access is the semantic gap between image data representation and their interpretation by humans [151]. Not surprisingly, the user experience with (visual only) video retrieval is one of frustration. Therefore, a new paradigm of semantics is required when aiming for access to video archives.

*This chapter is adapted from [160].

In a quest to narrow the semantic gap, recent research efforts have concentrated on automatic detection of semantic concepts in video. The feasibility of mapping low-level (visual) features to high-level concepts was proven by pioneering work, which distinguished between concepts such as *indoor* and *outdoor* [167], and *cityscape* and *landscape* [173]. The introduction of multimedia analysis, coupled with machine learning, has paved the way for generic indexing approaches [3, 10, 40, 41, 109, 154, 157–159]. Currently yielding concept lexicons bounded by 101 concepts [154], and expected to evolve into multimedia ontologies [15] containing as much as 1,000 concepts soon [62]. The speed at which these lexicons grow offers great potential for future video retrieval systems. At present the lexicons are not large enough, so they are no alternative yet for either the visual or textual retrieval paradigm. However, the availability of gradually increasing concept lexicons, raises the question: how to augment query-by-concept for effective interactive video retrieval?

We start from the premise that a video search engine should begin with off-line learning of a large lexicon of multimedia concepts. In order to be effective in its use, a video search engine should employ query-by-example, query-by-keyword, and interaction with an advanced user interface to refine the search until satisfaction. We propose a *lexicon-driven paradigm* to video retrieval. The uniqueness of the proposed paradigm lies in its emphasis on automatic learning of a large lexicon of concepts. When the lexicon is exploited for query-by-concept and combined with query-by-keyword, query-by-example, and interactive filtering using an advanced user interface, a powerful video search engine emerges, which we call the *MediaMill* semantic video search engine. To demonstrate the effectiveness of our lexicon-driven retrieval paradigm, the interactive search experiments with the MediaMill system are evaluated within the 2004 and 2005 NIST TRECVID video retrieval benchmark [147, 149].

The organization of this paper is as follows. First, we formulate the problem in terms of related work in Section 7.2. The blueprint of our lexicon-driven video retrieval paradigm is presented in Section 7.3, where we describe the MediaMill system. We present the experimental setup in which we evaluated our paradigm in Section 7.4. We show the results of our experiments in Section 7.5.

7.2 Problem Formulation and Related Work

We aim at providing users with semantic access to video archives. Specifically, we investigate whether this can be reached by machine learning. Then the question is how this large lexicon of learned concepts can be combined with query-by-keyword, query-by-example, and interactive manipulation to achieve effective video retrieval?

In response to this question, we focus on methodologies that advocate the combination of lexicon learning, query-by-example, query-by-keyword, and interaction for semantic access [4, 10, 30, 40, 168]. We observe that these state-of-the-art video search systems are structured in a similar fashion. First, they include an engine that indexes video data on a visual, textual, and semantic level. Systems typically apply similarity functions to index the data in the visual and textual modality. This similarity index facilitates retrieval in the form of query-by-example and query-by-keyword. Video search engines often employ a semantic indexing component to learn a lexicon of concepts and accompanying probability from provided examples. All indexes are typically stored in a database at the granularity of a video shot. A second component that all systems have in common is a retrieval engine, which offers users an access to the stored indexes and the video data. The system has an interface to compose queries, e.g. using query-by-keyword, query-by-example, and query-by-concept. The retrieval engine handles the query requests, combines

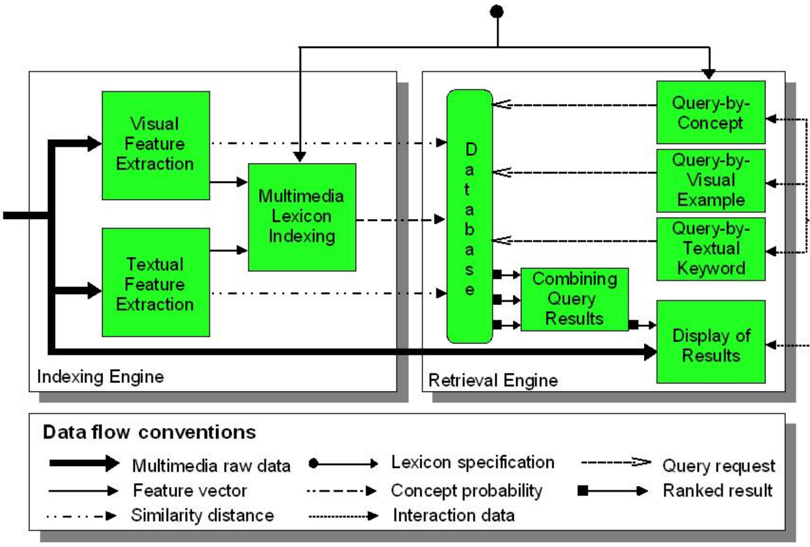


Figure 7.1: General framework for an interactive video search engine. In the indexing engine, the system learns to detect a lexicon of semantic concepts. In addition, it computes similarity distances. A retrieval engine then allows for several query selection methods. The system combines requests and displays results to a user. Based on interaction a user refines search results until satisfaction.

the results, and displays them to an interacting user. A general framework for interactive video search engines is presented in figure 7.1. While proposed solutions for effective video search engines share similar components, they stress different elements in reaching their goal.

The interactive video retrieval system proposed by Adcock *et al.* [4] combines textual analysis with an advanced user interface. Their textual analysis automatically segments recognized speech transcripts on a topic-based story level. They argue that search results should be presented in these semantically meaningful story units. Therefore, they present query-by-keyword results as story key frame collages in the user interface. Their system does not support query-by-example and query-by-concept.

In contrast to [4], Taskiran *et al.* [168] stress visual analysis for retrieval, in particular similarity of low-level color features. In addition, the authors provide users with a lexicon containing 1 concept, namely *face*. Obviously, a single concept can never address a wide variety of search topics. Thus, user interaction with the data is required. To that end, segmented shots are represented as an hierarchy of clustered frames. The authors combine this representation with query-by-example and query-by-concept by offering users query results in a so called similarity pyramid. While users browse through the pyramid they are offered a sense of the video archive at various level of (visual) detail. Unfortunately its effectiveness remains unclear, as a verification on interactive retrieval experiments is missing.

In addition to visual analysis, Fan *et al.* [40] emphasize the utility of a lexicon, containing 5 concepts, for video retrieval. The authors exploit a hierarchical classifier to index the video on shot, scene, and cluster level, allowing for hierarchical browsing of video archives on concept-level and visual similarity. Unfortunately, similar to [168], the paper lacks an evaluation of the utility of the proposed framework for interactive video retrieval. A lexicon of 5 concepts aids for interactive video retrieval, but is still limited.

Table 7.1: Overview of state-of-the-art video retrieval systems, their key-components, and evaluation details, sorted by lexicon size. Our contribution is denoted in bold.

Reference	Query-by-keyword	Query-by-example	Query-by-concept	Lexicon size	Display of results	Evaluation
Adcock *et al.* [4]	✓			0	Story board	TRECVID 2004
Taskiran *et al.* [168]		✓	✓	1	Similarity pyramid	Specific
Fan *et al.* [40]		✓	✓	5	Hierarchical summarization	Specific
Christel *et al.* [30]	✓	✓	✓	10	Story board	TRECVID 2003
Amir *et al.* [10]	✓	✓	✓	17	Grid browser	TRECVID 2003
Snoek *et al.*	✓	✓	✓	**32**	**Grid browser**	**TRECVID 2004**
Snoek *et al.*	✓	✓	✓	**101**	**Cross browser**	**TRECVID 2005**

One of the first systems to combine query-by-keyword, query-by-example, query-by-concept, and advanced display of results is the Informedia system [30, 63, 180]. It is especially strong in interactive search scenarios. In [30], the authors explain the success in interactive retrieval as a consequence of using storyboards, i.e. a grid of key frame results that are related to a keyword-based query. As queries for semantic concepts are hard to tackle using the textual modality only, the interface also supports filtering based on semantic concepts. The filters are based on a lexicon of 10 pre-indexed concepts with mixed performance [63]. Because the lexicon is limited in terms of the number of concepts, the filters are applied after a keyword-based search. The disadvantage of this approach is the dependence on keywords for initial search. Because the visual content is often not reflected in the associated text, user-interaction with this restricted answer set results in limited semantic access. To limit the dependence on keywords, we emphasize query-by-concept in the interactive video retrieval process, where possible.

A system for generic semantic indexing is proposed by Naphade *et al.* in [3, 10, 109]. The system exploits consecutive aggregations on features, multiple modalities, and concepts. Finally, the system optimizes the result by rule-based post filtering. They report good benchmark results on a lexicon of 17 concepts. In spite of the use of this lexicon, interactive retrieval results with the web-driven *MARVEL* system [10] are not competitive with [4, 30]. This is surprising, given the robustness of the concept detectors. Hence, *MARVEL* has difficulty in properly leveraging the concept detection results for interactive retrieval. A drawback of the interactive system is the lack of speed of the web-based grid browser. Moreover, it has no video playback functionality. However, the largest problem is the complex query interface that offers too many possibilities to query on low-level (visual) features and prevents users from quick retrieval of video segments of interest. We adopt and extend their ideas related to semantic video indexing, but we take a different road for interactive retrieval.

From the need to quantify effective video retrieval, we note that it has always been a delicate issue. Video archives are fragmented and mostly inaccessible due to copyrights and the sheer volume of data involved. As a consequence, many researchers evaluate their video retrieval methodologies on specific data sets, e.g. [40, 168]. To make matters worse, as the evaluation requires substantial effort, they often evaluate sub-modules of the complete framework only. This is hampering progress because methodologies can not be valued on their relative merit with respect to interactive video retrieval performance. To tackle the evaluation problem, the American National Institute of Standards and Technology (NIST) started organizing the TRECVID video retrieval benchmark. The benchmark aims to promote progress in video retrieval via open, metrics-based evaluation [147,149]. TRECVID provides video archives, a common shot segmentation, speech transcripts, and search topics that need to be solved by benchmark participants. Finally, they perform an independent examination of results using standard information retrieval evaluation measures. Because of its widespread acceptance in the field [147, 149], resulting in large participation of teams from academic labs, e.g. Carnegie Mellon University and Tsinghua University, and corporate research labs, e.g. IBM Research and FX Palo Alto Laboratory, the TRECVID benchmark can be regarded as the *de facto* standard to evaluate performance of video retrieval research.

To answer the questions related to combining video retrieval techniques and their joint evaluation in an interactive video retrieval setting, we first summarize our analysis of related work in table 7.1. It shows that interactive video retrieval methodologies stress different components indeed. We argue that a large lexicon of concepts matters most, i.e. query-by-concept should receive more emphasis in favor of traditional retrieval techniques. In this paper, we propose a lexicon-driven retrieval paradigm to equip users with semantic access to video archives (denoted in bold in table 7.1). The paradigm combines learning of a large lexicon - currently containing 32 concepts and 101 concepts respectively - with query-by-keyword, query-by-example, and interaction using an advanced display of results. We introduce the MediaMill semantic video search engine, which exploits a grid browser and a cross browser for display of results, to demonstrate the effectiveness of the proposed paradigm. Since the search engine combines several techniques, we will not discuss in-depth technical details of individual components, nor will we evaluate them. In contrast, we focus on the performance of the combined approach to interactive video retrieval using accepted benchmarks. To that end, we evaluate our lexicon-driven retrieval paradigm within the 2004 and 2005 NIST TRECVID benchmark. Interactive retrieval using the proposed paradigm facilitates effective and efficient semantic access to video archives.

7.3 The MediaMill Semantic Video Search Engine

With the MediaMill search engine we aim to retrieve from a video archive, composed of n unique shots, the best possible answer set in response to a user information need. To that end, the search engine combines learning of a large lexicon with query-by-keyword, query-by-example, and interaction. The system architecture of the search engine follows the general framework as sketched in figure 7.1. We now explain the various components of the search engine in more detail, where needed we provide pointers to published papers covering in-depth technical details.

7.3.1 Indexing Engine

Textual & Visual Feature Extraction

To arrive at a similarity distance for the textual modality we first derive words from automatic speech recognition results, obtained with standard tools, e.g. [47]. We exploit standard machine translation tools [83] in case the videos originate from non-English speaking countries. This allows for a generic approach. We remove common stop words from the English text using the SMART's English stop list [136]. We then construct a high dimensional vector space based on all remaining transcribed words. We rely on latent semantic indexing [35] to reduce the search space to 400 dimensions. While doing so, the method takes co-occurrence of related words into account by projecting them onto the same dimension. The rationale is that this

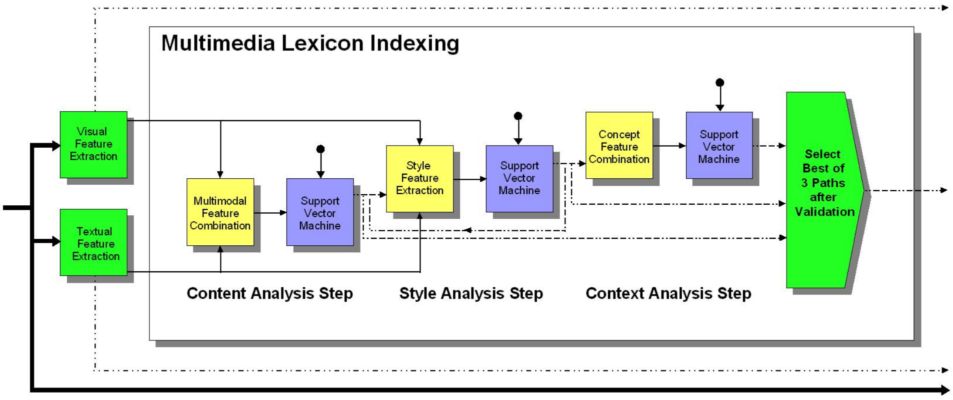

Multimedia Lexicon Indexing
Visual Feature Extraction
Textual Feature Extraction
Multimodal Feature Combination
Support Vector Machine
Style Feature Extraction
Support Vector Machine
Concept Feature Combination
Support Vector Machine
Select Best of 3 Paths after Validation
Content Analysis Step
Style Analysis Step
Context Analysis Step

Figure 7.2: Multimedia lexicon indexing is based on the semantic pathfinder [158], Chapter 6. We highlight its successive analysis steps in the detail from figure 7.1. The semantic pathfinder selects for each concept a best analysis path after validation.

reduced space is a better representation of the search space. When users exploit query-by-keyword as similarity measure, the terms of the query are placed in the same reduced space. The most similar shots, viz. the ones closest to the query in that space, are returned, regardless of whether they contain the original query terms.

In the visual modality the similarity query is by example. For all key frames in the video archive, we compute the perceptually uniform *Lab* color histogram [51] using 32 bins for each color channel. Users compare key frames with Euclidean histogram distance.

Multimedia Lexicon Indexing

Generic semantic video indexing is required to obtain a large concept lexicon. In literature, several approaches are proposed [3,10,40,41,109,154,158,159]. The utility of supervised learning in combination with multimedia content analysis has proven to be successful, with recent extensions to include video production style [159] and the insight that concepts often co-occur in context [3, 10, 109]. We combine these successful approaches into an integrated video indexing architecture.

The design principle of our architecture is derived from the idea that the essence of produced video is its creation by an author. Style is used to stress the semantics of the message, and to guide the audience in its interpretation. In the end, video aims at an effective semantic communication. All of this taken together, the main focus of semantic indexing must be to reverse this authoring process, for which we proposed the semantic pathfinder [154, 158], Chapter 6. The semantic pathfinder is composed of three analysis steps, see figure 7.2. The output of an analysis step in the pathfinder forms the input for the next one. We build this architecture on machine learning of concepts for the robust detection of semantics. An in depth discussion of the various techniques used is presented in Chapter 6.

7.3.2 Retrieval Engine

Video retrieval engines are often dictated by technical possibilities rather than actual user needs [87]. Frequently this results in an overly complex system. To shield the user from technical complexity, while at the same time offering increased efficiency, we store all computed indexes in a database. Users interact with the retrieval engine based on query selection methods. Each query method acts as a ranking operator

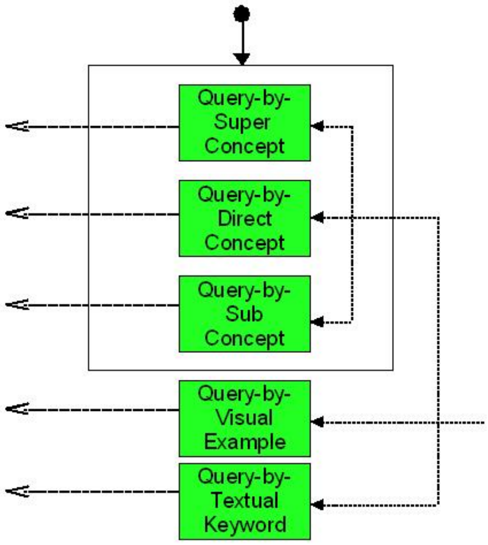


Figure 7.3: The MediaMill video search engine offers interacting users several methods for query selection. In the detail from figure 7.1, we highlight three query-by-concept methods, together with query-by-example, and query-by-keyword.

on the video archive. After a user issues a query it is processed and combined into a final result, which is presented to the user. The elements of our retrieval engine are now discussed in more detail.

Query Selection

The set of concepts in the lexicon forms the basis for interactive selection of query results. We identify three ways to exploit the lexicon for querying, i.e. *query-by-direct concept*, *query-by-sub concept*, and *query-by-super concept*. Users may rely on query-by-direct concept for search topics related directly to concepts from the lexicon. In case the lexicon contains the concept *aircraft*, all information needs related to *aircrafts* benefit from query-by-direct concept. This is an enormous advantage for the precision of the search. Users can also make a first selection when a query includes a sub-concept or a super-concept of a concept in the lexicon. For example, when searching for *sports* one can exploit query-by-sub concept using the available sport sub-concepts *tennis*, *soccer*, *baseball*, and *golf* from the lexicon. In a similar fashion, users may exploit query-by-super concept using *animal* to retrieve footage related to *ice bear*. To aid the user in the selection of the query we make lexicon concepts available in the form of a subset of the WordNet [42] taxonomy. This helps the user to take well-established concept relations into account. The layout of the interface has the same order as WordNet for maximum comfort. In this way, the lexicon of concepts aids users in various ways in specifying their queries.

For search topics not covered by the concepts in the lexicon, users have to rely on similarity distances in the form of query-by-keyword and query-by-example. Applying query-by-keyword in isolation allows users to find very specific topics only if they are mentioned in the transcription from automatic speech recognition. Based on query-by-example, on either provided or retrieved images, key frames that exhibit a similar color distribution can augment results further. This is especially fruitful for repetitive key frames that contain similar visual content throughout the archive, such as previews, graphics, and commercials.

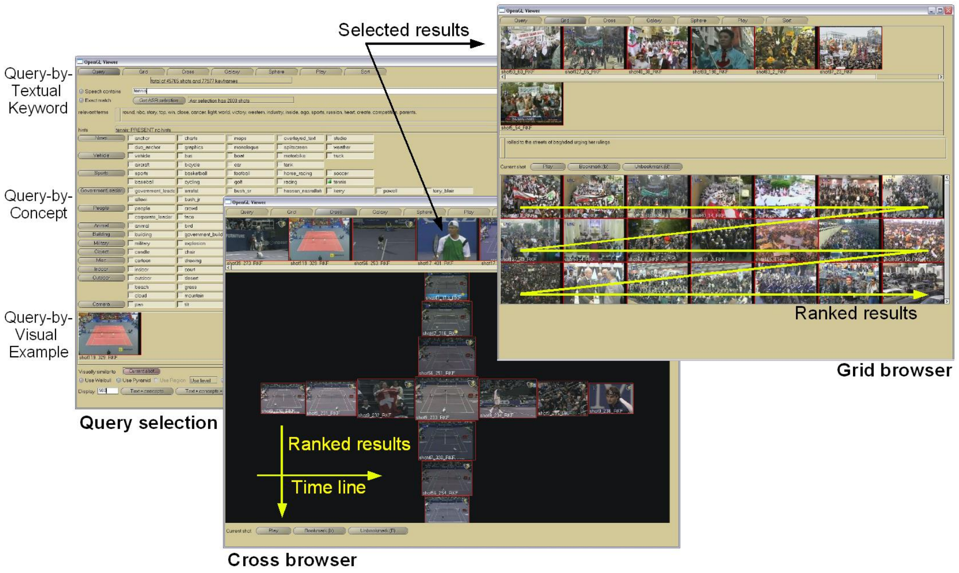


Figure 7.4: Interface of the MediaMill semantic video search engine. The system allows for interactive query-by-concept using a large lexicon. In addition, it facilitates query-by-keyword and query-by-example. For display of results users may rely on a cross browser or a grid browser.

Naturally, the search engine provides users the possibility to combine query selection methods. This is helpful when a concept is too general and needs refinement. For example when searching for Microsoft stock quotes, a user may combine query-by-concept *stock quotes* with query-by-keyword *Microsoft*. While doing so, the search engine exploits both the concept lexicon and the multimedia similarity distances. We summarize the methods for query selection in figure 7.3.

Combining Query Results

To rank results, query-by-concept exploits semantic probabilities, while query-by-keyword and query-by-example use similarity distances. When users mix query interfaces, and hence several numerical scores, this introduces the question how to combine the results. As noted in Section 7.2, one solution is to query the system in a sequential fashion. In such a scenario, a user may start with query-by-keyword, results obtained are subsequently filtered using query-by-concept. The disadvantage of this approach is the dependence on the accuracy of the initial query method. Therefore, we opt for a combination method that provides us the possibility to exploit query results in parallel. Rankings offer us a comparable output across various query results. Various ranking combination methods exist [68]. We employ a standard approach, using summation of linear rank normalizations [88], to combine query results.

Display of Results

The search engine supports two modes for displaying results. In the traditional *grid browser* a ranked list of key frame results is visualized as a lattice of thumbnails ordered left to right, top to bottom. However, ranking is a linear ordering. So, ideally it should be visualized as such. This leaves room to use the other dimension for visualization of the chronological series, or story, of the video program from which a key frame is selected. This makes sense as frequently other items in the same broadcast are relevant to a query also [4, 30]. Therefore, we also employ a *cross browser*, which facilitates quick selection of relevant results. If requested, playback of specific shots is also possible. We rely on interaction by a user to select query methods and combine retrieval results. Technically, the interface of the search engine is implemented in *OpenGL* to allow for easy query selection and swift visualization of results. We depict the various aspects of the user interface of the MediaMill video search engine in figure 7.4.

7.4 Experimental Setup

We investigate the impact of the proposed lexicon-driven paradigm for interactive video retrieval by performing 2 experiments with the MediaMill semantic video search engine:

- **Experiment 1:** *Interactive video retrieval with a 32 concept lexicon*;

In the first experiment, we evaluate video retrieval effectiveness using the MediaMill search engine in combination with a 32 concept lexicon and the grid browser.

- **Experiment 2:** *Interactive video retrieval with a 101 concept lexicon*;

In the second experiment, we evaluate video retrieval effectiveness using the MediaMill search engine in combination with a 101 concept lexicon and the cross browser. Finally, we compare interactive retrieval results obtained using the MediaMill search engine with a dozen other video retrieval systems. To allow for comparison, we perform all experiments as part of the interactive search tasks of the 2004 and 2005 NIST TRECVID benchmark.

7.4.1 Interactive Search

The goal of the interactive search task is to satisfy a number of video information needs. Given such a need, in the form of a search topic, a user is engaged in an interactive session with a video search engine. Based on the results obtained, a user rephrases queries; aiming at retrieval of more and more accurate results. To limit the amount of user interaction and to measure search system efficiency, all individual search topics are bounded by a 15-minute time limit. The 2004 interactive search task contains 23 search topics in total, the 2005 edition has 24. In line with the TRECVID submission procedure, a user was allowed to submit, for assessment by NIST, up to a maximum of 1,000 ranked results for the various search topics.

The 2004 video archive includes 184 hours of ABC World News Tonight and CNN Headline News. The training data contains approximately 120 hours covering the period of January until June 1998. The test data holds the remaining 64 hours, covering the period of October until December 1998. The 2005 archive contains 169 hours with 287 episodes from 13 broadcast news shows from US, Arabic, and Chinese sources, recorded during November 2004. The test data contains approximately 85 hours. Together with the video archives came automatic speech recognition results donated in 2004 by LIMSI [47] and in 2005 by a US government contractor. CLIPS-IMAG [129] and the Fraunhofer Institute [123] provided a camera shot segmentation, in 2004 and 2005 respectively. The camera shots serve as the unit for retrieval. Similar to concept detection, TRECVID uses *average precision* to determine the retrieval accuracy on individual search topics, see Section 6.2.2. As an indicator for overall search system quality TRECVID computes the mean average precision over all search topics from one run by a single user.

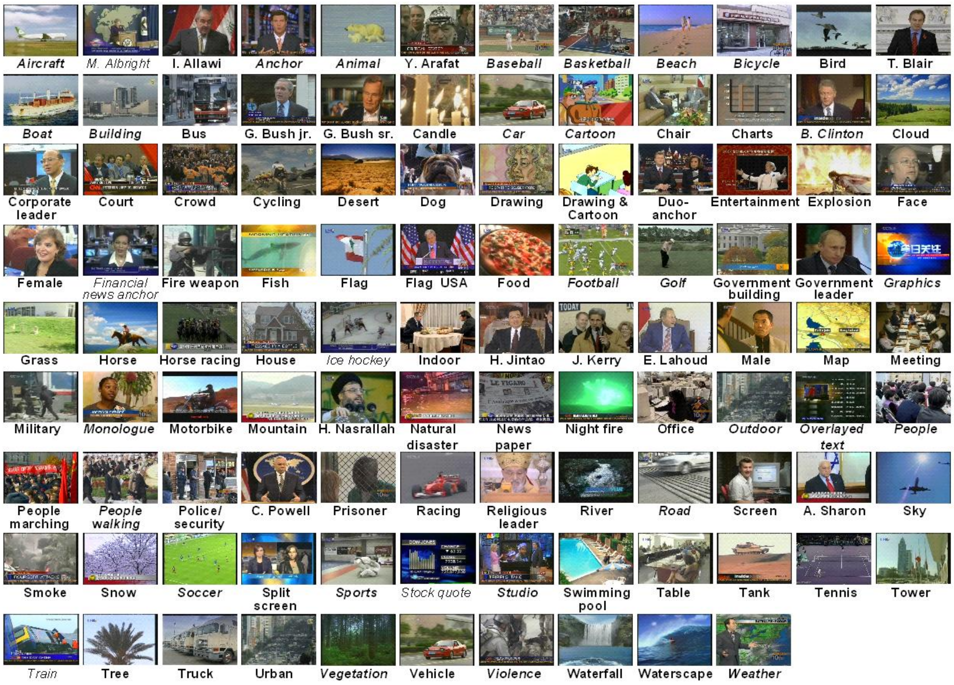


Figure 7.5: Instances of the concepts from the lexicons used. The lexicon of 32 concepts (TRECVID 2004) is given in italics, the 101 concept lexicon (TRECVID 2005) is denoted in bold. Concepts which appear in both lexicons follow two conventions.

7.4.2 Lexicon Specification

We automatically detect a lexicon of semantic concepts in both the TRECVID 2004 and 2005 data using the semantic pathfinder, as discussed in Section 7.3.1. In the 2004 data we detect a lexicon of 32 concepts, in the 2005 data a lexicon of 101 concepts[†]. We select concepts by following a predefined concept ontology for multimedia [112] as leading example. Concepts in this ontology are chosen based on presence in WordNet [42] and extensive analysis of video archive query logs. Where concepts should be related to program categories, setting, people, objects, activities, events, and graphics. In addition, a primary design choice was that concepts need to be clear by looking at a static key frame only. We visualize instantiations of the detected concepts in both lexicons in figure 7.5, additional details for 2004 data are in [158], for 2005 data are in [154].

It should be noted that although we have a large lexicon of concepts, with state-of-the-art results for generic indexing [158], performance of them is far from perfect. This often results in noisy detection outcomes. To give an indication of performance, we highlight our official TRECVID concept detection results on test data in table 7.2. The TRECVID procedure prescribes that 10 pre-defined concepts are evaluated. Hence, for each year, we can report the official benchmark results for 10 concepts in our lexicon only. The benchmark concepts are, however, representative for the entire lexicons.

We stress that the various topics became known only a few days before the deadline of submission. Hence, they were unknown at the time we developed our semantic concept detectors. Moreover, the test set performance of the concepts was unknown at the time we performed our interactive search experiments. To show the potential of our lexicon-driven paradigm we performed an experiment with a

[†] [Online]. Available: `http://www.mediamill.nl/challenge/` [162].

Table 7.2: MediaMill average precision results for the TRECVID 2004 [158] and 2005 [154] concept detection task.

TRECVID 2004		**TRECVID 2005**	
Concept	*Average Precision*	*Concept*	*Average Precision*
Aircraft	0.065	Building	0.235
M. Albright	0.136	Car	0.213
Basketball	0.209	Explosion	0.041
Beach	0.020	Flag USA	0.100
Boat	0.117	Map	0.142
B. Clinton	0.150	Mountain	0.220
People walking	0.170	People walking	0.199
Road	0.138	Prisoner	0.005
Train	0.062	Sports	0.342
Violence	0.086	Waterscape	0.201

single expert user, which is common procedure in TRECVID, e.g. [4, 10, 30]. Our expert user had no experience with the topics nor with the test data. The user did have experience with the MediaMill system and its concept lexicons, but only on training data, which is conform TRECVID guidelines.

7.5 Results

7.5.1 Retrieval with a 32 Concept Lexicon

We plot the complete numbered list of search topics used in our first experiment in figure 7.6. In addition, we plot the benchmark results for 61 users with 14 present-day interactive multimedia retrieval systems. The results give us insight in the contribution of the proposed paradigm for individual search topics when using a lexicon of 32 concepts.

For most search topics, the user of the proposed paradigm for interactive multimedia retrieval scores above average. Furthermore, the user of our approach obtains the highest average precision for seven search topics (Topics: 3, 14, 15, 16, 18, 20, 21). We explain the success of our interactive retrieval paradigm in this experiment in part by the lexicon used. In our lexicon, there was an (accidental) overlap with the requested concepts from some search topics; for example *ice hockey*, *bicycle*, and *Bill Clinton* (Topics: 6, 16, 20), where performance is very good. Implying that there is much to be expected from a larger set of concepts in the lexicon. For other topics, the user could use query-by-super concept for filtering, e.g. *sporting event* for tennis player (Topic: 18) and *animal* for horses (Topic: 21). So in our method, abstract concepts make sense even when they are referred to indirectly. As an exception, for search topics related to the concept *building* (Topics: 2, 22), our retrieval method performed badly compared to the best results. We explain this behavior by the fact that building was not the distinguishing concept in these topics, but rather concepts like *flood* and *fire*, implying that some concepts are more important than others.

The user of the paradigm performed moderate for search topics that did not have a clear overlap with the concepts in the lexicon. For topics related to wheelchairs (Topic: 19), umbrellas (Topic: 17), and person X who were not in the lexicon (Topics: 4, 9, 10, 11, 13), query-by-keyword is the only viable alternative.

When a user recognizes an answer to a search topic as a commercial or signature tune, query-by-example is particularly useful. Search topics profiting from this

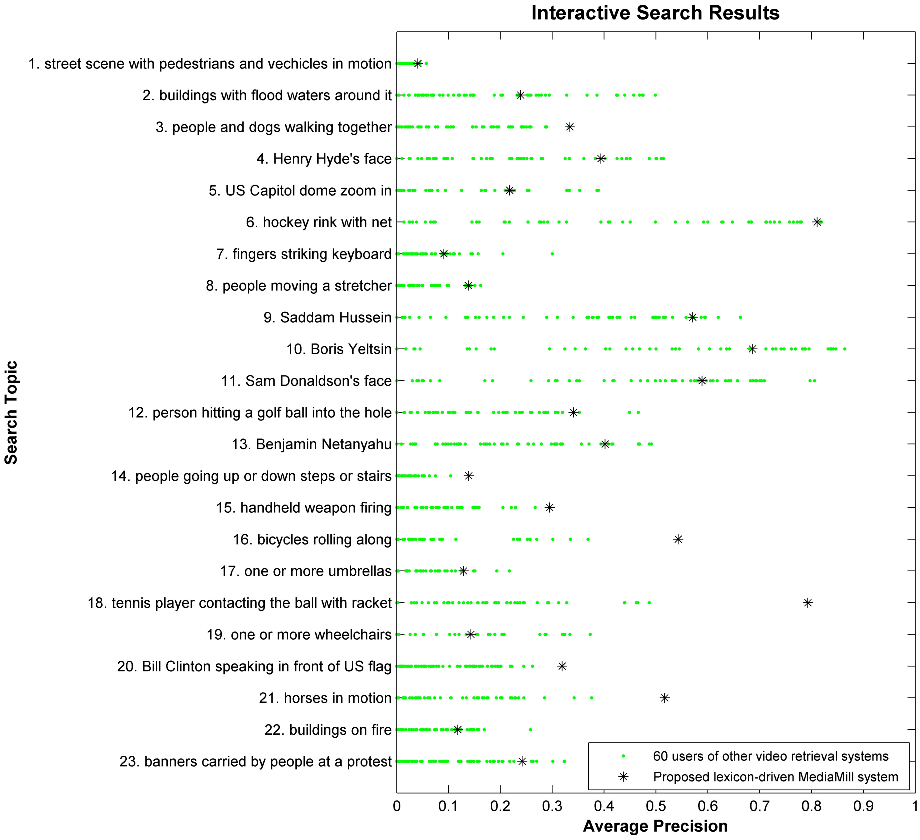


Figure 7.6: Comparison of interactive search results for 23 topics performed by 61 users of 14 present-day video retrieval systems. Results for the user of the proposed paradigm, with a 32 concept lexicon, are indicated with special markers.

observation are those related to bicycle and tennis player (Topics: 16, 18). Since these fragments contain similar visual content throughout the archive, they are easily retrievable with query-by-example.

After this first experiment, we conclude that for search topics related to concepts in the lexicon, query-by-concept is a good starting point. Query-by-keyword is effective when the (visual) content is described in the speech signal. If a user is interested in footage that is repeated throughout the archive, query-by-example is the way to go. With a lexicon containing only 32 concepts we already diminish the influence of traditional video retrieval techniques in favor of query-by-concept.

7.5.2 Retrieval with a 101 Concept Lexicon

We again plot the complete numbered list of search topics in figure 7.7 for our second experiment, where we use a lexicon of 101 concepts. Together with the topics, we plot the benchmark results for 49 users using 16 present-day interactive video search engines.

The results confirm the value of a large lexicon for interactive video retrieval. For most search topics the user of the proposed paradigm scores excellent, yielding a top 3 average precision for 17 out of 24 topics. Furthermore, our approach obtains the highest average precision for five search topics (Topics: 26, 31, 33, 36, 43). In our lexicon, there was an (accidental) overlap with the requested concepts from almost

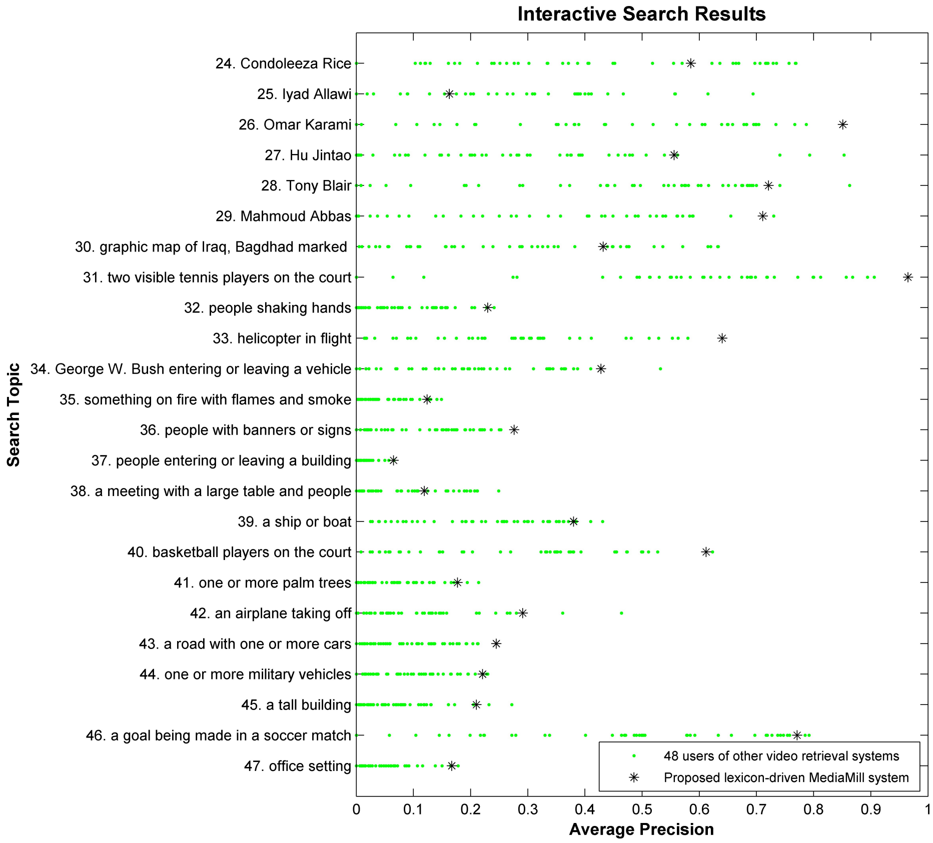
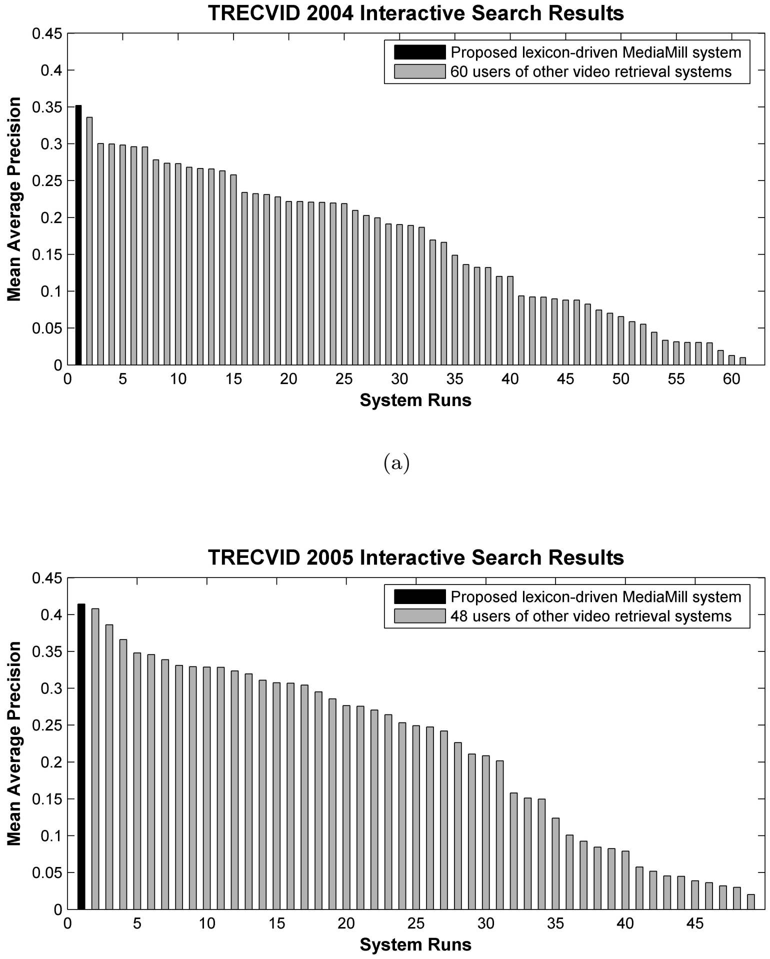


Figure 7.7: Comparison of interactive search results for 24 topics performed by 49 users of 16 present-day video retrieval systems. Results for the user of the proposed paradigm, with a 101 concept lexicon, are indicated with special markers.

all search topics; for example *tennis*, *people marching*, and *road* (Topics: 31, 36, 43), where performance is very good. These results demonstrate that many video search questions are solvable without using query-by-keyword and query-by-example.

The search engine performed moderate for topics that were not in the lexicon (Topic: 24), or which yielded very poor concept detection (Topic: 25). For these topics our user had to rely on query-by-keyword. In addition, we also performed less than expected for topics that require specific instances of a concept, e.g. maps with Bagdhad marked (Topic: 30). Although the concept *map* was part of our lexicon, our user was unable to exert this advantage. When search topics contain combinations of several reliable concepts, e.g. meeting, table, people (Topic: 38), results are also not optimal. This indicates that much is to be expected from a more intelligent combination of query results.

For some topics, the MediaMill search engine may be exploited in an unexpected way. By the use of common sense, the lexicon is also useful for topics that do not have a clear one-to-one relation with a concept. One of the search topics profiting from this observation is people shaking hands (Topic: 32). For this topic, the concept *government leader* is helpful. Indeed, government leaders shake hands quite often when visiting or welcoming fellow foreign leaders, which is often broadcasted in news items. For the topic on finding one or more palm trees (Topic: 41), query-by-direct concept on *tree* was not specific enough. Here our user exploited common sense by using the fact that by searching on *military* the system returns a lot of shots



(a)

(b)

Figure 7.8: Overview of all interactive search runs submitted to TRECVID 2004 (a) and TRECVID 2005 (b), ranked according to mean average precision.

from the war in Iraq. Indeed often containing palm trees. Lebanese former prime minister Omar Karami (Topic: 26) was not included in our lexicon. For this topic we combine common sense with the cross browser. Omar Karami appears often in long interviews. Thus, when a single shot from such an interview is localized, the cross browser offers an opportunity to select a large amount of relevant shots easily. When users employ common sense, the lexicon-driven paradigm becomes even more powerful.

Our second experiment shows that a large lexicon is the most valuable resource for interactive video retrieval. With a lexicon of 101 concepts almost all search topics are solvable directly, or indirectly, with good performance. Hence, the value of the lexicon-driven paradigm is evident. In fact, it diminishes the value of traditional techniques such as query-by-keyword and query-by-example to purely supportive tools for topics that can not be addressed by concepts from the lexicon. Using a large lexicon implies a paradigm shift for interactive video retrieval.

7.5.3 Benchmark Comparisons

To gain insight in the overall quality of our lexicon-driven interactive retrieval paradigm, we compare the mean average precision results of our lexicon-driven MediaMill video search engine with other state-of-the-art systems. For TRECVID 2004 we compare against 13 other retrieval systems. For TRECVID 2005 we compare against 15 present-day video search engines. Our approach is unique with respect to lexicon size, most others emphasize traditional retrieval paradigms. We visualize the results for all submitted interactive search runs of TRECVID 2004 in figure 7.8a, and TRECVID 2005 in figure 7.8b.

The results show that the proposed search engine obtains a mean average precision of 0.352 in TRECVID 2004, and 0.414 in TRECVID 2005. In both cases the highest overall score. In [66] the authors showed that the top 2 TRECVID 2004 systems significantly outperform all other submissions. What is striking about these results, is that we obtain them by using a lexicon of only 32 concepts. When we increase the concept lexicon to 101 concepts in TRECVID 2005, only 3 users stay within a difference of 0.05 mean average precision. These users employed a video retrieval system based on rapid serial visual presentation of search results [64]. In such a scenario a user is bombarded with as much key frames as possible. While effective in terms of TRECVID performance, this demanding approach is suited for limited domains only. The benchmark results demonstrate that lexicon-driven interactive retrieval yields superior performance relative to state-of-the-art video search engines.

7.6 Conclusion

In this paper, we combine automatic learning of a large lexicon of semantic concepts with video retrieval methods into an effective video search system. The aim of the combined system is to narrow the semantic gap for the user. The foundation of the proposed approach is to learn a lexicon of semantic concepts. Where it should be noted that we have used a generic machine learning system and no per-concept optimizations. Based on this learned lexicon, query-by-concept offers users a semantic entrance to video repositories. In addition, users are provided with an entry in the form of textual query-by-keyword and visual query-by-example. Interaction with the various query interfaces is handled by an advanced display of results, which provides feedback in the form of a grid browser or a cross browser. The resulting *MediaMill* semantic video search engine limits the influence of the semantic gap.

We investigate the impact of the proposed lexicon-driven paradigm for interactive video retrieval by performing 2 experiments with the MediaMill semantic video search engine. In our first experiment, with a lexicon of only 32 concepts, we already outperform state-of-the-art systems in 7 out of 23 random queries on 64 hours of US broadcast news. When we increase the lexicon to 101 concepts, in our second experiment, we obtain a top 3 average precision for 17 out of 24 topics and top performance for 5 topics on a 85 hours international news video archive. The key insight resulting from these experiments is that from all factors that play a role in interactive retrieval, a large lexicon of semantic concepts matters most. This is best demonstrated when we compare our lexicon-driven approach against the 2004 and 2005 NIST TRECVID benchmark. In both cases our MediaMill system obtains superior performance relative to a dozen other state-of-the-art video search engines, which still adhere to traditional video retrieval paradigms.

Retrieval results with the proposed paradigm range from 'poor' for topics like "find street scenes with pedestrians and vehicles in motion" to excellent for non-trivial topics like “find two visible tennis players on the court”. However, under all topics the performance is good relative to other systems and best overall. Fluctuating performance of multimedia retrieval technology is unacceptable for highly demanding applications, such as military intelligence. However, when used in a less demanding commercial search scenario, the proposed paradigm provides already valuable semantic information.

Keyterms in this chapter

Query-by-super-concept, query-by-direct-concept, query-by-sub-concept, query-by-visual-example, query-by-textual-keyword, query result combination, mean average precision

Bibliography

[1] TREC video retrieval evaluation online proceedings. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[2] S. Abney. Part-of-speech tagging and partial parsing. In S. Young and G. Bloothooft, editors, *Corpus-Based Methods in Language and Speech Processing*, pages 118–136. Kluwer Academic Publishers, Dordrecht, 1997.

[3] W. H. Adams, G. Iyengar, C.-Y. Lin, M.R. Naphade, C. Neti, H.J. Nock, and J.R. Smith. Semantic indexing of multimedia content using visual, audio, and text cues. *EURASIP J. Applied Signal Processing*, (2):170–185, 2003.

[4] J. Adcock, M. Cooper, A. Girgensohn, and L. Wilcox. Interactive video search using multilevel indexing. In *CIVR*, volume 3569 of *LNCS*, pages 205–214. Springer-Verlag, 2005.

[5] D.A. Adjeroh, I. King, and M.C. Lee. A distance measure for video sequences. *Computer Vision and Image Understanding*, 75(1):25–45, 1999.

[6] Ph. Aigrain, Ph. Joly, and V. Longueville. *Medium Knowledge-Based Macro-Segmentation of Video Into Sequences*, chapter 8, pages 159–173. Intelligent Multimedia Information Retrieval. AAAI Press, 1997.

[7] A.A. Alatan, A.N. Akansu, and W. Wolf. Multi-modal dialogue scene detection using hidden markov models for content-based multimedia indexing. *Multimedia Tools and Applications*, 14(2):137–151, 2001.

[8] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.

[9] Y. Altunbasak, P.E. Eren, and A.M. Tekalp. Region-based parametric motion segmentation using color information. *Graphical models and image processing*, 60(1):13–23, 1998.

[10] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M.R. Naphade, A.P. Natsev, C. Neti, H.J. Nock, J.R. Smith, B.L. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.

[11] J. Baan, A. van Ballegooij, J.-M. Geusebroek, D. Hiemstra, J. den Hartog, J. List, C. Snoek, I. Patras, S. Raaijmakers, L. Todoran, J. Vendrig, A. de Vries, T. Westerveld, and M. Worring. Lazy users and automatic video retrieval tools in (the) lowlands. In E.M. Voorhees and D.K. Harman, editors, *Proc. 10th Text REtrieval Conference*, volume 500-250 of *NIST Special Publication*, Gaithersburg, USA, 2001.

[12] N. Babaguchi, Y. Kawai, and T. Kitahashi. Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Trans. Multimedia*, 4(1):68–75, 2002.

[13] H.E. Bal et al. The distributed ASCI supercomputer project. *Operating Syst. Review*, 34(4):76–96, 2000.

[14] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(7):711–720, 1997.

[15] A.B. Benitez, J.R. Smith, and S.-F. Chang. MediaNet: A multimedia information network for knowledge representation. In *Proc. SPIE Conf. Internet Multimedia Management Syst.*, volume 4210, Boston, USA, 2000.

[16] M. Bertini, A. Del Bimbo, and P. Pala. Content-based indexing and retrieval of TV news. *Pattern Recognition Letters*, 22(5):503–516, 2001.

[17] D. Bikel, R. Schwartz, and R.M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[18] Blinkx Video Search, 2006. `http://www.blinkx.tv/`.

[19] J.M. Boggs and D.W. Petrie. *The Art of Watching Films.* Mayfield Publishing Company, Mountain View, USA, 5th edition, 2000.

[20] R.M. Bolle, B.-L. Yeo, and M.M. Yeung. Video query: Research directions. *IBM Journal of Research and Development*, 42(2):233–252, 1998.

[21] A. Bonzanini, R. Leonardi, and P. Migliorati. Event recognition in sport programs using low-level motion indices. In *IEEE Int'l Conf. Multimedia Expo*, pages 1208–1211, Tokyo, Japan, 2001.

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

[23] M. Brown, J. Foote, G. Jones, K. Sparck-Jones, and S. Young. Automatic content-based retrieval of broadcast news. In *ACM Multimedia*, San Francisco, USA, 1995.

[24] R. Brunelli, O. Mich, and C.M. Modena. A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, 10(2):78–112, 1999.

[25] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(8):1026–1038, 2002.

[26] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998.

[27] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[28] S.-F. Chang, W. Chen, H.J. Men, H. Sundaram, and D. Zhong. A fully automated content-based video search engine supporting spatio-temporal queries. *IEEE Trans. Circuits Syst. Video Technol.*, 8(5):602–615, 1998.

[29] P. Chiu, Girgensohn, W. Polak, E. Rieffel, and L. Wilcox. A genetic algorithm for video segmentation and summarization. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 1329–1332, 2000.

[30] M. Christel, C. Huang, N. Moraveji, and N. Papernick. Exploiting multiple modalities for interactive video retrieval. In *IEEE Int'l Conf. Acoust., Speech, Signal Processing*, volume 3, pages 1032–1035, Montreal, Canada, 2004.

[31] M. Christel, A. Olligschlaeger, and C. Huang. Interactive maps for a digital video library. *IEEE Multimedia*, 7(1):60–67, 2000.

[32] C. Colombo, A. Del Bimbo, and P. Pala. Semantics in visual information retrieval. *IEEE Multimedia*, 6(3):38–53, 1999.

[33] Convera, December 2001. `http://www.convera.com`.

[34] G. Davenport, T.G. Aguierre Smith, and N. Pincever. Cinematic principles for multimedia. *IEEE Comput. Graph. Appl.*, 11(4):67–74, 1991.

[35] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. American Soc. Inform. Sci.*, 41(6):391–407, 1990.

[36] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(2):121–132, 1997.

[37] N. Dimitrova, L. Agnihotri, and G. Wei. Video classification based on HMM using text and faces. In *European Signal Processing Conference*, Tampere, Finland, 2000.

[38] S. Eickeler and S. Müller. Content-based video indexing of TV broadcast news using hidden markov models. In *IEEE Int'l Conf. Acoust., Speech, Signal Processing*, pages 2997–3000, Phoenix, USA, 1999.

[39] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. Speech/music discrimination for multimedia applications. In *IEEE Int'l Conf. Acousts, Speech, and Signal Processing*, pages 2445–2448, Istanbul, Turkey, 2000.

[40] J. Fan, A.K. Elmagarmid, X. Zhu, W.G. Aref, and L. Wu. *ClassView*: hierarchical video shot classification, indexing, and accessing. *IEEE Trans. Multimedia*, 6(1):70–86, 2004.

[41] J. Fan, H. Luo, and A.K. Elmagarmid. Concept-oriented indexing of video databases: Toward semantic sensitive retrieval and browsing. *IEEE Trans. Image Processing*, 13(7):974–992, 2004.

[42] C. Fellbaum, editor. *WordNet: an electronic lexical database*. The MIT Press, Cambridge, USA, 1998.

[43] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *ACM Multimedia*, pages 295–304, San Francisco, USA, 1995.

[44] M.M. Fleck, D.A. Forsyth, and C. Bregler. Finding naked people. In *European Conference on Computer Vision*, volume 2, pages 593–602, Cambridge, UK, 1996.

[45] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[46] B. Furht, S.W. Smoliar, and H.-J. Zhang. *Video and Image Processing in Multimedia Systems*. Kluwer Academic Publishers, Norwell, USA, 2th edition, 1996.

[47] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Commun.*, 37(1–2):89–108, 2002.

[48] J. Gemmel, G. Bell, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: fulfilling the memex vision. In *Proceedings of the tenth ACM international conference on Multimedia*, 2002.

[49] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(12):1338–1350, 2001.

[50] Th. Gevers and A. W. M. Smeulders. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE Trans. Image Processing*, 9(1):102–119, 2000.

[51] Th. Gevers and A.W.M. Smeulders. Content-based image retrieval: An overview. In G. Medioni and S.B. Kang, editors, *Emerging Topics in Computer Vision*. Prentice Hall, 2004.

[52] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith. Query by humming – musical information retrieval in an audio database. In *ACM Multimedia*, San Francisco, USA, 1995.

[53] Y. Gong, L.T. Sin, and C.H. Chuan. Automatic parsing of TV soccer programs. In *IEEE Int'l Conf. Multimedia Computing and Systems*, pages 167–174, 1995.

[54] Google Video Search, 2006. `http://video.google.com/`.

[55] B. Günsel, A.M. Ferman, and A.M. Tekalp. Video indexing through integration of syntactic and semantic features. In *Third IEEE Workshop on Applications of Computer Vision*, Sarasota, USA, 1996.

[56] A. Gupta and R. Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, 1997.

[57] N. Haering, R. Qian, and I. Sezan. A semantic event-detection approach and its application to detecting hunts in wildlife video. *IEEE Trans. Circuits and Systems for Video Technology*, 10(6):857–868, 2000.

[58] A. Hampapur, R. Jain, and T. Weymouth. Feature based digital video indexing. In *IFIP 2.6 Third Working Conference on Visual Database Systems*, Lausanne, Switzerland, 1995.

[59] A. Hanjalic, G. Kakes, R.L. Lagendijk, and J. Biemond. Dancers: Delft advanced news retrieval system. In *IS&T/SPIE Electronic Imaging 2001: Storage and Retrieval for Media Databases 2001*, San Jose, USA, 2001.

[60] A. Hanjalic, R.L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):580–588, 1999.

[61] A. Hanjalic, G.C. Langelaar, P.M.B. van Roosmalen, J. Biemond, and R.L. Lagendijk. *Image and Video Databases: Restoration, Watermarking and Retrieval*. Elsevier Science, Amsterdam, The Netherlands, 2000.

[62] A.G. Hauptmann. Towards a large scale concept ontology for broadcast video. In *CIVR*, volume 3115 of *LNCS*, pages 674–675. Springer-Verlag, 2004.

[63] A.G. Hauptmann, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H.Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, R. Yang, and H.D. Wactlar. Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.

[64] A.G. Hauptmann, M. Christel, R. Concescu, J. Gao, Q. Jin, W.-H. Lin, J.-Y. Pan, S. M. Stevens, R. Yan, J. Yang, and Y. Zhang. CMU Informedia's TRECVID 2005 skirmishes. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2005.

[65] A.G. Hauptmann, D. Lee, and P.E. Kennedy. Topic labeling of multilingual broadcast news in the informedia digital video library. In *ACM DL/SIGIR MIDAS Workshop*, Berkely, USA, 1999.

[66] A.G. Hauptmann and W.-H. Lin. Assessing effectiveness in video retrieval. In *CIVR*, volume 3569 of *LNCS*, pages 215–225. Springer-Verlag, 2005.

[67] A.G. Hauptmann and M.J. Witbrock. Story segmentation and detection of commercials in broadcast news video. In *ADL-98 Advances in Digital Libraries*, pages 168–179, Santa Barbara, USA, 1998.

[68] T.K. Ho, J.J. Hull, and S.N. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(1):66–75, 1994.

[69] L. Hollink. *Semantic Annotation for Retrieval of Visual Resources.* PhD thesis, Vrije Universiteit Amsterdam, November 2006.

[70] L. Hollink, A. Th. Schreiber, B. Wielinga, and M. Worring. Classification of user image descriptions. *International Journal of Human Computer Studies*, 61(5):601–626, 2004.

[71] L. Hollink, A.Th. Schreiber, J. Wielemaker, and B. J. Wielinga. Semantic annotation of image collections. In *Proc. K-Cap 2003 Workshop on Knowledge Markup and Semantic Annotation*, Sanibel Island, FL, October 2003.

[72] T. Hori and K. Aizawa. Context-based video retrieval system for the life-log applications. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 31–38, 2003.

[73] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E.K. Wong. Integration of multimodal features for video scene classification based on HMM. In *IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, 1999.

[74] Eero Hyvönen, Samppa Saarela, Kim Viljanen, Eetu Mäkelä, Arttu Valo, Mirva Salminen, Suvi Kettula, and Miikka Junnila. A cultural community portal for publishing museum collections on the semantic web. In *ECAI Workshop on Application of Semantic Web Technologies to Web Communities*, Valencia, Spain, 2004.

[75] I. Ide, K. Yamamoto, and H. Tanaka. Automatic video indexing based on shot classification. In *First Int'l Conf. Advanced Multimedia Content Processing*, volume 1554 of *LNCS*, pages 87–102, Osaka, Japan, 1999. Springer-Verlag.

[76] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):4–37, 2000.

[77] R. Jain and A. Hampapur. Metadata in video databases. *ACM SIGMOD*, 23(4):27–33, 1994.

[78] P.J. Jang and A.G. Hauptmann. Learning to recognize speech by watching television. *IEEE Intelligent Systems*, 14(5):51–58, 1999.

[79] R.S. Jasinschi, N. Dimitrova, T. McGee, L. Agnihotri, J. Zimmerman, and D. Li. Integrated multimedia processing for topic segmentation and classification. In *IEEE Int'l Conf. Image Processing*, pages 366–369, Thessaloniki, Greece, 2001.

[80] O. Javed, Z. Rasheed, and M. Shah. A framework for segmentation of talk & game shows. In *IEEE Int'l Conf. Computer Vision*, Vancouver, Canada, 2001.

[81] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database – query by visual example. In *Proc. Int'l Conf. Pattern Recognition*, volume 1, pages 530–533, The Hague, The Netherlands, 1992.

[82] J.R. Kender and B.L. Yeo. Video scene segmentation via continuous video coherence. In *CVPR'98, Santa Barbara, CA*. IEEE, IEEE, June 1998.

[83] K. Knight and D. Marcu. Machine translation in the year 2004. In *IEEE Int'l Conf. Acoust., Speech, Signal Processing*, volume 5, pages 965–968, Philadelphia, USA, 2005.

[84] V. Kobla, D. DeMenthon, and D. Doermann. Identification of sports videos using replay, text, and camera motion features. In *SPIE Conference on Storage and Retrieval for Media Databases*, volume 3972, pages 332–343, 2000.

[85] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using MLC++: A machine learning library in C++. In *Proceedings of the 8th International Conference on Tools with Artificial Intelligence.*, pages 234–245, 1996. `http://www.sgi.com/Technology/mlc`.

[86] Y.-M. Kwon, C.-J. Song, and I.-J. Kim. A new approach for high level video structuring. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 773–776, 2000.

[87] H. Lee and A.F. Smeaton. Designing the user-interface for the Físchlár digital video library. *J. Digital Inform.*, 2(4), 2002.

[88] J.H. Lee. Analysis of multiple evidence combination. In *Proc. ACM SIGIR*, pages 267–276, 1997.

[89] S.-Y. Lee, S.-T. Lee, and D.-Y. Chen. *Automatic Video Summary and Description*, volume 1929 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, Berlin, 2000.

[90] D. Li, I.K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. *Pattern Recognition Letters*, 22(5):533–544, 2001.

[91] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Trans. Image Processing*, 9(1):147–156, 2000.

[92] R. Lienhart, C. Kuhmünch, and W. Effelsberg. On the detection and recognition of television commercials. In *IEEE Conference on Multimedia Computing and Systems*, pages 509–516, Ottawa, Canada, 1997.

[93] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Scene determination based on video and audio features. In *Proc. of the 6th IEEE Int. Conf. on Multimedia Systems*, volume 1, pages 685–690, 1999.

[94] C.-Y. Lin, B.L. Tseng, and J.R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.

[95] T. Lin and H.-J. Zhang. Automatic video scene extraction by shot grouping. In *Proceedings of ICPR '00*, Barcelona, Spain, 2000.

[96] G. Lu. Indexing and retrieval of audio: a survey. *Multimedia Tools and Applications*, 15:269–290, 2001.

[97] W.Y. Ma and B.S. Manjunath. NeTra: a toolbox for navigating large image databases. *Multimedia Syst.*, 7(3):184–198, 1999.

[98] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, USA, 1999.

[99] J.M. Martinez, R. Koenen, and F. Pereira. MPEG-7 the generic multimedia content description standard, part 1. *IEEE Multimedia*, april-june 2002.

[100] K. Minami, A. Akutsu, H. Hamada, and Y. Tomomura. Video handling with music and speech detection. *IEEE Multimedia*, 5(3):17–25, 1998.

[101] H. Miyamori and S. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 26–30, Grenoble, France, 2000.

[102] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(4):349–361, 2001.

[103] S. Moncrieff, C. Dorai, and S. Venkatesh. Detecting indexical signs in film audio for scene interpretation. In *IEEE Int'l Conf. Multimedia Expo*, pages 1192–1195, Tokyo, Japan, 2001.

[104] F. Nack and A.T. Lindsay. Everything you always wanted to know about MPEG-7: Part 1. *IEEE Multimedia*, 6(3):65–77, 1999.

[105] F. Nack and A.T. Lindsay. Everything you always wanted to know about MPEG-7: Part 2. *IEEE Multimedia*, 6(4):64–73, 1999.

[106] J. Nam, M. Alghoniemy, and A.H. Tewfik. Audio-visual content-based violent scene characterization. In *IEEE Int'l Conf. Image Processing*, volume 1, pages 353–357, Chicago, USA, 1998.

[107] J. Nam, A. Enis Cetin, and A.H. Tewfik. Speaker identification and video analysis for hierarchical video shot classification. In *IEEE Int'l Conf. Image Processing*, volume 2, pages 550–553, Washington DC, USA, 1997.

[108] M.R. Naphade. On supervision and statistical learning for semantic multimedia analysis. *J. Visual Commun. Image Representation*, 15(3):348–369, 2004.

[109] M.R. Naphade and T.S. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Trans. Multimedia*, 3(1):141–151, 2001.

[110] M.R. Naphade and T.S. Huang. Extracting semantics from audiovisual content: The final frontier in multimedia retrieval. *IEEE Trans. Neural Networks*, 13(4):793–810, 2002.

[111] M.R. Naphade, I.V. Kozintsev, and T.S. Huang. A factor graph framework for semantic video indexing. *IEEE Trans. Circuits and Systems for Video Technology*, 12(1):40–52, 2002.

[112] M.R. Naphade, J.R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[113] H.T. Nguyen, M. Worring, and A. Dev. Detection of moving objects in video using a robust motion similarity measure. *IEEE Trans. Image Processing*, 9(1):137–141, 2000.

[114] L. Nigay and J. Coutaz. A design space for multimodal systems: concurrent processing and data fusion. In *INTERCHI'93 Proceedings*, pages 172–178, Amsterdam, The Netherlands, 1993.

[115] D.W. Oard. The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3):141–178, 1997.

[116] H. Pan, P. Van Beek, and M.I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *IEEE Int'l Conf. Acoust., Speech, Signal Processing*, 2001.

[117] N.V. Patel and I.K. Sethi. Audio characterization for video indexing. In *Proceedings SPIE on Storage and Retrieval for Still Image and Video Databases*, volume 2670, pages 373–384, San Jose, USA, 1996.

[118] N.V. Patel and I.K. Sethi. Video classification using speaker identification. In *IS&T SPIE, Proceedings: Storage and Retrieval for Image and Video Databases IV*, San Jose, USA, 1997.

[119] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, USA, 1988.

[120] A.K. Peker, A.A. Alatan, and A.N. Akansu. Low-level motion activity features for semantic characterization of video. In *IEEE Int'l Conf. Multimedia Expo*, New York City, USA, 2000.

[121] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Int'l Conf. Computer Vision and Pattern Recognition*, Seattle, USA, 1994.

[122] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *Int'l J. Comput. Vision*, 18(3):233–254, 1996.

[123] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2004.

[124] S. Pfeiffer, S. Fischer, and W. Effelsberg. Automatic audio content analysis. In *ACM Multimedia*, pages 21–30, Boston, USA, 1996.

[125] S. Pfeiffer, R. Lienhart, and W. Effelsberg. Scene determination based on video and audio features. *Multimedia Tools and Applications*, 15(1):59–81, 2001.

[126] T.V. Pham and M. Worring. Face detection methods: A critical evaluation. Technical Report 2000-11, Intelligent Sensory Information Systems, University of Amsterdam, 2000.

[127] J.C. Platt. Probabilities for SV machines. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[128] Praja, December 2001. `http://www.praja.com`.

[129] G.M. Quénot, D. Moraru, L. Besacier, and P. Mulhem. CLIPS at TREC-11: Experiments in video retrieval. In E.M. Voorhees and L.P. Buckland, editors, *Proc. 11th Text REtrieval Conference*, volume 500-251 of *NIST Special Publication*, Gaithersburg, USA, 2002.

[130] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[131] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(1):23–38, 1998.

[132] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for TV baseball programs. In *ACM Multimedia*, pages 105–115, Los Angeles, USA, 2000.

[133] Y. Rui, T.S. Huang, and S. Mehrotra. Constructing table-of-content for videos. *Multimedia Systems, Special section on Video Libraries*, 7(5):359–368, 1999.

[134] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Trans. Circuits Syst. Video Technol.*, 8(5):644–655, 1998.

[135] E. Sahouria and A. Zakhor. Content analysis of video using principal components. *IEEE Trans. Circuits and Systems for Video Technology*, 9(8):1290–1298, 1999.

[136] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, USA, 1983.

[137] C. Saraceno and R. Leonardi. Identification of story units in audio-visual sequences by joint audio and video processing. In *IEEE Int'l Conf. Image Processing*, Chicago, USA, 1998.

[138] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, and S. Satoh. Video OCR: Indexing digital news libraries by recognition of superimposed caption. *Multimedia Syst.*, 7(5):385–395, 1999.

[139] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. *IEEE Multimedia*, 6(1):22–35, 1999.

[140] D.D. Saur, Y.-P. Tan, S.R. Kulkarni, and P.J. Ramadge. Automated analysis and annotation of basketball video. In *SPIE's Electronic Imaging conference on Storage and Retrieval for Image and Video Databases V*, volume 3022, pages 176–187, San Jose, USA, 1997.

[141] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *IEEE Computer Vision and Pattern Recognition*, Hilton Head, USA, 2000.

[142] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Int'l J. Comput. Vision*, 56(3):151–177, 2004.

[143] A. Th. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga. Ontology based photo annotation. *IEEE Intelligent Systems*, 16(3):66–74, 2001.

[144] F.J. Seinstra, C.G.M. Snoek, D. Koelma, J.M. Geusebroek, and M. Worring. User transparent parallel processing of the 2004 NIST TRECVID data set. In *Int'l Parallel Distrib. Processing Symposium*, Denver, USA, 2005.

[145] K. Shearer, C. Dorai, and S. Venkatesh. Incorporating domain knowledge with video and voice data analysis in news broadcasts. In *ACM Int'l Conf. Knowledge Discovery and Data Mining*, pages 46–53, Boston, USA, 2000.

[146] J. Shim, C. Dorai, and R. Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *IEEE Int'l Conf. Pattern Recognition*, pages 618–620, 1998.

[147] A.F. Smeaton. Large scale evaluations of multimedia information retrieval: The TRECVid experience. In *CIVR*, volume 3569 of *LNCS*, pages 19–27. Springer-Verlag, 2005.

[148] A.F. Smeaton, W. Kraaij, and P. Over. The TREC VIDeo retrieval evaluation (TRECVID): A case study and status report. In *Proc. RIAO 2004*, Avignon, France, 2004.

[149] A.F. Smeaton, P. Over, and W. Kraaij. TRECVID: Evaluating the effectiveness of information retrieval tasks on digital video. In *ACM Multimedia*, New York, USA, 2004.

[150] A.F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–180, August 1996.

[151] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(12):1349–1380, 2000.

[152] J.R. Smith and S.-F. Chang. Visually searching the web for content. *IEEE Multimedia*, 4(3):12–20, 1997.

[153] C.G.M. Snoek. *The Authoring Metaphor to Machine Understanding of Multimedia.* PhD thesis, Univ. of Amsterdam, 2005.

[154] C.G.M. Snoek, J.C. van Gemert, J.M. Geusebroek, B. Huurnink, D.C. Koelma, G.P. Nguyen, O. de Rooij, F.J. Seinstra, A.W.M. Smeulders, C.J. Veenman, and M. Worring. The MediaMill TRECVID 2005 semantic video search engine. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2005.

[155] C.G.M. Snoek and M. Worring. Multimedia event-based video indexing using time intervals. *IEEE Trans. Multimedia*, 7(4):638–647, 2005.

[156] C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools Applicat.*, 25(1):5–35, 2005.

[157] C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, and F.J. Seinstra. The MediaMill TRECVID 2004 semantic video search engine. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2004.

[158] C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, F.J. Seinstra, and A.W.M. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(10):1678–1689, October 2006.

[159] C.G.M. Snoek, M. Worring, and A.G. Hauptmann. Learning rich semantics from news video archives by style analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2(2):91–108, May 2006.

[160] C.G.M. Snoek, M. Worring, D.C. Koelma, and A.W.M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *IEEE Trans. Multimedia*, 2007. In press.

[161] C.G.M. Snoek, M. Worring, J. van Gemert, J.M. Geusebroek, D. Koelma, G.P. Nguyen, O. de Rooij, and F. Seinstra. Mediamill: Exploring news video archives based on learned semantics. In *Proceedings of ACM Multimedia*, Singapore, 2005.

[162] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.M. Geusebroek, and A.W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of ACM Multimedia*, Santa Barbara, USA, October 2006.

[163] R.K. Srihari. Automatic indexing and content-based retrieval of captioned images. *IEEE Computer*, 28(9):49–56, 1995.

[164] M. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval of Image and Video Databases III*, pages 381–392. SPIE Press vol. 2420, 1995.

[165] G. Sudhir, J.C.M. Lee, and A.K. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *IEEE International Workshop on Content-Based Access of Image and Video Databases, in conjunction with ICCV'98*, Bombay, India, 1998.

[166] H. Sundaram and S.-F. Chang. Determining computable scenes in films and their structures using audio visual memory models. In *Proceedings of the 8th ACM Multimedia Conference*, Los Angeles, CA, 2000.

[167] M. Szummer and R.W. Picard. Indoor-outdoor image classification. In *IEEE Int'l Workshop Content-based Access Image Video Databases*, Bombay, India, 1998.

[168] C. Taskiran, J.-Y. Chen, A. Albiol, L. Torres, C.A. Bouman, and E.J. Delp. *ViBE*: A compressed video database structured for active browsing and search. *IEEE Trans. Multimedia*, 6(1):103–118, 2004.

[169] B.T. Truong and S. Venkatesh. Determining dramatic intensification via flashing lights in movies. In *IEEE Int'l Conf. Multimedia Expo*, pages 61–64, Tokyo, Japan, 2001.

[170] B.T. Truong, S. Venkatesh, and C. Dorai. Automatic genre identification for content-based video categorization. In *IEEE Int'l Conf. Pattern Recognition*, Barcelona, Spain, 2000.

[171] S. Tsekeridou and I. Pitas. Content-based video parsing and indexing based on audio-visual interaction. *IEEE Trans. Circuits and Systems for Video Technology*, 11(4):522–535, 2001.

[172] B.L. Tseng, C.-Y. Lin, M.R. Naphade, A. Natsev, and J.R. Smith. Normalized classifier fusion for semantic visual concept detection. In *IEEE Int'l Conf. Image Processing*, volume 2, pages 535–538, Barcelona, Spain, 2003.

[173] A. Vailaya, M.A.T Figueiredo, A.K. Jain, and H.-J. Zhang. Image classification for content-based indexing. *IEEE Trans. Image Processing*, 10(1):117–130, 2001.

[174] A. Vailaya and A.K. Jain. Detecting sky and vegetation in outdoor images. In *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases VIII*, volume 3972, San Jose, USA, 2000.

[175] A. Vailaya, A.K. Jain, and H.-J. Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31:1921–1936, 1998.

[176] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, USA, 2th edition, 2000.

[177] J. Vendrig and M. Worring. Systematic evaluation of logical story unit segmentation. *IEEE Transactions on Multimedia*, 4(4):492–499, 2002.

[178] E. Veneau, R. Ronfard, and P. Bouthemy. From video shot clustering to sequence segmentation. In *Proceedings of ICPR '00*, volume 4, pages 254–257, Barcelona, Spain, 2000.

[179] Virage, December 2001. `http://www.virage.com`.

[180] H.D. Wactlar, M.G. Christel, Y. Gong, and A.G. Hauptmann. Lessons learned from building a terabyte digital video library. *IEEE Computer*, 32(2):66–73, 1999.

[181] Y. Wang, Z. Liu, and J. Huang. Multimedia content analysis using both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6):12–36, 2000.

[182] T. Westerveld. Image retrieval: Content versus context. In *Content-Based Multimedia Information Access, RIAO 2000 Conference*, pages 276–284, Paris, France, 2000.

[183] I.H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with java implementations.* Morgan Kaufmann Publishers, 2000.

[184] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

[185] L. Wu, J. Benois-Pineau, and D. Barba. Spatio-temporal segmentation of image sequences for object-oriented low bit-rate image coding. *Image Communication*, 8(6):513–544, 1996.

[186] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and systems for segmentation and structure analysis in soccer video. In *IEEE Int'l Conf. Multimedia Expo*, pages 928–931, Tokyo, Japan, 2001.

[187] M.-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(1):34–58, 2002.

[188] M.M. Yeung and B.-L. Yeo. Video content characterization and compaction for digital library applications. In *IS&T/SPIE Storage and Retrieval of Image and Video Databases V*, volume 3022, pages 45–58, 1997.

[189] M.M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *Computer Vision and Image Understanding*, 71(1):94–109, 1998.

[190] H.-J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[191] H.-J. Zhang, S.Y. Tan, S.W. Smoliar, and Y. Gong. Automatic parsing and indexing of news video. *Multimedia Systems*, 2(6):256–266, 1995.

[192] T. Zhang and C.-C. Jay Kuo. Hierarchical classification of audio data for archiving and retrieving. In *IEEE Int'l Conf. Acoust., Speech, Signal Processing*, volume 6, pages 3001–3004, Phoenix, USA, 1999.

[193] D. Zhong and S.-F. Chang. Structure analysis of sports video using domain models. In *IEEE Int'l Conf. Multimedia Expo*, pages 920–923, Tokyo, Japan, 2001.

[194] Y. Zhong, H.-J. Zhang, and A.K. Jain. Automatic caption localization in compressed video. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(4):385–392, 2000.

[195] W. Zhou, A. Vellaikal, and C.-C. Jay Kuo. Rule-based video classification system for basketball video indexing. In *ACM Multimedia*, Los Angeles, USA, 2000.

[196] W. Zhu, C. Toklu, and S.-P. Liou. Automatic news video segmentation and categorization based on closed-captioned text. In *IEEE Int'l Conf. Multimedia Expo*, pages 1036–1039, Tokyo, Japan, 2001.