A Survey on: Content-based Access to Image and Video Databases

Kjersti Aas Line Eikvil

March 1997

Contents

Chapter 1

Introduction.

Traditional database management systems are designed to deal with alphanumeric information only. While they offer excellent facilities for efficient searching, multi-user access, transaction management and crash recovery, they do not deal very well with non-alphanumeric data. Some recent systems can store an image as a binary large object and associate it with software to display the image, and some others allow the user to describe images with keywords, but few have the power to extract visual information from images and perform image searches based on these features.

The amount of visual information is increasing, and efficient tools for management, search and retrieval are now needed. In image and video databases the material has to be organized, stored, retrieved, visualized and distributed effectively and efficiently. Moreover, there may be a need for the image and video documents to be accessed both through their contents (i.e. visual features) as well as by any other information associated with the image. Like today's text databases can be searched with text queries, image and video databases should enable search with combined text and visual queries.

Efficient retrieval from image databases is useful for those who have a large collection of images, e.g. image providers, digital librarians, scientists dealing with satellite images, managers of large volumes of medical images, etc. A video retrieval system may be useful for video archiving, video editing, video production, interactive video and multimedia information systems used by broadcasting companies, film industries, security agencies, national archives, libraries and educational and training institutions.

Our survey will focus on applications and methods for content-based access to image and video archives. In addition, a total video database system requires efficient database management, communication protocols and user interfaces, but these topics will not be treated in this report. We start the survey by giving an introduction to content-based analysis for image and video databases in Chapter 2. In Chapter 3 we give some examples of application areas and a brief overview of existing systems. A few commercial systems for content-based retrieval from image archives exist, but for video archives there are currently

no such systems on the market. Most of the reviewed systems are therefore prototypes developed in research institutions. Chapter 4 and 5 go deeper into the different methods used for content-based image and video indexing and retrieval, and review different approaches tested for the problem.

Chapter 2

Image and video databases

Advances in computing and communication technology have increased the dominance of visual information. It will be difficult to cope with this explosion of visual information unless it is organized in a way that allows for fast search and retrieval. A similar situation occurred for textual data, and led to the creation of computerized database management systems. However, these systems will only work efficiently with alphanumeric information. This means it's time for a new type of database management systems which are able to handle information in the type of images, video, audio etc.

2.1 Image databases

A database of images may be useful for many applications where you would be interested in searching for images of specified objects. You may for instance want an image for illustrating an article about horseriding in a magazine. Then you would probably like to be able to specify that you need an image of a horse, and maybe also the kind of setting you would like; on a farm, in the city, in the woods etc. With the appropriate key words, you would make a search into the database for a suitable match. To be able to make such a search, it is necessary to have some information about the contents associated with the images.

The information about the images in the database could be entered by hand, but in most images there are literally hundreds of objects that could be referenced. Each image object will also have a long list of attributes; the horse is a black pony standing in a field of grass. Even worse, spatial relationships defining the layout of the image may be important in understanding the image contents. To enter all the information manually is difficult, and the result is that users enter only the minimum number of annotations required. Consequently, the resulting labels are not rich enough or consistent enough to cover all sorts of queries, and the database has to be re-annotated for each problem. Hence,

it would be desirable if some of the labelling could be performed automatically.

Through the automatic image annotation, the different regions of the images may be identified and information about shape, colour and texture may be extracted for each region. To find an image of a black horse out in a green field, you could search the image database for images of all black objects on a green background. To determine what the different regions represent, is a different and much more difficult task. Recognition of objects requires that the system knows what the objects look like. For 2-dimensional objects (e.g. characters) good descriptions of the objects can be obtained, but for 3-dimensional objects, which can attain any position, this is very difficult. Hence, unless the image has been manually indexed with the tag "horse", you would not be able to find images of horses automatically.

Sometimes, when searching an image database you may not be able to describe in words what you are looking for. You may for instance be looking for images of objects with a certain pattern or a certain texture. In these cases it would be convenient to be able to pose your query by using another image containing a pattern or texture similar to what you are looking for. To pose queries through images, would also be useful for instance for security applications with the need for access control systems. In these cases you may have a database of images of people cleared for access, a camera at an entrance captures images which are compared to the database.

2.2 Video databases

In its simplest form, a video database is created by digitizing the video, and inserting it into a database along with some textual information describing the contents. If the objective for the later retrieval from the video database is to fetch an entire video, where a short description of the contents or the name and date of the production is sufficient, the database and a video digitizer is all you will need. This may be the situation for a movie database, where you would generally be looking for a movie based on the title or the cast. However, if you want to search a video database containing documentaries, news clips or educational productions, you will probably need a more direct access to relevant material.

Consider the problem of TV news, which are often broadcasted at a particular time. If you are not in front of a TV at that time, the information becomes virtually inaccessible. And even if you are there to watch it, you will have to watch all the stories - with no possibility to select which to skip and which to study in more detail. However, suppose that each sequence was analysed and the information in it was stored in a database with pointers to the relevant frames. This database could then be used to view the news of choice to the desired depth and in the desired sequence.

With traditional video this has not been a possibility due to the video's sequential nature. To access a particular segment of interest on a tape, one must spend significant time

searching for the segment. Digital video presents new possibilities regarding direct access to selected frames or frame sequences.

Instead of representing the video as a set of ordered single frames, it is convenient to segment the video into shorter sequences of similar frames. A natural approach is to segment the video into shots, where a shot defines the sequence of frames captured during one run of the camera. A shot will then contain a sequence from *one* scene where the camera may have moved (pan, tilt, zoom). There will be a cut at the end of the shot, but no cuts within the shot. By analysing changes between consecutive frames in the digitized video, it is therefore possible to detect the shot boundaries automatically.

The next step would be to assign information about the contents to the shot segments. The shots are often represented by one or a few key-frames. The single key-frames can be analysed in much the same way as single images to obtain some description of the contents. In addition, the temporal information of the video, including camera motion, moving objects, changes in brightness and colour etc., can be exploited in the contents analysis of the frames. Speech recognition can also be very efficient for extracting information about the contents. In a recorded TV debate, word spotting may for instance be used to identify the debated topics. Audio analysis can also be used to extract characteristics of the video segments, by determining whether there is music, speech or silence.

When making queries into a video database, you will probably want the answer to your query presented in a somewhat different way than you would for a database of single images. Instead of just a collection of selected single frames, it might be convenient to be able to see the structure of the video document matching the query. By matching similar shots and using the temporal information of the video document, it is possibly to automatically build a graph representing the changes between scenes.

Chapter 3

Applications and Systems

In this chapter, we will first present some applications where there is a need for flexible and efficient retrieval from image or video databases. In the last section we will list some of the existing systems for content-based access to image and video databases. As only a handful commercial systems exist, the majority of the systems presented are non-commercial prototypes developed at universities or similar research institutions.

3.1 Image databases

For image archives the new possibilities given by content-based access lies in the ability to perform "queries-by-example", meaning that you can present an image of an object, pattern, texture etc., and fetch the images in the database that most resembles the example of the query.

3.1.1 Security applications

Face matching is useful for security applications. Imagine that a person stands at the access control in front of a system console with a built-in CCD-camera and identifies himself by a personal identification number or ID-card. The snapshot of the face may then be compared to the stored image corresponding to the person. This means that you need a database where you can perform the query by presenting a sample image.

Forensic investigations often require a search for faces in large facial databases. If for instance a victim or an eye witness describes the facial features of the perpetrator of a crime to a forensic composite technician, the technician can make a sketch of the criminal's face which may be matched against the faces in the database. Another example is the situation where a suspected criminal doesn't want to reveal his true identity to a criminal investigator. The investigator can then capture an image of the criminal's face

and compare it to the images in the facial database.

3.1.2 Catalogue searches

This is another application area where "query-by-example" is very important. You have an image of an object, a pattern or a texture, and you want to search for a similar object or pattern. Suppose that you want a new wallpaper in your apartment and you have some idea of which colour and/or pattern you want it to have. Then it would be convenient to be able to search an on-line catalogue of wallpaper types.

An archaeologist that has found an old tool during excavation and want to date it, or a botanist that has found a rare herb that she wants to know more about, may capture an image of what they have found and match it against images in a database.

3.1.3 Medical applications

In medicine you often get large databases of images, and the ability to search the database through images taken of the current patient with images from previous cases may be helpful. This could aid a physician in making a diagnosis, and would allow the physician to view images that are similar (along some dimension) to the image in question. Queries might specify that the similarity be in terms of blood vessel structure, the types of lesions in the image, the corresponding pathology, etc. Also, geometric properties of anatomical organs are hard to describe precisely in words; it may therefore be easier to use images.

3.2 Video applications

For video databases the new possibilities lie in the ability to access directly selected segments of the video. This opens up for applications where interactive and selective access to video documents is needed.

3.2.1 Interactive video

Interactive video covers applications where the user would want to select and access video scenes in any order. News-on-demand service is an example of an interactive video application. The users of such a service may want to select news items based on topic, and possibly on image contents, and may request the ability to decide in which sequence the news items should be played. Similarly could segments from educational video productions be selected interactively to adapt to each user's level.

3.2.2 Video archives

These are applications where you have a stock of recorded video and audio material, into which you would search for sequences on certain topics, from certain places or with certain persons. This is especially interesting to film or television directors and television journalists who need to retrieve video with certain contents - e.g. to find video shots of the new prime minister. Broadcasting corporations today have large archives of video tapes containing the corporation's complete video production. The contents of the videos are usually described in a free-text database, but there is no direct linkage between the text-based information and the video documents. Therefore, search for specific video segments from the archive is time-consuming and often requires specific knowledge of the archive.

3.2.3 Video libraries

From a video library, you would usually like to extract longer video documents telling a more complete story on a selected subject. For larger firms a video library describing different operations can be used for training of new personell. Museums may want to have a video library documenting rituals of ancient tribes, or the arts of old and forgotten hand-craft techniques.

3.2.4 Video-On-Demand services

VOD services allow the user to search for videos and movies stored on a digital video server. The entire movie is presumed to be the unit of interest and, thus, selection is mainly based on bibliographic data such as title, genre, or director. Users may start, stop, pause, or play back a part of the video, but they usually don't need functionality to skip parts of the video, change the sequence of scenes, or search for parts of the video with specific contents. This means, that the importance of content-based retrieval for this particular application is low.

3.3 World-Wide Web applications

Search engines such as Lycos, Alta Vista, and Yahoo index documents by their textual contents. These systems periodically scour the Web, record the text on each page, and condense it into compact and searchable indexes. Lycos for instance, simply extracts keywords using an algorithm that considers characteristics like word placement, word frequencies, etc. The user, by entering query terms and/or by selecting subjects, uses these search engines to more easily find the desired Web documents. Visual information (images, graphics, bitmaps, annotations and videos) is published both embedded in Web-documents and as stand-alone objects. In order to catalogue this information, an efficient

automated system is needed that regularly traverses the Web, detects visual information and processes it in a way that allows for efficient search and retrieval. A similar system for audio information is also needed.

There are currently no commercial systems for searching images and videos on the World Wide Web, but Smith et. al. [34] have recently developed an experimental image and video search engine.

3.4 Systems

During the last few years some content-based systems for image retrieval have been developed. There are currently very few commercial systems, and none of these handles content-based retrieval from video. A list of existing systems is given in Table 3.1. A more detailed description of each system is given in the following sections. Section 3.4.1 describes the systems for image retrieval, while Section 3.4.2 contains a description of systems for video retrieval.

Of the systems listed here, only two of the systems for content-based access to image databases are commercial products. And, as far as we have found, there are currently no commercial systems supporting content-based segmentation, archiving and retrieval from video databases. Most of the systems described here are prototypes developed at different universities.

System	*Comm*	*Video*	*Image*	*Firm*
QBIC	YES	-	YES	IBM (US)
VIRAGE	YES	-	YES	Virage Inc. (US)
CHABOT	NO	-	YES	UC Berkely (US)
CANDID	NO	-	YES	Los Alamos National Lab. (US)
Photobook	NO	-	YES	MIT (US)
VisualSEEK	NO	-	YES	Columbia Univ. (US)
CVEPS	NO	YES	-	Columbia Univ. (US)
JAKOB	NO	YES	-	Univ. of Palermo (IT)
VISION	NO	YES	-	Univ. of Kansas (US)
SWIM	NO	YES	-	Univ. of Singapore (SG)

Table 3.1: *Retrieval Systems.*

3.4.1 Systems for image retrieval

The QBIC system [10, 25] uses colour, shape, and texture to match images in a database with a user's query, which has the form "find more pictures like this one". The user can make a sketch of a shape, select colours and colour distributions from a colour wheel, and select textures from a predetermined selection. The system returns a ranked list of the best matches to the user's query.

The VIRAGE system [6] offers retrieval by colour, texture or shape, either singly or in combination. The system is designed as a series of independent modules, which can be combined in a variety of ways.

The CHABOT system [26] This system uses a relational database management system (POSTGRES) for storing and managing the images and their associated textual data. To implement retrieval from the database, CHABOT integrates the use of stored text and other data types with content-based analysis of the images to perform "concept queries".

The CANDID system [17] is a content-based system where the images in the database are characterized by a global signature that can represent features such as local textures, shapes, and colours. When a user performs a query to the database to retrieve images that are similar to a given example image, a global signature for that example image is first computed, and this signature is compared to the signatures of all images in the database. All database images are ranked with respect to their similarity to the query image.

Photobook [28] works by comparing features associated with images, not the images themselves. These features are in turn the parameter values of particular models fitted to each image. These models are commonly colour, texture, and shape, though the system will work with features from any model. Photobook includes FourEyes, an interactive learning agent which selects and combines models based on examples from the user. This makes Photobook different from tools like QBIC and VIRAGE, which support search on various features but offer little assistance in actually choosing one for a given task.

VisualSEEK [8, 35] is a fully automated content-based image query system which allows users to search images by colours and spatial layout. It includes a Java-based Web interface for interactive visual content specification, integrated visual and textual search and performance evaluation.

3.4.2 Systems for video retrieval

CVEPS [8] is a software prototype of a video indexing and manipulation system which supports automatic video segmentation, video indexing based on key frames or objects, and compressed video editing.

JAKOB [18] splits the video documents into shots using a shot extractor. A few representative frames are selected from each shot and described in terms of colour and texture contents. Motion features related to short sequences are also computed to give a dynamic description. When a query, direct or by example, is put to the system, the query module interprets it and arranges the query parameters for the match. The most similar shots are rendered. The user can browse the resulting shots and she can iterate the query, changing if necessary the query parameters (only on the selected shots).

VISION [19] is a system in which video processing and speech recognition are integrated. The videos are automatically segmented into a number of logically meaningful video clips by a two-step algorithm based on video and audio contents. A subtitle decoder and a word-spotter is being incorporated into the system to extract textual information to index video clips by their contents.

SWIM [48] is an integrated system for parsing, retrieval, and browsing. The video is segmented into shots and each shot is abstracted into key-frames. Visual features, such as colour and texture, are used to represent the contents of key-frames. In addition, temporal information is obtained from variations among key-frames from the same shot, and camera movements. Indexing is then supported by a clustering process which classifies key-frames into different visual categories. Retrieval may be based on the annotated index or low-level image features.

Chapter 4

Database creation

The contents of books are often organized into chapters, sections, and subsections. The contents of videos may also be organized into hierarchical structures in similar ways. Traditional film theory defines a three level hierarchy consisting of shots, scenes, and sequences. A *shot* represents a piece of video recorded in one contiguous operation; a *scene* is often defined as a complete, continuous chain of actions at one location, and a *sequence* represents a group of scenes linked together by a common thread of action.

To enable the search of video segments, the video clips have to be properly segmented into meaningful units. To make available the text attributes, and to provide audio-visual features, the contents of the video has to be characterized. Manual segmentation and characterization are time consuming, skill and knowledge dependent, and potentially limited to only the attributes that have text equivalents. However, computer-assisted content-based indexing is still a research area and currently a bottleneck in the productive use of video resources. This chapter gives an overview of the research on analysis of video contents, describing the two processes involved in the creation of a video database: *video parsing* and *video indexing*. By video parsing we mean the segmentation of a video program into elemental units and by video indexing the analysis of each unit for those features which will form the basis for the entries in an index structure.

4.1 Video Parsing

The process of making a video involves two major steps. The first is the production of shots, *shooting*, and the other is the assembling of shots into final cuts, *editing*. A *shot* is the sequence of images that is captured by a camera between the *record* and *stop* operations. The editing process consists of selecting shots, deciding their temporal order and implementing the edits between the shots. The latter may introduce additional frames into the final cut to produce fades and dissolves.

Video parsing research has so far been directed towards the problem of detecting shots, and Section 4.1.1 is a review of this work. Having segmented the video into shots, the next step is to find the appropriate data representation for a shot. This issue is treated in Section 4.1.2.

4.1.1 Shot segmentation

In the video retrieval literature, the boundaries between shots are commonly called *scene changes*, and the act of segmenting a video into shots is commonly called *scene change detection*. After the editing process, shots may be separated by cuts or edited transitions such as a fade or dissolve. This gives rise to two different types of shot boundaries: *abrupt* and *gradual*. In the first case, the change of contents occurs from one frame to the next, while in the other case, the change occurs over a longer period of time.

The vast majority of work on detection of shot boundaries are based on uncompressed data, but there are also recent efforts in performing the segmentation on compressed video. In this section we review the work done on both compressed and uncompressed data. Despite the fact that sound is a central element in video, only a few papers have addressed the issue of using both visual and audio information for segmentation. Some examples are given here.

Approaches for uncompressed video

There are two major categories of shot segmentation techniques for uncompressed video: *data driven* and *model driven*. The data driven approaches typically involve the application of various low level image processing operations, while the model driven approaches utilize the inherent structure of video.

Data driven techniques. The data driven approaches pose the problem of shot segmentation as one of detecting camera motion breaks in arbitrary image sequences. The solutions that have been presented typically involve the application of difference metrics to evaluate the changes between successive frames.

Nagasaka et al. [24] have evaluated a number of image processing measures for detecting shot boundaries in digital video. Their conclusion is that the best measurement is a sub-window-based histogram comparison between frames.

Zhang et al. [46] have also presented the evaluation of different image processing techniques for detection of cut edits. They used dual thresholds to detect gradual transitions like fades and dissolves.

Shararay [33] detects abrupt and gradual scene changes based on motion-controlled temporal filtering of the disparity between consecutive frames. Each block of the first frame

is matched to the second frame to find the best "fitting" region. A non-linear filter is then used to generate a global match value. Gradual transition is detected by identification of sustained lowlevel increases in matched values.

Ardizzone et al. [3] introduce a neural architecture for scene cut detection. This method is fast, but it is not able to detect gradual shot boundaries.

Hsu et al. [16] model scene changes and activities as motion discontinuities. Characterization of activities is performed by considering the sign of the Gaussian and mean curvature of the spatio-temporal surfaces. Clustering and split-and-merge approaches are then used to segment the video.

Otsuji et al. [27] propose a projection detection filter for reliable video cut detection.

Model driven techniques. As opposed to the data driven techniques, the model driven techniques try to utilize the natural structure of the video. The approaches differ in what they try to model.

Hampapur et al. [12, 13] use models of edit effects. They design feature detectors for detecting three classes of edits; cuts, chromatic effects and spatial effects. The feature detectors are applied to the individual frames of the video, and each frame is classified to either "edit" or "shot".

Zhang et al. [45] have developed model-based tools which take advantage of prior knowledge of the video contents. The tools have been tested in a system that can segment a news program into shots and classify the shots as anchor-person shots, news shots, commercial breaks etc.

Swanberg et al. [38] used a language-based model to match the incoming video sequence with the expected grammatical elements of a news broadcast.

Approaches for compressed video

A typical video can be compressed down to 2-3 gigabytes, but may occupy over 200 gigabytes when uncompressed. If one was able to analyse video directly in compressed format, one would save both the auxiliary storage for decompressed data and the computational costs for decompression. Several methods have been proposed for scene change detection on compressed MPEG [22, 32, 20, 47, 41] and Motion JPEG [4] data. These methods have proved to be sufficiently accurate for segmentation of the majority of shots in a video sequence.

Before describing the approaches for MPEG data, a brief description on the MPEG standard can be useful. MPEG defines three different types of frames. *I-frames* (intracoded frames) are fully and independently encoded using a DCT (Discrete Cosine Transform) based scheme. *P-frames* (predictive frames) are coded relative to the previous I- or P-

frame by using motion vectors. *B-frames* (bi-directional frames) use the previous and the next I- or P-frame to code current frames. (For more on MPEG, see Appendix A.)

Meng et al. [22] use the variance of DC coefficients in I- and P-frames and motion vector information to characterize scene changes. Sethi et al. [32] use only the DC coefficients of I-frames to perform hypothesis testing using the luminance histogram, while Liu et al. [20] make use of only information in P- and B-frames to detect scene changes. Zhang et al. [47] detect abrupt transitions between shots by counting the number of valid motion vectors in P- or B-frames, while a full-frame approach is taken to detect gradual transitions. Yeo et al. [41] have developed algorithms to identify both abrupt and gradual scene transitions using the DC coefficients of an encoded (Motion JPEG or MPEG) video sequence.

Approaches using audio information

The audio track provides a very rich source of information which can be exploited to segment a video into more meaningful clips. For example, by using a speaker identification algorithm, the segmentation procedure could identify a session given by a lecturer in a conference as a complete video clip without cutting it into several segments based purely on scene changes. There are not very many researchers studying both vision and audio, but a few examples exist.

Li et al. [19] have developed a two-step segmentation algorithm which combines audio analysis with video-based segmentation methods. It performs the segmentation task in two steps: initial segmentation based on scene changes in video is followed by a merger based on features in the audio track. For the video segmentation they use pixel-by-pixel and colour histogram differences between successive frames. The audio-based merger put some of the segments resulting from the video segmentation back together using information about low energy sections of the audio track.

In a system described by Hauptmann et al. [14] the segmentation of video is based on both video and audio semantic primitives. First, the video is segmented into shots using optical flow and colour histogram analysis. Then, the audio part of a raw digitized video tape is fed through speech analysis routines, which produces a transcript of spoken text. The speech signal is also analysed for low energy sections (silence) that indicate acoustic paragraphs. The segment breaks produced by image processing are examined along with the boundaries identified by the speech analysis and an improved set of segment boundaries are heuristically derived to partition the video into shots. The transcript is processed to identify important keywords that can be used to index the shots.

Yow et al. [44] use the loudness of the audio track to detect highlights in a soccer match [44]. In soccer and other sports, exciting scenes are often punctuated by the commentators' loud, surprised and at times almost delirious voices. The level of excitement also correlate strongly with the loud cheers of spectators.

4.1.2 Shot representation

In the analysis of video it is, as we have seen, natural to segment the video into shots. Within the video database, it is also necessary to have a way of representing the shot to facilitate later searches and queries into the database. The problem is then to find the appropriate abstraction for a shot.

In the literature, there are two main approaches for abstracting a shot. In the first approach, called *key-frame extraction*, one or if there is much motion and action in the shot, a small number of representative frames are extracted. In the other approach, called *mosaicking*, the frames in the shot are superimposed on top of each other to obtain salient stills.

Key-frame extraction

The key-frame extraction process is usually rule-based. In the work by Zhang et al. [48], the first and last frames are always extracted from the shot. The number of keyframes in addition to these two is specified by the user. Ardizzone et al. [2] extract one frame for every second of the video segment.

Arman et al. [5] extract what they call Rframes. Each Rframe consists of a body, four motion tracking regions and shot length indicators. The body is a frame (typically the 10th) chosen from the video shot. The four motion tracking regions trace the motion of boundary pixels through time, providing the user with the sense of motion within the shot.

Mosaicking

Assuming that there is no independent motion (only camera motion) in a shot, then by finding the transformation between subsequent images, the frames can be superimposed on top of each other to obtain a new image. If there is independent motion, the moving object(s) have to be segmented from the background and the high-resolution image of the background can still be reconstructed - along with images and trajectories of the moving objects. Teodosi et al. [40] and Sawhney et al. [31] propose techniques for obtaining such still images.

Yow et al. [44] use salient stills to present a larger view of selected highlights in soccer games. The salient stills are constructed by superimposing spatially and temporally the individual frames of a shot on top of each other, taking into account the global camera actions (pan and zoom). An example is shown in Figure 4.1 (from [44]).

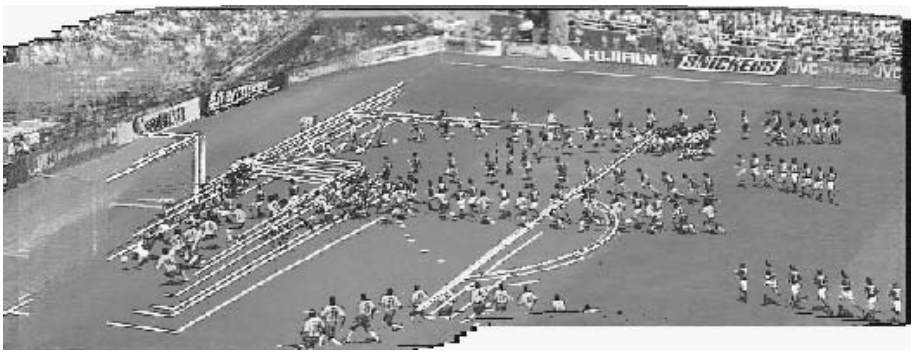

Figure 4.1: *Mosaic of the frames in a soccer sequence.*

4.2 Video Indexing

When the video has been partitioned into shots and the shots have been abstracted into keyframes, the next step is to compute features, from which the shot can be indexed, compared and classified. There are two major categories of features: *primitive* and *logical*. Primitive image features such as colour or shape can be extracted automatically or semi-automatically. Logical features are abstract attributes of the images such as the presence of specific types of buildings. Some logical features may be synthesized from primitive features, whereas others can only be obtained through considerable human involvement.

4.2.1 Primitive features

When it comes to primitive features, significant progress has been made by exploiting the results of computer vision research. Experimental and even a few commercial systems now exist which can provide content-based retrieval of still images on the basis of primitive features such as colour or shape. Some types of features, particularly colour, have proved to be more effective as retrieval cues than others. The primitive features can be divided into two parts which are complementary and can be used either separate or together: *key-frame features* and *temporal features*. The latter are important features of a video shot. For instance, temporal variance can be used to segment a news video, utilizing that the anchor person and interview shots always have small temporal variance, while other shots like sports and commercials usually have large temporal variance. A summary of the different features is given in Table 4.1, and descriptions of features are given in the following sections.

Colour

Retrieval by colour was first introduced by Swain et al. [37] and has proved to have excellent discrimination power in image retrieval systems [25, 26, 37].

Key-frame features	*Temporal features*
Colour	Camera operations
Texture	Temporal variance
Shape	Statistical motion features
	Object motion

Table 4.1: *Primitive features.*

The most common colour feature is the *colour histogram*. It is invariant under translation and rotation about the view axis, and changes only slowly under the change of angle of view, scale and occlusion [37].

In most images there are certain colours that are more frequent than others. These *dominant colours* can be easily identified from colour histograms of key-frames, and experiments have shown that using only a few dominant colours will not degrade the performance of colour image matching [37]. Another way of representing the colour distribution is to use moments (i.e. average, variance, and skewness) [36].

Texture

Although no precise definition of texture exists, certain intuitive concepts of texture can be defined. Texture is a property of a local region in an image and it involves the spatial distribution of the grey-levels or colours in the scene. In the description of texture, the following qualitative properties are important: Uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency and phase. Because there are so many qualitative properties which may describe the texture of a region, a texture model applicable to all types of images can not be established.

Retrieval by texture, comparing values of automatically extracted features as defined by Tamura [39], has been achieved with some degree of success, particularly when used in conjunction with colour matching [25].

Another model which is popular and effective in image retrieval is the SAR (Simultaneous Auto Regressive) model, which describes a pixel in terms of its neighbouring pixels. The multiresolution SAR (MSAR) model [21] describes textures at different resolutions in order to model different granularities.

Shape

Shape retrieval has proved to be a great challenge, despite considerable research into the topic. At present, most shape retrieval systems operate by using well established

image processing techniques to compute a set of shape features for identified regions of interest within a stored image. Two main types of shape features have been used; *global features* such as aspect ratio, circularity and moment invariants [25], and *local features* such as points of high curvature or sets of consecutive boundary segments. Evaluation results reported with IBM's QBIC system [9] show that the retrieval effectiveness of shape matching cannot compare with that of colour matching.

Temporal features

Key-frame features utilize only the spatial information and ignore to a large extent the temporal nature of a video clip. If only one key-frame is chosen to represent a shot, shots that are very similar may be judged to be different because the key-frames are different. Hence, in a video shot where object and camera motions are prominent, one key-frame is not sufficient. In addition to the key-frame features, features that represent the temporal characteristics of the shot should therefore be used.

Temporal features may be extracted from the analysis of camera motion. The camera operations (i.e. panning, tilting and zooming) can be detected by analysing either motion fields or spatio-temporal images.

Temporal variation of brightness and colours may be important, and can be represented by the mean and variance of average brightness and a few dominant colours calculated over all frames in shot.

Zhong et al. [49] state that the variance of colour histogram over all the frames in a shot can be used as a metric for the temporal variance of a shot. They also use certain types of statistical features derived from optical flows to describe motions in shots.

Ardizzone et al. [2] use motion features that are based on optical flow analysis.

4.2.2 Logical features

By logical features we mean derived or abstract attributes of an image, such as the presence of specified types of buildings or animals in a scene. Matching of this kind of features requires not just feature analysis, but a complex inferential process, making reference to some stored knowledge of object paradigms. Colour, texture or shape can not tell whether a particular image represents a dog as dogs appear in many shapes, sizes and colours.

There are many who consider that the only way to provide retrieval at this level is to employ human indexers to add suitable keywords [11]. However, there have been some attempts to solve the problem. Rabitti et al. [29] developed a system that was capable of retrieving line drawings of objects of a given type. The system analysed object drawings, labelling each with a set of possible interpretations and their probabilities. These were then used to derive likely interpretations of the scene within which they appeared. More

recently, Hermes et al. [15] have used similar techniques to derive logical descriptors from images of outdoor scenes.

If the query is as abstract as "find a romantic picture", the image cannot even be retrieved using a logical reasoning process, and it is difficult to see how retrieval at that abstract level could ever be automated.

4.2.3 Other features

As stated in the previous sections, visual features do not capture the full logical contents of the image, and some information like who created the image where and when, cannot be derived from the image. Hence, keywords are important and cannot be entirely replaced by visual information. In addition, searching by keywords is much faster than searching by visual features. To provide textual annotation, information from bibliographic data and audio tracks may be used. For example word spotting or speech recognition may be performed on the audio track to extract information about the contents. In addition, speech analysis can be performed to identify speakers; audio analysis can be used to classify the audio characteristics of a video segment (i.e. music, speech, silence). Photobook [28] is an example of a system that uses both content based features and text annotations for querying, while the system VISION [19] is both video and audio based.

4.3 Beyond the shots

An hour of video is typically composed of a few hundred shots. Thus, the number of images in a video library will rapidly increase with the size of the library. Searching large video databases then amounts to searching large image databases. For example, a video database of 100 hours resulting from four days of continuous broadcasting from one TV station contains about 10 million frames. Assuming that the key-frames (one for each shot) on average constitute about 0.5% of the data, the video is still represented by about 50,000 images. Hence, to reduce the amount of information to be presented to the user, the shots should be grouped into larger units, e.g. scenes.

To capture repetitions of shots with similar contents, time-constrained clustering [42] of the shots may be performed. The clustering process takes into account both visual characteristics and temporal locality of the shots. The latter to prevent that two shots that are far apart in time, but have similar contents, are clustered together. In [43] a scene transition graph (STG) is built from the clustering results and the temporal information associated with each shot. The nodes of the graph capture the core contents of the video while the edges represent the progress of the story from one scene to the next. Story units are finally extracted by finding cut edges in the graph. These units may be further analysed in a hierarchical manner. A STG depicting the story units and flow of story for a half-hour sitcom or movie segment can be laid out and displayed on a single computer

screen, meaning that the user can get a quick overview of this material.

An example of a Scene Transition Graph constructed using time-constrained clustering of video shots is shown in Figure 4.2 (from [42]). There are 27 story units found through segmentation of the graph at cut-edges. In the figure, the cut-edges are thicker than the other edges. Each cut-edge represents a unique point of transition from a shot in an earlier story unit to the consecutive shot of the next story unit. Each story unit represents a semantically meaningful unit of the video and approximates a scene.

A visual summary like the STG should be used for condensing the visual contents and representing the temporal flow of video that has underlying story structures. Examples include sitcoms, movies, news and documentaries. For sports videos, other types of visual summaries, such as salient stills (See Section 4.1.2), may be more appropriate.

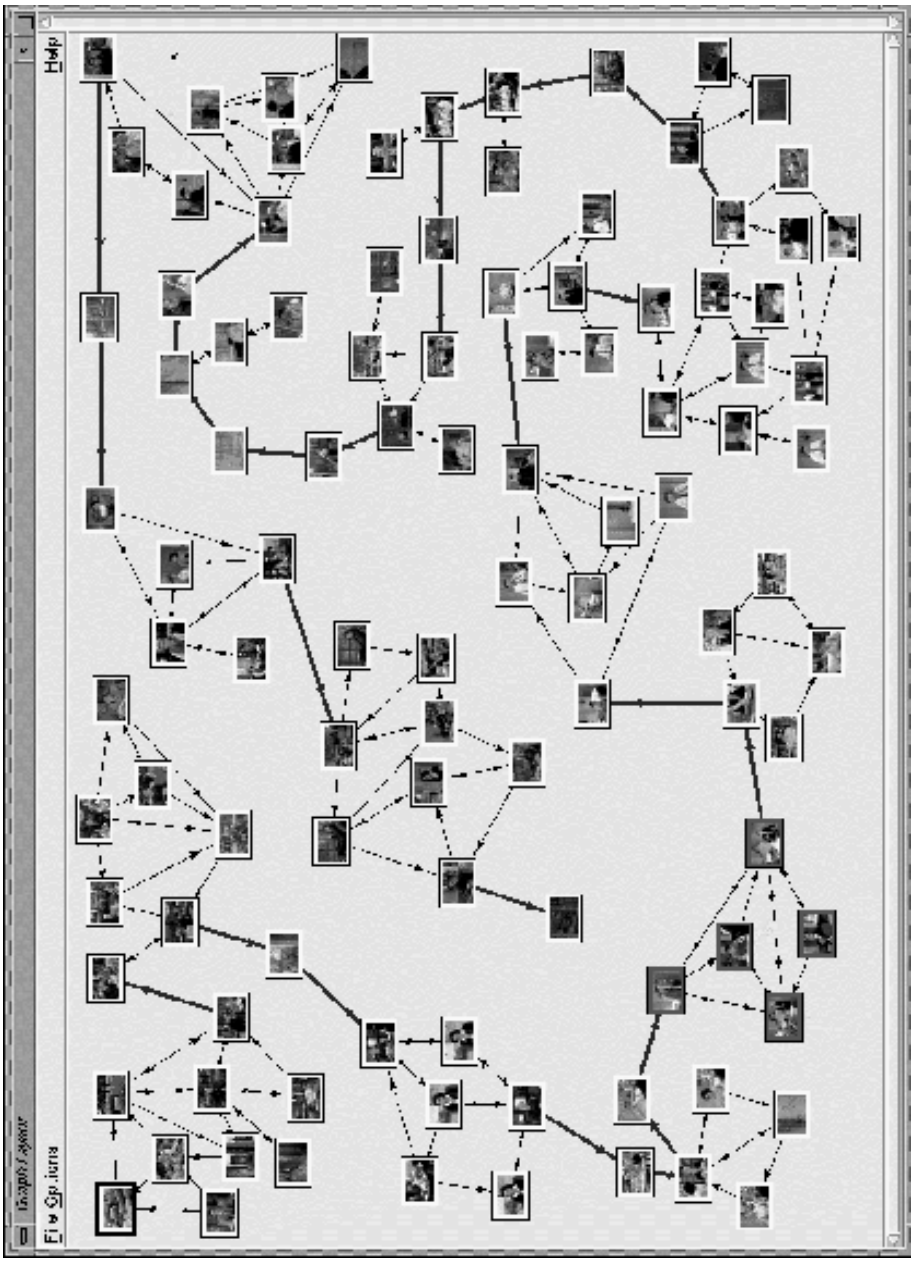

Figure 4.2: *Example of Scene Transition Graph. The nodes of the graph capture the core contents of the video while the edges represent the progress of the story from one scene to the next. Cuts between story units are marked with thicker edges.*

Chapter 5

Database queries

A video query is more complicated than a traditional query in text-based databases. For a video clip, there are visual and audio information, and temporal dynamics associated with this information. The prime concern of any video query process is that the query is natural and easy to formulate, that the user interface assists in a user-friendly way in the query formulation process, and that the query results are presented in an organized and sensible fashion. Another important aspect is that the search should be performed quickly.

This chapter is divided into three sections, corresponding to the three stages involved in a video query process [7]: *navigating, searching,* and *browsing.* The purpose of the navigating step, which is presented in Section 5.1, is to bring the candidate list of videos down to a manageable size. A way of achieving this is to categorize the videos when creating the database. Section 5.2 treats the searching step, which is the main stage of the query process. Various search keys can be used: text, visual and audio information. The result of the search phase is a series of video clips, and in the browsing phase, which is presented in Section 5.3, high-level overviews of the visual contents in the candidate videos are presented to the user.

5.1 Navigating

Video may be classified into categories like news, sports, entertainment etc. If such a categorization is available, the user can, as the first step in a query process, ask for the desired category. The initial query should be formulated in textual form, meaning that the advances that have been made in recent years in the area of text retrieval can be utilized.

A categorization of video may create problems for certain video items. It is for instance not clear whether news about a sporting event belong to the category of *news* or the category of *sports*. This means that certain videos may be classified into multiple categories or the

user will have to select multiple categories when searching.

5.2 Searching

The searching step is the main stage of the query process. The result of the search is a list of candidate videos and the ultimate goal of the search is to make this list as short as possible without missing the video(s) of interest.

Rowe et al. [30] collected sample queries by surveying a variety of users. Three types of indexes that can be used to answer the types of queries users would want to ask were identified:

- **Bibliographic data.** This category includes information about the video (e.g. title, abstract, subject, genre) and the individuals involved in the production of the video (e.g. producer, director, cast).
- **Structural data.** This category includes information about the structure of the video (e.g. the number of shots, the lengths of each shot, temporal relations between shots).
- **Contents data.** This category includes information based on the contents of the video (e.g. key-frame sequences, detecting objects and actors in scenes).

A video library should support all three types of queries addressed above, and given the multimedia nature of video source material it is important that the retrieval of the data should be multimedia-based. Text is important for all three types of queries, but especially to retrieve bibliographic information. To retrieve videos based on their contents, visual features may give the best result. Table 5.1 summarizes how the modalities that are available for video can be used for different types of queries.

	Text	*Visual*	*Audio*
Bibliographic data	x		
Structural data	x	x	x
Contents data	x	x	x

Table 5.1: *Different modalities are used for different types of queries.*

5.2.1 Query by textual information

Almost all video documents have some text associated with them, and using text-based search as a first step in a video query serves as a good search filter. The text can be

bibliographic information (the date of creation, the title, the source, etc.) or it may contain information about the structure (number of shots) or the contents (recognized spoken words and objects) of the video. The text information in the database may be entered by hand or it may be derived from automatic analysis of the video.

5.2.2 Query by visual information

Once a video has been abstracted to key-frames, search becomes a matter of identifying those key-frames from the database which are similar to the query. There are three main types of queries for retrieval of shots: The first, *query by image example*, allows users to specify a complete image or an image region as the query key. Specific images can be retrieved based on the similarity with the input image itself or the image features derived from the input image.

The second type is *feature-based image retrieval*. Visual features like colour and texture are extracted in the indexing stage and compared in the search stage to find similar images. The formulation of input features to the search engine can be provided by a drawing or sketch from the user or by a selection from system templates.

In the last type of query, *semantic retrieval*, the query is based on derived or abstract attributes of the image (the presence of specified types of buildings or animals in a scene). This kind of matching requires not just feature analysis, but a complex inferential process. Usually, the semantic information is provided by users in the indexing stage.

5.2.3 Query by audio information

Sound is a central element in video, but there are very few researchers studying both vision and audio. However, the research results from speech recognition and speaker identifivation could be very useful for video/audio database applications. For example, rather than having to listen to an entire audio document to find the sections of interest, the user can use speaker change markers to jump to the next speaker. If a talker is known, the user can choose to listen to only those portions of the audio spoken by that talker. An argument between two speakers can be detected by noting a rapid alternation between them. Word spotting can be used to find discussions of topics of interest by searching for user-specified keywords.

5.3 Browsing

Browsing the contents in video is a highly important feature in a video query. This is because textual descriptions can only supply a limited part of the attributes of the video.

It is often hard to formulate a query in text equivalents and the choices of wordings are subjective and highly dependent on the individual.

In the browsing phase, good high-level overviews of the visual contents in the candidate videos should be available. A user should be able to get a quick and brief understanding and browse through many videos in a relatively short period of time. Ideally, the user should also have random access to any point of any video. This means that a *visual summary*, i.e. a condensed abstraction and presentation of the contents in a video segment is necessary, or at least very desirable.

Today's standard technique for browsing is *storyboard browsing*, in which the video information is condensed into meaningful snapshots. Early browsers like the one developed at Apple [23] divide the video into equal length segments and denote the first frame of each segment as the key frame. Such browsers do not detect scene transitions and they may both miss important information and display repetitive frames.

Many video documents have story structures which are reflected in the visual contents, and this should be utilized by the browser. The fundamental unit of the production of video is the shot which captures continuous action. A scene is usually composed of a small number of interrelated shots that are unified by location or dramatic incident. Feature films are often divided into three acts, each of which consists of a half-dozen scenes. The act-scene-shot decomposition forms a hierarchy for understanding the story. News footage also have a similar structure. A news program is divided into stories, each of which typically starts with a common visual cue signalling the start of a new story. Each story in turn will contain several shots and perhaps multiple scenes. Thus, at the lower levels of the hierarchy, a scene may consist of alternating shots of the two main characters or an interviewer and an interviewee. At the higher levels of abstraction, an act in a movie or a story in a news show may be signalled by a visual motif, such as an establishing shot of the locale or a shot of the anchor in front of a title card.

A first step toward content-based browsing is to utilize the contents information obtained from video parsing including segment boundaries, camera operations and key-frames. Zhang et al. [48] have designed a hierarchical browser. At the top of the hierarchy, a whole video is represented by N pictures, each corresponding to a segment consisting of an equal number of consecutive shots. As one descends through the hierarchy, the focus is on smaller groups of shots, single shots, the key-frames of a shot, and finally a sequence of frames represented by one keyframe.

In [48], the shots at high levels are grouped only according to their sequential relations, and not their contents. As a result, though random access is provided, a user has to go down to the second or third level to get a sense of the contents of all shots in a group. The hierarchical scene transition graph described in Section 4.3 is a further step toward content-based browsing. It permits rapid nonlinear browsing and navigation of programs in digital libraries. A user can get a quick overview by looking at the graph, and then navigate to a particular story unit of interest. She can zoom into the story contents, and

select the individual nodes to look at the constituent video shots. She can go on to view any shots and listen to audio tracks.

Chapter 6

Summary

Visual information has always been an important source of knowledge. With the advances in computing and communication technology, this information, in the type of digital images and digital video, is highly available also through the computer. To be able to cope with what seems to be an explosion of visual information, an organization of the material which allows for fast search and retrieval is required. This calls for systems which in some way can provide content-based handling of the visual information. In this survey we have looked at the current status of content-based access to image and video databases with regard to applications, existing systems, and research.

An image retrieval system is necessary for users that have large collections of images, e.g. image providers, digital librarians, scientists dealing with satellite images, managers of large volumes of medical images, etc. For image archives the new possibilities given by content-based access lies in the ability to perform "queries-by-example", meaning that you can present an image of an object, pattern, texture etc., and fetch the images in the database that most resemble the example of the query.

During the last few years some content-based systems for image retrieval have been developed, and a few of these are commercially available. These systems generally offer retrieval by colour, texture or shape, and smart combinations of these features can help you in finding the image you are looking for. However, how to automatically handle logical queries, like finding all images containing a dog, is still very much a matter of research. Abstract queries, like search for gloomy pictures or romantic pictures, are even more difficult, and we can not see how automatic retrieval at this level can ever be achieved.

A video retrieval system is useful for video archiving, video editing, video production, interactive video, and multimedia information systems. For video databases the new possibilities lie in the ability to access directly selected segments of the video. This opens up for applications where interactive and selective access to video documents is needed. Construction of an effective video library requires a set of tools which support the tasks of parsing, indexing, retrieval, and browsing of video documents. Automatic methods for

shot and scene segmentation are then essential.

A few prototype systems providing content-based access to video databases exist, but currently there are no commercial systems available. The prototype systems provide tools for automatic segmentation of video into shots, and these shots can be represented by extracted key-frames. Content-based access to key-frames is enabled through low level features as colour, texture or shape, in the same way as for still images. Through clustering of similar shots and scenes, a graphical representation of the flow of the video can be obtained to give the user a quick overview of the video document. However, there is still a long way before it is possible to condense the course of events in a video automatically. Research on recognition of action and combination of audio and video, may bring us somewhat closer to the solution of this problem.

Content-based access to images, video and audio is in many respects a new way of thinking regarding access to non-textual information, and it opens up to a lot of new applications which have not previously been possible.

Appendix A

MPEG compression

The international standard for video compression, MPEG (Moving Pictures Expert Group) has been widely used both for storage and transmission purposes. It takes advantage of the fact that adjacent frames are highly correlated by coding most of the frames as differences relative to neighbouring frames. The MPEG standard defines three types of frames; I-, P- and B-frames. These three types of frames are combined to form a group of frames.

The I-frame (intra-coded frame) contains most information. It is coded using only information present in the frame itself and it's very similar to the JPEG frame. The coding model is described in Section A.1. The P-frame (predictive-coded frame) is coded relative to the previous I- or P-frame and the B-frame (bidirectionally predictive-coded frame) uses the previous and the next I- or P-frame to code current frames. The coding of P- and B-frames is based on a technique called *motion compensation*, which is treated in Section A.2.

A.1 Intra-frame coding

The method used to encode I-frames is based on JPEG compression and it mainly includes the following steps:

1. 2D Discrete cosine transform (DCT)
2. Quantization
3. Run-length encoding

The DCT transforms 8x8 blocks of pixels to their frequency components. After converting the input block of 8x8 pixels into an 8x8 block of DCT coefficients, quantization is applied to the DCT block, i.e. the DCT coefficients are quantized into a certain amount of levels.

The combination of DCT and quantization results in many of the frequency coefficients being zero. To take maximum advantage of this, the coefficients are organized in a zigzag order to produce long runs of zeroes before they are run-length encoded. Finally, Huffmann encoding is applied to the run-length encoded sequence.

A.2 Inter-frame coding

P-frames obtain predictions from temporally preceding I- or P-frames in the sequence, whereas B-frames obtain predictions from the nearest preceding and/or upcoming I- or P-frames in the sequence. Different regions of the B-frames may use different predictions, and may be predicted from preceding frames (forward prediction), upcoming frames (backward prediction), both (bi-directional prediction), or neither (intra-frame coding). Similarly P-frames may be predicted from preceding frames or use no prediction.

The forward prediction, backward prediction and bi-directional prediction techniques are based on a process called *motion compensation*. When a region of the frame is compressed by motion compensation, the output is a set of *motion vectors* and *error terms*. The motion vectors describe the direction and amount of motion between the reference frame and the frame being coded, while the error terms contain the contents differences between the two frames.

Not all the information in a frame can be predicted from a previous frame. Consider a scene in which a door opens. The visual details of the room behind the door cannot be predicted from a previous frame in which the door was closed. This situation can be handled by coding these parts of the frame without reference to adjacent frames, that is, by intra-frame coding. However, in many cases it can also be handled by predicting backward in time, assuming that an upcoming frame already has been coded and transmitted.

Bibliography

[1] A. Aigran, P. Joly. *The automatic real-time analysis of file editing and transition effects and its applications.* Computer and Graphics, Vol. 18, pp. 93–103, January 1994.

[2] E. Ardizzone, M. La Cascia, D. Molinelli. *Motion and Colour-Based Video Indexing and Retrieval.* Proceedings ICPR'96, Vol C, pp. 135–139.

[3] E. Ardizzone, G.A.M. Gioiello, M. La Cascia, D. Molinelli. *A real-time neural approach to scene cut detection* In Proceedings of IS&T/SPIE - Storage & Retrieval for Image and Video Databases IV, San Jose 1996.

[4] F. Arman, A. Hsu, M. Chiu. *Image processing on compressed data for large video databases.* In Proceedings of First ACM International Conference on Multimedia, pp. 267–272, August 1993.

[5] F. Arman et al. *Content-Based Browsing of Video Sequences.* Proceedings of ACM Int. Conf. on Multimedia'94, San Fransisco, Oct. 1994.

[6] J. R. Bach et al. *The Virage image search engine: An open framework for image management.* In Storage and Retrieval for Still Image and Video Databases IV, SPIE, February 1996.

[7] R. M. Bolle, B. L. Yeo, M. M. Yeung. *Video Query: Beyond the keywords.* IBM Research Report, RC 20586, October 1996.

[8] S.-F. Chang, J. R. Smith, J. Meng. *Efficient Techniques for Feature-Based Image/Video Access and Manipulation.* In Proceedings, 33rd Annual Clinic on Library Applications of Data Processing: Digital Image Access and Retrieval, Illinois, March, 1996.

[9] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz. *Efficient and effective querying by image content.* Journal of Intelligent Information Systems, Vol 3, pp. 231–262, 1994.

[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steale, P. Yanker. *Query by Image and Video Content: the QBIC System.* IEEE Computer, Sept. 1995.

[11] V.N. Gudivada, V.V. Raghavan. *Content-Based Image Retrieval Systems.* IEEE Computer-Special Issue, Vol. 28, pp. 18–62, September 1995.

[12] A. Hampapur, T. Weymouth, R. Jain. *Digital Video Segmentation.* Proceedings of ACM Multimedia'94, ACM Press.

[13] A. Hampapur, R. Jain, T. E. Weymouth. *Production Model Based Digital Video Segmentation.* Multimedia Tools and Applications, Vol. 1, pp. 1–38, 1995.

[14] A. G. Hauptmann, M. A. Smith. *Text, Speech and Vision for Video Segmentation: The Informedia Project.* In AAAI-95 Fall Symposium on Computational Models for Integrating Language and Vision, November, 1995.

[15] T. Hermes et al. *Image retrieval for information systems.* In Storage and Retrieval for Image and Video Databases III, Proc. SPIE 2420, pp. 394–405, 1995.

[16] P. R. Hsu, H. Harashima. *Detecting scene changes and activities in video databases.* In ICASSP 94, Vol. 5, pp. 33–36, April 1994.

[17] P. M. Kelly, M. Cannon, D. R. Hush. *Query by image example: the comparison algorithm for navigating digital image databases (CANDID) approach.* In SPIE Proc. Storage and Retrieval for Image and Video Databases III, pp.238–249, 1995.

[18] M. La Cascia, E. Ardizzone. *JACOB: Just a content-based query system for video databases.* In ICASSP, Atlanta, May 1996.

[19] W. Li, S. Gauch, J. Gauch, K. M. Pua. *VISION: A Digital Video Library.* The 1st ACM Digital Libraries(ACM DL'96). Bethesa, MD, March 23–25, 1996.

[20] H. C. Liu, G. L. Zick. *Scene decomposition of MPEG compressed video.* In Digital Video Compression: Algorithms and Technologies, Vol. SPIE 2419, pp. 26–37, February 1995.

[21] J. Mao, A. K. Jain. *Texture classification and Segmentation Using Multiresolution Simultanous Autoregressive Models.* Pattern Recognition Vol. 25, No. 2, pp. 173–188, 1992.

[22] J. Meng, Y. Juan, S. F. Chang. *Scene change detection in a MPEG compressed video sequence.* In Digital Video Compression: Algorithms and Technologies, Vol. SPIE 2419, pp. 14–25, February 1995.

[23] M. Mills, J. Cohen, Y. Y. Wong. *A magnifier tool for video data.* In Proceedings of ACM Computer Human Interface (CHI), May 1992.

[24] A. Nagasaka, Y. Tanaka. *Automatic Video Indexing and Full Video Search for Object Appearances.* In W.E. Knuth, editor, IFIP Trans., Visual Database Systems II, pp.113–128, 1992.

[25] W. Niblack et al. *The QBIC project: Querying images by colour, texture and shape.* IBM Research Report RJ-9203.

[26] V. E. Ogle, M. Stonebraker. *Chabot: Retrieval from a relational database of images.* IEEE Computer, Vol. 28, No. 9, September 1995.

[27] K. Otsuji and Y. Tomumara. *Projection detecting filter for video cut detection.* In Proceedings of First ACM International Conference on Multimedia, pp. 251–257, August, 1993.

[28] A. Pentland, R.W. Picard, S. Sclaroff. *Photobook: Tools for Contentbased Manipulation of Image Databases.* Proceedings SPIE: Storage and Retrieval of Image and Video Databases II, No. 2185, pp. 34–47, San Jose, Feb. 1994.

[29] F. Rabitti, P. Stanchev. *GRIM_DBMS: a graphical image database management system.* In Visual Database Systems, pp. 415–430, Elsevier, Amsterdam.

[30] L. A. Rowe, J. S. Boreczky, C. A. Eads. *Indices for user acess to large video databases.* In Storage and Retrieval for Image and Video Databse II, IS&T/SPIE, pp. 150–161, February 1994.

[31] H. S. Sawhney, S. Ayer, M. Gorkani. *Model-based 2D&3D dominant motion estimation for mosaicking and video representation.* Technical report, IBM Almaden Research Lab, December 1994.

[32] I. K. Sethi, N. Patel. *A statistical approach to scene change detection.* In Storage and Retrival for Image and Video Databases III, Vol. SPIE 2420, pp. 329–338, February 1995.

[33] B. Shararay. *Scene change detection and content-based sampling of video sequences.* In Digital Video Compression: Algorithms and Technologies, Vol. SPIE 2419, pp. 2–13, February 1995.

[34] J. R. Smith, S.-F. Chang. *An Image and Video Search Engine for the World-Wide Web.* In Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases V, San Jose, CA, February 1997. IS&T/SPIE.

[35] J. R. Smith, S.-F. Chang. *VisualSEEk: a fully automated contentbased image query system.* ACM Multimedia '96, November, 1996.

[36] M. Stricker, M. Orengo. *Similarity of Colour Images.* Proc. IS&T/SPIE. Conf. on Storage and Retrieval for Image and Video Databases III, San Jose, CA, 1995.

[37] M. J. Swain, D. H. Ballard. *Colour Indexing.* International Journal of Computer Vision, Vol. 7, pp. 11–32, 1991.

[38] D. Swanberg, C. F. Shu, R. Jain. *Knowledge guided parsing in video databases.* In Storage and Retrieval for Image and Video Databases, Vol. SPIE 1908, pp. 13–25, 1993.

[39] H. Tamura, S. Mori, T. Yamawaki. *Texture Features Corresponding to Visual Perception.* IEEE Trans. on Syst., Man, and Cybern. Vol. 6, No. 4, pp. 460–473, 1979.

[40] L. Teodosio. W. Bender. *Salient Video Stills: Content and Context Preserved.* ACM Multimedia, Anaheim, CA, 1993.

[41] B. L. Yeo, B. Liu. *Rapid scene analysis on compressed videos.* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 5, pp. 533-544, December 1995.

[42] M. M. Yeung, B. L. Yeo. *Time-constrained Clustering for Segmentation of Video into Story Units.* Proceedings ICPR'96, Vol C, pp. 375–380.

[43] M. M. Yeung, B. L. Yeo, W. Wolf, B. Liu. *Video browsing using clustering and scene transitions on compressed sequences.* In Multimedia Computing and Networking, Vol. SPIE 2417, pp. 399–413, Feb. 1995.

[44] D. Yow, B. L. Yeo, M. Yeung, B. Liu *Analysis and presentation of soccer highlights from digital video.* In Proceedings Second Asian Conference on Comp. Vis., 1995.

[45] H. J. Zhang, Y. H. Gong, S. W. Smoliar, S. Y. Yan. *Automatic Parsing of news video.* In Internatinal Conference on Multimedia Computing and Systems, pp. 45–54, 1994.

[46] H. Zhang, A. Kankanhalli, S. W. Smoliar. *Automatic partitioning on full-motion video.* Mulimedia Systems, Vol. 1, pp. 10–28, July 1993.

[47] H. J. Zhang, C. Y. Low, S. W. Smoliar, *Video parsing and browsing using compressed data.* Multimedia tools and applications, Vol. 1, pp. 89–111, Mars, 1995.

[48] H.J. Zhang, C. Y. Low, S. W. Smoliar, J. H. Wu. *Video Parsing, Retrieval and Browsing: An integrated and Content-Based Solution.* Proc. of ACM Multimedia'95, San Francisco, Nov.7–9, 1995.

[49] D. Zhong, H. J. Zhang, S. F. Chang. *Clustering Methods for Video Browsing and Annotation.* SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, February 1996.