17

Video Analysis and Indexing

Hui Ding, Wei Pan and Yong Guan
Capital Normal University
China

1. Introduction

In recent years, hardware technologies and standards activities have matured to the point that it is becoming feasible to transmit, store, process, and view video signals that are stored in digital formats, and to share video signals between different platforms and application areas. Manual annotation of video contents is a tedious, time consuming, subjective, inaccurate, incomplete, and - perhaps more importantly - costly process. Over the past decade, a growing number of researchers have been attempting to fulfil the need for creative algorithms and systems that allow (semi-) automatic ways to describe, organize, and manage video data with greater understanding of its semantic contents.
In this chapter, we study methods for automatic segmentation in digital videos. We also demonstrate their suitability for indexing and retrieval. The first section is related to granularity and addresses the question: what to index, and reviews existing techniques in video analysis and indexing. The modalities and their analysis are presented in section 2. Pre-processing, feature extraction and representation are discussed in Section 3. We discuss the actual process of developing models for semantic concepts in Section 4. Finally, directions for future research and conclusions are presented in Section 5.

1.1 What is a video database?

Multimedia data, such as text, audio, images and video, is rapidly evolving as the main form for the creation, exchange, and storage of information in the modern era. Numerous types of videos are created in the world. Conservative estimates state that there are more than 6 million hours of video already stored and this number grow at a rate of about 10 percent a year (Hjelsvold, 1995). Projections estimate that by the end of 2010, 50 percent of the total digital data stored worldwide will be video and rich media (Brown, 2001). The amount of video information stored in archives worldwide is huge. Because the amount of video we all must manage is growing at an exponential rate, it makes the creation of the video databases all the more essential. And consequently, the design and implementation of video database systems has become a major topic of interest.
What is a Video Database? The video database is a storehouse of information on the various aspects related to the videos. The video database is one of the most accessed types of databases. One way to think about a digital video system is in the context of a database. Such a database contains a large collection of information elements of a certain granularity. These elements are described according to a range of attributes and can be accessed according to the intent of the user.

Source: Digital Video, Book edited by: Floriano De Rango,
ISBN 978-953-7619-70-1, pp. 500, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

The combination of the growing number of applications for video-intensive products and solutions - from personal video recorders to multimedia collaborative systems - with the many technical challenges behind the design of contemporary video database systems. Progress in visual information analysis has been fostered by many research fields (Fig. 1), particularly: (text-based) information retrieval, image processing and computer vision, pattern recognition, multimedia database organization, multidimensional indexing, data mining, machine learning, and visualization, psychological modeling of user behavior, man-machine interaction, among many others.

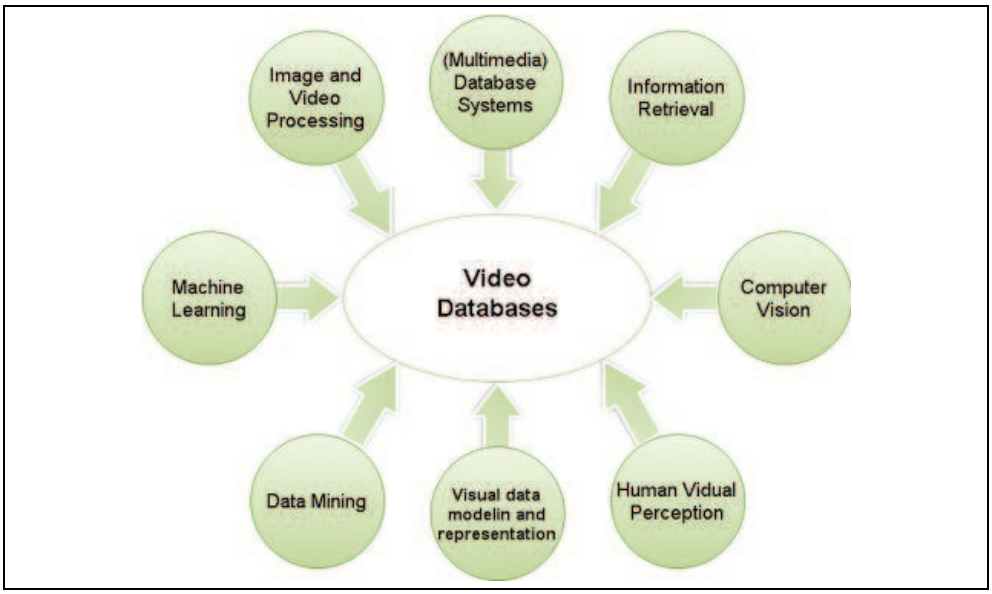

Fig. 1. Visual information retrieval blends together many research disciplines

1.2 A short overview of related work

However, raw video data by itself has limited usefulness, since it takes far too long to search for the desired piece of information within a videotape repository or a digital video archive. Attempts to improve the efficiency of the search process by adding extra data (henceforth called metadata) to the video contents do little more than transferring the burden of performing inefficient, tedious, and time-consuming tasks to the cataloguing stage. The challenging goal is to devise better ways to automatically store, catalog, and retrieve video information with greater understanding of its contents. Researchers from various disciplines have acknowledged such challenge and provided a vast number of algorithms, systems, and papers on this topic during recent years.

As in conventional information retrieval, the purpose of a Visual Information Retrieval (VIR) system is to retrieve all the images (or image sequences, video) that are relevant to a user query while retrieving as few non-relevant images as possible. The emphasis is on the retrieval of information as opposed to the retrieval of data. Similarly to its text-based counterpart a visual information retrieval system must be able to interpret the contents of the documents (images) in a collection and rank them according to a degree of relevance to the user query.

The interpretation process involves extracting (semantic) information from the documents (images) and using this information to match the user needs (Baeza-Yates & Ribeiro-Neto, 1999). VIR systems can be classified in three main generations, according to the attributes used to search and retrieve a desired image or video file:

- First-generation VIR systems: Use query by text, allowing queries such as "all pictures of red Ferraris" or "all images of Van Gogh's paintings". They rely strongly on metadata, which can be represented either by alphanumeric strings, keywords, or full scripts.
- Second-generation (CB)VIR systems: Support query by content, where the notion of content, for still images, includes, in increasing level of complexity: perceptual properties (e.g., color, shape, texture), semantic primitives (abstractions such as objects, roles, and scenes), and subjective attributes such as impressions, emotions and meaning associated to the perceptual properties. Many second-generation systems use content-based techniques as a complementary component, rather than a replacement, of text-based tools.
- Third-generation (SB)VIR systems (Zhang, 2006): The semantic gap has dictated that solutions to image and video indexing could only be applied in narrow domains using specific concept detectors. When recent advances in data-driven image and video analysis on the one hand, and top-down ontology engineering and reasoning on the other hand, are combined we are reaching the point where the semantic gap can be bridged for many concepts in broad domains such as film, news, sports and personal albums. The research in these disciplines has traditionally been within different communities.

2. Techniques of video analysis

In order to organize the video database, a segmentation algorithm is applied to the video data to obtain video intervals. Parsing, which segments the video stream into generic clips? These clips are the elemental index units in the database. Ideally, the system decomposes individual images into semantic primitives. On the basis of these primitives, a video clip can be indexed with a semantic description using existing knowledge-representation techniques.

2.2 Architecture for video data

Generally, data represents the facts or observations on any phenomenon that is worth formulating and recording. Even though there can be many data structures for the same application, there are some fundamental features that the structure should reflect. Unfortunately, current video characterization techniques rely on image representations based on low-level visual primitives (such as color, texture, and motion), while practical and computationally efficient, and fails to capture most of the structure that is relevant for the perceptual decoding of the video.

Video metadata are data used for the description of video data, including the attributes and the structure of videos, video content and relationships that exist within a video, among videos and between videos and real world objects. A key aspect for the definition of a video metadata model is the imposed video structure. Video data are often represented either as a set of still images that contain salient objects or as clips that have specific spatial (e.g. color, position etc.) or temporal (e.g. motion) features or are related to semantic objects (Li & Özsu,

1997) (Dağtas et al., 2000) (Al-Khatib et al., 1999). More sophisticated approaches are either a hierarchical representation of video objects (Analyti & Christodoulakis, 1995) (Yeo & Yeung, 1997) (Kyriakaki, 2000) based on their structure, or an event-based approach that represents a video object as a set of (non-contiguous, even overlapping) video segments called strata or temporal cohesions (Hacid et al, 2000) that correspond to individual events.

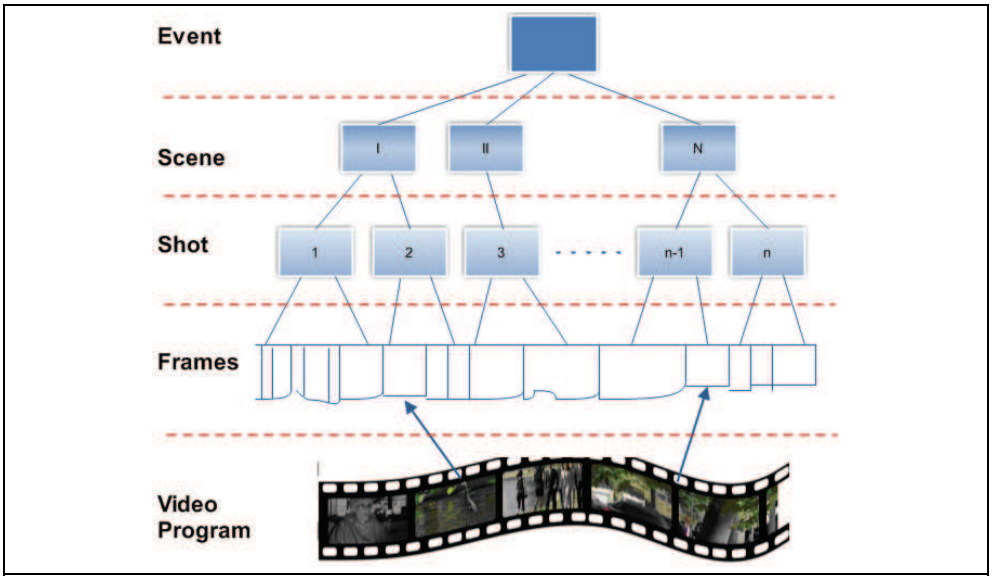

Fig. 2. General structure of a video data

The imposed structure of a video is shown in Fig. 2: A video is represented as an instance of the (Video) Program class and is comprised of a set of Stories. Each story is a logical section of the video object and is further divided in a set of Scenes. A scene represents an event and may be either composite or simple: a simple scene contains a simple event and is comprised of video Shots, while a composite one contains a composite event and is comprised of other scenes. Shots are sets of "similar" consequent frames that are usually recognized using automatic segmentation techniques.

2.3 Temporal segmentation of digital video

Following Fig. 2, temporal video segmentation is the essential first step towards automatic analysis of digital video sequences. Its goal is to divide the video stream into a set of meaningful and manageable segments (shots). Shots are considered to be the primitives for higher level content analysis. A video shot is defined as a series of interrelated consecutive frames taken contiguously by a single camera and representing a continuous action in time and space.

According to whether the transition between shots is abrupt or not, the shot boundaries are categorized into two types: cuts and gradual transitions. Abrupt transitions (cuts) are simpler. In the case of shot cuts, the content change is usually large and easier to detect than the content change during a gradual transition (Lienhart, 1999). Many metrics and the classification algorithms have been proposed in the litera-ture during the past decade, e.g.,

starting with the initial work of (Zhang et al., 1993) (Hampapur et al., 1994) to some recent work of (Smeaton, 2007), (Teng & Tan, 2008). In the following paragraphs of this section, we shall briefly review the metrics used and the classification strategy adopted.

2.3.1 Pixels difference metrics

Metrics classified as pixels difference metrics (PDM) are based on the intensity variations of pixels in equal position in consecutive frames. Temporal segmentation techniques using interframe differences based on color are conceptually similar, but they are not very popular as they have a greater computational burden and almost the same accuracy of their intensity based counterpart.

A basic PDM is the sum of the absolute differences of intensity of the pixels of two consecutive frames (Ford et al., 1997). In particular, indicating with Y(x, y, j) and Y(x, y, k) the intensity of the pixels at position (x, y) and frames j and k, the metric can be expressed in the following way:

$$\Delta f = \sum_x \sum_y \left| Y(x,y,j) - Y(x,y,k) \right| \tag{1}$$

where the summation is taken all over the frame.

The same authors propose other metrics based on first and second order statistical moments of the distributions of the levels of intensity of the pixels. Indicating with μ_k and σ_k respectively the values of mean and standard deviation of the intensity of the pixels of the frame k, it is possible to define the following interframe metric between the frames *j* and *k*:

$$x_2 + y_2 = z_2 \lambda = \frac{\left[\frac{\sigma_j + \sigma_k}{2} + \left(\frac{\mu_j - \mu_k}{2} \right)^2 \right]^2}{\sigma_j \sigma_k} \tag{2}$$

This metric has been also used in (Dugad et al., 1998), (Sethi & Patel, 1995), and is usually named likelihood ratio, assuming a uniform second order statistic.

2.3.2 Histogram difference metrics

Metrics classified as histograms difference metrics (HDM) are based on the evaluation of the histograms of one or more channels of the adopted color space. As it is well known, the histogram of a digital image is a measure that supplies information on the general appearance of the image. With reference to an image represented by three color components, each quantized with 8 bit/pixel, a three-dimensional histogram or three one-dimensional histograms can be defined. Although the histogram does not contain any information on the spatial distribution of intensity, the use of interframe metrics based on image histograms is very popular because it represents a good compromise between the computational complexity and the ability to represent the image content.

In recent years several histogram-based techniques have been proposed (Gargi et al., 2000), (Dailianas et al., 1995); some of them are based only on the luminance channel, others on the conversion of 3-D or 2-D histograms to linear histograms (Ardizzone & Cascia, 1997). In what follows some of the most popular HDM metrics are reviewed. In all the equations *M* and *N* are respectively the width and height (in pixels) of the image, *j* and *k* are the frame

indices, L is the number of intensity levels and $H[j,i]$ the value of the histogram for the *i-th* intensity level at frame *j*. A commonly used metric is the bin-to-bin difference, defined as the sum of the absolute differences between histogram values computed for the two frames:

$$f_{db2b}(j,k) = \frac{1}{2MN}\sum_{i=0}^{L-1}\left|H(j,i) - H(k,i)\right| \tag{3}$$

The metric can easily be extended to the case of color images, computing the difference separately for every color component and weighting the results. For example, for a RGB representation we have:

$$f_{db2b}(j,k) = \frac{r}{s} f_{db2b}(j,k)^{(red)} + \frac{g}{s} f_{db2b}(j,k)^{(green)} + \frac{b}{s} f_{db2b}(j,k)^{(blue)} \tag{4}$$

where *r*, *g* and *b* are the average values of the three channels and *s* is:

$$s = \frac{r+g+b}{3} \tag{5}$$

Another metric is called intersection difference and is defined in the following way:

$$f_{d\,\mathrm{int}}(j,k) = 1 - \frac{1}{MN}\sum_{i=0}^{L-1}\min\left[H(j,i), H(k,i)\right] \tag{6}$$

In other approaches [28], the chi-square test has been used, which is generally accepted as a test useful to detect if two binned distributions are generated from the some source:

$$f_{dchi2}(j,k) = \sum_{i=0}^{L-1}\frac{\left[H(j,i) - H(k,i)\right]^2}{\left[H(j,i) + H(k,i)\right]^2} \tag{7}$$

Also the correlation between histograms is used:

$$f_{dcorr}(j,k) = 1 - \frac{\mathrm{cov}(j,k)}{\sigma_j \sigma_k} \tag{8}$$

where $\mathrm{cov}(j,k)$ is the covariance between frame histograms:

$$\mathrm{cov}(j,k) = \frac{1}{L}\sum_{i=0}^{L-1}\left[H(j,i) - \mu_j\right]\left[H(k,i) - \mu_k\right] \tag{9}$$

and μ_j and σ_j represent the mean and the standard deviation, respectively, of the histogram of the frame *j*:

$$\mu_j = \frac{1}{L}\sum_{i=0}^{L-1}H(j,i); \quad \sigma_j = \sqrt{\frac{1}{L}\sum_{i=0}^{L-1}\left[H(j,i) - \mu_j\right]^2} \tag{10}$$

All the metrics discussed so far are global, i.e., based on the histogram computed over the entire frame. Both PDM and HDM techniques are based on the computation of a similarity

measure of two subsequent frames and on comparison of this measure with a threshold. The choice of the threshold is critical, as too low threshold values may lead to false detections and too high threshold values may cause the opposite effect of missed transitions. To limit the problem of threshold selection, several techniques have been proposed.
It has been pointed out that the aim of temporal segmentation is the decomposition of a video in camera-shots. Therefore, the temporal segmentation must primarily allow to exactly locating transitions between consecutive shots. Secondarily, the classification of the type of transition is of interest. Basically, temporal segmentation algorithms are based on the evaluation of the quantitative differences between successive frames and on some kind of threshold. In general an effective segmentation technique must combine an inter-frame metric computationally simple and able to detect video content changes with robust decision criteria.

2.4 Techniques operating on compressed video

The Moving Picture Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio and their combination. So far MPEG has produced:

- MPEG-1, the standard for storage and retrieval of moving pictures and audio on storage media (approved Nov. 1992)
- MPEG-2, the standard for digital television (approved Nov. 1994)
- MPEG-4, the standard for multimedia applications
- MPEG-7 the content representation standard for multimedia information search, filtering, management and processing (to be approved July 2001).
- MPEG-21, the multimedia framework.

MPEG uses two basic compression techniques: 16×16 macroblock-based motion compensation to reduce temporal redundancy and 8×8 Discrete Cosine Transform (DCT) block-based compression to capture spatial redundancy. An MPEG stream consists of three types of pictures, I, P and B, which are combined in a repetitive pattern called group of picture (GOP).

- I (Intra) frames provide random access points into the compressed data and are coded using only information present in the picture itself. DCT coefficients of each block are quantized and coded using Run Length Encoding (RLE) and entropy coding. The first DCT coefficient is called DC term and is proportional to the average intensity of the respective block.
- P (Predicted) frames are coded with forward motion compensation using the nearest previous reference (I or P) pictures.
- B (Bi-directional) pictures are also motion compensated, this time with respect to both past and future reference frames

A well known approach to temporal segmentation in the MPEG compressed domain, useful for detecting both abrupt and gradual transitions, has been proposed in (Yeo & Liu, 1995) using the DC sequences. Since this technique uses I, P and B frames, a partial decompression of the video is necessary. The DC terms of I frames are directly available in the MPEG stream, while those of B and P frames must be estimated using the motion vectors and the DCT coefficients of previous I frames. This reconstruction process is computationally very expensive.

Both PDM and HDM metrics are suited as similarity measures, but pixel differences-based metrics give satisfactory results as DC images are already smoothed versions of the corresponding full images. Gradual transitions are detected through an accurate temporal analysis of the metric.

3. Content-based analysis of digital video

Video content analysis is a strongly multidisciplinary research area. The ever increasing amount of multimedia data creates a need for new sophisticated methods to retrieve the information one is looking for. Analysis of the ground truths provided for development data revealed that the important sections to be included in summarized video were of four types: shots containing camera motion, shots of people entering or leaving a scene, shots showing certain objects, and shots of distinct events. Since high-level features can be indicators of the relative importance of a particular video segment (Todd, 2005), appropriate features were extracted to capture these four types. The video analysis (Ojala et al., 2001) task consists in recovering the object shape, object texture, and object motion from the given video, shown in Fig. 3.

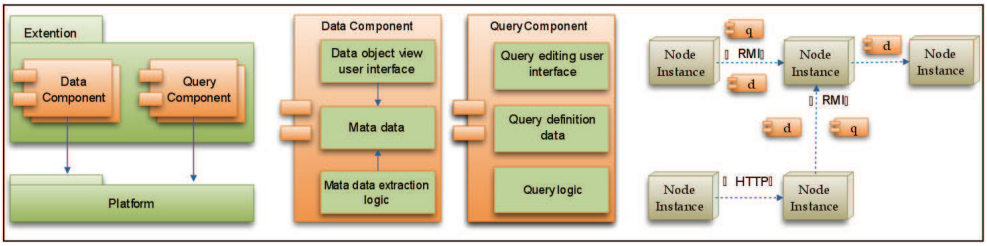

Fig. 3. Principal components of the CMRS architecture described using UML notation. (a) CMRS architecture comprises of a media independent platform and media specific extensions, which contain one or more Data and Query components. (b) Data and Query components encapsulate the data, user interface and operations related to physical media items and queries, respectively. (c) Runtime view of the platform.

3.1 Feature extraction and selection

Current efforts in the automatic video content analysis are directed primarily towards the decomposition of video streams into meaningful subsequences. By reducing the dimensionality of the input set correlated information is eliminated at the cost of a loss of accuracy (Addison, 2003). Dimensionality reduction can be achieved either by eliminating data closely related with other data in the set, or combining data to make a smaller set of features. The feature extraction techniques used in this study are principal components analysis and auto-associative neural networks. Feature selection is achieved by the use of genetic algorithms, sensitivity analysis.

3.1.1 Linear principal components analysis

Using the example of projecting data from two dimensions to one, a linear projection requires the optimum choice of projection to be a minimisation of the sum-of-squares error. This is obtained first by subtracting the mean $\bar{x}$ of the data set. The covariance matrix is

calculated and its eigenvectors and eigenvalues are found. The eigenvectors corresponding to the M largest eigenvalues are retained, and the input vectors x^n are subsequently projected onto the eigenvectors to give components of the transformed vectors z^n in the M-dimensional space. Retaining a subset $M < d$ of the basis vectors u_i so that only M coefficients z_i are used allows for replacement of the remaining coefficients by constants b_i. This allows each x vector to be approximated by an expression of the form:

$$\tilde{\chi} = \sum_{i=1}^{M} z_i u_i + \sum_{l=M+1}^{d} b_i u_i \tag{11}$$

Where u_i represents a linear combination of d orthonormal vectors.

3.1.2 Auto Associative Networks (AAN)

An auto associative network (AAN) consists of a multi-layer perception with d inputs, d outputs, and M hidden units with $M < d$. (Fausett, 1994). The targets used to train the network are the input vectors themselves, which means the network is attempting to map each input vector onto itself. Because the number of units in the middle layer is reduced, a perfect reconstruction of the input vectors may not always be possible. The network is then trained using a sum of squares error of the following form.

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{d}\left\{y_k(x^n) - x_k^n\right\}^2 \tag{12}$$

where N is the number of patterns in the sample and x_k^n represents the target value for the output unit k when the input vector is x^n. The error minimisation here performs a form of unsupervised training, even though we are using a supervised architecture as no independent target data is provided. Such networks perform a non-linear principal components analysis, which has the advantage of not using linear transformations.

3.1.3 Genetic algorithms

We use the Holland (Holland, 1975) algorithm. The algorithm can be expressed as follows.

1. Set generation counter I ← 0
2. Create initial population, Pop(i), by random generation of N-individuals
3. Apply objective function to the individual, record the value found (determines data fitness)
4. Increment next generation, I ←I +1
5. Select N individuals randomly from the previous population Pop (I-1) based on their fitness
6. Select R parents from new population to form new population to form new children by applying the genetic operators
7. Evaluate fitness of newly formed children by applying the objective function
8. If I < maximum number of generations to be considered, go to step (4)
9. Write out best solution found

3.1.4 Sensitivity analysis

Sensitivity analysis (SA) is the study of how the variation (uncertainty) in the output of a mathematical model can be apportioned, qualitatively or quantitatively, to different sources

of variation in the input of a model (Bishop, 1997). We conducted sensitivity analysis by treating each input variable in turn as if it were "unavailable" (Hunter et al., 2000). A neural network is trained using all of the input attributes, and the values of training and test set errors are produced. Afterwards the network is "pruned" of input variables whose training and verification errors are below the threshold. In this way variables can be assessed according to the deterioration effect they have upon network performance if removed.

3.2 Content comparison techniques

"Content-based" means that the search will analyze the actual contents of the frame image. Unfortunately, automatic indexing and feature extraction from digital video is even harder than still-image analysis. Several techniques are proposed to automatically segment digital video into scenes, shots and subshots based on color histogram, motion, texture and shape features (Volker, 1999). The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the frame image itself.

- *Color* represents the distribution of colors within the entire image. This distribution includes the amounts of each color.
- *Texture* represents the low-level patterns and textures within the image, such as graininess or smoothness. Unlike shape, texture is very sensitive to features that appear with great frequency in the image.
- *Shape* represents the shapes that appear in the image, as determined by color-based segmentation techniques. A shape is characterized by a region of uniform color.

3.2.1 Color

Color reflects the distribution of colors within the entire frame image. A color space is a mathematical representation of a set of colors. The three most popular color models are RGB (used in computer graphics); YIQ, YUV or YCbCr (used in video systems); and CMYK (used in color printing). However, none of these color spaces are directly related to the intuitive notions of hue, saturation, and brightness. This resulted in the temporary pursuit of other models, such as HIS and HSV, to simplify programming, processing, and end-user manipulation.

- RGB Color Space

The red, green, and blue (RGB) color space is widely used throughout computer graphics. Red, green, and blue are three primary additive colors (individual components are added together to form a desired color) and are represented by a three-dimensional, Cartesian coordinate system (Figure 3.1). The indicated diagonal of the cube, with equal amounts of each primary component, represents various gray levels. Table 1 contains the RGB values for 100% amplitude, 100% saturated color bars, a common video test signal.

	Nominal Range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	0 to 255	255	255	0	0	255	255	0	0
G	0 to 255	255	255	255	255	0	0	0	0
B	0 to 255	255	0	255	0	255	0	255	0

Table 1. 100% RGB Color Bars

The RGB color space is the most prevalent choice for computer graphics because color displays use red, green, and blue to create the desired color. However, RGB is not very efficient when dealing with "real-world" images. All three RGB components need to be of equal band-width to generate any color within the RGB color cube. The result of this is a frame buffer that has the same pixel depth and display resolution for each RGB component. Also, processing an image in the RGB color space is usually not the most efficient method. For these and other reasons, many video standards use luma and two color dif ference signals. The most common are the YUV, YIQ, and YCbCr color spaces. Although all are related, there are some differences.

- YCbCr Color Space

The YCbCr color space was developed as part of ITU-R BT.601 during the development of a world-wide digital component video standard. YCbCr is a scaled and offset version of the YUV color space. Y is defined to have a nominal 8-bit range of 16–235; Cb and Cr are defined to have a nominal range of 16–240. There are several YCbCr sampling formats, such as 4:4:4, 4:2:2, 4:1:1, and 4:2:0.

RGB - YCbCr Equations: SDTV

The basic equations to convert between 8-bit digital R´G´B´ data with a 16–235 nominal range and YCbCr are:

$$\begin{aligned} Y_{601} &= 0.2999R' + 0.587G' + 0.114B' \\ Cb &= -0.172R' - 0.399G' + 0.511B' + 128 \\ Cr &= 0.511R' - 0.428G' - 0.083B' + 128 \end{aligned} \tag{13}$$

$$\begin{aligned} R' &= Y_{601} + 1.371(Cr - 128) \\ G' &= Y_{601} - 0.698(Cr - 128) - 0.336(Cb - 128) \\ B' &= Y_{601} + 1.732(Cb - 128) \end{aligned}$$

When performing YCbCr to R´G´B´ conversion, the resulting R´G´B´ values have a nominal range of 16–235, with possible occasional excursions into the 0–15 and 236–255 values. This is due to Y and CbCr occasionally going outside the 16–235 and 16–240 ranges, respectively, due to video processing and noise. Note that 8-bit YCbCr and R´G´B´ data should be saturated at the 0 and 255 levels to avoid underflow and overflow wrap-around problems.

Table 2 lists the YCbCr values for 75% amplitude, 100% saturated color bars, a common video test signal.

	Nominal Range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
SDTV									
Y	16 to 235	180	162	131	112	84	65	35	16
Cb	16 to 240	128	44	156	72	184	100	212	128
Cr	16 to 240	128	142	44	58	198	212	114	128
HDTV									
Y	16 to 235	180	168	145	133	63	51	28	16
Cb	16 to 240	128	44	147	63	193	109	212	128
Cr	16 to 240	128	136	44	52	204	212	120	128

Table 2. 75% YCbCr Color Bars.

RGB - YCbCr Equations: HDTV

The basic equations to convert between 8-bit digital R´G´B´ data with a 16–235 nominal range and YCbCr are:

$$\begin{aligned} Y_{709} &= 0.213R' + 0.751G' + 0.072B' \\ Cb &= -0.117R' - 0.394G' + 0.511B' + 128 \\ Cr &= 0.511R' - 0.464G' - 0.047B' + 128 \end{aligned} \tag{14}$$

$$\begin{aligned} R' &= Y_{709} + 1.540(Cr - 128) \\ G' &= Y_{709} - 0.459(Cr - 128) - 0.183(Cb - 128) \\ B' &= Y_{709} + 1.816(Cb - 128) \end{aligned}$$

When performing YCbCr to R´G´B´ conversion, the resulting R´G´B´ values have a nominal range of 16–235, with possible occasional excursions into the 0–15 and 236–255 values. This is due to Y and CbCr occasionally going outside the 16–235 and 16–240 ranges, respectively, due to video processing and noise. Note that 8-bit YCbCr and R´G´B´ data should be saturated at the 0 and 255 levels to avoid underflow and overflow wrap-around problems.

Table 2 lists the YCbCr values for 75% amplitude, 100% saturated color bars, a common video test signal.

- HSI, HLS, and HSV Color Spaces

The HSI (hue, saturation, intensity) and HSV (hue, saturation, value) color spaces were developed to be more "intuitive" in manipulating color and were designed to approximate the way humans perceive and interpret color. They were developed when colors had to be specified manually, and are rarely used now that users can select colors visually or specify Pantone colors. These color spaces are discussed for "historic" interest. HLS (hue, lightness, saturation) is similar to HSI; the term lightness is used rather than intensity. The difference between HSI and HSV is the computation of the brightness component (I or V), which determines the distribution and dynamic range of both the brightness (I or V) and saturation(S). The HSI color space is best for traditional image processing functions such as convolution, equalization, histograms, and so on, which operate by manipulation of the brightness values since I is equally dependent on R, G, and B. The HSV color space is preferred for manipulation of hue and saturation (to shift colors or adjust the amount of color) since it yields a greater dynamic range of saturation.

3.2.2 Texture

Texture reflects the texture of the entire image. Texture is most useful for full images of textures, such as catalogs of wood grains, marble, sand, or stones. A variety of techniques have been developed for measuring texture similarity. Most techniques rely on comparing values of what are known as second-order statistics calculated from query and stored images. These methods calculate measures of image texture such as the degree of contrast, coarseness, directionality and regularity; or periodicity, directionality and randomness (Liu & Picard, 1996). Alternative methods of texture analysis for image retrieval include the use of Gabor filters and fractals (Kaplan, 1998). Gabor filter (or Gabor wavelet) is widely adopted to extract texture features from the images for image retrieval and has been shown to be very efficient. Manjunath and Ma (Manjunath & Ma, 1996) have shown that image

retrieval using Gabor features outperforms that using pyramid-structured wavelet transform (PWT) features, tree-structured wavelet transform (TWT) features and multiresolution simultaneous autoregressive model (MR-SAR) features.
Haralick (Haralick, 1979) and Van Gool (Gool et al., 1985) divide the techniques for texture description into two main categories: statistical and structural. Most natural textures can not be described by any structural placement rule, therefore the statistical methods are usually the methods of choice. One possible approach to reveal many of the statistical texture properties is by modelling the texture as an autoregressive (AR) stochastic process, using least squares parameter estimation. Letting s and r be coordinates in the 2-D coordinate system, a general causal or non-causal auto-regressive model may be written:

$$y(s) = \sum_{r \in N} \theta_r y(s-r) + e(s) \tag{15}$$

Where $y(s)$ is the image, θ_r are the model parameters, $e(s)$ is the prediction error process, and N is a neighbour set. The usefulness of this modelling is demonstrated with experiments showing that it is possible to create synthetic textures with visual properties similar to natural textures.

3.2.3 Shape

Shape represents the shapes that appear in the image. Shapes are determined by identifying regions of uniform color. In the absence of color information or in the presence of images with similar colors, it becomes imperative to use additional image attributes for an efficient retrieval. Shape is useful to capture objects such as horizon lines in landscapes, rectangular shapes in buildings, and organic shapes such as trees. Shape is very useful for querying on simple shapes (like circles, polygons, or diagonal lines) especially when the query image is drawn by hand. Incorporating rotation invariance in shape matching generally increases the computational requirements.

4. Semantic-based annotation for digital video

An annotation represents any symbolic description of a video, or an excerpt of a video. Semantic concepts do not occur in isolation. There is always a context to the co-occurrence of semantic concepts in a video scene. To locate and represent the semantic meanings in video data is the key to enable intelligent query. This section presents the methodology for knowledge elicitation and, in particular, introduces a preliminary semantic representation scheme that bridges the information gap not only between different media but also between different levels of contents in the media.

4.1 Framework of object-based video indexing

Object segmentation and tracking is a key component for new generation of digital video representation, transmission and manipulations. The schema provides a general framework for video object extraction, indexing, and classification. By video objects, here we refer to objects of interest including salient low-level image regions (uniform color/texture regions), moving foreground objects, and group of primitive objects satisfying spatio-temporal constraints (e.g., different regions of a car or a person). Automatic extraction of video objects at different levels can be used to generate a library of video data units, from which various

functionalities can be developed. For example, video objects can be searched according to their visual features, including spatio-temporal attributes. High-level semantic concepts can be associated with groups of low-level objects through the use of domain knowledge or user interaction.

As mentioned above, in general, it is hard to track a meaningful object (e.g., a person) due to its dynamic complexity and ambiguity over space and time. Objects usually do not correspond to simple partitions based on single features like color or motion. Furthermore, definition of high-level objects tends to be domain dependent. On the other hand, objects can usually be divided into several spatial homogeneous regions according to image features. These features are relatively stable for each region over time. For example, color is a good candidate for low-level region tracking. It does not change significantly under varying image conditions, such as change in orientation, shift of view, partial occlusion or change of shape. Some texture features like coarseness and contrast also have nice invariance properties. Thus, homogenous color or texture regions are suitable candidates for primitive region segmentation. Further grouping of objects and semantic abstraction can be developed based on these basic feature regions and their spatio-temporal relationship. Based on these observations, we proposed the following model for video object tracking and indexing (Fig. 4).

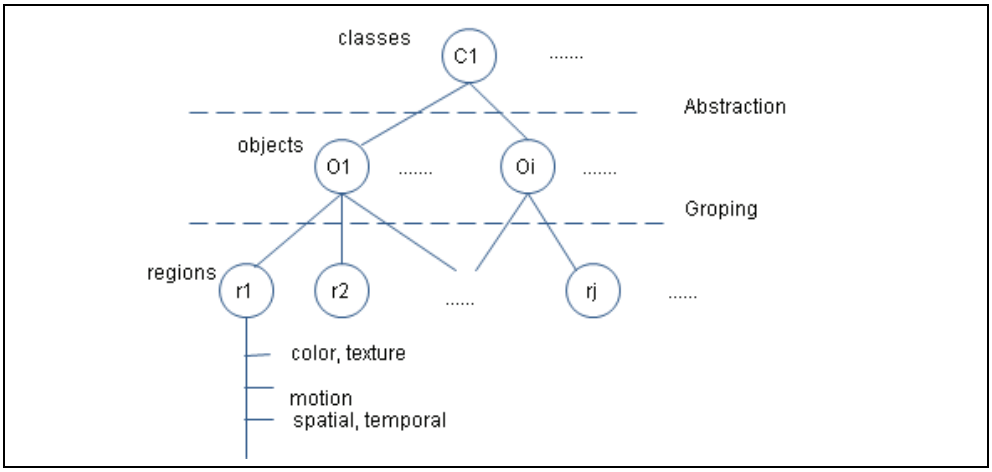

Fig. 4. Hierarchical representation of video objects

At the bottom level are primitive regions segmented according to color, texture, or motion measures. As these regions are tracked over time, temporal attributes such as trajectory, motion pattern, and life span can be obtained. The top level includes links to conceptual abstraction of video objects. For example, a group of video objects may be classified to moving human figure by identifying color regions (skin tone), spatial relationships (geometrical symmetry in the human models), and motion pattern of component regions. We propose the above hierarchical video object schema for content-based video indexing. One challenging issue here is to maximize the extent of useful information obtained from automatic image analysis tasks. A library of low-level regions and mid-level video objects can be constructed to be used in high-level semantic concept mapping. This general schema can be adapted to different specific domains efficiently and achieve higher performance.

4.2 Video indexing using face detection and face recognition methods

In this section, we will construct such a signature by using semantic information, namely information about the appearance of faces of distinct individuals. We will not concern ourselves with the extraction of face-related information, since ample work has been performed on the subject. Instead we will try to solve the problems of consistency and robustness with regards to face-based indexing, to represent face information with minimal redundancy, and also to find a fast (logarithmic-time) search method. All works on face-related information for video indexing until now have focused on the extraction of the face-related information and not on its organization and efficient indexing. In effect, they are works on face recognition with a view to application on indexing. The face detection stage is presented in Fig. 5.

Fig. 5. Detection algorithm scheme

The skin pixel detection is performed with a simple colormap, using the YCbCr color space. The second block corresponds to the segmentation algorithm which is performed in two stages, where the chrominance and luminance information is used consecutively. For each stage an algorithm which combines pixel and region based color segmentation techniques is used. After the segmentation, a set of connected homogenous skin-like regions is obtained. Then, potential face candidates (FC) are obtained by an iterative merging procedure using an adjacency criterion. Once the set of FC is built, it is necessary to remove the ones that do not match to any face. To that end some constrains regarding shape, size and overlapping are used.

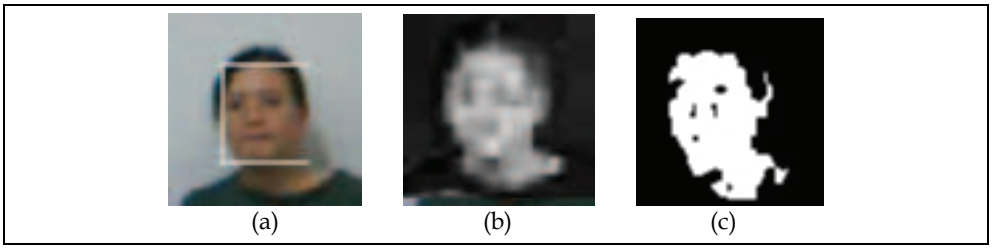

Fig. 6. Results of face detection: (a) the capture frame image; (b) result of similarity; (c) the binary result

Some results are shown in Fig. 6. Notice that the algorithm is able to produce good face candidates but also provides some candidates which do not correspond to any face. Most of these erroneous face candidates will be not recognized in the face recognition stage and will be discarded, but at the expenses of increasing unnecessarily the computational cost. Thus, if there are many erroneous face candidates, the overall system becomes inefficient.

Face areas are characterized to have a homogeneous chrominance component. Taking into account this fact, a new selection criterion has been designed in order to remove all the FC composed of regions whose average color differs substantially, as is the case when hair and face are included in the same candidate.

4.3 Text extraction for video indexing

The increasing availability of online digital video has rekindled interest in the problems of how to index multimedia information sources automatically and how to browse and manipulate them efficiently (David, 1998) (Snoek & Worring, 2005) (Zhu et al., 2006). The need for efficient content-based video indexing and retrieval has increased due to the rapid growth of video data available to consumers. For this purpose, text in video, especially the superimposed text, is the most frequently used since it provides high level semantic information about video content and it has distinctive visual characteristic.

Extraction of text information involves detection, localization, tracking, extraction, enhancement, and recognition of the text from a given video. Although a large number of techniques have been done on solving this problem, very little work has ever considered the temporal aspects of video. Text may span tens or even hundreds of frames, providing a tremendous amount of redundant information. In this section we address some of the efficient and reliable aspects of the text matching algorithms, by using fast correlation, rectangular sub-images techniques in a multi-resolution scheme, which enables real-time text tracking and locating applications on a standard personal computer (Ding et al., 2008).

4.3.1 Matching strategy

Similarity is the guiding principle for solving the matching problem. Among the different similarity measures proposed in the literature, Normalized Cross-Correlation (NCC) is widely used to their robustness in template matching. It has also been shown that NCC tends to give better results (Wu et al., 1995). Suppose T is a template to be tracking in an image I. Template matching by normalized correlations computes the following value at each point (x,y) of the image I:

$$c(x,y)=\frac{\mathrm{cov}_{x,y}(T,I)}{\sqrt{Q_{x,y}(T)Q_{x,y}(I)}} \tag{16}$$

in which,

$$\begin{aligned}\mathrm{cov}_{x,y}(T,I)&=\sum_{u,v}T(u,v)I(x+u,y+v)\\ Q_{x,y}(T)&=\sum_{u,v}T^2(u,v)\\ Q_{x,y}(I)&=\sum_{u,v}I^2(x+u,y+v)\end{aligned} \tag{17}$$

where the summations are over all template coordinates. A large value of $c(x,y)$ indicates a likely match at the coordinate (x,y). It can be shown that a match that maximizes c is identical to the template T up to scaling.

4.3.2 Computational optimisation

In this section we outline the optimization techniques adopted to avoid redundant calculations. Fig. 7 shows the coordinate relations between template and candidate images. The arrow denotes the moving direction of template. Expansion (16):

$$\mathrm{cov}_{x,y}(T,I)=\sum_{u=1}^{W}\sum_{v=1}^{H}T(u,v)I(x+u,y+v) \tag{18}$$

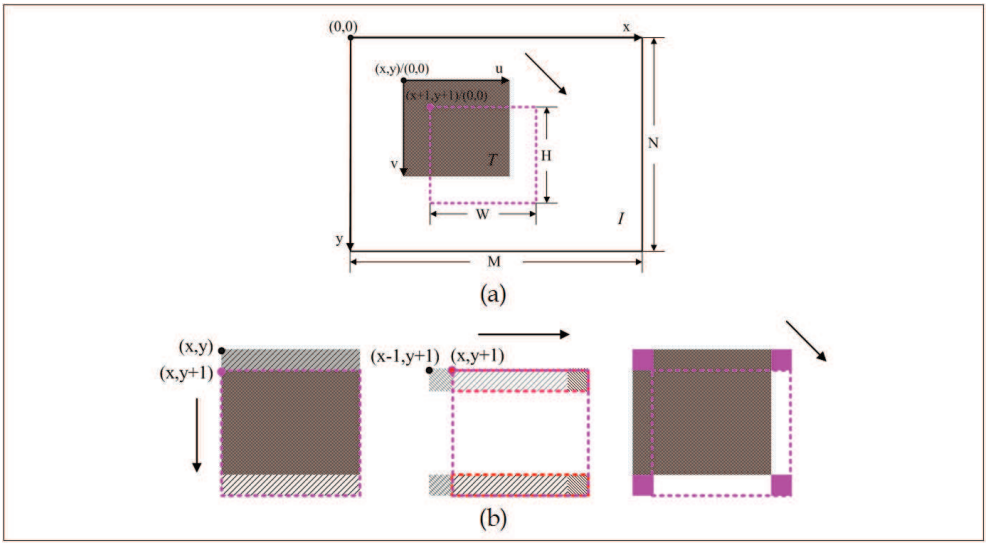

Fig. 7. (a)The coordinate relations between template and candidate images; (b) the steps of the template movement

Observing Figure 2, and following Eq. (18), suppose that $\mathrm{cov}_{x,y}(T,I)$ is the NCC score between a template T and a candidate image I at point (x,y). It is easy to notice that $\mathrm{cov}_{x,y+1}(T,I)$ can be attained from $\mathrm{cov}_{x,y}(T,I)$:

$$\mathrm{cov}_{x,y+1}(T,I) = \mathrm{cov}_{x,y}(T,I) + diff_{x,y+1}(T,I) \tag{19}$$

With $diff_{x,y+1}(T,I)$ representing the difference between the $c(x,y)$ associated with the lowermost and uppermost rows of the matching window (bias region shown in Figure 2(a)):

$$diff_{x,y+1}(T,I) = \sum_{u=1}^{W}[T(u,H)I(x+u,y+H+1) - T(u,1)I(x+u,y+1)] \tag{20}$$

And

$$\begin{aligned} diff_{x,y+1}(T,I) = diff_{x-1,y+1}(T,I) + [T(W,H)I(x+W,y+H+1) \\ - T(1,H)I(x,y+H+1)] + [T(W,1)I(x+W,y+1) - T(1,1)I(x,y+1)] \end{aligned} \tag{21}$$

This allows for keeping complexity small and independent of the size of the matching window, which is the size of the template, since only four elementary operations are needed to obtain the NCC score at each new point.

With the optimized algorithm, which is associated with the 4 purple points of Figure 2.The computational scheme of Eq. (16) and (21) makes use of a vertical recursion and a horizontal recursion to obtain the updating term.

When using the conventional NCC method, there is a computation of ($M \times N \times W \times H$) times sum-of-squares operations for each matching point, however, this computation will be degraded to ($M \times N + W \times H$) times operations with the optimized algorithm assuming the diff is unknown, otherwise only 4 times operations needed, similarly calculating Q.

With the optimized algorithm, which is associated with the 4 purple points of Figure 2.The computational scheme of Eq. (16) and (21) makes use of a vertical recursion and a horizontal recursion to obtain the updating term.

4.3.3 Multi-Resolution matching and result

The NCC-based text matching is region-based and its computational cost is considerable when tracking a large text line. The coarse-to-fine technique uses a hierarchical representation of image called Gaussian Pyramid (Bergen et al., 1992).

We collected a large mount of video sequences for experiments on text tracking from a wide variety of video sources, including movie credits, TV programs and news etc. A description of running times and match score of the algorithm on different video frames is listed in Table 3.

Algorithm	Candidate Image Size	Template Image Size	Match Percentage	Runtime (second)
Fast NNC	640×480	428×34	98.783%	0.0239
cvMatchTemplate	640×480	428×34	98.784%	0.1670
Fast NNC	352×288	173×31	98.088%	0.0283
cvMatchTemplate	352×288	288×384	98.088%	0.1698
Fast NNC	512×384	370×30	92.906%	0.0258
cvMatchTemplate	512×384	370×30	92.907%	0.1710

Table 3. Running times and match score

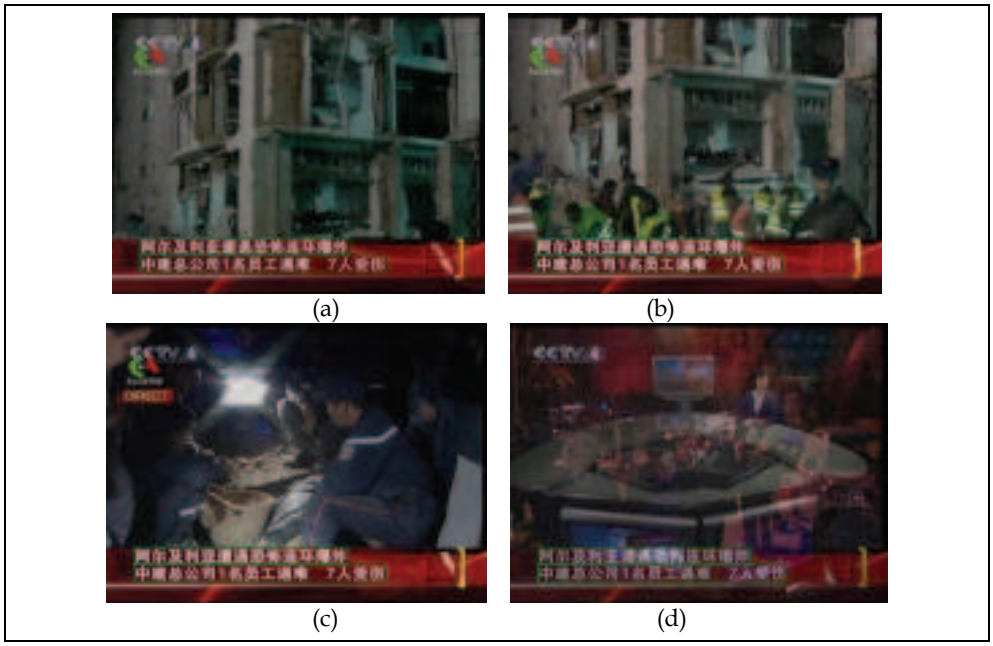

Fig. 8. Results of news detection: (a) template (b)1057th frame, 97.348%*(c)1158th frame, 97.859%* (d)1280th frame, 95.586%*

The time spent in the stage for obtaining coefficients for the normalized cross-correlation is obviously reduced comparing to the cvMatchTemplate in the same search window size, but the match score is almost invariant. Various types of videos have been measured by using our proposed method. Figure 4 shows result of news in 25 fps. Where the green box in Fig. 8 (a) is the text region referred to as template, it is the text detection result of frame 996. Where * denote the match percentage.
The measure results of American TV programs (the Apprentice) in 25fps. The shot, as well as background, is change frequently in this video. Where the green boxes in Figure 8(a) are the text detection result of frame 5564.
Generally, when the match score is lower than 90% we think the result can not be confident and new detection needed. Unfortunately, quantitative evaluation of tracking accuracy is not easy because of the lack of truth data. Our experiments show that the matcher can work well when the text is in simple (rigid, linear) motion, even in complex backgrounds.

5. Future research and conclusions

We have presented a novel method for performing fast retrieval of video segments based on the output of face detectors and recognizers, with possible uses to indexing of video databases. Finally, we have developed a fast text matching method using fast correlation in the coarse-to-fine framework. By using the normalized cross-correlation (NCC) similarity measure rather than the simple SSD or SAD, the reliability of the algorithm is increased. The fast cross-correlation has been realized by using the recursive technique. These systems are similar to ours in that they use features from the visual data to segment and index video content. We also focus on content-based retrieval that consumers would want in their homes such as automatic personalization of content retrieval based on user profiles.
During the last thirty years we have witnessed a tremendous explosion in research and applications in the visual communications field. The field is now mature as is proven by the large number of applications that make use of this technology. There is no doubt that the beginning of the new century revolves around the "information society." Technologically speaking, the information society will be driven by audio and visual applications that allow instant access to multimedia information.

6. Acknowledgments

This work is supported by the research and application of intelligent equipment based on untouched techniques for children under 8 years old of BMSTC & Beijing Municipal Education Commission (No. 2007B06 & No. KM200810028017).

7. References

Hjelsvold, R. (1995). *VideoSTAR – A database for video information sharing*, Ph.D. Thesis. Norwegian Institute of Technology

Brown, K. (2001). *A rich diet: Data-rich multimedia has a lot in store for archiving and storage companies,* Broadband Week

Baeza-Yates R. & Ribeiro-Neto B. (1999). *Modern Information Retrieval,* Addison-Wesley / ACM Press, ISBN-13: 978-0201398298, New York

Zhang Y.J. (2006). *Semantic-Based Visual Information Retrieval*, IRM Press, ISBN-13: 978-1599043715, USA

Li J. Z. & Özsu M. T. (1997). STARS: A Spatial Attributes Retrieval System for Images and Videos, *Proceedings of the 4th International Conference on Multimedia Modeling* (MMM'97), pp 69-84, Singapore

Dağtas, Al-Khatib W.; Ghafoor A. & Kashyap R. L. (2000). Models for Motion-Based Indexing and Retrieval, I*EEE Transactions on Image Processing*, Vol. 9, No. 1, January 2000

Al-Khatib Q.; Day F.; Ghafoor A. & Berra B. (1999). Semantic Modelling and Knowledge Representation in 7 Multimedia Databases, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, January/February 1999

Analyti A. & Christodoulakis S. (1995). Multimedia Object Modeling and Content-Based Querying, *Proceedings of Advanced Course – Multimedia databases in Perspective*, Netherlands 1995.

Yeo B-L. & Yeung M. (1997). Retrieving and Visualizing Video, *Communications of the ACM*, Vol. 40, No. 12, December 1997

Kyriakaki G. (2000) *MPEG Information Management Services for Audiovisual Applications*, Master Thesis, Technical University of Crete, March 2000

Hacid M-S.; Decleir C. & Kouloumdjian J. (2000). A Database Approach for Modeling and Querying Video Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 5, September/October 2000

Lienhart, R. (1999). Comparison of automatic shot boundary detection algorithms. *In: SPIE Conf. on Storage and Retrieval for Image and Video Databases VII*, vol. 3656, pp. 290–301

Zhang H. J.; Kankanhalli A. & Smoliar S. W. (1993). Automatic Partitioning of Full Motion Video, *Multimedia Systems*, vol.1, pp:10 – 28

Hampapur A.; Jain R. & T. Weymouth. (1994). Digital Video Segmentation. *In Proc. ACM Multimedia,* pp: 357 – 364

Teng S.H. & Tan W.W. (2008). Video Temporal Segmentation Using Support Vector Machine, *Contact Information Retrieval Technology*, pp: 442-447

Smeaton A.F. (2007). Techniques used and open challenges to the analysis, indexing and retrieval of digital video. *Information Systems,* vol.32(4), pp: 545–559

Ford R. M.; Robson C.; Tample D.; & Gerlach M. (1997). Metrics for Scene Change Detection in Digital Video Sequences, I*n IEEE International Conference on Multimedia Computing and Systems (ICMCS '97)*, pp: 610–611, Ottawa, Canada

Dugad R.; Ratakonda K. & Ahuja N. (1998). Robust Video Shot Change Detection, *IEEE Second Workshop on Multimedia Signal Processing*, pp: 376–381, Redondo Beach, California

Sethi K. & Patel N. V. (1995). A Statistical Approach to Scene Change Detection, *in IS&T SPIE Proceedings: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 329–339, San Jose

Gargi U.; Kasturi R. & Strayer S. H. (2000). Performance Characterization of Video-Shot-Change Detection Methods, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 1–13.

A. Dailianas,R. B. Allen, and P. England, Comparison of Automatic Video Segmentation Algorithms, in Proceedings of SPIE Photonics West, 1995.SPIE, Vol. 2615, pp. 2–16, Philadelphia, 1995.

Ardizzone E. & La Cascia M. (1997). Automatic Video Database Indexing and Retrieval, *Multimedia Tools and Applications*, Vol. 4, pp. 29–56

Yeo L. & Liu B. (1995). Rapid Scene Analysis on Compressed Video, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5, No. 6, pp. 533–544

Todd R. R. (2005). *Digital Image Sequence Processing, Compression, and Analysis*, CRC Press, ISBN: 0849315263, 9780849315268, USA

Ojala T.; Kauniskangas H.; Keränen H.; Matinmikko E.; Aittola M.; Hagelberg K.; Rautiainen M. & Häkkinen M. (2001). CMRS: Architecture for content-based multimedia retrieval. *Proc. Infotech Oulu International Workshop on Information Retrieval*, Oulu, Finland, pp: 179-190.

Addison J. F. D.; MacIntyre J. (editors). (2003). Intelligent techniques: A review, *Springer Verlag (UK) Publishing Company*, 1st Edition, Chapter 9

Fausett, L. (1994). *Fundamentals of Neural Networks*, Englewood Cliffs, NJ: Prentice Hall.

Holland J. (1975). *Adaptation in Natural and Artificial systems*. MIT Press.

Bishop C. M. (1997). *Neural networks for pattern recognition*, Oxford University Press, pp: 6-9

Hunter A.; Kennedy L.; Henry J. & Ferguson R.I. (2000). Application of Neural Networks and Sensitivity Analysis to improved prediction of Trauma Survival, *Computer Methods and Algorithms in Biomedicine 62*, pp: 11-19

Volker R. (1999). Content-based retrieval from digital video, *Image and Vision Computing*,Volume 17, Issue 7, May 1999, pp: 531-540

Liu F. & Picard R. W. (1996). Periodicity, directionality and randomness: Wold features for image modelling and retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), pp: 722-733

Manjunath B. S. & Ma W. Y. (1996). Texture features for browsing and retrieval of large image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (Special Issue on Digital Libraries), Vol. 18 (8), August 1996, pp: 837-842.

Kaplan L. M. et al. (1998). Fast texture database retrieval using extended fractal features, *In Storage and Retrieval for Image and Video Databases VI (*Sethi, I K and Jain, R C, eds), Proc SPIE 3312, pp: 162-173

Haralick R. M. (1979). Statistical and structural approaches to texture. *Proc. IEEE.* Vol 67, pp: 786-804

Gool L. V. & Dewaele P. and Oosterlinck A. (1985). Texture analysis anno 1983, *Computerr Vision, Graphics and Image Processing,* vol 29, pp: 336-357.

Ding H.; Ding X. Q.; Wang S. J. (2008). Texture Fusion Based Wavelet Transform Applied to Video Text Enhancement, *Journal of Information and Computational Science*, pp: 2083-2090

Ding H.; Ding X. Q.; Wang S. J. (2008). Fast Text Matching in Digital Videos, *International Symposium on Computational Intelligence and Design 2008*, China, pp: 273-276, 2008.

David D. (1998). The indexing and retrieval of document images: a survey, *International Journal of Computer Vision and Image Understanding*, 70(3), pp: 287-298.

Snoek C.G.M. & Worring M. (2005). Multimodal Video Indexing: A Review of the State-of-the-art, *Multimedia Tools and Applications*, 25(1), pp: 5-35

Zhu Q.; Yeh M.C. & K.T. Cheng. (2006). Multimodal fusion using learned text concepts for image categorization, *Proc. 14th ACM Conf. on Multimedia*, Santa Barbara, CA, 2006, pp: 211–220

Wu Q.X.; McNeill S.J. & Pairman D. (1995). Fast algorithms for correlation-relaxation technique to determine cloud motion fields?, *Proc. Digital Image Computing: Techniques and Applications,* Brisbane, Australia,1995, pp: 330-335

Schweitzer H.; Bell J.W. & F. Wu. (2002). Very fast template matching, *Proc. 7th European Conf. on Computer Vision*, Copenhagen, Denmark, pp: 358-372.

Gonzalez R.C.; Woods R.E. (1992). *Digital Image Processing*, Massachusetts: Addison-Wesley,

Bergen J.R.; Anandan P.; Hanna K.J. & R. Himgorani. (1992). Hierarchical Model-based Motion Estimation, *Proc. 2nd European Conf. on Computer Vision*, Santa Margherita Ligure, Italy, pp: 237-252.

Digital Video

Edited by Floriano De Rango

ISBN 978-953-7619-70-1
Hard cover, 500 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

This book tries to address different aspects and issues related to video and multimedia distribution over the heterogeneous environment considering broadband satellite networks and general wireless systems where wireless communications and conditions can pose serious problems to the efficient and reliable delivery of content. Specific chapters of the book relate to different research topics covering the architectural aspects of the most famous DVB standard (DVB-T, DVB-S/S2, DVB-H etc.), the protocol aspects and the transmission techniques making use of MIMO, hierarchical modulation and lossy compression. In addition, research issues related to the application layer and to the content semantic, organization and research on the web have also been addressed in order to give a complete view of the problems. The network technologies used in the book are mainly broadband wireless and satellite networks. The book can be read by intermediate students, researchers, engineers or people with some knowledge or specialization in network topics.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821