Arranging Pixels in a DBMS
When Vision and Databases Come Together

Themistoklis Palpanas

themis@cs.toronto.edu

Department of Computer Science

University of Toronto

10 King's College Road, Toronto

Ontario, M5S 3G4, CANADA

Department of Computer Science

University of Toronto

Technical Report CSRG-404

Abstract

The rapid technological advances augmented our capabilities for generating and storing multimedia content data. The large amount of image and video data makes it very cumbersome to manually search, browse, and annotate them. Therefore, the need for systems that would automate the aforementioned procedures is now evident more than ever. The focus of this paper is on image and video databases. We investigate what techniques are used in order to achieve the goals of searching, browsing, and annotating multimedia content. We analyze the state of the art methods employed for extracting interesting features from the data (colour moments, texture and shape, high level representation, motion components, camera operations). Most of the above methods originate from the vision community. Yet, many interesting problems arise when we try to apply these methods in conjunction with a DataBase Management System (DBMS). Thus, in many cases, the database community has derived new techniques to deal with the problem of indexing multimedia data. Note, that this survey does not intend to be exhaustive, but rather indicative of the state of the art trends, and the future research directions in both the vision and database communities.

1 Introduction

The rapid technological advances augmented our capabilities for generating and storing multimedia content data. Research centres are storing scientific images and video, and television channels are archiving every single frame of broadcasted images. The large amount of image and video data makes it a tedious and hard job to manually search, browse, and annotate them, just by exhaustively going through the whole corpus. Therefore, the need for systems that would automate the aforementioned procedures is now evident more than ever.

In this paper we will present an overview of the existing work in the area which is called "multimedia databases". The focus will be on image and video databases, and we will investigate what techniques are used in order to achieve the goals of searching, browsing, and annotating multimedia content. We will analyze the state of the art methods employed for extracting interesting features from the data (colour moments, texture and shape, high level representation, motion components, camera operations). Most of the above methods originate from the vision community. Yet, many interesting problems arise when we try to apply these methods in conjunction with databases. Thus, in many cases, the database community has derived new techniques to deal with the problem of indexing multimedia data. Note, that this survey does not intend to be exhaustive, but rather indicative of the state of the art trends, and the future research directions in both the vision and databases communities.

2 The Heritage

It was during this century that a really mass production of images started to take place. Even before the invention of digital media (and digital storage in particular) the amount of image and video produced worldwide was overwhelming.

This includes photograph archives of news agencies, involving every aspect of everyday life; research centres, dealing with animal and plant species on earth, meteorology, earth observation from satellites, astronomy; private institutions, including museum collections and artistic photography. The video (film) archives were, and still are, more voluminous, because of the nature of the medium per se. Cinema movies, documentaries, and television broadcasts account for a significant amount of the culture that humanity has produced in the last century.

2.1 The Digital Age

The advent of computers boosted the image production of all sorts and kinds. This did not happen suddenly, but rather gradually, as the new digital medium gave the opportunity to automate the process of image acquisition and storage. Nowadays, there are ongoing research projects that produce image data at rates much higher than the human analysts can deal with. This amount of data is being stored in digital format, but is still examined manually by humans, in a painstaking and time-consuming effort.

Despite the major advances in computer systems, and especially in database technology, during the last years, the field of automatic image manipulation has not benefited. The work done in the databases area is very significant, and has proved to provide extremely efficient and versatile solutions for a wide variety of applications. Nevertheless, the research in this area proved to be not mature enough to encompass image and video data. Hence, any archives storing the aforementioned data types either do not have any practically useful set of metadata associated with them. Those systems require human agents in order to perform even simple operations such as image classification. Moreover, they are neither scalable nor flexible, rendering the work of maintaining and analyzing image archives very tedious and cumbersome.

3 The State of the Art

In this section we will review a number of different approaches that try to bridge the gap between the vision and the database communities [SKG98]. Image and video databases form a novel and very active area of research that has drawn a lot of attention from people of both fields.

Researchers from the vision community are working on ways to break down an image (either still image, or part of a series of frames in a video sequence) into more primitive elements, in terms of the objects represented therein. That involves detecting any active agents in the image and separating them from the background. Identifying human beings, both bodies as a whole and just faces, since this is of special interest in many applications. Following moving objects in some scene, even in the case when the camera itself is moving.

Another area of interest in the vision community is the automatic classification of images. This classification may be performed at various levels of abstraction and according to different criteria. Some of the possible applications are thematic classification of images, and summarization and fast browsing of movies.

The work in the database field moves along a direction perpendicular to the one described above. The interesting issues here are the efficient and scalable storage and retrieval of images. In order to achieve that we must first be able to devise methods for the extraction of all the interesting features, and the concise representation of them. Then, specialized structures can index this information, and provide fast answers to user queries. The queries can be as simple as specific questions requiring an exact answer, or more complex ones asking for neighborhood of similar answers. In the latter case it is not enough to provide an index that will allow fast answering. The same index should also be built in a way that similar objects are close together in the index as well. The similarity function is user defined, and there is virtually no restriction on the form that it may have.

In the following paragraphs we will present some of the relevant work in both the vision and the database research areas. The focus though, will be on research associated with images and video involving human agents. The work that we will discuss is very recent, and almost no part of it has found its way into commercial applications.

3.1 The Vision Perspective

We can categorize the work in this area in three classes:

Object Recognition The goal here is to examine an image and determine whether a specific object exists in the image or not. We will focus on the detection of human beings, for which there are different approaches. Some of the research in this area is specialized, in that it considers only face detection.

Motion Detection and Tracking The objective is to examine a series of images (i.e., frames from a video sequence), identify the existing motions, and be able to follow the moving objects as they move around in the scene. Sometimes the aforementioned procedure must be preceded by an object recognition phase.

Clustering and Classification There are two main streams in this field. First, clustering of video frames in a hierarchical structure, in order to summarize the contents of a movie, and enable fast browsing. Second, automatic classification of images or video sequences based on their semantic content.

3.1.1 Object Recognition

The work done by Nelson and Selinger [NS98] describes an appearance-based object recognition system. This work does not deal with human figures, but rather focuses on correctly identifying an object from a small database of 3-dimensional shapes, with occlusion and clutter resistance.

The novel idea in this research effort is the way objects are represented in the knowledge base of the system. The representation of an object is a loosely structured combination of a number of *local context* regions, each one describing a *key feature*. Key features are the distinctive features that an object may have (e.g., the handle of a cup), and the image patch which includes a key feature along with its neighborhood constitutes a local context region.

The way the object recognition procedure works is the following. First, they take sample images of an object, that cover the region of interest on the viewing sphere. Then, key features are extracted from these images, and the corresponding local context regions are normalized and stored in the database. Currently, only one type of feature is being used, namely robust boundary fragments of the object. Note, that not all the key features are saved, but only the most representative ones are selected. Figure 1 shows an example of patches generated for a cup image, and the kind of fragmentation that the model

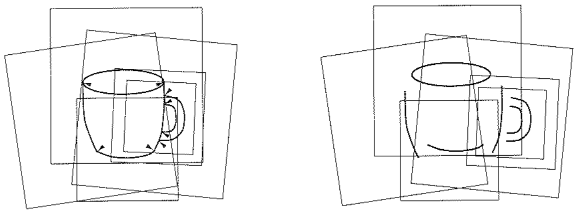


Figure 1: Example of local context regions generated by fragmenting an image of a cup (the arrows indicate fragment endpoints).

allows, while preserving loose global relationships.

When the target image arrives, local contexts are extracted once again, and matched against the database. Probabilistic hypotheses are formed about key features and the configuration (i.e., position, size, orientation) of the object that could have produced them. Subsequently, loosely consistent groupings of the most prevalent hypotheses are identified. The grouping that ends up having the highest score is selected as the most probable answer.

The experiments show that the method achieves high success rates (up to 99.6% for 6 objects), and is fairly robust to occlusion and background clutter, since it is not based on any global properties of the objects. The reported recognition rate is around 90% for 75% clutter and 25% occlusion (when the database has 6 objects). The above results are very encouraging, though the main concerns here are ones of scalability. Given the complex environment that such a system would have to operate, it should be able to function with a database consisting of at least several hundreds of objects. It would be very interesting to test whether this technique scales to these numbers, while maintaining similar accuracy levels, and low computational complexity.

The study of Chalom and Bove [CJ96] treats a variation of the recognition problem. They segment an image or image sequences into regions corresponding to objects. The segmentation scheme takes into account multiple image characteristics, develops a multi-modal statistical model of object regions, and the bootstrapping process is based on user-supplied training data.

The problem of image segmentation boils down to assigning each pixel of the image to one of the objects that are contained in the image. In order to achieve that, every pixel is associated to a feature vector which describes various characteristics of the pixel. The features used in this study include colour and luminance information, position in the image, and motion estimation. The latter is computed using a dense optical flow technique, or even more naive block matching algorithms.

The proposed method employs *probability density functions* (PDF) that parametrically describe the feature vectors of each image segment. Preliminary experiments showed that for many "natural" image sequences most of the features' distributions could be approximated by a mixture of Gaussian PDFs. The actual segmentation of the image is driven by the user, who specifies a small number of training points that determine the different objects in the image. Then, the system computes a multi-modal PDF that estimates each segment. It is obvious that the higher number of modes the model is using the better the fit to the observed data. In order to avoid the problem of model over fitting an information theoretic approach is used. The number of modes in the model is determined by maximizing the *benefit* of increasing the complexity of the model. The benefit is defined as the normalized Kullback-Leibler distance[1] between a particular instance of the model and the real data. At this point, the remaining problem is to classify the rest of the points in the image (i.e., excluding the training points) into one of the regions of the image. This is done by maximizing the a posteriori probability $P(R_i|\hat{f})$ of the pixel with feature vector $\hat{f}$ to belong to region R_i. In the case of sequences of images, the PDF estimates of each region are updated at each frame by tracking the training points, using some motion estimation technique.

The experiments (Figure 2) have shown that the overall segmentation is quite robust, and varies slightly if a particular feature calculation is modified. The low level of user interaction and the simplicity of training required make this approach very appealing to use.

Rowley, Baluja, and Kanade [RBK98] studied the problem of face detection. They propose a neural network-based solution, that detects faces at any degree of rotation in the image plane (Figure 3). The system is based on face templates, which are constructed using image intensities of training face images.

The neural network consists of two levels. The first level, the *router network* takes as input a 20x20

[1] The Kullback-Leibler distance is known in the literature as the relative entropy [KK92] and measures the closeness of two probability distributions. It is defined as: $D(p|q) = \sum_X p(X) \log \frac{p(X)}{q(X)}$, where p, q are the two probability distributions of interest.

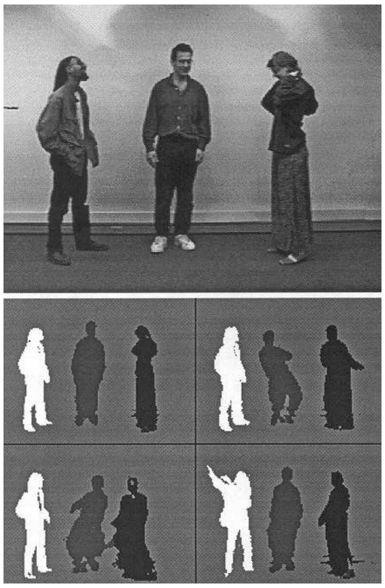


Figure 2: Segmentation of a series of images, using training data from the first frame only.

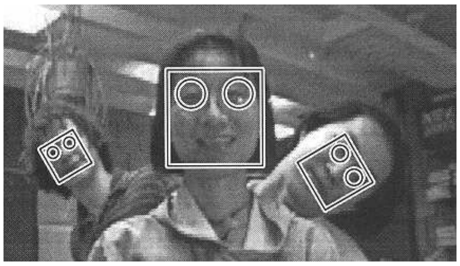


Figure 3: Sample output of the system.

pixel window of the original image, and returns a rotation angle. Then, the window is rotated by that angle, in order to get he potential face to the upright position. Finally, the second layer of the neural network, the *detector network* decides whether a face exists in the image window or not. Because the detector network is only applied once at each location in the image, the processing is significantly faster than exhaustively trying all the possible orientations. To account for faces larger than the window size, the input image is repeatedly subsampled, and the window filter is applied at each scale. An overview of the system is depicted in Figure 4.

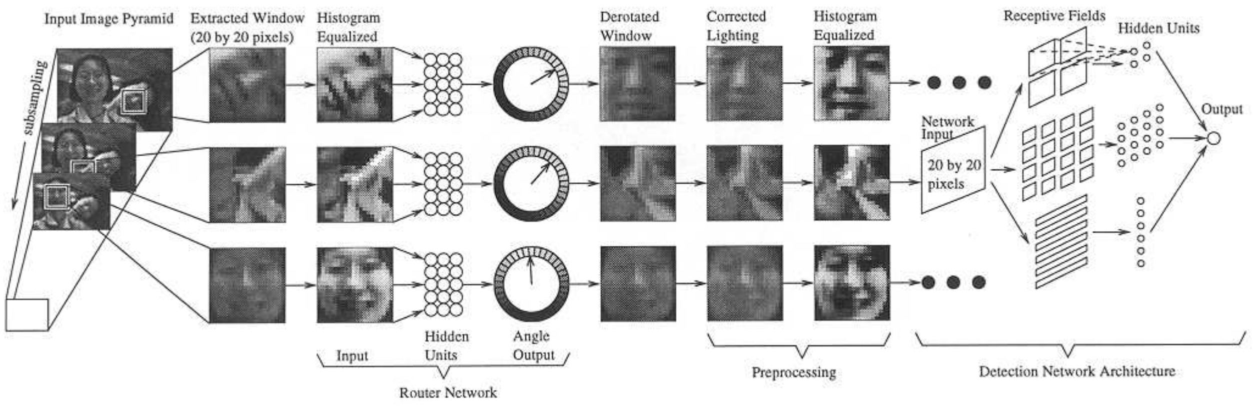


Figure 4: Overview of the algorithm.

The experiments show that the system correctly identifies 80% of the faces in a corpus of more than 180 real-life images, with very few false positives. The same method has also been applied for detecting faces rotated out of the image plane, such as profiles and semi-profiles. These results are preliminary, yet promising.

3.1.2 Motion Detection and Tracking

The interesting and difficult problem of motion estimation is treated by MacLean, Jepson, and Frecker [MJF94]. More specifically, they study the recovery of egomotion, and the segmentation of independent object motion using the *Expectation-Maximization* (EM) algorithm[2] [MK97]. Egomotion is defined as the image motion induced by an observer moving through a static environment, while motion due to independently moving objects is defined as the image motion induced by the movement of an object relative to the observer when that object is not stationary with respect to the environment at large.

[2]The EM algorithm has been successfully applied to a variety of problems involving incomplete data, such as training neural networks, and learning mixture models. The EM associates a given incomplete-data problem with a simpler complete-data problem, and iteratively finds the maximum likelihood estimates of the data. In a typical situation, the EM converges monotonically to a fixed point in the state space, usually a local maximum.

In this study the optical flow information is used in order to estimate each of the (3-dimensional) motions in the image. Then, the problem is one of segmenting the optical flow in distinct clusters, so that each cluster corresponds to each one of the independent motions. The clusters can be modeled as a mixture of probability distributions (this study uses Gaussian distributions) with different parameters.

Once the optical flow is recovered, linear and bilinear constraints are generated for each sample point. The linear constraints involve only the translational motion, while the bilinear both the translational and rotational. All these constraints are handled by the EM algorithm in order to assign the corresponding points to one of the clusters that describe the independent motions. As a result, the authors can recover the motion of the camera as well as any other independent motions. This process also yields an inverse relative-depth map, which reveals the 3-dimensional placement of the objects of the image.

The aforementioned work deals with an important aspect of the research in the vision community. It is evident that the problem it tackles has many unknown parameters, is under-specified, and thus gives rise to approximate answers. This study makes the assumption that the number of motions in the sequence of images is known in advance. However, in order to move to large scale processing of images we will need to automate this step as well. Another interesting direction of future work would be to study the efficiency of the algorithm as the number of independent motions in the image increase.

People are the central element in a large fraction of the multimedia content production, which renders the need to track and interpret human action essential. The ability to find and follow people's head, hands, and body is therefore a substantial visual problem. This is the focus of the work done by Azarbayejani, Wren, and Pentland [AWP96]. They utilize 2-dimensional image analysis techniques in order to recognize the image of a person and track her movements, and they subsequently use this information to recover 3-dimensional models of the human actors in their environment. The latter problem is more related to applications such as *visually guided animation*, *avatars*, and *telepresence*, where the common characteristic is that we want to make a computer animated human character reproduce the moves of an actual person. Hence, we will only look closer at the way this work deals with the former problem.

Although this study deals exclusively with images involving human actors, it is very similar in essence to the work that partitions an image to segments corresponding to different objects [CJ96], which we discussed earlier. The approach used in this study is essentially the same. The system uses Gaussian probability distributions to describe the human actors in the image, using colour, position, and texture information. However, in this specialized case of *recognizing and tracking human motion* the model can be refined, and customized. The knowledge that the objects of interest are persons leads to the construction of a coarse human model, upon which the system's representation is based.

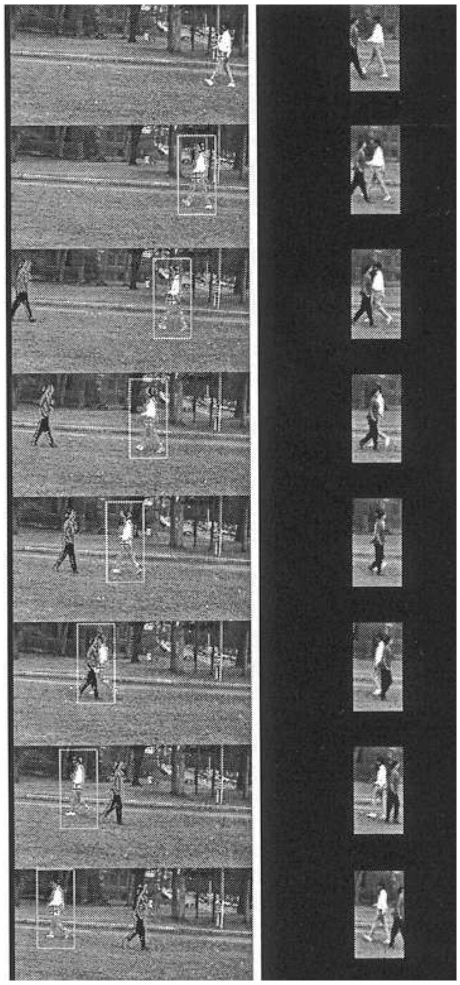
The system works in the following way. It first learns the background scene, by acquiring a sequence of images that do not contain a person. Then, when a person is introduced the system detects the change, and tries to build up a model that follows the general characteristics of the human model which is broken down to head, upper body, hands, legs, and feet. This is a refinement process, starting with a single blob covering the whole person, and successively splitting to obtain more detail. If some part of the human body is occluded the corresponding blob is deleted from the person model, to be recomputed once the occluded part reappears. All the model parameters are predicted and updated for the future images in the sequence, which enables real-time processing.

The interesting point in this work is the translation of the implicit knowledge in terms of the model, which makes the procedure faster, and more reliable. Nevertheless, more research has to be done for the problem of identifying human actors in a scene, without any prior knowledge (except for the fact that we are looking for a human).

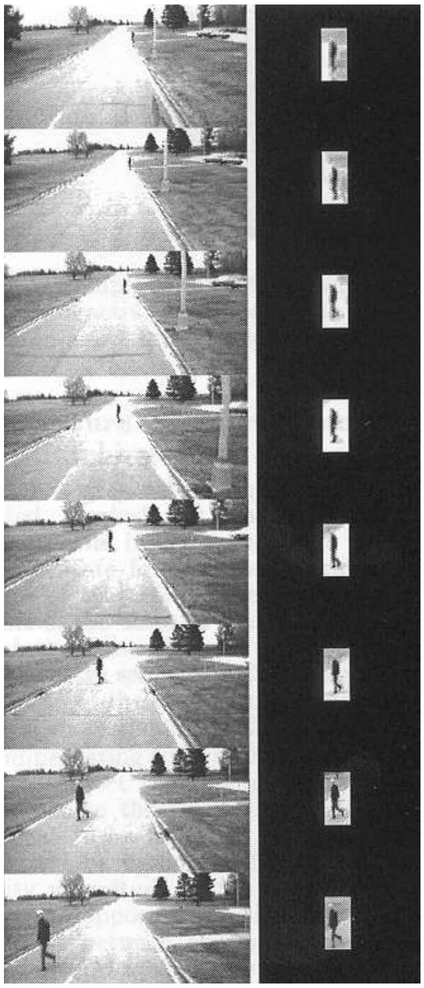
Polana and Nelson take a step further, and treat the problem of detecting and recognizing periodic, non rigid motion [PN]. Although we will discuss their work in relation to human motion, the work they have done is more general, extending to any periodic motion.

The interesting point in this study is that they deviate from the norm in this area. Previous studies were trying to identify different kinds of human motion by tracking a number of feature points, and classifying their trajectories. This involved a fine-grained analysis and segmentation of the human actor, in order to recognize and track specific points, such as the joints. The above requirement made the whole process delicate, computationally expensive, and cumbersome. The current study demonstrates that repetitive motion is a cue strong enough to enable the recognition of the motion with only high-level analysis. This is achieved by matching the motion against a set of known spatiotemporal motion templates.

Prior to the matching procedure a number of essential preprocessing steps must take place [PN94]. These steps are the segmentation of the moving actor, and the normalization of its motion, both in the space and time dimensions, so that it can be compared to the motion templates. The first step makes sure that the system isolates the segments of the image where a moving actor exists, using information based on the optical flow [Nel91]. Once the object of interest is identified, the pixels of the image that correspond to it are clustered, and tracked. The algorithm uses information from the last k image frames to estimate the centroid velocity of the object, and its future position. This makes the process robust to the presence of other moving objects, as well as temporary occlusions (Figure 5.a). The second step, the space normalization procedure, accounts for the changes in the scale of the moving object, as it moves (i.e., comes closer to the camera or moves away from it) in subsequent frames. An example of the output of this algorithm is depicted in Figure 5.b. The above procedure results in an



(a) Successfully tracking a walking person in the case of multiple moving objects and occlusion.



(b) Changing scale for a person crossing a street.

Figure 5: Tracking and normalizing procedures.

image sequence consisting of the actor at the centre of the image frame and at the same distance from the camera throughout the image sequence. The study does not suggest any efficient algorithm for the temporal scale invariance, which is achieved just by matching the test template against the reference template at all possible temporal translations. The core of the system is the procedure that performs the periodic motion recognition according to the following steps:

1. Compute optical flow for the pixels in the normalized image sequence.

2. Divide the image in a grid, and compute the flow magnitude of the centroid of each cell.

3. Compute the discrete Fourier transform of the signal produced by the variation of the flow magnitude of each cell across time, and identify the dominant frequency of the motion.

4. Divide the motion in T time segments, and XY spatial segments per time segment. Construct a vector of size XYT by computing some motion statistic (this study uses the sum of motion magnitude) for the segments, involving only the pixels that exhibit the dominant frequency.

5. Match the resulting feature vector to the vectors of known motions in the knowledge base. The closest match is the winner.

Figure 6 shows an example of the feature vectors produced by the system.

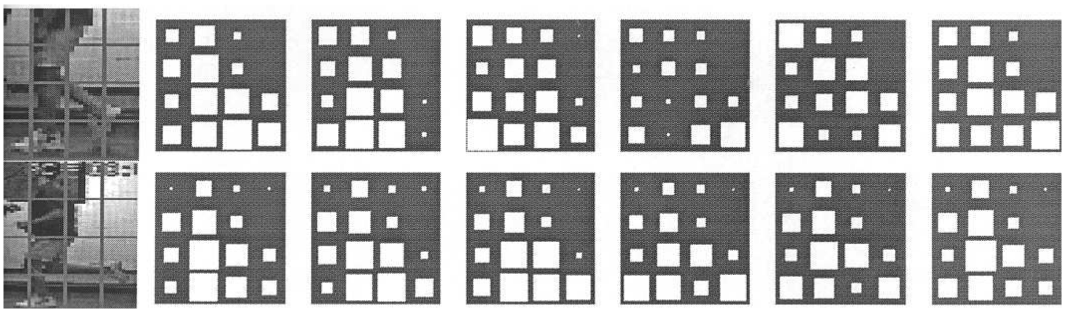


Figure 6: Total motion magnitude feature vector for a sample of walk (top) and a sample of run (bottom).

The algorithms presented in this work carry out the difficult task of periodic motion recognition, without having high computational complexity. In fact, the authors demonstrate a real-time implementation of the system. The procedure also proved to be quite robust to noise, such as a scene where leaves were fluttering in the background. Nevertheless, all the system tests involved the classification of some motion into one of the eight distinct classes in the knowledge base. Further research is needed in order to determine the behaviour of the system as the size of the knowledge base increases, especially since the confusion matrix of the experiments indicates that there might be a problem.

3.1.3 Clustering and Classification

Zhang et al. [ZLSW95] present a solution for computer assisted video parsing and content-based video retrieval and browsing. This is a different direction of research from the work we discussed above, yet, it has some very interesting and useful applications. The goal here is to temporally segment and abstract a video source, so that each segment has some semantic continuity, and can be represented by a *key-frame* (selected from the frames of the image sequence). Such a representation enhances the functionality of multimedia systems by enabling the construction of indexes and the fast browsing of the video content.

It is evident that the extremely high volume of data associated with video storage and analysis calls for smart indexing and browsing techniques that will keep only the necessary pieces of information. Indexing video data is a very challenging endeavor exactly because it is not possible (with the current resources) to index every single detail of a video source. This study proposes instead to identify several key-frames in the video and index only those. Each key-frame will be representative of the semantic content for a small interval of the whole video sequence. Once key-frames are extracted they can also be used for hierarchical browsing of the video source. That is, the user may quickly scan through the key-frames in order to jump to the part of the video she is interested in.

We will now review the pivotal operations of the system. The temporal segmentation is the first step of the analysis of the video source. Two of the techniques used are the inspection of the characteristic patterns of motion vectors, or the examination of the spatiotemporal energy models[3] [AB85] of the image sequence. In addition, the way colours change between subsequent frames are indicative of content change (e.g., a dramatic change in the colour space is usually due to a scene change). This segmentation divides the video into *camera shots* which are a collection of one or more frames recorded contiguously and representing a continuous action in time and space. Then, for each segment a small number of representative key-frames are selected. The selection procedure makes use of both the colour features of the images and the motion therein. Colour features include distribution of colour intensities histograms, average brightness, colour moments, and dominant colour. These properties can guide the key-frame extraction process to only select frames that are significantly different from each other. The motion analysis can identify camera operations, and take actions accordingly. For example a zoom shot will be abstracted by three frames, the first, the middle, and the last one. The key-frames are subsequently indexed along many dimensions:

Colour moments: They characterize the distribution of colours in the image. The first three moments are used (for each colour used for the representation of the image), namely the average intensity, the variance, and the skewness.

Texture: Information for texture is useful both for colour and grey scale images. The system uses the

[3] According to this approach, the subsequent frames of a video sequence are considered in a 3-dimensional space where the third dimension is time. Then, motion corresponds to a 3-dimensional orientation in this space, and is attributed an amount of energy. Physiology and psychophysics suggest a set of filters that lead to motion detection and explanation in a natural fashion (i.e., the way humans do). It is interesting to note here that this technique does not require any image processing. Yet, it may detect even random motion, like the (non-)picture we get on television when it is not tuned to any channel.

Multiresolution Simultaneous AutoRegressive (MSAR) [MJ92] model, which can capture texture information at different granularities.

Shape and edges: Shapes are identified by semi-automatic colour-based segmentation algorithms, and edges by using edge detection filters.

After all these steps of preprocessing of the video content, the system allows two types of search: querying or browsing the multimedia database. When querying, the user may use a visual template (specifying colour distributions, texture maps, etc.), or an actual image, and ask for similar frames. The user may also ask for shots that exhibit certain temporal features, such as specific camera operations. In the browsing mode, the user is presented with a number of key-frames organized in a hierarchical manner (Figure 7) according to the temporal segmentation algorithm (this is more an abstraction procedure,

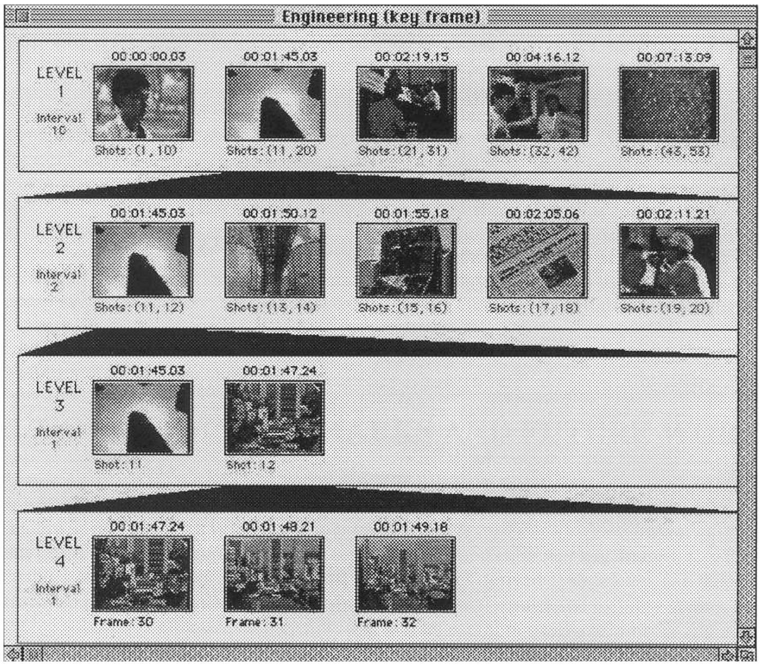


Figure 7: Key-frame-based hierarchical video browser.

rather than clustering since similar shots that are not consecutive are not unified).

This work demonstrates that video content can be processed and organized in a variety of ways, which substantially increase the functionality of a multimedia system. There are some primitive notions of clustering, in the sense that frames belonging to the same shot are represented by the same key-frame, and that the system can associate images having some common characteristics. However, the authors do not comment at all on the implementation issues involved in such an endeavour. There are many technical problems related to the index structures that are required for the plethora of metadata they use, and the fast processing of similarity queries, that are open research problems by themselves.

A study presented by Zhong, Zhang, and Chang [ZZC96] is very similar to the one described above. The aim is to build a system that will allow users to browse through the contents of a video source in a hierarchical manner, and without having to see every single frame.

This work employs techniques analogous to the previous study. The video source is first temporally segmented into the different camera shots. Key-frames are extracted for each segment based on content variation of each shot. Then, characteristic features are extracted from the key-frames, and these features are represented by a l-dimensional vector, where l is the number of different features taken into account.

The new factor in this study is that a clustering algorithm is applied to the key-frames of all the segments (i.e., regardless of their temporal position in the source) in order to construct the hierarchical browsing tree. The clustering algorithm is the traditional K-means algorithm, which assigns each data point to one of the k (prespecified) clusters according to some distance metric. In addition, a *fuzzy* clustering criterion is used. Thus, if a key-frame is almost equally close to more than one clusters, it will be assigned to all of them.

The result of this procedure is a very concise representation of the video sequence, which abstracts many of the details, and tries to produce a summary. However, the authors do not discuss at all the implications of using K-means in this context. This approach is not compared to any other clustering algorithm, and moreover, they do not comment on the selection of, and the effect of altering k which is a user-specified parameter.

Vailaya et al. [VFJZ99] attack a variation of the previous problem. They propose a scheme for classifying still images in a hierarchical classification tree. The purpose is to be able to give more accurate answers to similarity queries, or content based retrieval. When a query comes, the system selects answers only from the class that the query image belongs in.

The system, in the first level, is able to classify images as *indoor*, *outdoor*, or other (e.g., faces). The outdoor images are subsequently classified as *city*, *landscape*, or other. Then, the landscape images are classified as *sunset*, *mountain* and *forest*, or other. At the last level, the system discriminates between the mountain and the forest images. The image classifier is based on a Bayesian Network[4] [Jen96]. There

[4] A Bayesian Network is a concise representation of a joint probability distribution defined on a finite set of discrete random variables. The representation is a directed acyclic network consisting of nodes which correspond to random variables, and arcs which correspond to probabilistic dependencies between the variables. A conditional probability distribution is associated with each node and describes the dependency between the node and its parents. The networks are most often

is no unique classifier, but rather one classifier for each level of the classification tree. The probabilistic models required for the Bayesian classifiers are estimated during a training phase, when the system sees images and is told which class they belong to. Overfitting is avoided by using the *Minimum Description Length* (MDL) principle.[5] The image features that the classifiers use are the following:

Indoor vs. Outdoor: Spatial colour and intensity distributions since outdoor images tend to have uniformity (e.g., blue sky on the top), while indoor images exhibit larger variations.

City vs. Landscape: Distribution of edges, because cities tend to have only vertical and horizontal edges, characteristic of man-made structures, while non-city images have edges randomly distributed in all directions.

Sunset vs. Forest vs. Mountain: Global colour distributions and saturation values since sunsets have typically yellow, orange, and reddish colours, the green dominates in forest images, and images of mountains have the blue of the sky at the top.

The reported experiments show that the system classifies images with an accuracy of as low as 75%, yet, the typical performance is around 90%. These numbers are quite encouraging, especially when considering the fact that they tend to increase as the training set becomes larger. The performance of the system though, is heavily dependent on the quality of the training set. Therefore, the task of collecting a representative set of images for the training phase of the algorithm becomes problematic. Another important issue is the choice of the distinctive features for each pair of classes. There is no robust or automated way for doing so, and it seems that this procedure will get harder as the number of classes increase.

Two studies by Iyengar and Lippman [IL98b] [IL98a] discuss techniques for the classification of video sources into one of two predetermined classes. Specifically, they present a system which classifies movies as *action* or *character* movies, and video sequences as *sports* or *news*. This work proposes the use of *Hidden Markov Models* (HMM)[6] for the classification task. The input to the HMM model for both

used in expert systems which reason under uncertainty.

[5] "We may take it as an axiom that a model or a model class, which permits the shortest encoding of the data, captures best all the properties in the data we wish to learn" [Ris89]. More formally, the best model M describing a set of observations z_i is the one that minimizes $l(M) + l(z_1, \ldots, z_n|M)$, where $l(M)$ is the length in bits of a machine-readable representation of M, and $l(z_1, \ldots, z_n|M)$ is the number of bits needed to encode the observations with respect to M. The first quantity measures the complexity of the model, while the second the degree to which the model accounts for the observations.

[6] The difference from a simple Markov Model is that in the case of HMM neither the transitions between states, nor the states themselves are known. Therefore, the algorithm has to determine the number of states as well.

classification tasks is the distribution of motion energy. For the movies case, an additional feature based on the colour histograms of successive frames is used.

The experiments show that the system achieves a correct classification rate of 90%. Yet, once again, the issues of feature selection, training, and more importantly scalability of the solution are prominent.

A variation of the previous approach is presented by Vasconcelos and Lippman [VL98]. They describe a Bayesian architecture for content characterization of movies. The key-difference here is that they do not try to classify a movie in its entirety, but rather characterize each segment of the movie, and associate it with one of the known classes.

The system first segments the movie in camera shots, and then classifies each segment into *action* (A), *close-up* (D), *crowd* (C), and *setting* (S). The Bayesian network that the system uses is depicted in Figure 8. The information for the classification come from the three modules represented at the

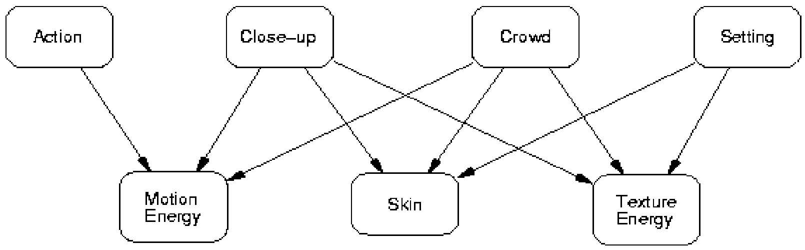


Figure 8: The Bayesian network of the system.

bottom of the figure. The first one accounts for the level of activity present in the shot by measuring the motion energy. The second module identifies human actors based on the colour features of the skin. The third one detects natural scenes as opposed to man-made environments by performing a wavelet decomposition and measuring the ration between the energy in the diagonal bands and that in the horizontal and vertical ones (large ratios indicate natural scenes). As is evident form the figure, the system is capable of integrating information from all three modules during inference, rendering it more flexible.

When the system analyses a movie, it produces a semantic timeline. That is, a brief semantic characterization of the movie in granularity of camera shots. Figure 9 depicts the semantic timelines for two movies. This kind of representation can quickly give information about the content of a video source. In this figure for example, it is evident that the movie on top is a character movie, with many close-up shots and nearly no action, while the second movie has lots of action in a natural setting. The

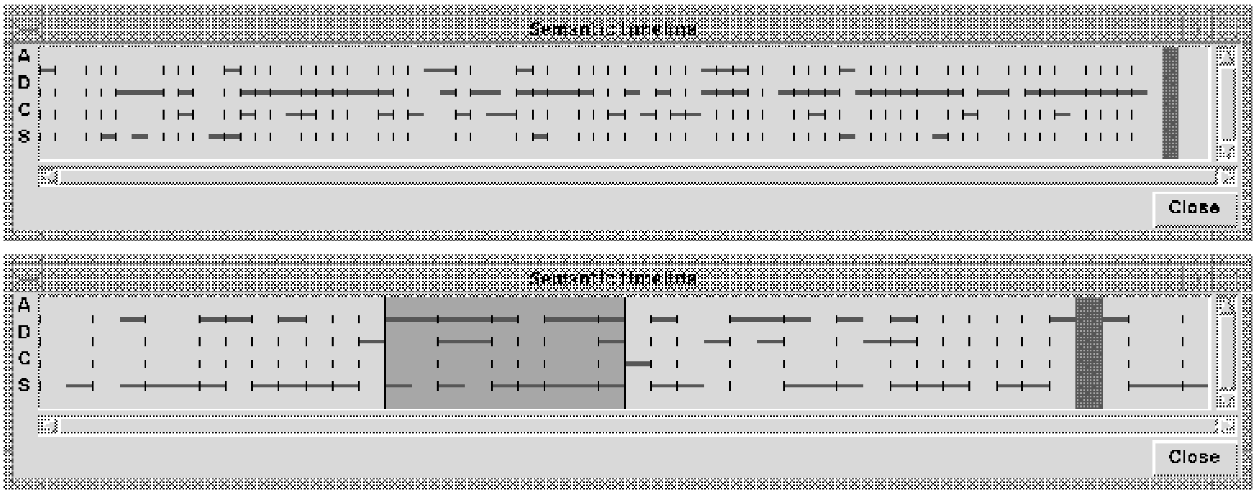


Figure 9: Semantic timelines for two movies.

system may also be used as a browsing tool since it allows the users to swiftly move to parts of a movie with some specific content.

3.2 The Database Perspective

The work on multimedia systems in the database community is not as extensive and diverse as the work that researchers in vision have carried out. There are a few reasons that may explain this phenomenon. First of all, the multimedia database systems are a fairly new trend in the database area. It is only in the last few years that people started to investigate more closely the problems associated with the integration of traditional database technologies and new media such as images and video. We also have to keep in mind that multimedia systems are generally extremely demanding in every aspect: computationally expensive, with considerable storage requirements, and extremely hard to manage in an efficient and systematic way. For many years the database field was not mature enough to deal with this kind of demands.

Lately though, the technological advances coupled with the increasing need for multimedia database systems attracted much attention to this area. Nevertheless, the main focus of the work is the devise of efficient representations and data structures, that will allow the flexible and fast management of any multimedia object. In the following paragraphs we will review some of the work in this area. It will be apparent to the reader that the following studies use techniques that are less vision-oriented and more database-driven. The presentation of the work is divided in two sections. The first one consists of studies dealing with the analysis of image characteristics, while the second proposes general architectures for multimedia database systems.

3.2.1 Image-Based Analysis

Faloutsos et al. [FBF+94] describe the *Query By Image Content* (QBIC) system which involves methods to query large on-line image databases using the images' content (i.e., colour, texture, shape) as the basis of the queries. This paper discussed a set of novel features, and set the grounds for similar work in image databases by bridging the gap between vision and databases.

The QBIC system utilizes some of the techniques studied in the vision community for image analysis and characterization. Specifically, the features captured by the system include colour histograms, shape information (such as area, circularity, eccentricity, major axis orientation of the objects in the image), and texture which translates to coarseness, contrast, and directionality. All the above features are, for each image, converted into vectors which become the representation of the image in the QBIC system. Then, when a similarity query arrives the system simply compares these vectors in order to give the answer. The problems that arise is such a situation are the following. As the database grows bigger, a simple linear scan of all the feature vectors (so as to produce the answer) becomes prohibitively expensive. Therefore, an indexing method must be applied. However, the feature vectors have high dimensionality, which reduces the effectiveness of indexes. Moreover, all the known indexing methods are only applicable to Euclidean feature spaces [SRF97], which is not the case with the distance functions used in this context. For example, the distance function for colour histograms has a full quadratic form involving all cross terms. (In what follows we will not discuss the texture features since they can be accommodated by a weighted Euclidean distance function in three dimensions, and pose no further research problems.)

This study proposes viable solutions to the above issues. The major idea is to use a signature approach, that is to create a filter that will allow some false positives (i.e., as few as possible), but *no* false dismissals. Thus, the goal is to find a mapping of the feature vector $\vec{X}$ into a vector $\vec{X}' = f(\vec{X})$, where $\vec{X}'$ is a vector in a more suitable space, with a distance function $D'()$ which will underestimate the actual distance: $D'(\vec{X}', \vec{Y}') \leq D(\vec{X}, \vec{Y})$. The answerset can then be computed by first applying the fast approximate distance function on the indexed data, retrieve a small subset of the database, and subsequently make a sequential scan to identify the false positives.

For the case of the colour histograms this is achieved by replacing the original distance function d_{hist} with $d_{avg} \leq d_{hist}$. The average colour distance d_{avg} is defined on a different (and significantly smaller) colour feature than the colour histogram. It is merely the distance of the averages of each colour in the colour space over the the entire image. This metric is much cheaper to compute and store, than colour histograms. In addition, it follows the condition that it underestimates the real histogram distance d_{hist}.

For the second problem, of efficiently representing and searching the space of shape features, a dimensionality reduction technique is applied. According to this approach, a high dimensional feature vector is mapped into another vector of equally high dimensionality. The gain is that the transformed vector has most of the information (or energy) in the first few coefficients. Thus, we use only the first few coefficients for indexing those vectors (in a substantially lower dimensionality), which also results in an underestimation of the real distance. The suitable transformations of the above kind form two large families [Pra91]:

- Data dependent transforms, like the Karhunen Loeve transform, which needs a sample of the data to perform statistical analysis, but results in a very good approximation of the original data.
- Data independent transforms, like the Discrete Cosine, Harr, Fourier, or wavelet transform, which yield larger errors, but there is no need for recomputation or update when the data change.

The experiments show that the procedure proposed in this work can considerably reduce the CPU time, as well as the number of I/Os involved in the computation of similar queries in an image database. This results in a significant speed-up for the responses of the system, which may operate in an interactive way with the user.

It is often the case that users cannot precisely express their queries, even when sophisticated graphical user interfaces are provided. Usually, such systems require the user to give a sample query, and to specify the relative importance of colour, texture, and shape attributes. The study of Ishikawa, Subramanya, and Faloutsos [ISF98] proposes an alternative to the aforementioned issue by allowing the user to give several examples, and optionally the corresponding importance scores. Then, the system identifies which attributes are important, it recovers any hidden correlations among them, and it also assigns weights to the parameters under consideration.

The basic idea is to find an unknown distance function that allows not only for different weights of each attribute, but also for correlations. As a means of comparison, the straight Euclidean distance has circles for isosurfaces; a weighted Euclidean distance has ellipses, whose major axis is aligned with the coordinate axis. The authors describe a form of distance functions that result in ellipses, but are not necessarily aligned with the coordinate axis.

The proposed distance function is: $D(\vec{x}, \vec{q}) = (\vec{x} - \vec{q})^T M(\vec{x} - \vec{q})$, where $\vec{q}$ is the "ideal" data point the user is looking for, $\vec{x}$ is one of the data points in the database, and M is a generalized ellipsoid distance matrix (which has the straight and weighted Euclidean distances as special cases). Then, the problem is formulated as following minimization problem: $min_{M,\vec{q}} \sum_{i=1}^{N} u_i(\vec{x_i} - \vec{q})^T M(\vec{x_i} - \vec{q})$, subject to the constraint $det(M) = 1$ (so that the zero matrix is not a solution). The goal is to find the $\vec{q}$ (that the user has in mind) and an M such that the distance of $\vec{q}$ to the N examples $\vec{x_i}$ (that the user has selected) weighted by u_i is minimized. In other words, $\vec{q}$ and M define a generalized ellipsoid in the search space, and we are interested in the data points lying in it.

This method proves to efficiently learn correlated user queries, and converge fairly quickly to the correct solution. Experiments run on a real dataset showed that as few as five examples were enough to drive the system. The algorithm made a very good initial guess, and after six iterations the convergence was almost perfect. Furthermore, it turns out that the proposed distance function is readily supported by the available multidimensional indexing structures [SK97]. However, it is not obvious what features in the image and video databases context this technique will be useful for, and how it is going to be exploited.

A method for analyzing the structure of a video sequence, which is orthogonal to all the approaches proposed so far, is described by Kobla, Doermann, and Faloutsos [KDF97]. This technique reduces a sequence of MPEG encoded video frames to a trail of points in a low dimensional space. Careful manipulation of the frames in this space can reveal structural properties of the video content. For example, we are able to detect gradual edits between camera shots.

The approach of this work is to extract physical features from the consecutive frames in a video clip (stored in MPEG format), and represent them with a sequence of points in a low dimensional space. This procedure will lead to clusters of points whose frames are similar. Consequently, each cluster will correspond to parts of the video source where little or no change in content is present.

The algorithm starts by extracting the DC coefficients[7] of the MPEG frames which refer to the luminance and chrominance components. This yields 1800 DC coefficients per frame (because there are four luminance and two chrominance DC coefficients per macroblock per frame) which composes the feature vector. Since this is an extremely high dimensional space, the authors employ a dimensionality reduction technique, called *FastMap*[8] [FL95], which can effectively reduce the dimensionality to a manageable size, at a very low computational cost (linear to the size of the dataset). An example of the output of the above process is depicted in Figure 10.a. For visualization purposes the frames are mapped to a 3-dimensional space, and the whole sequence of points is referred to as *VideoTrail*.

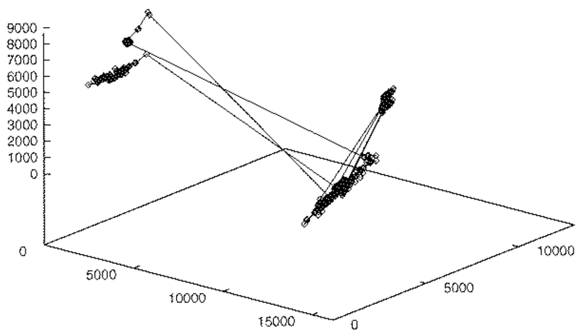
[7]Out of the 64 Discrete Cosine Transform coefficients, the coefficient with zero frequency in both dimensions is called "DC coefficient", while the remaining 63 are called "AC coefficients". Note, that in order to get the DC coefficient there is no need to decompress the MPEG sequence, making the process fast.

[8]An important notion regarding FastMap is that the points and their coordinates in the new lower dimensionality space do not carry any special meaning. It is the relative distances among points that remain consistent. Thus, we can decide on the similarity between points by comparing these relative distances.

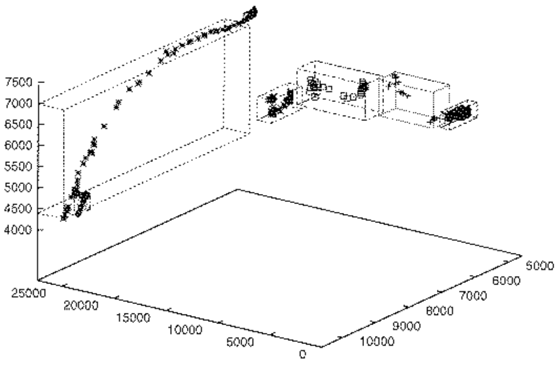
The video clip comes from a news interview. It is evident from the figure that the representation has captured the high-level semantic structure of the video sequence by forming three discrete clusters of points: two for the shots of the two persons involved in the interview, and one more for the shots that both of them were in the picture at the same time.

The aim of the system is to automatically analyze VideoTrails and determine clusters of low activity (i.e., individual shots), and high activity (i.e., transitions and gradual edits). The segmentation of a VideoTrail into clusters is achieved by using a cost function which detects the boundaries of each cluster. Then, the classification of each trail segment into low or high activity is based on the combination of various (weighted) geometric criteria: directional monotonicity, sparsity, convex hull volume ratio, and the shape of the Minimum Bounding Rectangle (MBR)[9].

The system is able to detect most of the transitions occurring in video clips, even when they are not linear, and correctly classify VideoTrails as stationary or transitional (90% accuracy). The large cluster of the trail at the left of Figure 10.b is an example of a fade of one picture into another. The application use of the system discussed in this study is rather high-level, though, there are a



(a) VideoTrail of a news interview, comprised of 3 distinct camera shots.



(b) VideoTrail of a documentary clip containing several shots, and fades.

Figure 10: Examples of VideoTrails.

few promising directions that need further research: classification of action and conversational scenes, key frame selection of camera shots (possibly the centre or the centroid of the MBR of each cluster), and more general video classification (by using graph matching techniques against some known graph-representations of VideoTrails).

[9]Minimum Bounding Rectangle is termed the smallest hyper-rectangle (in an n-dimensional space) that completely and exactly contains a specific set of points.

The work of Shivakumar [Shi99] addresses the problem of finding pirated video sequences. This is essentially the same as answering the question of finding similar copies of video sequences in a huge repository of movies. Though, similarity has a stricter interpretation here, meaning *essentially the same* copies. The main challenges of a video copy detection system are to make the process accurate (i.e., low false positives and false negatives), and scalable to hundreds of thousands of video clips. In addition, the system should be resilient to video source modifications, such as change of format, frame rate, aspect ratio, resolution, and segmentation.

Since scalability is one of the main concerns in this work, traditional techniques used in the vision community are not applicable. For example, features based on optical flow are very laborious to compute and manipulate. This study proposes the use of *temporal signatures* based on shot transitions. Under this scheme, each video clip v is represented by a timing sequence $T(v) = [t_1, t_2, \ldots, t_n]$, where $t_i (1 \leq i \leq n)$ denotes the time (in seconds) at which the i^{th} shot transition occurs in v. The intuition behind this approach is that different movies, even of the same director, will have significantly different temporal signatures.

Then, the question we have to answer is how to efficiently measure similarity between two temporal signatures, taking into account that the algorithm which detects shot transitions may induce timing errors or even misreport transitions. (The shot transition algorithm operates on the DC coefficients of the sequence of images when the video source is in MPEG format. Otherwise, it is based on the Kullback-Leibler distance of the colour histograms of two consecutive frames of the video sequence. Note, that unlike the former option the second approach is not a real-time process.) The similarity function that the author suggests (expressed as a mathematical formula) handles time precision errors by decaying the time differences, and accounts for missing shots by matching each of one video clip with the closest (in time) shot of the other.

The whole database of video clips is indexed using *Locality Sensitive Hashing* (LSH)[10] [IM98]. It turns out that representing each video sequence with 1000 bits is enough for LSH to effectively store them and answer similarity queries (the answer time is less than a second even for several thousands of video clips). The tests revealed that the classes of similar and unrelated movies are well separated and easily identifiable. However, the study does not discuss the problem of finding similar segments of movies.

[10] Locality Sensitive Hashing was recently proposed as an indexing structure for high-dimensional data. It is a hashing scheme with the additional capability to efficiently answer similarity queries.

3.2.2 General Architectures for Multimedia Databases

The *DIStributed Image database MAnagement system* (DISIMA) project [OIÖ98] describes a general framework for multimedia database systems. This framework allows for content-based queries on any multimedia object stored in the database. Yet, this work does not consider at all the problem of semantic extraction. Instead, it proposes an approach for storing and indexing multimedia data, as well as for supporting the content-based queries. This project aims at providing a multimedia storage and retrieval system with functionality from the database perspective. Hence, once the research in the vision and artificial intelligence communities is mature enough to enable us to derive semantic information from the pixels in an image, we will just have to put everything together to make it work. Figure 11 illustrates this integrated image processing environment. The DISIMA project is only focusing on the DataBase

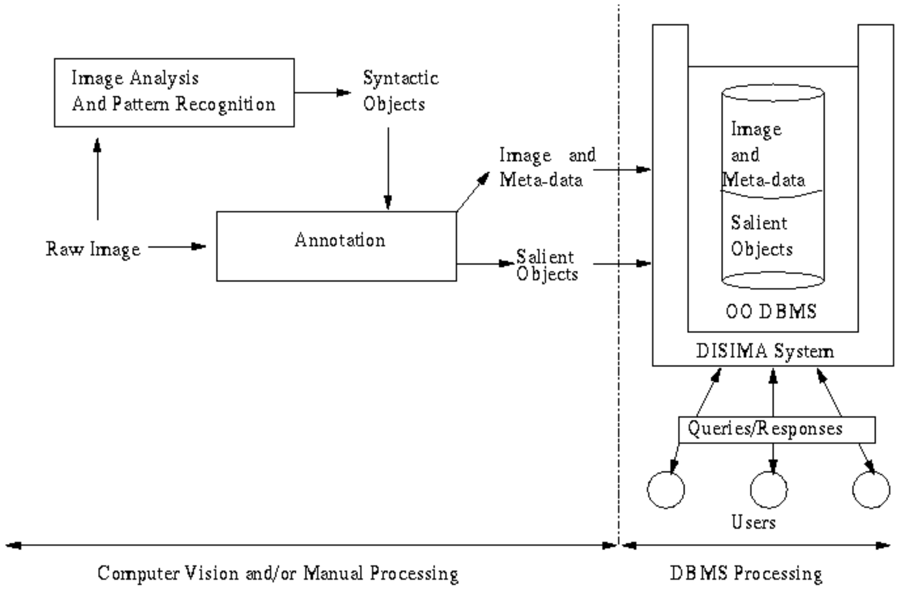


Figure 11: Image processing environment.

Management System (DBMS) related issues, that is the DBMS processing part at the right of the figure.

The system is designed to support a wide range of queries, including queries on the semantics of the image objects. Examples of such queries are the following: find images similar to a sample image (typical similarity query), find images that contain a house, find video clips where Maria Callas is walking behind a parked car. Evidently, in order to answer these questions we need to store in the database a lot of additional meta-information. The DISIMA object model [OÖLL97] tries to capture all this information by decoupling the representation of the image objects from the semantic content of the image. Then, each object that appears in the image is associated with an instance of the corresponding object in the semantic classes hierarchy. For example, a specific set of pixels in the original image which form the head of Maria Callas, is linked to an instance of the class *head*, which in turn is part of the class *human*. In addition, the same set of pixels correspond also to an instance of the class *opera singer*, which is part of *artist* and then *person*. Note, that the management of all the classes and the relationships among them is entirely handled by the database system.

Once we have the objects comprising an image, we can also express in a formal way the spatial relationships that hold among them. The DISIMA system uses a temporal interval algebra [LÖS96] and a compatible query mechanism [LÖ97]. The formulation accommodates both relationships among still objects (i.e., in still images), and objects that change position with time (i.e., in video sequences) [LÖS97]. Then, spatial queries expressed in interval algebra are naturally supported by the database system. Furthermore, a special data structure was developed [NÖL99] in order to support more efficiently the directional and topological relationships that interval algebra allows.

Overall, the DISIMA framework provides an integrated solution for multimedia content representation, indexing, and retrieval. The interesting point is that it provides an architecture that extends the functionality of a multimedia database to encompass semantic information about the objects, and the ability to query this information. Nevertheless, this additional volume of semantic information can quickly grow to unmanageable sizes, even when employing the techniques described in the DISIMA project. Thus, further research towards this direction is necessary.

Zaïane et al. [ZHLH98] [ZHL+98] take a step further in the area of multimedia database systems. They investigate the potential benefits of the application of data warehousing and data mining technologies on multimedia data. This work employs *On-Line Analytical Processing* (OLAP)[11], tools, that allow the user to efficiently create different views of the data. Subsequently, data mining techniques are applied in order to gain a deeper insight of the characteristics of the data.

The system described in this work uses information about the multimedia object file name (i.e., a one word description of the image), its size and format, a list of associated keywords, colour information, and major edge orientations. The data mining modules operate on these data to produce association rules[12] and classifications. Association rules reveal relationships among the data that are not explicit

[11] OLAP requires that the data are logically organized in a multidimensional model. The data of interest are represented through a set of *facts* which are organized according to a set of *dimensions* particular to each application. Semantic hierarchies are also imposed on each dimension. The OLAP model provides efficient ways for doing aggregation, summarization and consolidation of the underlying data. The above operations may be applied on any dimension, and at any level of the hierarchies.

[12] Association rules are rules of the form *if X then Y*, where X and Y are sets of data predicates. Each association rule is assigned a *support* level, which is the percentage of facts in the database that make $X \cup Y$ true, and a *confidence*, which is the percentage of facts in the database containing X that also contain Y.

or easily identifiable. An example of an association rule is that "most of the images related to the sky that are large in size have a light blue colour, while if the images are small it is most probable to have a dark blue colour".

The additional advantage of the data analysis and manipulation techniques presented in this study is that thanks to the OLAP architecture they may operate on different views of the data. These views can be carefully selected to correspond to particular features of interest, and to various levels of the dimension hierarchies. Therefore, the analysis can be as general or as focused as the user requires, and the results more helpful. Yet, the choice of storing all the relevant information in one structure, which renders the data analysis procedure so powerful, is also imposing significant limitations on the scalability of the system. When one tries to remedy this problem by reducing the number of dimensions (i.e., data features) the system keeps track of, the system loses much of its added value. Nevertheless, this area of research is still in its infancy and looks very promising. This approach can be particularly useful especially in the case where semantic information about the multimedia objects is embedded in the system.

4 The Roads of the Future

The work presented in the previous sections is strong evidence that both the vision and the database communities have made considerable progress during the last years. Image processing is a fairly well-studied area that provides the basis for the subsequent analysis phases. It is indeed image analysis that has attracted a lot of attention recently, and seems to be very promising for the future as well.

By image analysis we mean all the high-level manipulation of the primitive characteristics of an image, such as image segmentation, object recognition, hierarchical clustering, and classification. The research in image segmentation into different objects of interest can only allow semi-automatic operation. User interaction is indispensable, and there are no robust techniques that would work across camera shots, where the variability of luminance, position, distance from the camera, and even orientation would vary drastically.

Evidently, object recognition can play an important role in image segmentation. In the ideal case, it would identify the components of an image for different camera shots, but leave the intra-shot segmentation to the less heavy-weight image segmentation algorithms. Object recognition will provide solid indications of object segments irrespective of environment conditions and partial occlusion. There is ongoing research in this area, yet, the performance is limited, especially because of the computation cost. The applications of these kind of technologies are numerous, one example of which is the *Hyperlink*

Television [BDAC98]. A working prototype of this system makes it feasible for users to point and click on objects shown on the television and get back information about them.

The classification and clustering techniques can be an effective way of feature abstraction and summarization. The results of the research in this area are used for quick and efficient browsing of video content. They can also be useful when answering similarity queries since they have the potential to restrict the search space and provide fast and qualitative answers. Future research should focus on ways to make clustering and classification more rigorous by identifying those characteristics of multimedia data that are the most distinctive.

One of the problems inherent in the work of the vision community is scalability. Image analysis requires a substantial amount of computational power, and most of the work presented herein can only handle small scale domains of knowledge. Nevertheless, the scalability problem is of paramount importance for real world applications. Recent research in the database community attempts to answer some of the problems that arise in large multimedia data collections. Yet, these efforts are rather concentrated on some specific topics, which proves the difficulties associated with image analysis and manipulation. It is evident, that this new (for the database world) domain calls for novel indexing structures and algorithms, specifically targeted to image datasets. One of the major challenges is to come up with meaningful and effective feature extraction methods, as well as to design appropriate similarity functions. Novel techniques for doing the aforementioned tasks could also lead to different approaches for asking queries to multimedia databases.

The DISIMA project is the first integrated approach, and rather than providing final solutions to problems it opens several new directions for research. Languages that express spatiotemporal queries about the image objects must be designed, and be supported by both special indexing structures and algorithms that would allow fast processing. The research in the vision community can already support a relatively large portion of the functionality envisioned for the DISIMA system, by providing routines for object recognition, motion detection, tracking, and recognition. The goal to be able to answer semantic queries sets high expectations. This would automate many of the procedures that are now manual and cumbersome. One example is the automatic annotation of multimedia sources, which gets more important as the archives of such data types grow larger. The methodology that would enable a multimedia object to be processed so as to allow querying by concept rather than querying by "pixel" (i.e., using image statistics) is still an open research problem.

A large multimedia database would also serve as a repository of reusable multimedia components. This would allow users to pick objects from the database and combine them to produce new multimedia content. Relevant research [WA94] already provides some functionality for decomposing video clips in layers (with different objects belonging to different layers), and for playing any combination of these layers at will. Once we have a multimedia database, another important question is how to perform information extraction, and exploit the data mining techniques that have been proposed in the databases literature.

5 Conclusions

In this paper we presented an overview of the existing work in the area of image and video databases. We discussed several of the techniques employed for searching, browsing, and annotating multimedia content. Most of this work originates from the vision community. Though, it is not readily applicable to the database context. Therefore, the database community has derived new techniques, as well as adapted the old ones, in order to deal with the complex problems associated with indexing and supporting multimedia data.

Nevertheless, the research area of multimedia databases is in its infancy, and a great amount of work still needs to be done by both sides. The vision and the database communities need to work closer together in order to better understand the problems that exist on the other side, and propose more viable solutions. Multimedia databases are just now beginning to offer a minimal functionality and get the attention of the industry, where the demand for such systems is high. Multimedia databases have certainly a bright future.

Acknowledgements

All the images presented herein come from the corresponding papers.

References

[AB85] Edward H. Adelson and James R. Bergen. Spatiotemporal Energy Models for the Perception of Motion. *Optical Society of America A*, 2(2):284–299, 1985.

[AWP96] Ali Azarbayejani, Christopher Wren, and Alex Pentland. Real-Time 3-D Tracking of the Human Body. In *IMAGE'COM*, Bordeaux, France, May 1996.

[BDAC98] V. Michael Bove, Jr. Jonathan Dakss, Stefan Agamanolis, and Edmond Chalom. Adding Hyperlinks to Digital Television. In *SMPTE Technical Conference*, Pasadena, CA, USA, October 1998.

[CJ96] Edmond Chalom and V. Michael Bove Jr. Segmentation of an Image Sequence Using Multi-Dimensional Image Attributes. In *International Conference on Image Processing*, Lausanne, Switzerland, September 1996.

[FBF+94] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and Effective Querying by Image Content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[FL95] C. Faloutsos and K. Lin. A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In *ACM SIGMOD International Conference*, pages 163–174, San Jose, CA, USA, May 1995.

[IL98a] Giridharan Iyengar and Andrew B. Lippman. Models for Automatic Classification of Video Sequences. In *Storage and Retrieval VI*, San Jose, CA, USA, January 1998.

[IL98b] Giridharan Iyengar and Andrew B. Lippman. Semantically Controlled Content-Based Retrieval of Video Sequences. In *Multimedia Storage and Archiving III, Voice, Video and Data*, Boston, MA, USA, November 1998.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards removing the Curse of Dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613, Dallas, TX, USA, May 1998.

[ISF98] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. MindReader: Querying Databases Through Multiple Examples. Technical Report CMU-CS-98-119, School of Computer Science, Carnegie Mellon University, April 1998.

[Jen96] F. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag, 1996.

[KDF97] Vikrant Kobla, David Doermann, and Christos Faloutsos. VideoTrails: Representing and Visualizing Structure in Video Sequences. In *ACM Multimedia*, Seattle, WA, USA, November 1997.

[KK92] J. N. Kapur and H. K. Kesavan. *Entropy Optimization Principles with Applications*. Academic Press Inc, 1992.

[LÖ97] John Z. Li and M. Tamer Özsu. STARS: A Spatial Attributes Retrieval System for Images and Videos. In *International Conference on Multimedia Modeling*, pages 69–84, Singapore, Singapore, November 1997.

[LÖS96] John Z. Li, M. Tamer Özsu, and Duane Szafron. Spatial Reasoning Rules in Multimedia Management Systems. In *International Conference on Multimedia Modeling*, pages 119–133, Toulouse, France, November 1996.

[LÖS97] John Z. Li, M. Tamer Özsu, and Duane Szafron. Modeling of Moving Objects in a Video Database. In *International Conference on Multimedia Computing and Systems*, pages 336–343, Ottowa, Canada, June 1997.

[MJ92] J. Mao and A. K. Jain. Texture Classification Using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25(2):173–188, 1992.

[MJF94] W. James MacLean, Allan D. Jepson, and Richard C. Frecker. Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM Algorithm. In *British Machine Vision Conference*, pages 175–184, Leeds, U.K., 1994.

[MK97] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons Inc, 1997.

[Nel91] Randal Nelson. Qualitative detection of Motion by a Moving Observer. *International Journal of Computer Vision*, 7(1):33–46, November 1991.

[NÖL99] Youping Niu, M. Tamer Özsu, and Xiaobo Li. 2-D-S Tree: An Index Structure for Content-Based Retrieval of Images. In *SPIE Conference on Multimedia Computing and Networking*, San Jose, CA, USA, January 1999.

[NS98] Randal C. Nelson and Andrea Selinger. A Cubist Approach to Object Recognition. In *International Conference on Computer Vision*, pages 614–621, Mumbai (Bombay), India, January 1998.

[OIÖ98] Vincent Oria, Paul J. Iglinski, and M. Tamer Özsu. A Framework for Multimedia Database Systems. In *African Conference on Research in Computer Science*, Dakar, Senegal, October 1998.

[OÖLL97] Vincent Oria, M. Tamer Özsu, Ling Liu, and Xiaobo Li. Modeling Images for Content-Based Queries: The DISIMA Approach. In *International Conference on Visual Information Systems*, pages 339–346, San Diego, CA, USA, December 1997.

[PN] Ramprasad Polana and Randal Nelson. Detection and Recognition of Periodic, Nonrigid Motion. *International Journal of Computer Vision*, To appear.

[PN94] Ramprasad Polana and Randal Nelson. Low Level Recognition of Human Motion (or How to Get Your Man Without Finding His Body Parts. In *IEEE Computer Society Workshop on Motion of Nonrigid and Articulate Objects*, Austin, TX, USA, 1994.

[Pra91] William K. Pratt. *Digital Image Processing.* John Wiley and Sons, Inc., 1991.

[RBK98] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Rotation Invariant Neural Network-Based Face Detection. In *Computer Vision and Pattern Recognition*, pages 38–44, 1998.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific, 1989.

[Shi99] Narayanan Shivakumar. *Detecting Digital Copyright Violations on the Internet.* PhD thesis, Department of Computer Science, Stanford University, August 1999.

[SK97] Thomas Seidl and Hans-Peter Kriegel. Efficient User-Adaptable Similarity Search in Large Multimedia Databases. In *VLDB International Conference*, pages 506–515, Athens, Greece, August 1997.

[SKG98] Arnold W. M. Smeulders, Martin L. Kersten, and Theo Gevers. Crossing the Divide Between Computer Vision and Data Bases in Search of Image Databases. In *Visual Databases*, Italy, 1998.

[SRF97] Timos K. Sellis, Nick Roussopoulos, and Christos Faloutsos. Multidimensional Access Methods: Trees Have Grown Everywhere. In *VLDB International Conference*, pages 13–14, Athens, Greece, August 1997.

[VFJZ99] Aditya Vailaya, Mário Figueiredo, Anil Jain, and HongJiang Zhang. Content-Based Hierarchical Classification of Vacation Images. In *International Conference on Multimedia Computing and Systems*, Florence, Italy, June 1999.

[VL98] Nuno Vasconcelos and Andrew Lippman. Bayesian Modeling of Video Editing and Structure: Semantic Features for Video Summarization and Browsing. In *International Conference on Image Processing*, Chicago, IL, USA, October 1998.

[WA94] John Y. A. Wang and Edward H. Adelson. Representing Moving Images with Layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[ZHL+98] Osmar R. Zaïane, Jiawei Han, Ze-Nian Li, Sonny H. Chee, and Jenny Y. Chiang. MultiMediaMiner: A System Prototype for Multimedia Data Mining. In *ACM SIGMOD International Conference*, Seattle, WA, USA, June 1998.

[ZHLH98] Osmar R. Zaïane, Jiawei Han, Ze-Nian Li, and Jean Hou. Mining Multimedia Data. In *CASCON Meeting of Minds*, pages 83–96, Toronto, Canada, November 1998.

[ZLSW95] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution. In *ACM Multimedia*, San Fransisco, CA, USA, 1995.

[ZZC96] Di Zhong, HongJiang Zhang, and Shih-Fu Chang. Clustering Methods for Video Browsing and Annotation. In *SPIE Conference on Storage and Retrieval for Image and Video Databases*, San jose, CA, USA, February 1996.

Contents