Content-Based Retrieval from Digital Video

For a Special issue on Content-based Image Indexing and Retrieval
Image and Vision Computing Journal

Author:
Dipl.-Inform. Volker Roth

Fraunhofer Gesellschaft
Institut für Graphische Datenverarbeitung
Rundeturmstraße 6
D-64283 Darmstadt
Germany

Abstract

There is already a huge demand for efficient image indexing and content-based retrieval. With TV going digital, advances in real-time video decompression, easy access to the Internet and the availability of cheap mass storage and fast graphics adaptor cards, digital video will become the next “big” media. Unfortunately, automatic indexing and feature extraction from digital video is even harder than still-image analysis. Presently, automatic analysis of digital video is mostly restricted to automatic detection of scene changes. In this paper we present a framework suitable to immediately explore the consequences of content-based video retrieval with a high granularity of video content. The frameworks employs *semantic networks* to represent video contents on a high level of abstraction and uses time-varying *sensitive regions* to link objects in a video to the knowledge base. A prototype was implemented under NEXTSTEP, exploiting the rich user-interface capabilities of this platform to feature drag & drop queries and authoring of the video retrieval system.

Keywords

Digital Video, Content-Based Retrieval, Semantic Networks, Spreading of Activation.

1 Introduction

Electronic document processing has allowed simple and easy storage of documents – which has led to a sea of documents. With cheap mass storage, the Internet and the World Wide Web at everyone’s fingertips, the volume of easily available multimedia data has mushroomed which makes finding a parrticular bit of information rather hard.

Among the multimedial data, retrieval for text is understood best. Text retrieval actually had a long tradition even before libraries entered the electronic age. A number of techniques such as *inverted indexes, stop-word lists, clustering, relevance feedback* and *thesauri* were developed to aid in automatic indexing and retrieval of electronic texts.

When databases became “multimedia” aware – mostly by adding images to the databases’ contents – the first approach was to annotate the images with text and use the annotations as a basis for retrieval. It was soon recognized that textual queries for images were inadequate and queries were required which are appropriate for the retrieved medium. For instance the fingerprint database of the FBI, for example, contains fingerprint images of more than 25 million individuals [12]. Since the purpose of this database is to identify persons by their fingerprints, queries other than those consisting of fingerprint images do not make sense.

Content-based retrieval of images borrows from fields such as computer vision, pattern matching, cognitive psychology, and many more, in order to extract intrinsic image features suitable for automatic indexing and retrieval. These features are used to reduce the complexity of image comparisons and to improve the organization of image databases.

Unfortunately, automatic retrieval of suitable features is very hard; it is usually only feasible for retrieval systems that incorporate a high degree of domain-specific knowledge about the type of image contents to be retrieved. Features are suitable if they support the computation of similarity measures that are roughly capable of mimicking the human judgement of similarity; after all, content-based retrieval is meant to be for humans, and not for computers.

One such retrieval system is *Photobook* [15], a content-based image retrieval project of the Massachusetts Institute of Technology's Media Laboratory. Photobook comes in three flavours: Face Photobook, Shape Photobook for recognizing images of tools and Texture Photobook. Each category is based on a different content and retrieval model. Face Photobook uses *eigenfaces* and the *distance-from-face-space* calculation for determining the viewing geometry [15, 14, 19]. Shape Photobook uses a *finite element model* of each tool's shape to determine the deformation energy required to align the shapes of two tools [18]. All images must be normalized according to a variety of parameters; for example, Face Photobook requires the images to be normalized for position, scale, brightness, contrast and similar effects. The types of images that can be retrieved with Photobook are quite restricted and each category requires a seperate content representation and retrieval model.

Other content-based retrieval projects such as the *Query By Image Content* (QBIC) project [13] rely on a combination of automatic and semi-automatic image indexing [11]. QBIC uses color features, texture features, shape features and sketch features which are stored along with the images. Furthermore Wang *et. al.* implemented Wavelet-based indexing and sketch retrieval within the QBIC system [22] which supports partial sketches.

Digital video consists of a sequence of images; if content-based retrieval of single images is already hard, we have to question what content-based retrieval from digital video is going to be like. Consequently, the state of the art in video retrieval is considerably humbler in its demands. Current efforts in the automatic video content analysis are directed primarily towards the decomposition of video streams into meaningful subsequences. In this context the term *meaningful* denotes self-contained sequences (scenes) which are perceived by the viewer as continuous action.

A designated frame which is called a *representational frame*, can be chosen from a scene to act as the scene's representative. Browsing the representational frames enables the user to scan through a video in a manner superior to the traditional *fast forward* or *rewind* methods of conventional video cassette recorders.

Several techniques are proposed to automatically segment digital video into scenes, shots and subshots based on color histogram, motion, texture and shape features [23, 2, 3]. The *VideoQ* system described by Chang *et. al.* segments video based on global motion, and tracks objects based on color, motion and edge information [7]. The presented retrieval system also supports sketch queries based on animated sketches, and thus includes motion and temporal aspects of objects occuring in a video. Considerable work is done also in the area of keyframe selection and automatic indexing of digital video directly from the compressed domain of MPEG videos. Kobla *et. al.* describe a scene detection method which analyses the flow information in compressed MPEG videos based on the DCT coefficients and motion vector information [20]. A special problem for scene change detection pose gradual transitions due to fades, dissolves an other special effects edits which are usually found in videos. Kobla *et. al.* also describe an approach dedicated to such transitions which shows good results [21].

The correctness of scene breaks which are detected automatically can be verified visually by the flow of the boundary pixels of subsequent frames [4]. The flow of the boundary pixels for the top edges of a scene's frames is created by arranging the top rows of consecutive frames vertically. The resulting image is called a *motion tracking region*. For undetected scene changes, bars are likely to appear which run parallel to the corresponding frame edge.

Media Streams [9] is the prototype of an application which enables the user to create multi-layered iconic annotations of the video content. An *iconic language* serves to bridge the gap between the natural languages and hence facilitates the interoperability of an application. The objective of Media Streams is to provide a representation of the video content which enables search and retrieval from large video databases.

As a first conclusion, the level of granularity in current video retrieval systems appears to be fairly low and automatic content analysis of video streams is not yet feasible. However, there are advances. Although being based on expensive special purpose hardware such as systolic array processors, experimental autopilots for cars are performing quite impressive. The MIPS per Dollar rate has grown exponentially in the past twenty years and will probably not stagnate in the next few years. Content analysis of digital video might thus become feasible in the future. For the time being, though, we want to settle for a retrieval method that allows to investigate content-based retrieval from digital video *now*. A method that will unfortunately have to build on manual indexing but which is scalable to semi-automatic or automatic indexing when the appropriate technology becomes available.

An appropriate content representation should be rich in its expressive power and it should support a high granularity of the represented contents. We chose *semantic networks* to represent video contents on a high level of abstraction. In Section 2 we briefly continue to discuss content-based retrieval from a more general point of view. Section 3 shows how the parts fit together and explains our general framework for content-based retrieval from video. This framework has also been implemented at the *Center for Computer Graphics*, Darmstadt, Germany; the prototype is presented in Section 5.

2 Content-Based Retrieval in General

Most image and video data are mappings of real-world entities to a binary form. Such data contains two basic types of information; those which refer to attributes and relations of real-world entities (the content) and those which are specific to the real-world entities' binary representations (the encoding). Whichever binary representation (encoding) we choose, the content of the image or video should remain the same.

A scene's *content* remains the same, no matter whether the scene is represented by an array of TIFF images or by a MPEG stream. Neither does a scene's content depend on image resolution. Low image quality and resolution might just make it harder to extract descriptive *features*. The important point is that image and compression type, resolution or color mappings are specific to the binary encoding of the video and not to the content.

Some *features* can be derived from the entities' binary encoding by the application of appropriate decoding procedures, others cannot because no appropriate decoding procedures exist. Those for which a decoding procedure exists are referred to as *content–based non–information–bearing* features; otherwise they are referred to as *content–based information–bearing* [10] features. Decoding procedures for content–based non–information–bearing features are actually affected by the quality and type of the representation's binary encoding.

Content-based information bearing features include semantic information about a video whose extraction requires an amount of knowledge and experience only a human has. The Cyc project [8], an ambitious project in the field of artificial intelligence which is already running since 1984, has the objective to provide a computer with enough common sense knowledge to understand and reason about common texts such as newspapers. To make a computer understand digital video is even more ambitious.

Since content-based retrieval is to support humans, the judgement of what the content is and which contents are similar (for retrieval purposes) should deliver the same results that human judgement does. There is a problem, though; if a retrieval system is given a query showing a particular waterfall, should it retrieve *similar* images of other waterfalls or other images of the same waterfall, maybe from a slightly different viewing angle? The first query is for a set of images that show instances of the same *class* of objects where the membership of this subset is governed by the existence of particular features such as visual appearence. The second query is for other images of the same *instance* that the query contains. A human might recognize the image as one of a particular waterfall and retrieve other images of it.

For content-based retrieval, basically the following types of queries can be identified:

Directly: The user knows exactly what he is searching for, and he knows the exact keys the system uses to identify that particular item.

By Similarity: The user selects one or several documents or parts of a document which are "similar" to the kind of document that he is searching for. This approach is taken in several image retrieval systems such as the QBIC Project [13]. This project employs similarity measures based on color distribution and texture.

By Prototype: This technique is related to the previous one. The prototype may be a rough "sketch" created by the user at query-time or an item that is interpreted by the system to produce a particular representation. This representation is then matched against the database's contents using a similarity measure [13, 22].

One additional retrieval mechanism that should be provided is *browsing*. Ideally the browsing mechanism should work together with the retrieval mechanisms mentioned above so that users can browse the whole database as well as the results of a query. Moreover the user should be able to choose examples from the database browser and input them to queries.

The presentation of query results is also important. Rather than displaying full versions of the (possibly huge number of initially) retrieved documents, significant representations of them should be used instead. In general, such representations are icons, miniatures of the original or descriptions.

3 Framework

For a readily explorable framework for content-based video retrieval we decided to use *semantic networks* for the representation of video contents. A *semantic* *network* is one knowledge representation among a variety of possible others, each having advantages and disadvantages with respect to certain applications. Other examples of knowledge representations are *logic, frame-like representations* and *rule-based production systems*.

Semantic networks were introduced as a model for the human cognition in the cognitive psychology field. The basic assumption is that semantically related concepts are connected by *associative links*. Such models are usually called *associative memory models*. Recall is achieved as follows: if a concept is activated activation spreads from this concept via the associative links to the related concepts. If the activation received by a concept reaches saturation (by exceeding the threshold of consciousness) it is "remembered".

Associative models can be portrayed by networks of vertices which are connected by edges. Several different possibilities exist to represent inheritance, attributes, concepts and relations. The variant we chose is best descibed as *propositional network* (see [1] for details). A detailed introduction to knowledge representations is for example [16]. A recommended introduction to logic and automated theorem proving is [5] which covers a broad range of topics. A discussion of knowledge representation from cognitive psychology's point of view can be found in [1]. For the remainder of the paper we assume that the reader is familiar with the notion of propositional networks.

In order to support video retrieval the concepts of the knowledge base must be linked in some way to the video(s) represented by it. At least the principal entities visible in a video, their actions and their attributes should be represented by the knowledge base. These entities are referred to as *objects*. We furthermore demand that the granularity of retrieval should enable retrieval on the level of object occurrence. This means that a query for frames showing Indiana Jones fighting off snakes with a torch should retrieve exactly the frames of a movie which show this kind of action (provided that the knowledge base contains information about the query's subject in the desired detail). In addition to that it should be possible to highlight the objects returned by a query within the frames they occur in.

The demands can be fulfilled by combining propositional networks with *sensitive regions* [6]. Sensitive regions are a generalization of the *anchor* concept common in hypermedia models towards a mechanism which is also applicable to time-varying media such as video and audio data. An anchor (in the hypermedia sense) is a sub-space of a document which can be activated by the user. Sensitive regions can be thought of as "hot–spots" superimposed over video frames to delineate the outline of the regions of interest of those frames. Video sequences exist in three dimensions. These three dimensions consist of the two dimensions of the plane on which the frames are displayed plus the time axis. Objects visible in the video sequences must have an anchor representation for each frame in which they appear. Therefore, the sensitive regions can be described in a video sequence by a three-dimensional volume. A simple implementation of sensitive video regions involves polygonal areas and a linear inter-frame interpolation in order to reduce the number of required polygones. Morphing is used to interpolate between polygons with different numbers of control points.

By linking concept instances in the knowledge base to the visible occurrences of the represented objects with sensitive regions and vice versa, knowing a sensitive region is made equivalent to knowing the instance node it belongs to. Hence, each sensitive region references the concept instance it represents, and each concept instance with a graphical counterpart visible in the video might have a (reference to its) sensitive region. This is illustrated in Figure 1. In general, every instance with a counterpart visible in the video might have a sensitive region; Torch-1 thus may also have a sensitive region if a higher grade of detail is desired.

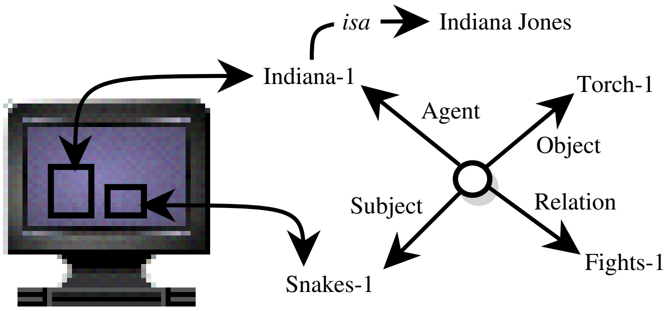

Figure 1: Concept instances accessed through sensitive regions.

If, as the result of a query, the node for the instance Indiana-1 is activated then the sensitive region of this node is immediately known. This sensitive region identifies firstly the frames showing Indiana Jones fighting off snakes with a torch, and secondly the exact location of that particular occurence of Indiana Jones within the frames. The node's name is Indiana-1 rather than Indiana because it represents Indiana Jones in a particular state. Indiana-1 is therefore an instance of the class of occurrences of Indiana Jones in the video. Representing Indiana Jones driving a truck named Truck-1 later in the video requires a node of the same class which might have the name Indiana-2.

Those concepts that do not have a visible counterpart in the video are represented by a graphical icon (e. g. a shaded sphere) to allow their graphical manipulation (such as querying and linking).

Concepts are linked simultanously in a class hierarchy as well as to concepts to which they are related. In our prototype implementation we actually do not

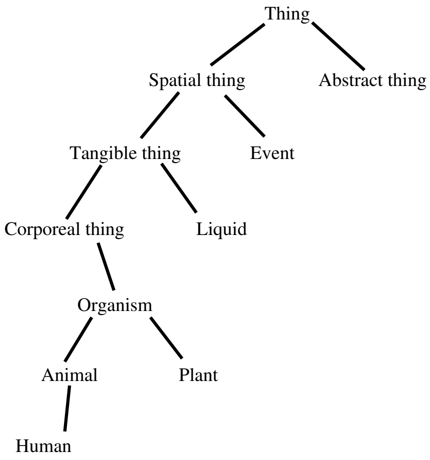

Figure 2: An example of an ontologic concept hierarchy taken from [16]

differentiate between class links and links to related oncepts. All links are untyped but may be weighted with a factor. The factors of an example knowledge base are set to prefer direct relations over inheritance.

The concept hierarchies to be used should be structured according to the basic units of human perception and human thinking. Such hierarchies are called *ontologic*; An example of an ontologic concept hierarchy is shown in Figure 2.

Basically, two approaches exist for querying semantic networks which are *subnet matching* and *spreading of activation*.

With subnet matching, queries to a semantic network representation are mapped onto a query-network; this means that the query itself is formulated as a network which might contain variable nodes. If the query represented by a query–network can be answered with “yes” then the instantiation of the variable nodes is the result of the query. The mapping process tries to determine if the query–network is a subnet of the semantic network by matching nodes and links of equal type starting with an arbitrary (non-variable) node of the query-network. If the query-network can be made equal to a subnet of the semantic network the query is answered “yes”, otherwise it is answered “no”.

However, the nodes of concept classes can be interpreted as to match their subclasses. Hence, the matching process may be guided by rules of inference (e.g. the transitivity of the *isa* relation). The possibility to issue queries with variable nodes results in a high complexity of the matching process. The functionality of the matching process is comparable to that of an automatic theorem prover. The semantic net and the inference rules resemble the axiom system and the query-network resembles the theorem which is to be proven.

In contrast to the matching mechanism described above which only supports queries with variable nodes, spreading of activation is able to determine paths between arbitrary nodes of a network. The underlying principle is rather simple. Starting from the nodes activated by a query, activation is propagated to the direct neighbors. This process is iterated for the neighbors and their neighbors and so forth. In analogy to biological neuron cells the activation propagated from one node to another is called a *spike* of activation. If a node is activated by two different spikes a path between two query nodes is found. The meeting spikes should be deleted because nothing new can be learned from a further propagation of them. However, further propagation may return additional paths of other spikes. The spreading of activation is thus similar to the algorithms for finding the shortest path between vertices in a graph known from graph theory.

The longer the path from one concept to another is, the smaller is the concepts' semantic closeness. Spreading of activation can therefore be used to determine the semantic closeness between concepts. It may also be desirable to restrict the paths the activation can take according to certain rules. Paths may for example be limited to a maximum length. The links between the nodes may also have weights, for instance in the range from 0.1 to 0.9; a weight which is less than 1 guarantees that the activation is automatically lowered with an increasing path length.

If multiple nodes are combined in a query then these nodes might be processed in order with their spikes being superimposed on the net. Every node which receives activation must register with a filter procedure; this procedure stores references to the n nodes which received the most activation. The nodes with the most activation are then the candidates for the query result. The filter procedure may also turn down nodes of an undesired type. Using this approach the query might be refined gradually according to intermediate results.

4 Discussion

The basic principle of semantic networks is that nodes stand for concepts and links stand for relations between concepts. This has several advantages and disadvantages. Certainly, the representation of some aspects of knowledge such as negation are complicated with semantic networks especially if for example a negation should be applied to a link rather than a node. Nevertheless, semantic networks have the same expressional power as other representations such as logic have.

An important aspect of semantic networks is the reflection of the semantic closeness between concepts based on the links. The longer the path from one concept to another one is, the less close are these concepts semantically. This results in a localization or clustering of related concepts which reduces the search complexity for queries considerably in contrast to unstructured representations such as logic. In an unstructured representation such as logic it is very difficult to restrict the number of possible combinations that must be put into consideration for an inference. Whereas in logic the number of possible combinations explodes, semantic nets merely require a local search. Furthermore semantic nets are well suited for parallel architectures particularly if spreading of activation is used as a query mechanism.

Spreading activation also enables queries which, if expressed in logic, would require at least second-order logic. Unfortunately, the unification in second-order logic is not decidable [5]; therefore it is not possible to create a general automatic theorem prover for the second-order logic.

Another advantage of the semantic networks over the first-order logic is the simplicity of the inference mechanism in comparison to first-order logic. In this context it should be noted that languages such as Prolog which are based on the resolution mechanism, actually resemble only a subset of first-order logic for which queries are decidable. For a discussion of these topics see [5].

Sensitive regions are an easy-to-use mechanism for issuing queries which is compatible to current hypermedia systems such as the World Wide Web. Our prototype implementation features drag & drop issuing of queries and authoring of knowledge bases (e. g. creating associative links between concepts).

The proposed framework can be complemented using existing image analysis technology towards a semi-automated indexing system in a number of ways. Firstly, object tracking and separation could be used to generate and propose sensitive regions automatically. Ideally no human interaction is needed. However, a tool which allows to mark the boundaries of the first and last occurrence of a concept in a video sequence, and which computes the intermediate polygones automatically using object tracking, would facilitate the annotation process significantly. Secondly, sensitive regions, wether they were generated automatically or manually, may guide the automatic classficiation and recognition of meaningful objects in a video by excluding the disturbing “noise” outside the sensitive regions.

Automated object recognition might also be facilitated by extensions to the proposed framework. For this purpose the concept class nodes might also have sensitive regions. These regions then point to a particularly crafted video showing normalized sequences of prototypical objects. Every such sensitive region then links a concept node to a sequence showing the prototype of the class it represents. A hypothetical object recognition algorithm may thus descend the hierarchy by deciding which concept prototype is most similar to the object in question. If the concept hierarchy more or less forms a tree (and has not degenerated to a linear list) this may reduce the number of picture-to-picture comparisons to a number which is logarithmic instead of linear in the total number of images. The hierarchy might thus be used to implement a method comparable to *clustering* techniques.

Alternatively, the nodes can be modelled and implemented in an object-oriented way. Thus the nodes of a concept might include methods to decide if a given video sequence delineated by the node's sensitive region fits in the class represented by the node. These methods might also be subject of inheritance. The methods employed by the concept nodes might furthermore use prototype representations other than bitmaps defined by sensitive regions.

5 Prototype Implementation

A prototype of the framework described in Section 3 was implemented at the *Center for Computer Graphics* in Darmstadt under NEXTSTEP using Objective-C. The implementation is completely object-oriented and makes use of the rich user interface cababilities of the NEXTSTEP environment. Almost all important authoring and querying steps are easily done by dragging & dropping rectangular clips from a video, from a browser that shows query results, and other user interface components.

Only the basic functioning of the prototype can be described here. For a more complete description see [17]. The user interface consists of five components:

Hierarchy Browser: This tool is for browsing and authoring the concept hierarchy. Concept nodes of the propositional network that represents a video's contents can be created, subclassed and deleted easily using this tool. Dragging an iconic representation of a concept (such as a clip from the frame shown in the video viewer) onto the hierarchy browser highlights the class path from the root of the hierarchy to the represented concept. Clicking on a concept in the browser selects it for inspection and detail editing in the *inspector*.

Inspector: The inspector allows the creation of associative links to other concepts by simply dragging e. g. a clip of another concept onto the list of links shown in the inspector window. Furthermore, sensitive regions can be added to concepts and edited by adding control polygones. These are shown and can be adjusted by dragging them in the *video viewer*.

Video Viewer: The video viewer is used to play a video. It offers a simple VCR like interface. The sensitive regions that delineate the objects which are represented by concepts in the knowledge base, are superimposed onto the video images. The sensitive regions consist of rectangular areas in a sequence of video frames that can be dragged off the frames to various other user interface components.

Query Window: Dragging concept references such as clips from sensitive regions onto the query window activates these concepts and activation spreads through the propositional network. Concepts which receive more activation than a certain threshold register themselves with the *dock view*.

Dock View: The dock view shows the results of a query. Each concept which is part of the query result is represented by a clip of its sensitive region or a shaded sphere if the concept does not have one. The list is ordered left-to-right which means that concepts that are more relevant to a query are shown further to the left. Clicking on a concepts icon in the dock view plays back exactly the video sequence in which the concept appears in the video viewer.

Queries are implemented by spreading of activation. Starting with the query node, an initial spark of activation is spread using iterative breadth-first traversal of the weighted links originating from the query node. Activation spread over a link is multiplied with the link's weight. The activation spread by successive activation of nodes (e. g. by successively dropping nodes onto the query window) is superimposed until a new query is started by clicking on the *new query* button in the query window. With each query, a query number is increased by one, which eliminates the need to clear the semantic net prior to issuing a query. The depth of queries is limited by a parameter which specifies the maximum path length.

5.1 A Sample Knowledge Base and Query

In this section a sample query is described. For demonstration purposes five scenes from the motion picture "True Lies" were recorded and modelled with a semantic network representation using the prototype application.

5.1.1 Sample Scenes

The movie "True Lies" is a spy story; the main character is a secret agent with the name Harry Tasker. Additional characters occuring in the sample scenes are Harry's wife Helen, Juno Skinner, who is an accomplice of the movie's "bad guys", an anynomous couple, and a guard. The scenes recorded for demonstration are as follows:

1. Harry Tasker dances with Juno Skinner. This is an early scene in the movie, playing at some party in a Swiss villa.

2. Another man and a woman at the party are dancing.

3. Harry Tasker leaves the Swiss villa and is approached by a guard who wants to verify Harry Tasker's invitation card (which he does not have). In order to derange the villa's security Harry detonates a bomb he had prepared on his secret approach to the villa. The device used to trigger the explosion is a radio transmitter hidden in the cigarette case Harry holds in his left hand.

4. Helen is forced to mimic a prostitute dancing for a customer in the apartment of a hotel.

5. Harry kisses his wife Helen he just rescued from her confinement by the terrorists. While he kisses her a nuclear warhead explodes in the scene's background.

Three of the five scenes involve dancing and two scenes involve explosions such that the effect of queries to various occurrences of a similar event can be investigated. Scene number three can be used furthermore to demonstrate a causal relationship.

5.1.2 Example Knowledge Base

All principal characters visible in the movie are represented in the knowledge base. This includes all occurrences of Harry Tasker, Helen Tasker, Juno Skinner, the guard, an unidentified woman and an unidentified man. Harry Tasker appears in three of the five scenes playing at three different points in the movie's subjective time. Therefore, the concept *Harry_Tasker* representing the movie character Harry Tasker, is a concept class with three concept instances numbered from one to three. The concept instance *Harry_Tasker2* for example represents Harry Tasker when he dances with Juno Skinner. In general, the concept instances of a concept class with more than one instance are labeled with the name of the concept class concatenated with a running number. The instances of the abstract relation *Cause* are thus labeled *Cause1, Cause2,* and so forth.

The most prominent actions of these characters are also represented in the knowledge base. Among the represented actions are *dancing, pressing devices, holding* and *wearing* items of different kinds.

Several items such as the necklace worn by Juno Skinner, the detonator held by Harry Tasker, and the flashlight held by the guard appearing in the video pictures are also represented in the knowledge base. Additionally, various clothes worn by the movie characters are represented.

The modelling of complex events might require the representation of causal dependencies between actions and events. An example of such a causal dependency is the explosion Harry Tasker triggers using the detonator. This fact is

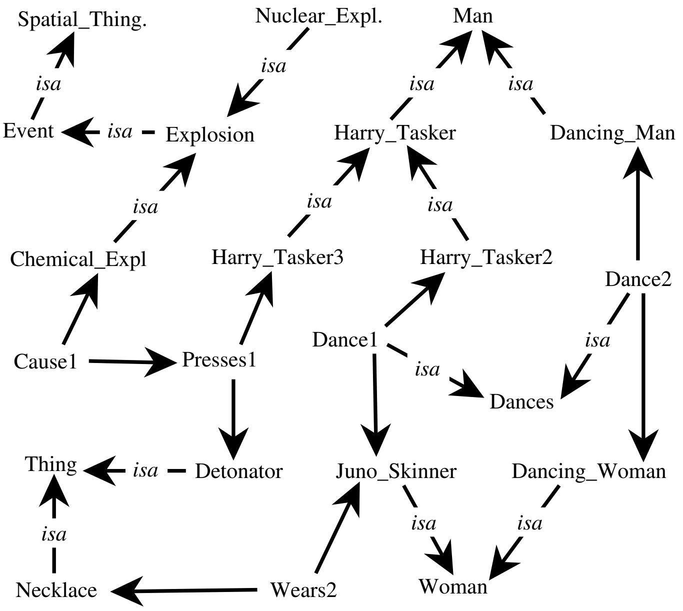

Figure 3: An excerpt of the semantic network representation of scenes from the motion picture "True Lies"

represented by the abstract concept class *Cause*. The representation of causal dependencies enables queries such as *why* something happened in a movie and what the *result* of a character's action was.

Temporal aspects of concepts are modelled by distinguishing different temporal states of a concept and creating an instance for each of these states. Each instance is then connected exactly to those concepts it is related to *at the time of its validity*. Relations which are valid independently of a concept's state are represented by connections to the concept class.

A two-dimensional layout of the example knowledge base would be rather confusing. Therefore only an excerpt of the example knowledge base is given in Figure 3. The concept hierarchy of the example knowledge base builds on the ontologic concept hierarchy outlined in Figure 2. An excerpt of the hierarchy is given in Figure 4.

Although the edges in Figure 3 are directed, activation is spread in either direction of the edges. The amount of activation spread over an edge depends on the edge's weights. An edge is a bidirectional connection consisting of two unidirectional *links* with separate weights. Every link in Figure 3 pointing in the arrow's direction has a weight of 0.95 except the *isa* links which have a weight of 0.5; a

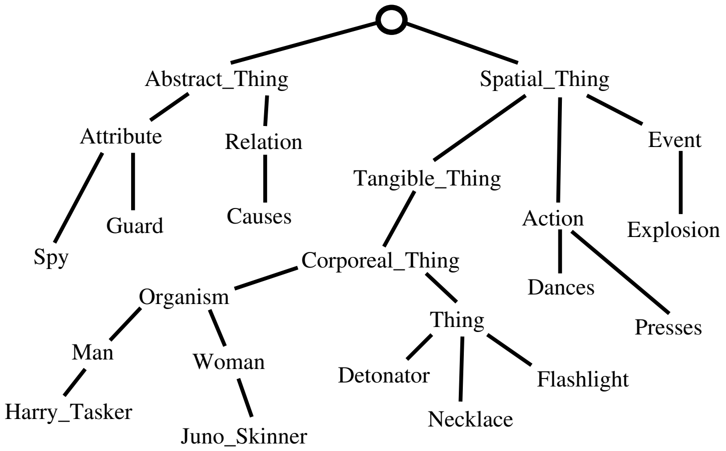

Figure 4: An excerpt of the concept hierarchy for the representation of scenes from the motion picture "True Lies"

weight of 0.9 is given to all links pointing in the opposite direction. Weighting *isa* links with 0.5, and weighting links pointing away from the semantic relations with 0.95 (such as the link from *Dance1* to *Juno_Skinner*) results in a preference of direct associations over inheritance. This improved the quality of query results for several sample queries.

5.1.3 Example Query

A query is issued by adding node references to the query window's browser. This can be done in a straightforward manner by dragging the icon of a node onto the browser. For nodes having a sensitive region these icons can be dragged directly from the video picture visible in the video viewer.

Dragging icons from the video viewer to the query browser is comparable to a *query by similarity* whereas dragging the icons of concept classes from the hierarchy browser to the query browser is comparable to a *query by prototype* (see Section 2).

We now describe a query–by–example for video sequences of the motion picture in which the movie's hero (Harry Tasker) is related to explosions. The desired video sequences are thus associated to Harry Tasker and to explosions and no other information is given. Therefore, the concepts *Harry_Tasker* and *Explosion* must be activated. The most straightforward method is to select two arbitrary video pictures, one showing Harry Tasker and another one showing an explosion. Two such pictures from the example scenes are shown in Figure 5. Picture (a) shows Harry Tasker dancing with Juno Skinner; both enclosed by sensitive regions. The

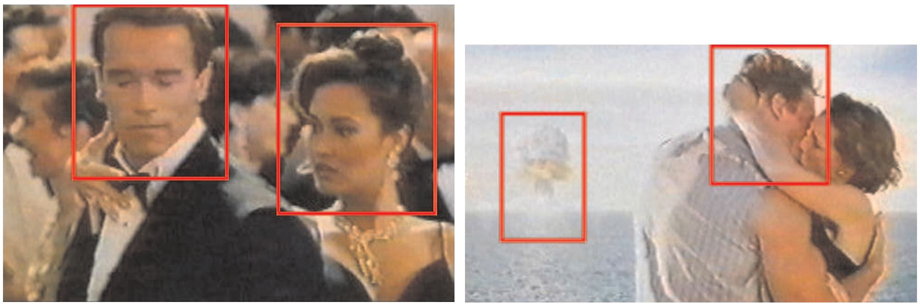

Figure 5: Two example pictures from two different scenes of the motion picture “True Lies”. The sensitive regions defined for these pictures are displayed as rectangles.

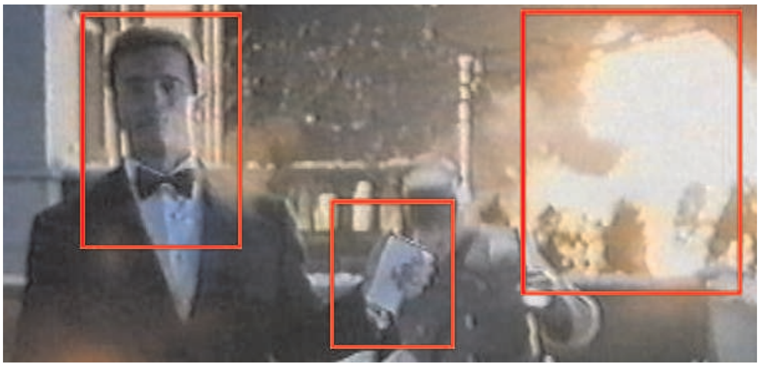

Figure 6: The last picture of a scene which is retrieved by the example query.

icon from the sensitive region enclosing Harry’s image is dragged to the query browser. The Dock View immediately responds by displaying the icons of concepts associated with Harry Tasker. From picture (b) the icon from the sensitive region which encloses the nuclear explosion is dragged to the query browser. The Dock View rearranges its display to show the results of the refined query.

The first four icons displayed in the Dock View are shown in Figure 7. They stand for the nodes representing the concepts *Nuclear_Expl, Chemical_Expl, Harry_Tasker3* and *Detonator*. Clicking on the icon labeled “Harry_Tasker3” plays back the scene in the Video Viewer which shows Harry Tasker leaving the Swiss villa which he burglarized. Then a guard approaches him which he distracts by exploding a bomb remotely with a radio detonator hidden in his cigarette case. The playback stops at the picture shown in Figure 6. The white rectangles visible in this picture are the sensitive regions associated with the concepts *Chemical_Expl, Harry_Tasker3* and *Detonator*. The original picture shows additional

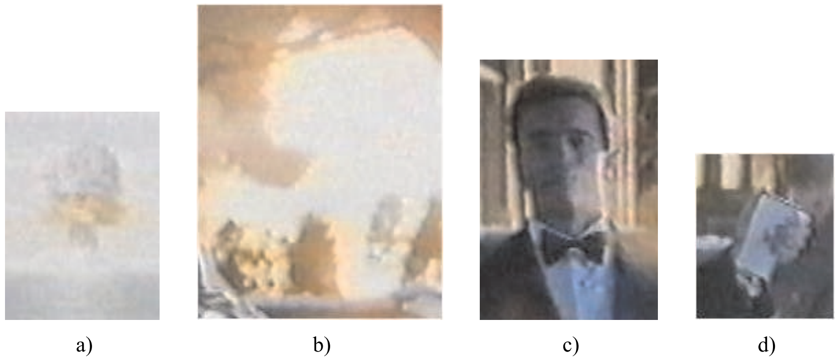

Figure 7: The four icons representing the concepts most relevant to the example query; (a) Nuclear_Expl, (b) Chemical_Expl, (c) Harry_Tasker3, (c) Detonator

sensitive regions which are left out in the figure for clarity.

Clicking the icon representing *Chemical_Expl* would play back the video sequence showing the explosion. Apart from the first picture shown in Figure 6 the explosion scene does not show Harry Tasker anymore.

The simplest possible query consists of a single concept. The result of such a query will consist of the concepts directly associated with the query node. A query consisting of the concept *Dancing_Man* thus returns the icons representing the concepts *Dancing_Man, Dancing_Woman, Harry_Tasker2, Harry_Tasker1, Harry_Tasker3* and *Security_Man* in order of decreasing relevance to the query.

The concept *Dancing_Woman* is connected with the query node by the semantic relation *Dance2*. The other retrieved concepts inherit from *Man* which is the concept class the query node belongs to. Because links other than *isa* links are preferred (see Section 5.1.2) the concept *Dancing_Woman* is considered more relevant to the query than the concepts inheriting from *Man*.

Adding the concept *Dancing_Woman* to the query retrieves the concepts *Dancing_Man, Dancing_Woman, Harry_Tasker2, Juno_Skinner, Helen_Tasker* and *Harry_Tasker1* in order of decreasing relevance. The query nodes are both related to *Dance2*. *Dancing_Man* inherits from *Man*, *Dancing_Woman* inherits from *Woman*. *Harry_Tasker2, Juno_Skinner* and *Helen_Tasker* are connected to instances of *Dance* and inherit from either *Man* or *Woman*.

This query can be verified graphically using Figure 3. The two queries discussed above demonstrate how queries are issued to the prototype's retrieval mechanism. The second example furthermore illustrates the processes that lead to a particular query result.

6 Conclusions

In their advertising campaigns companies such as Sun, Apple, Intel, IBM, or Microsoft fight for the goodwill of the customers by emphasizing the multimedia capabilities of their computer hardware or operating systems, and video is an integral part of the advertised multimedia capabilities.

Video conferences or playing music over an Ethernet is already common using Sun workstations. In the personal computer market 32 Bit sound, full-MIDI support, MPEG recording from live video sources and displaying TV channels in a window are common.

The increasing use of video requires the development of new technologies to manage video data. However, in particular the automatic analysis of video content poses great technical problems. The delicacy of the automatic analysis of video content is demonstrated already by the problems imposed by the automatic content analysis of generic pictures. Although several content-based image retrieval systems achieve impressive results the performance of these systems is often based on a rigorous restriction of the type of manageable documents. Applications which retrieve generic pictures often build on semi-automatic content analysis because purely automatic mechanisms are not considered robust enough.

The automatic content analysis of video is even more complicated and less feasible than the automatic analysis of image content. The present state-of-the-art in the automatic content analysis is more or less restricted to automatic scene detection unless extraordinary fast and expensive hardware is involved.

This paper proposes a framework for the content-based video retrieval which might enable the unison of various efforts in this area. This framework is applicable on average workstations and PCs. It furthermore satisfies the demand for a semantic representation of video content and a query mechanism that are based on a model of human cognition. This includes the ability to model changes of video objects over time and causal dependencies between distinct events occurring in a video.

The organization of the proposed knowledge representation furthermore features an inherent clustering which reduces the amount of information that must be taken into consideration for a query. Additionally this organization is well-suited for parallel computation and it is adaptable to distributed environments.

As soon as the mechanisms for automatic content analysis become sufficiently robust these mechanisms can be integrated into the proposed framework such as the automatic extraction of objects from video. This mechanism could be used, for instance, to create the sensitive regions automatically.

The user interface enables the making of queries and the authoring of the knowledge base in a straightforward manner using primarily Drag & Drop operations. The user does not have to learn a complicated query language.

We consider the approach taken in this paper a step towards feasible user-friendly content-based retrieval from video. Content-based retrieval is a worthwhile research subject, and the human fascination of moving pictures and the desire for a quick and easy access to the most favorite of these moving pictures will ensure a long-lasting interest in the content-based retrieval from video.

References

[1] ANDERSON, J. R. *Cognitive Psychology and its Implications*. Freeman and Company, New York, 1990.

[2] ARDIZZONE, E., CASIA, M. L., GESÚ V. D., AND VALENTI, C. Content-based indexing of image and video databases by global and shape features. In *Proceedings of the International Conference on Pattern Recognition* (1996), pp. 140–144.

[3] ARDIZZONE, E., CASIA, M. L., AND MOLINELLI, D. Motion and color-based video indexing and retrieval. In *Proceedings of the International Conference on Pattern Recognition* (1996), pp. 135–139.

[4] ARMAN, F., DEPOMMIER, R., HSU, A., AND M.-Y. CHIU. Content-based browsing of video sequences. In *Proceedings of ACM International Conference on Multimedia '94* (1994).

[5] BIBEL, W. *Deduktion*. R. Oldenbourg Verlag, Wien, 1992.

[6] BURRILL, V., KIRSTE, T., AND WEISS, J. Time-varying sensitive regions in dynamic multimedia objects. *Information and Software Technology 36*, 4 (1994), 213–223.

[7] CHANG, S.-F., CHEN, W., MENG, H. J., SUNDARAM, H., AND ZHONG, D. VideoQ: An automated content based video search system using visual cues. In *Proceedings of The Fifth ACM International Multimedia Conference* (November 1997).

[8] CYCORP. Cycorp, Inc. (CYC), Version current March 6th 1998. Web page at URL <http://www.cyc.com>.

[9] DAVIS, M. Media Streams: an iconic visual language for video annotation. *Telektronikk 89*, 4 (1993). Norwegian Telecom Research.

[10] GROSKY, W. I. Multimedia information systems. *IEEE Multimedia 1* (1994).

[11] J. ASHLEY *et. al.* Automatic and semiautomatic methods for image annotation and retrieval in QBIC. In *Storage and Retrieval for Image and Video Databases III* (1995), vol. 2420 of *Proc. SPIE*, pp. 24–25.

[12] JAMES DOWE III. Content-based retrieval in multimedia imaging. In *Proc. SPIE* (1993), vol. 1908.

[13] M. FLICKNER *et. al.* Query by image and video content: The QBIC System. *IEEE Computer 28*, 9 (Sep. 1995).

[14] PENTLAND, A., MOGHADDAM, B., AND STARNER, T. View-based and modular eigenspaces for face recognition. Perceptual Computing Technical Report 245, Massachusetts Institute of Technology Media Laboratory, Cambridge, MA02139, 1994.

[15] PENTLAND, A., PICARD, R. W., AND SCLAROFF, S. Photobook: Tools for Content-Based Manipulation of Image Databases. In *Storage and Retrieval of Image and Video Databases II* (1994), vol. 2185 of *Proc. SPIE*, pp. 34–47.

[16] REIMER, U. *Einführung in die Wissensrepresentation.* Teubner Verlag, Stuttgart, 1991.

[17] ROTH, V. Content-Based Retrieval from Video. Diploma Thesis, Technische Hochschule Darmstadt, February 1995.

[18] SCARLOFF, S., AND PENTLAND, A. A finite-element framework for correspondence and matching. In *4th International Conference on Computer Vision* (Berlin, Germany, May 1993), pp. 308–313. Also available as: M.I.T. Media Laboratory Perceptual Computing Technical Note No. 201.

[19] TURK, M., AND PENTLAND, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* (May 1991).

[20] V. KOBLA, D. S. DOERMANN, K., AND FALOUTSOS, C. Compressed domain video indexing techniques using DCT and motion vector information in MPEG video. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases V* (February 1997), vol. 3022, pp. 200–211.

[21] V. KOBLA, D.S. DOERMANN, K.-I. L., AND FALOUTSOS, C. VideoTrails: Representing and visualizing structure in video sequences. In *Proceedings of The Fifth ACM International Multimedia Conference* (November 1997).

[22] WANG, J. Z., WIEDERHOLD, G., FIRSCHEIN, O., AND WEI, S. X. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries* (May 1997).

[23] ZHANG, H. J., LOW, C. Y., SMOLIAR, S. W., AND WU, J. H. Video parsing, retrieval and browsing: An integrated and content-based solution. In *Proceedings of the ACM Multimedia Conference* (1995), pp. 15–24.