

|||
Consiglio Nazionale delle Ricerche

|||
**ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE**

PISA

IST. EL. INF.
BIBLIOTECA
Posiz. *ARCHIVIO*

&
Considerations on a Model for Video Data

Giuseppe Amato Carlo Meghini

Nota Interna B4-42

Dicembre 1995

Considerations on a Model for Video Data

Giuseppe Amato Carlo Meghini
IEI-CNR, Via S. Maria 46 - 56126 Pisa - Italy
Email: {amato.meghini}@iei.pi.cnr.it

Abstract:

Video data are rich of implicit information that cannot be efficiently managed by a computer when they are stored as unstructured linear streams. To allow a computer to efficiently deal with this kind of information is necessary to add some further information that offers a high level representation of what is contained in video data.

This paper addresses the issues of defining a video data model. First a brief survey of the current approaches is presented then there is a proposal of a video model. The proposed model distinguishes between video data and video presentations. It allows to retrieve and browse video data and video presentations. and to build presentations.

1. Introduction

Video data belong to the typology of data commonly named Continuous Media (Cvl) since their content is a sequence of media quanta that have a meaning only if they are presented in time. Multimedia storage systems have to store video data at least at the same rate at which they are sent by recording devices and have to retrieve them at least at the same rate required by the output devices to avoid hiccups. Video data playback needs high transfer rate and they require a large amount of disk space.

Because of these severe constraints, much effort has been made to discover efficient input/output hardware, clever physical data organisation and effective compression techniques to make real time video data handling feasible. This advances in technology have made possible the development of many multimedia applications and systems [GemmeI95], [Berson94] for handling video data with the result to create huge repositories containing large amount of video data hard to be managed since videos are in general stored as unstructured linear streams and they loose at all the rich amount of implicit information that they contain. It is hard to find some video knowing its content, or given a video, it is difficult to search for a certain sequence within it.

People often try to move methodologies, which they use every day, to computers and this often lead to not exploiting the features offered by them.

For instance, people, because of the VCR philosophy where videos are sequentially stored on a video tape, still manage video data only as continuous sequence of bytes. A person looking for a certain sequence in a film, unless he remembers the actual position of the sequence, starts from the beginning of the stream and fast forward large amount of data, spending several minutes, until he sees on the screen something similar to what he is looking for.

Similar consideration could be done regarding video authoring since in many commercial systems the montage could be done only by placing video segment in the correct order, with few editing opportunities.

Video data have a rich content and a precise structure even if they are stored as unstructured linear stream. This structure is present either in the subject displayed in a presentation or in the spatial dimension of a presentation or in the temporal dimension of a presentation. A spectator, watching a film, can easily infer that there is a person standing near the table where other three persons are talking. A spectator can see that there is a main window in which an anchorman is speaking and overlapped there is a little window displaying images coming from the satellite. A spectator can notice that there are temporal jumps moving from a scene to another.

All this information cannot be efficiently managed by a computer while the video is stored simply as a not structured linear stream since the video is simply treated as a raw sequence of bytes. To allow a computer to efficiently deal with this kind of information is necessary to add to video data some further information that offers a high level representation of what is contained in them.

This paper addresses the issues of defining a model that allows to manage video data more easily exploiting as much as possible their features. The proposed model aims at the exploitation of the rich semantic information contained in a video by giving video data a structure and adding further information to represent video data content.

Section 2 presents a survey of current approaches to manage video data and section 3 give an intuitive description of the proposed model.

2. Current approaches

Here is a brief survey of the current approaches used to handle video data. Current approaches aims at defining models for creating multimedia presentations, models for browsing video data and models retrieving video data by content. Some of them suffer from moving manual techniques, commonly used for instance in cinematography, into computer area not exploiting further features that computers offer.

In the following we will describe some of the most common approaches and we will discuss their limits.

2.1. Stratification

In [Smith92] there is a proposal to annotate video streams using the idea of stratification. A stratum corresponds to a keyword associated with a sequence of frames. Sequence of frames of a stratum may overlap sequences belonging to different strata.

Using this technique it is possible to exploit the idea of context. For example, if the keyword child and the keyword garden have been defined, it is possible to search for a video sequence in which a child in a garden appears searching for the video sequence in which the stratum garden overlaps the stratum child.

This approach attempts to make retrieving of video segments by content possible, but it has not powerful construct to describe content of video sequence: it only uses keywords associated with strata corresponding to video segments.

Stratification does not define any hierarchical relationship among strata but the idea of context is obtained using the overlap and inclusion relationship that implicitly exist among the sequence of frames.

In this model the video stream is treated as an unstructured linear stream making any opportunity to manipulate video segments arduous.

2.2. Time Line

The most common approach to handle multimedia data adopted by existing systems [Semich92] is the time line approach. In such paradigm authors define their presentation placing video streams along axis representing time flow, explicitly defining time and space relationship among multimedia streams.

Time line based approach is very intuitive since it represents the natural way to see the final presentation, unfortunately it lacks of expressive and powerful construct to edit multimedia objects. It is hard to modify an existing presentation since this leads to redefine most of the previously defined relationship. When an author wants to use a different video sequence in place of the previously provided in a fixed range of time, if the new sequence consumes more time than the previous, all the remaining multimedia objects have to be moved from their position in the time axis. More in general, it is not always possible to foresee, during editing process, precise timing information. Sometime delays due to overload of some device, or delay due to interaction with the user can change any previously defined synchronisation.

The time line approach does not provide any feature to describe the content and structure of presentations being created or contents of the multimedia object used in a presentation. After a presentation has been created, it is a new unstructured linear stream.

2.3. Scripts

Another common approach adopted by authoring system is the use of scripts to define timing and space attribute of video in a presentation. This approach is conceptually similar to the time line approach. In time line based systems, authors use graphical tools to manipulate video stream. In script based system these actions are performed using a programming language. Eventually an author can still use a graphical interface then its actions are automatically translated in a script that can be successively edited.

The use of a programming language to manipulate multimedia objects allows to automate repetitive tasks that have to be performed explicitly by the authors elsewhere. For instance it is possible to do things like "repeat this sequence two times" or "go to the next sequence".

This approach is more flexible than the previous one but presents similar problems. An author may need to rewrite large amount of code if he wants to change a sequence synchronised with other sequences. Script approach does not offer any construct that allows to efficiently retrieve video data.

A system, called Video Schema, using scripts is described in [Matthews93]. Video Schema scripts can be written using Scheme programming language and a user can easily switch from timeline editing into script editing.

2.4. Structured models

A video presentation is normally composed of a set of video sequences each of which has a precise position on the screen and is synchronised with the others. When an author builds a presentation should be relieved of the burden of defining what happens in each instant of the presentation. An author, should think in high level terms in place to define what happens in the presentation in each instant in each position in the screen. The author should be interested in the overall information to use and in the overall layout of the presentation not in the low level information describing detailed information about synchronisation and timing.

During the authoring process the author uses and builds structure but all these structures are implicit. When an author builds a presentation, for instance for the television news, uses different video streams, each stream has its own identity and is maybe further structured, but the final presentation loses any explicit structure and is always an unstructured linear stream.

It would be better to make this information explicit. Losing any structure makes manipulating an existing presentation or reusing video streams hard. Using explicit structure when dealing with video data is an essential requirement.

Several models have been defined to give video streams a structure. In the following we will discuss some of them.

The Algebraic Video System [Weys95] uses an algebraic model to describe, manipulate and search video data. Video data is stored in a raw format as a continuous stream of frames and using operators of the video algebra a user can create new videos starting from the existing ones, can search for particular expressions identifying particular video sequences, can play video, etc.

The fundamental entity of the model is the *presentation*. All expressions of the video algebra represent presentations. There are four kinds of video algebra expressions: creation, composition, output and description expressions.

Basic expressions could be created specifying a raw video stream, which video data is stored in, and a range within the video stream.

Complex expressions could be created by composing simpler ones. The same set of frames of a video stream can belong to different expressions. This feature allows editing of video by reusing existing component avoiding to copy video streams that consume large amount of space.

A video expression can contain specification about the spatial layout of the window in which the corresponding presentation has to be displayed.

Using video algebra expressions, description can be associated with presentation. Descriptions can be used for content based retrieval of expressions. A content is a Boolean combination of attributes and attributes are pairs (field-name, value). The component expressions of a video expression inherit descriptions from their parent expressions so the descriptions associated with an expression are also associated with its sub expressions. There is a particular operation, the *hide-content* operation, that allows to define video expressions that do not contain any description (i.e., these expressions do not inherit descriptions from their parent expressions). Content based access is done searching for expressions that satisfy a set of user defined properties.

The algebraic video model offers a powerful set of operators to define presentation but it offers few constructs for retrieving video segments. In video algebra it is possible to associate a textual description to an expression and this description is inherited by the component expressions. This hardly satisfies all significant aspect that a user would model, a more expressive model would be needed. Video algebra

allows to give a description of the overall expression but it is difficult using its description construct to describe in detail the subjects appearing in a video sequence.

In [Hamakawa93] a model in which a multimedia object is represented as a temporal glue, an extension of the TEX glue, is proposed. In this model, multimedia objects can be stretch and shrink in temporal dimension further than in two-dimensional space. An object is in general composed of many multimedia objects. The top level object contains accurate information about the overall layout (both time and space positions), the component objects have a relative layout and their accurate layout is computed using the indications coming from the top level object of the hierarchy. This model has a powerful set of possibilities for crating presentations but it doesn't provide any construct to describe video sequence to allow video data retrieval. Also in [Cecelia92] there is a proposal to treat temporal dimension using the same ideas found in TEX spatial layout algorithm.

There are some structured multimedia models that use powerful sets of operator to manage different kind of multimedia data but lacks to give powerful construct to deal with single media. For example the models CMIF [Hardman93], [Rossum93] and MHEG [Meyer95] allow a structured composition of multimedia presentation and specification of synchronisation constraints between media items but video data are still unstructured linear streams of data.

3. A proposal for a video data model

Different users of a video database have different needs depending on their background. People would ask the video database for different questions and they would use differently video data. A customer, which uses a video on demand system, is interested in different aspects than a director making a movie or a student browsing course lectures. A video on demand user will formulate query about the title of a film, the actors, the genres. He is presumably only interested in playing the video and browsing it using the fast forward and fast backward operations. A director should formulate query about particular sequences of videos, specifying with precision the content of a sequence (for instance he wants to find the sequence in witch Bill Clinton meets Arafat). He would have the possibility to manipulate video, to change the ordering in which sequences are shown. A student would search for videos in which a particular argument has been discussed or for videos in which there are lessons taught by a certain professor. Our intent is to define a general model for dealing with video data taking in account all the features that could be useful for people managing video data.

The proposed model discriminates between video data and video presentations. Video data exist independently of the fact that they could be displayed since the same video or the same portion of video could be played in different presentation with different styles. For instance a service shot by a video reporter in Somalia could be used in a presentation about the war in Somalia as main service, could be used as background scene while an anchorman is specking in an other program and could be displayed in a small window overlapped to the main window. Video data stored in the database have no information describing their usage as part of some presentations but their usage is decided by the director when he defines a video presentation. Presentations consist of structures that describe the way some video data should be shown (timing, position, synchronisation, etc.). Video data used in a presentation should be not copied but they should be referred using pointers. The same video

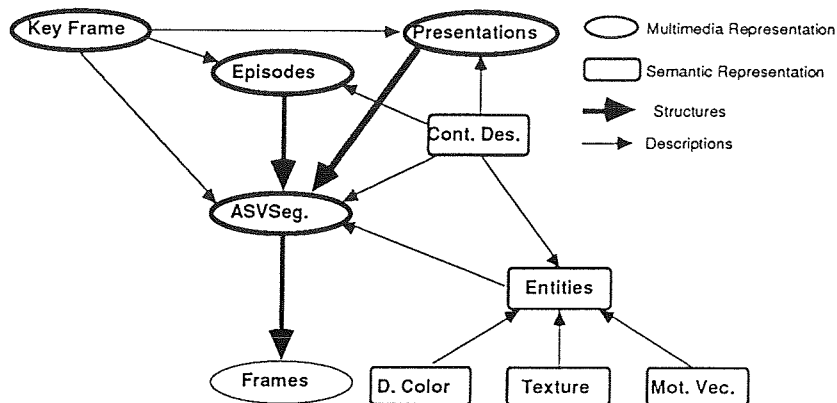


Fig.1 Video Data Model

sequence could be used in different presentations and there is only a copy of that video sequence.

In the following section a description of the structure of the proposed model is depicted. The description, for sake of simplicity, does not use formalisms and is accomplished using intuitive ideas. A complete formal description would require more space and would be beyond the aim of this paper.

3.1. Overview of the Model

In traditional databases, data are a structured representation of reality stored in a format suited to be efficiently managed and retrieved by a computer. In a multimedia database, data are a multimedia representation of the reality, as images, sounds, videos. These data are rich of information that is hardly manageable by a computer with the current technology, since they are stored in raw format, but is easily recognised by a human being. In a video in which two persons are walking hand by hand, a person can recognise that scene, but from the computer point of view, the video is a continuous stream of meaningless bytes. To make easy the computer to deal with video data is necessary to add in the database further data representing the same reality represented by the multimedia data but that are more easily manageable. We call the former *multimedia representation* and the latter *semantic representation* of the reality. Multimedia representation is used to model the management of video data as raw video streams. Semantic representation is the representation used to model the retrieving of video data by content. Further than these two representations, the mapping between them should be represented. The semantic information representing "a person whose name is John Smith running from the left to the right of the screen in Central Park in autumn" should be linked with the portion of the video sequence in which this situation emerges.

A picture summarising the overall structure of the proposed model is depicted in Fig. 1. Ellipses illustrate multimedia representation, rectangles illustrate semantic representation. Big arrows represent structure hierarchies among entities of the model, small arrows represent descriptive relationship among entities of the model.

3.1.1. Multimedia Representation

A video stream can be thought as a discrete sequence of frames and a frame can be thought as sets of coloured point of arbitrary form as in the model presented in [Meghini95]. During playback of a video stream, its frames have to be displayed at a given rate depending on the quality of the video.

People querying the video database are likely interested in a video sequence rather than a particular frame in a video stream since there are probably few differences among contents of consecutive frames. Eventually a user can search for a video sequence then he can choose a particular frame in the sequence, but the entity that the content is inferred from is a video segment. In the proposed model we use the notion of *Arbitrarily Shaped Video Segment (ASVS)* [Chang93], [Yun93]. An ASVS is an arbitrarily shaped unbroken sequence of frames with no abrupt scene change between contiguous frames. A continuous camera shot is an ASVS. A scene, shot using different cameras, results in a set of ASVSs belonging to the same episode. In the model we consider not only rectangular video segments but segments can in general have an arbitrary shape. For instance an ASVS can be a portion of a cartoon: a cartoon can be thought as a composition of many ASVSs. An ASVS can be the portion of video representing the anchorman of the weather forecast. In this video there is stored only the moving shape of the anchorman without background, which is a separate video and will be eventually add in the weather forecast presentation.

An ordered set of ASVSs related to the same scene is an *Episode*. An episode is the set of segments corresponding to significant events in a video. For instance a "walk in the park", shot from different point of view by different cameras generating many ASVSs, is an episode. Episodes can be organised hierarchically so the episode "walk in the park" may contain, for instance the episode "meting with friend" or the episode "play with a dog".

Segments of an episode are in general not played consecutively in a presentation but can be interlaced with segment belonging to different episodes. Segments of the episode "walk in the park" can be interlaced with segment belonging to different episodes somehow related. Modelling episodes explicitly, makes feasible to access them in the database also when their segments are used in presentations. Elsewhere any explicit structure among ASVSs and among episodes is lost after a director organises the montage of a film, even though that the structure is unconsciously perceived by spectators watching the film. If a director is interested to all segments of a certain episode he should have the possibility to access them even if in some presentation (maybe in the final version of the film) they are interlaced with many other segments belonging to different episodes.

An episode can be thought as a collection of segments related to the same event and it does not contain any indication about the way their segments have to be displayed. Segments can be used in different presentation using different layout so a different structure is needed to hold this information. A *Presentation* defines the way a video, composed of different ASVSs, has to be played back. It is a structure built over ASVSs used to organise them choosing their ordering, their position in the display, their style, their synchronisation. Storing explicitly distinct structures for presentations allows to avoid copying of video data and to reuse them. Video data can be simply referred by presentation in place of making a copy of them when they are used. This is an important feature in a multimedia system since video data consume large amount of disk space and copying could be troublesome. For instance, the television news of Wednesday 29 November is stored as a presentation in which

many ASVS are used, but each ASVS is independently stored in the database as belonging to its episode and can also be used in different presentation. Video data can be view from two different points of view: as data with their own content and as data that collaborate to build a new information. The former is reached using the notion of episode, the latter using the idea of presentation in which many ASVSs interact to form the overall information.

Episodes, segments and presentations are data rich of information but, as previously stated, this information cannot be immediately exploited with the current technology. It is hard to find a particular episode, or a particular sequence of an episode trying to exploit video content from its raw representation. The only way to find a particular sequence in a video stream is to playback it or eventually to fast forward it. Notice that the fast forward operation has technical limitations as the stream has to be played back at a speed higher than its usual consumption rate with the risk to overload the system. Some compression techniques as MPEG-2 [Li94], [Urabe93], [Patel93], allow the fast forward and fast backward operation reducing this problem that exists anyway.

The notion of *Key Frame* has been introduced in the model to make easier browsing of the video database. A key frame is a single frame summarising either the content of a segment, of an episode, of structured part of presentations. It is in few words the representative frame of a video sequence ([Arman93], [Arman94]). Key frames can be used to browse videos more easily than using the fast forward or fast backward operations. Searching for a certain sequence could be performed browsing representative frames, looking for some picture in which there is something similar to the sequence we are looking for and this can be performed with typical techniques used for retrieving images [Meghini95].

Representative frames have been used in many systems ([Gabbe94], [Otsuji93]) and powerful algorithms have been developed to automatically infer them given a video sequence. In [Teodosio93] there is a proposal about how to extract salient video still, images created to resume the most significant aspect of a video sequence.

3.1.2. Semantic representation

As previously stated, the multimedia representation furnishes a model to organise raw video stream, but it lacks in giving facility to manage contents for retrieving video data. Content information exists in raw data but it is expressed in a form not suitable to be efficiently managed. To make easier video retrieving by content, the proposed model provides an alternative representation of information contained in video data.

An episode or an ASVS contains many subjects and temporal events. Each significant subject and event should be extracted and represented separately. For instance let's assume to have an episode that has the following content described in natural language: *two persons, a man and a woman, are walking hand by hand in central park; along the path they are following there is a park bench, which an old man is sitting on; after they pass in front of the old man, they meet, on the garden on the left of the path, a black dog running toward the left; the two persons' names are John and Mary and this is happening in autumn.* Three different persons appear in this episode, that have their own peculiar characteristics. There is a black dog, a garden, a path. These subjects interact spatially and temporally: the man and the woman pass in front of the old man, the garden is on the left of the path, after they meet the old man they meet the dog. The subjects have static and dynamic features:

the dog is black, the dog is running toward the left of the screen, the garden has as dominant colour the yellow as the scene has been shot in autumn.

To identify different relevant subject of a segment, as for instance a person walking, we introduce *Entities*. An entity is the portion (both spatial and temporal) of the segment in which the subject to be represented appears. For instance the spatial and temporal portion of the segment in which the dog or the old man appears is an entity. An ASVS can be thought as partitioned in a set of entities. Notice that it is possible that different entities belonging to different ASVSs, eventually belonging to different episodes, correspond to the same subject. For instance the old man sitting in the park can have been shot from different point of view by different cameras resulting in different ASVSs so in different entities. Entities are only a hook to the portion of ASVSs in which a subject appears but they do not contain any content. When different entities represent the same subject these different entities can be associated with the same content description as will be discussed later.

Entities are associated with visual information as *Dominant Colour*, and *Texture*. For instance the entity corresponding to the black dog is associated with the colour black. The garden along the path is associated with the colour yellow and with a texture recalling yellow leaves on the ground.

Entities have a dynamic behaviour and the database could be interrogated exploiting dynamic features. For instance a query like "give me the ASVSs in which there is a black subject moving toward the left of the screen" could be submitted to the video database. To accomplish this in the proposed model each entity is associated with a *Motion Vector Table* describing its movement on the screen. This information could be, for instance, automatically extract from the MPEG compressed stream in which the inter frame dependencies are resolved finding blocks of pixels that are present in many frames in different positions [Dimitrova94]. These blocks are not explicitly stored in all frames in which they appear but they are stored only once and in the other frames there is a reference to the frame where the block is actually stored and a vector saying which is the new position of the block.

Using this technique an activity can be represented by its typical vector table. For instance a person running toward the left shot in profile has a typical vector table describing the movement of its arms and its legs. The database can be interrogated using this information by use of some similarity based algorithm.

The notions previously described allow to extract from a video stream much useful information for retrieving video data, but the model still lacks in describing the semantic content of a video stream. Up to now we can say that there is a black spot moving toward the left of the screen, but the system doesn't know that it is a dog that runs. The same is for all the other entities appearing in the video and for the event that they produce. The system doesn't know that John and Mary meet an old man. To obtain that, each entity, ASVS episode, presentation is associated with its content description. A content description is defined using both the semantic data model and the event calculus. The former offers a database like description the latter is useful to model events and interaction among described components.

Content description is not limited to entities but also ASVS, episodes and presentations should be described as they can hold information strictly related to them and information resulting by the composition of their components. For instance the presentation corresponding to the television news of Wednesday 29 November has its specific identity and needs to be described. This presentation can be associated with information describing the director the technical staff, the editorial staff; its description should also characterise that it is a television news.

4. Conclusions

The proposed model takes into considerations the aspects of retrieving, browsing and authoring videos. Using this model videos can be retrieved using semantic knowledge about their content, using attributes as colours and textures, using information about the motion of objects appearing in video sequences. The video database can be browsed using key frames or navigating the structure of the presentations. Presentations can be built using segments and do not alter the identity of the video segments themselves. The same segment can be used in different presentations and it is not copied but is referred by presentations using pointers.

In designing the model we tried to use notions that are available with the current technology to be able to effectively implement it.

5. Bibliography:

- [Arman93] Image Processing on Compressed Data for Large Video Databases, Farshid Arman, Arding Hsu and Ming-Yee Chiu, Proc of ACM Transaction on Multimedia 1993.
- [Arman94] Content-based Browsing of Video Sequences, F. Arman, R. Deponmmier, A.Hsu and M-Y. Chiu, Proc of ACM Transaction on Multimedia 1994.
- [Berson94] Design of a Scalable Multimedia Storage Manager, Steven Berson, A. Dashti, S. Golubchik, Richard R. Muntz and C. Shahabi, UCLA - Computer Science Department Technical Report CSD-940042, December 1994.
- [Cecelia92] Scheduling Multimedia Document Using Temporal Constraints, Proc. of Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, California, USA, November 1992, Springer Verlag, LNCS N. 712
- [Chang93] Transform Coding of Arbitrarily- Shaped Image Segments, Shih-Fu Chang and David G. Messerschmitt, Proc of ACM Transaction on Multimedia 1993.
- [Dimitrova 94] RX for Semantic Video Database Retrieval, Nevenka Dimitrova and Foouzan Golshani, Proc of ACM Transaction on Multimedia 1994.
- [Gabbe94] Towards Intelligent Recognition of Multimedia Episodes in Real-Time Applications, John D. Gabbe, Allen Ginsberg and Bethany S. Robinson, Proc of ACM Transaction on Multimedia 1994.
- [Gemmel95] Multimedia Storage Servers: A Tutorial, D. James Gemmel, Harrick M. Vin, Dilip D. Kandlur, P. Venkat Rangan and Lawrence A. Rowe, IEEE Computer, May 1995.
- [Hamakawa93] ObjectComposition and Playback Models for Handling Multimedia Data, Rei Hamakawa and Jun Rekimoto, Proc of ACM Transaction on Multimedia 1993.
- [Hardman93] Structured Multimedia Authoring, Lynda Hardman, Guido van Rossum and Dick C. A. Bulterman, Proc of ACM Transaction on Multimedia 1993.
- [Li94] MPEG-2 Coded- and Uncoded- Stream Synchronisation Control for Real-time Multimedia Transmission and Presentation over B-ISDN, Lian Li and Nicolas D. Georganas, Proc of ACM Transaction on Multimedia 1994.
- [Matthews93] VideoSchema: A programmable Video Editing System for Automation and Media Recognition, James Matthews, Peter Gloor and Fillia Makedon, Proc of ACM Transaction on Multimedia 1993.
- [Meghini 95] An Image Retrieval Model Based on Classical Logic, Proc of ACM SIGIR'95, Seattle WA USA 1995.
- [Meyer95] MHEG Explained, Thomas Meyer - Boudnik and Wolfgang Effelsberg, IEEE Multimedia, Spring 1995.
- [Otsuji93] Projection Detecting for Video Cut Detection, Kiyotaka Otsuji and Yoshinobu Tonomura, Proc of ACM Transaction on Multimedia 1993.

- [Patel93] Performance of a Software MPEG Video Decoder, Ketan Patel, Brian Smith and Lawrence A. Rowe, Proc of ACM Transaction on Multimedia 1993.
- [Rossum93] CMIFed: A presentation Environment for Portable Hypermedia Documents, Guido van Rossum, Jack Jensen, K. Sjoerd Mullender, Dick C.A. Bulterman, Proc of ACM Transaction on Multimedia 1993
- [Semich92] Multimedia Tools For Development Pros, J. William Semich, in Datamation, August 15, 1992
- [Smith92] The Stratification System. A design Environment for Random Access Video, Proc. of Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, California, USA, November 1992, Springer Verlag, LNCS N. 712
- [Teodosio93] Salient Video Stills: Content and Context Preserved, Laura Teodosio and Walter Bender, Proc of ACM Transaction on Multimedia 1993.
- [Urabe93] MPEGTool: An X Window Based MPEG Encoder and Statistics Tool, Toshiuki Urabe, Hassan Afzal and Grace Ho, Proc of ACM Transaction on Multimedia 1993.
- [Weiss95] Composition and Search with a Video Algebra, Ron Weiss, Andrzej Duda and David K. Gifford, IEEE Multimedia, 2(1): 12-25 Spring 1995.
- [Yun93] Architecture for Multi-Source Multi-User Video Compositing, Louis C. Yun and David G. Messerschmitt, Proc of ACM Transaction on Multimedia 1993.