The Canvass News Video Composition System[1]

G. Ahanger and T.D.C. Little

Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877
tdcl@bu.edu

MCL Technical Report No. 08-25-1999

Abstract– Canvass, or *Customized Access to News Video Archive Storage System*, is a proof-of concept prototype video composition system that allows users to retrieve automatically-composed video stories based on input selection criteria. Composition is achieved at the shot level. In essence, users are provided with a set of news stories that are composed, on-the-fly, from available news content, considering their individual needs. We envision this concept to be applicable to future personalized video delivery systems as video storage and delivery systems become more ubiquitous. This report describes the priciples of operation and architecture for Canvass.

[1] This work is supported by the NSF under grant No. IRI-9502702. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 What is Canvass?

Canvass, or *Customized Access to News Video Archive Storage System*, is a proof-of concept prototype video composition system that allows users to retrieve automatically-composed video stories based on input selection criteria. Composition is achieved at the shot level. In essence, users are provided with a set of news stories that are composed, on-the-fly, from available news content, considering their individual needs. We envision this concept to be applicable to future personalized video delivery systems as video storage and delivery systems become more ubiquitous. The Canvass demo was origially a web-based applicaiton; however, it is no longer maintained.

The system was built to demonstrate techniques developed during the course of National Science Foundation Grant number entitled "Fast Access to Video Information." Further details of these techniques can be found in the references.

In the remainder of this report we overview the principles of operation and system architecture of Canvass. We also describe how the video archive was created, present some example queries to the system, and discuss the metrics used to evaluate performance. Related publications are listed at the end.

2 Principles of Operation

Canvass uses two types of composition to create a video story:

1. Instance-based composition: A news item is generated from an instance (e.g., today's 7:00 PM news).
2. Period-based composition: A news item is generated over a period (e.g., news about Albright's visit to the Middle East).

These compositions are achieved using the following techniques:

1. Temporal ordering: Segments are organized on the timeline as a chronology according to their creation time and date. This scheme is applicable to period-based compositions.
2. Thematic composition: Segments are organized with related information to maintain temporal continuity. Concept similarity is used for the sequencing and the concept

vectors associated with the segments are used to find the similarity between the corresponding two segments. This technique yields better thematic flow in a video piece than temporal ordering by itself. A resulting video piece consists of segments from one or several threads, across which the information does not vary significantly. Thus, the resulting video piece has long forward jumps in time as some intermediate segments are dropped. This is applicable to both the instance-based and period-based compositions.

3. Thematic nearness composition: Segments are organized by finding the similarity between segments through normalizing the concept similarity with the creation time difference of the two segments. This technique is implemented so that segments from most of the threads present in a storyline are incorporated in a composition. A video piece resulting from this technique has smaller temporal jumps as fewer segments are considered dissimilar; however, it has lower thematic continuity than a piece resulting from the thematic composition technique.

A user can also specify playout time constraints in the queries. These constraints reduce to:

1. Ample time: When a user specifies more time than the total playout duration of the data matching the query, Canvass tries to fill the remaining time with filler shots at the end.
2. Limited time: When a user specifies less time than the total playout duration of the data matching the query, Canvass drops shots to fit the final presentation into the constraint.

2.1 Types of Queries Supported by Canvass:

Canvass supports several types of queries as described below:

1. Query the annotated metadata: News video items can be retrieved by applying user-generated search criteria to the annotated metadata of the video database. These annotations apply to the events associated with news items, people, and places seen in the visuals and discussed in the sound track. For example, the places are categorized as "country," "city," and"place." For these types of queries we do not merge news items

belonging to the same event but present them as individual news items. Basically, we treat these queries as instance-based queries (as a demonstration of instance-based results, otherwise, the news items belonging to same event can be merged by using the closed-caption data).

2. Query the closed-caption metadata: News items can be retrieved by applying user-generated search criteria to the indices generated from closed-caption data. A closed-caption-based query is considered a period based query. The segments retrieved are clustered into individual news stories; therefore, various news items belonging to the same story are merged and composed as a single story over a period of chronological time. In this type of query we do not allow clustering of the same type of segments, preventing transposition of facts (i.e., future developments in an event are presented before the preceding developments). A user can impose playout-time constraints under this type of query.

3. Query both the closed caption and annotated metadata: A user can simultaneously select annotated metadata and type keywords for search. We allow this type of query to increase the precision of the results. The set of segments returned from annotated metadata match is intersected with the set of segments returned from the match with the closed-caption data. Only the common segments are retained. We treat this type of query also as period-based. A user can impose playout-time constraints under this type of query.

3 System Architecture

The high level data flow diagram of the Canvass is shown in Fig. 3. The Canvass application is implemented in CGI-bin scripts that are accessed from the Web interface.

The system architecture of Canvass, a news digital video production system, is divided into annotation, indexing, and database population module, retrieval module including user interface and composition techniques, and video delivery module. The annotation and indexing module is used to annotate and index video data and transcripts, respectively, and populate a database with the annotated metadata. In the retrieval module we process queries using both annotated and transcript metadata and compose the resulting data. The video delivery module is used to schedule the playout of actual video segments within a composition.

3.1 Annotation and Indexing Module

Initially video data are in analog, NTSC/VHS format and these are next converted into MPEG-1 digital format and stored in digital archive (Fig. 1). To be able to search and retrieve digital video information from the archive we require information within video data. The information is extracted based on the concepts discussed in [6]. The annotated information/metadata is stored in a relational database, called miniSQL [13]. The database schema for Canvass is discussed in [6].

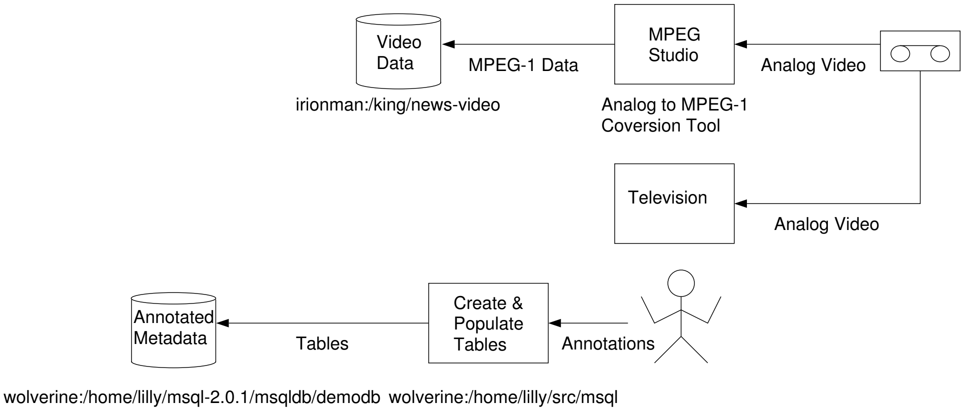
Video Data
MPEG-1 Data
MPEG Studio
Analog Video
irionman:/king/news-video
Analog to MPEG-1 Coversion Tool
Television
Analog Video
Annotated Metadata
Tables
Create & Populate Tables
Annotations
wolverine:/home/lilly/msql-2.0.1/msqldb/demodb
wolverine:/home/lilly/src/msql

Figure 1: Annotation, Conversion, and DB Population Processes

The following tables are used to store information and relationships among objects.

Database name:	demodb	
Tables contained:		
	object	All objects and information about their attributes is stored in this table.
	object_comp	Relationships among structural objects is stored in this table.
	physical_map	Information about the related video files, transcript files, and playout duration of segment is stored in this table.

The following steps are used to populate a database. Initially, a database called "demodb" is created using miniSQL commands [13]. Next, we create a script file called "create_news.msql" that is used to create tables in the database. This file contains all the information about the names of the tables and the fields in each table. The annotated metadata is also stored in script files (e.g., populate_news42.msql) according to miniSQL specifications and are used to populate the database tables. Following commands are used to create tables and populate them: To create tables: msql demodb < create_news.msql

To populate tables: msql deomdb < populate_news42.msql

In canvass we use the C language API's to retrieve video data.

Transcripts associated with structural objects (e.g., introduction, enclose, question/answer, and comment) are indexed using a tool called SMART [9, 14]. Initially, the closed-caption data are extracted from analog video format and stored as text file. The programs used to extract closed-caption data are as follows:

CNN regular news: CNNCollect
CBS, NBC, and ABC news: CBSCollect

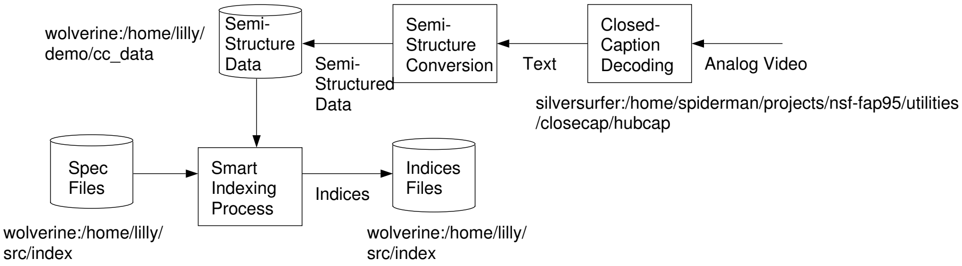

Figure 2: Closed-captioned Decoding and Indexing Processes

Next, the text associated with each structural object divided into separate sections (semi-structured) and each section is assigned an ID. The ID is the same as the ID of the corresponding video segments (object IDs in relational database) [3] as to establish correlation for manipulation during query processing. SMART is used for indexing and retrieval of semi-structured data. For indexing, SMART requires following files:

spec.data and spec.default: use to specify type of indexing required [3, 9]
doc_loc: contains names of the files to be indexed.

To index the semi-structured data the following command is executed:

"smart index.doc spec.data"

To bring up the query interface of SMART, use following the command:

"smart inter spec.data"

Next, the software architecture of the retrieval and video playout modules is discussed.

4 Software Architecture of Canvass Runtime

The high level data flow diagram of the Canvass is shown in Fig. 3. The Canvass application is implemented in CGI-bin scripts that are accessed from the Web interface.

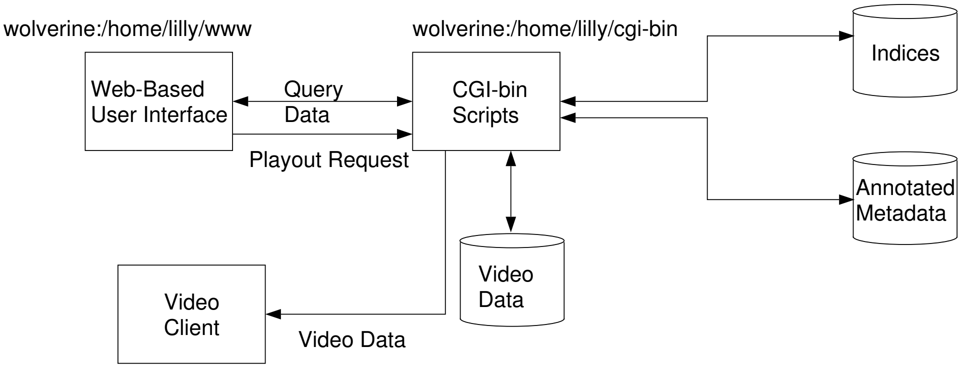

Figure 3: Hight Level Data flow Diagram of Canvass

The algorithms for user query processing [3] and video composition [4] are implemented in CGI-bin scripts in the C language and software architecture is shown in Fig. 4

The interface in implemented in the HTML language. Once the interface is accessed it calls a script to populate the interface with metadata from the database and provide mechanism to issues queries to the system. A user can issue following types of queries [3]:

1. Metadata-based query: A query is processed using the annotated metadata.
2. Semi-structured query: A query is processed using the semi-structured metadata/indices.
3. Composite query: A query is processed using both semi-structures and annotated metadata.

The retrieved segments are then composed and customized into a piece of video [4]. The conceptual compositions or information about a composition is passed to a user and is displayed in the user interface. The video playout is initiated at the user's request by passing the composition information to a video server (CGI-bin). The server in turn reads the corresponding video data and sends them to a video client for rendering. The communication between the video server and the client takes place via 100BaseT using TCP/IP protocols. We use MTV client [12] for video playout.

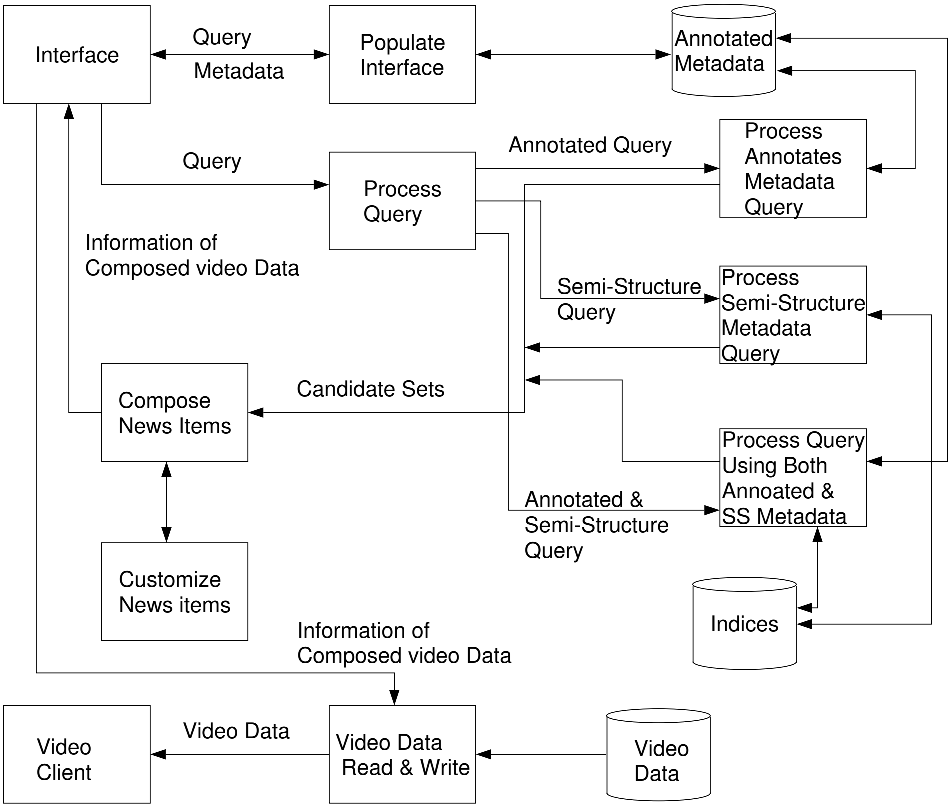

Figure 4: Detailed Software Architecture of Canvass

The following files comprise the canvass application source code: `cluster.c`, `composenews.c`, `initialize.c`, `socket.c`, `transitive.c`, `comparison.c`, `customize.c`, `listappend.c`, `stream.c`, `traversedb.c`, `composeduration.c`, `demo.c`, `temporalorder.c`, `util.c`.

4.1 System Implementation

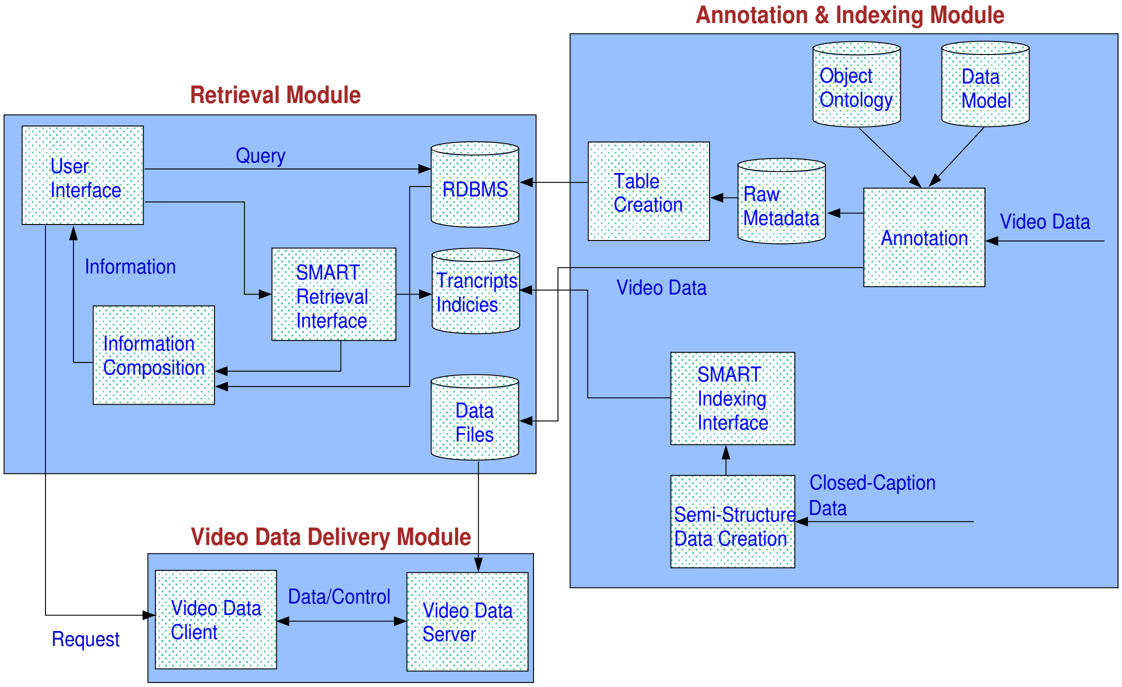

Figure 5: System Architecture of Canvass

Fig. 5 depicts the Canvass system architecture. `miniSQL` is the relational database manager of for the video metadata. The annotation is achieved via an annotation tool called Vane. SMART is used as the basis for text indexing of the closed-caption data from the video, and is also used to match user text-selection criteria with the stored indices. The user interface is Web-based and the Common Gateway Interface (CGI) scripts are written in the C language. In the prototype, video data are streamed using either an MPEG client or using a Real player.

4.1.1 The Video Data Set

A video data set was acquired to populate the video database with content suitable for automatic composition. Broadcast news video data were acquired from CNN, NBC, and

ABC over a period of 40 days from 20 January 1998 to 28 February 1998. During this period we recorded the 9:00 AM and 8:00 PM CNN (national) broadcasts (CNN1 and CNN2), the 6:30 PM NBC (national) broadcast, and the 12:00 PM ABC (local) broadcast.

Data were initially recorded in analog, VHS/NTSC, format and considerable effort was required to translate the data into a state suitable for resolving queries to yield candidate sets and composable segments. The analog video streams were first digitized into MPEG-1 format (and subsequently the Real Networks format) and then content and structural information/metadata were extracted. Segments were annotated based on the types of components within each news item. Content information such as conceptual and tangible entities (e.g., people, locations, cause and effects, and events) were annotated to support the generation of concept vectors.

The closed-caption data associated with each identified segment were decoded and individual documents and indexed using the SMART indexing tool developed at Cornell University.

In the following, we describe how to use the Canvass prototype.

4.1.2 How to Use Canvass for News Presentation

The Canvass prototype has a variety of options on its interface. As a guide through these features, the following steps are provided to explore each type of aforementioned queries.

1. Instance and annotated metadata-based query:
   (a) Select "Specific" and then select a field from "Select Time"
   (b) Select the date from the "End Date" option
   (c) Select "Query Database" and select one or more of the annotated fields
   (d) Execute "Submit Query" to generate the composition
2. Instance and annotated metadata-based query with time-constraint and segment-type presentation preference:
   (a) Select "Specific" and then select a field from "Select Time"
   (b) Select the date from the "End Date" option
   (c) Select "Query Database" and select one or more of the annotated fields

   (d) Select any duration from the ”Select Duration” option

   (e) Select preference from the ”Select Preference Type” option

   (f) Execute ”Submit Query” to generate the composition

3. Instance and annotated metadata-based query with the use of thematic composition in a news item:

   (a) Select ”Specific” and then select a field from ”Select Time”

   (b) Select the date from the ”End Date” option

   (c) Select ”Thematic” from the ”Select Composition” option

   (d) Select ”Query Database” and select one or more of the annotated fields

   (e) Execute ”Submit Query” to generate the composition

4. Period and closed-caption metadata-based query:

   (a) Select ”Period” and then select the start and end date from the ”Start Date” and ”End Date” options

   (b) Select ”Query Closed Caption” and enter keywords into the field

   (c) Select ”Associative Search” and ”Transitive Search” to improve recall of the system

   (d) Select ”Thematic” or ”Thematic Nearness” from the ”Select Composition” option to achieve something other than ”Temporal” composition

   (e) Select any duration from the ”Select Duration” option to introduce a playout duration constraint

   (f) Execute ”Submit Query” to generate the composition

4.1.3 Example Queries

Here we walk through some example queries to help illustrate the behavior of the system.

Example 1

1. Select ”Specific”

2. Select ”All” from the ”Select Time” option
3. Select ”1998/01/31” from the ”End Date” option
4. Click on the ”Submit Query”

Results: All the news item available for the date choose are displayed. Click on ”Play Video” to view the results.

Next, **in addition** to first three steps:

1. Select ”Scene Footage” from ”Select Source Type Preference” option
2. Click on ”Submit Query”

Results: The segments in the body of the news item displayed are previously clustered and the ”Scene Footage” segments are displayed first.

Next, **reset all** the options and

1. Select the ”Period” option
2. Select ”1997/11/16” from the ”Start Date” option
3. Select ”1998/02/18” from the ”End Date” option
4. Select the ”Query Closed Caption” option
5. Type in the keyword ”cars”
6. Select the ”Prim-Clustering” option
7. Click ”Submit Query”

Results: Segments comprised of two news item are retrieved. The segments are clustered into the individual storylines and displayed.

Repeat Example 1 with keywords ”oprah alabama clinic” for some variety.

Example 2 Repeat Example 1 with the addition of the following options:

1. Select the ”Transitive Search” option
2. Click ”Submit Query”

Results: Additional segments are retrieved that satisfy a transitive relationship in the metadata.

Example 3 Again, building on Example 1, repeat with the addition of the following:

1. Select the ”Associative Search” option
2. Select ”3 minutes” from the ”Select Playout Duration” option
3. Click ”Submit Query”

Results: You will notice a further increase in number of segments retrieved. But for the playout only segments that are accommodated in 3 minutes are sequenced. Click on ”Display Newscast Statistics” to see the statistics before and after this time-constrained playout.

For playout-duration-constrained queries you can select the ”Filler Algorithm” option. This attempts to use filler shots to consume the requested playout duration.

The default clustering algorithm used for the closed-caption query is fast but not most effective. By selecting ”Prim-Clustering”, more precision is obtained.

4.1.4 Statistics

Statistics characterizing the performance of the applied queries are presented on the results page by selecting the ”Display Newscast Statistics” option. These statistics characterize the following:

1. Information: The message conveyed by the content in a video segment (inference)
2. Thematic Continuity: The flow of information between consecutive segments in a composition

3. Temporal Continuity: The sequencing of segments according to the chronology in a composition

4. Period Span Coverage: The extent of the life of an event

5. Content Progression: The rate of change in information within a composition

6. Structural Continuity: The pattern of organization of segment types in a composition

5 Accessing the Demo System

The demonstration system requires a video playout client. We have tailored the application to function using both the MPEG-1 and Real formats.

5.1 Real Format

Install the RealPlayerG2 (or later perhaps a more recent version). Begin the demo using the Real format (http://wolverine.bu.edu/lilly/demo1.html).

5.2 MPEG-1

We designed the system using a program called mtv (it may work with other clients). Make sure that your browser MIME-types has an entry for your client. For mtv, it should be "video/MTV_File" and and application of "mtv -P5675". In this case the 5675 entry designates the TCP/IP port number for streaming. (The demonstration system is no longer maintained on line.)

6 Summary

Canvass represents a research prototype constructed to demonstrate techniques for automatic news video composition. This paper overviews some of its basic features and provides example queries to illustrate its operation. Please visit the Canvass demo at http://wolverine.bu.edu/canvas In the future we expect the Canvass functionality to be commonplace as video storage and delivery capabilities improve.

Acknowledgements

The following individuals participated in the development of Canvass. P. Basu, J. Begin, M. Carrer, J. Casebolt, L. Kuczynski, A. Narayanan, L. Ligresti, and K. Srinivasan,

References

[1] FAP Project URL http://hulk.bu.edu/projects/fap/summary.html.

[2] G. Ahanger and T.D.C. Little, "Data Semantics for Improving Retrieval Performance of Digital News Video Systems," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 13, No. 3, May/June 2001, pp. 352-360.

[3] G. Ahanger and T.D.C. Little, "Data Semantics for Improving Retrieval Performance of Digital News Video Systems," *Proc. 8th IFIP 2.6 Working Conference on Database Semantics*, January 1999.

[4] G. Ahanger and T.D.C. Little, "Automatic Composition Techniques for Video Production," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 10, No. 6, December 1998, pp. 967-987.

[5] G. Ahanger and T.D.C. Little, "A Language to Support Automatic Composition of Newscasts," *Computer and Information Technology*, Vol. 6, No. 3, September 1998, pp. 297-310.

[6] G. Ahanger and T.D.C Little, "A System for Customized News Delivery from Video Archives," *Proc. 4th Intl. Conf. on Multimedia Computing and Systems*, Ottawa, Canada, June 1997, pp. 526-533.

[7] G. Ahanger and T.D.C. Little, "A Survey of Technologies for Parsing and Indexing Digital Video," *Journal of Visual Communication and Image Representation*, special issue on Digital Libraries, Vol. 7, No. 1, March 1996, pp. 28-43.

[8] G. Ahanger, "Techniques for Automatic Digital Video Composition," Ph.D. Dissertation, Dept. of Electrical and Computer Engineering, Boston University, December 11, 1998.

[9] C. Buckley, "Implementation of the SMART Information Retrieval System," Computer Science Department, Cornell University, No. TR85-686, 1985.

[10] M. Carrer, L. Ligresti, G. Ahanger, T.D.C. Little, “An Annotation Engine for Supporting Video Database Population,” *Multimedia Tools and Applications*, Vol. 5, No. 3, November 1997, pp. 233-258. (Also reprinted in *Multimedia Technologies and Applications for the 21st Century*, Borko Furht (ed.), Kluwer Academic Publishers, 1998, pp. 161-184.)

[11] W. Klippgen, T.D.C. Little, G. Ahanger, and D. Venkatesh, “The Use of Metadata for the Rendering of Personalized Video Delivery,” in *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data,* Amit Sheth and Wolfgang Klas (eds.), McGraw Hill, March 1998, pp. 287-318.

[12] MPEG Playback Software, *www.mpegtv.com.*

[13] miniSQL: A Relational Database, *http://www.Hughes.com.au/.*

[14] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, New York, 1983.