AFRL-IF-RS-TR-2002-286
Final Technical Report
October 2002

INTELLIGENT ANALYST DIGITAL VIDEO LIBRARY PROTOTYPE

Sonic Foundry

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-286 has been reviewed and is approved for publication.

APPROVED: 

JONATHAN C. GREGORY
Project Engineer

FOR THE DIRECTOR: 

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 2002	Final Aug 98 – Jan 02

4. TITLE AND SUBTITLE
INTELLIGENT ANALYST DIGITAL VIDEO LIBRARY PROTOTYPE

5. FUNDING NUMBERS
C - F30602-98-2-0284
PE - 62702F
PR - 459E
TA - MC
WU - AR

6. AUTHOR(S)
Manju Ghai

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sonic Foundry
311 Craig Street
Pittsburgh Pennsylvania 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEC
32 Brooks Road
Rome New York 13441-4114

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-286

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Jonathan C. Gregory/IFEC/(315) 330-4294/ Jonathan.Gregory@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
This report provides details on the Intelligence Analyst Digital Video Library Prototype that was developed to incorporate various methods, techniques, and technology into a video indexing and retrieval system to enable Intelligence Analysts to more easily and effectively perform their jobs when exploiting video. It incorporated various automatic and user assisted indexing methods such as geoposition (GPS), moving object detection, and analyst annotations, date/time, etc.

14. SUBJECT TERMS
Video Indexing, Video Exploitation, Video Database

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

Overview

With the support of the Air Force, ISLIP Media, now Sonic Foundry Media Systems, to be referred to as Sonic from this point forward, was contracted to develop an *Intelligence Analyst Digital Video Library (IADVL) Prototype.* Sonic planned to integrate new technologies that emerged from Carnegie Mellon University's Informedia research lab and elsewhere, and, ultimately, commercialize them into a new set of products. Sonic chose to concentrate on technologies that it believed would most significantly improve the performance and productivity of defense intelligence analysts through enhancing their ability to create and work with digital content. The project was developed around progress in four areas:

- Global Positioning Systems as Retrieval Index
- Analyst Annotation of Library Content
- Moving Object Detection
- Mixed Media Search Techniques

Upon completion of this effort, Sonic delivered a prototype system designed to illustrate how the application of such technologies can aid the DoD's efforts to exploit the large body of extent and continuously produced audio, video and textual information from defense and civilian activities for intelligence purposes. Using core products, Sonic provided a technological means to transcribe textual or natural language queries to a database of video and audio information into a meaningful search of imagery data. The result of this development project is intended to substantially increase the responsiveness and the data mining abilities of defense intelligence analysts. Examples of defense relevance include:

- Ability to search and correlate vast amounts of broadcast news sources.
- Automated analysis of surveillance and reconnaissance video data with annotations.
- Capability for video-based training on demand.
- Enabling rapid generation of compelling briefing materials.

Infrastructure

In order to support the desired capabilities, the core infrastructure of the ISLIP technology had to be revamped. A new framework for searchable video that would support those new features, data structures, interfaces and applications – enabling the tasks as outlined for this Air Force sponsored project – was created.

Data Model Changes

At the core of this infrastructure lies a totally new data model revamped to support the features required for this project. For example, the new data model includes data definitions for geo-coordinates for frames of video and for segments of video thereby enabling location data to be associated with each frame of video for tracing motion on a map or for displaying regions associated with a particular story. Furthermore, the new data model was expanded to include data definitions for image indexes used for **Mixed Media Search Techniques** as well as *Moving Object Detection* and *Analyst Annotations of Library Content*.

Please see Appendix A for further detailed descriptions of the data model.

Interface Changes

In order to accommodate the new data model, a revised set of interfaces to allow for reading and writing data were created. The new interfaces include support for Global Positioning Systems as a Retrieval Index and Mixed Media Search Techniques. These interfaces are provided both in C++ and COM to enable access to the metadata using a variety of programming languages such as C++ and ASP.

Advanced Indexing Framework

In conjunction with our front end, real-time Indexing product, we have designed a fully automatic framework for offline indexing techniques such as those intended for this project. This framework, formerly known as *Builde*r but today simply referred to as *Advanced Indexing Modules*, provides a well-defined structure for pluggable indexing techniques. For example, the process of extracting audio from an audio/video digital file and running speech recognition to generate text is one Advanced Indexing technique. Another technique is to analyze video in order to detect and record where moving objects appeared or disappeared.

This Advanced Indexing infrastructure was designed to work in a fully automatic mode, taking the output from our front end, real-time indexing application, *Indexer*.

Global Positions Systems as a Retrieval Index

There has been a long-standing problem linking video content with the location where the footage was shot. This location data can be made available by use of equipment capable of embedding GPS (Global Positioning System) coordinates of the camera into the videotape while recording. Storing the GPS data during recording and using it during playback/indexing can enrich the indexed video data. This additional data can enhance the video searching and playback experience. Sonic created a system for the Air Force that could just that.

Sonic created two indexing methods that would facilitate the capture of this information.

Geospatial Referencing

Initial research and development was done which allowed us to build an Advanced Indexing Module, the *Geospatial Indexing Module,* which would create an index based on named locations within broadcasts rather than extracting GPS data from within the video. By indexing named locations first, we were able to bypass the issue of getting GPS data from video as standards for embedding GPS data into video were not highly agreed upon at this time[1].

This work was based on Informedia algorithms in which a video transcript is analyzed for keywords and natural language constructs so as to identify (with a large hit percentage/low false positives) geographic locations mentioned in the clip. These "places" can then be georeferenced. This georeferencing is a mapping between "place" and its physical location in a given coordinate system, usually Mercator-based latitude and

[1] As our research continued, we did, however, identify several cameras that embed GPS data into the Vertical Blanking Interval in video and would also work with the decoding box selected for purposes of the prototype.

longitude. Sonic expanded this algorithm to analyze not just the video transcript but also any manual annotations that were associated with the video, Ie. description information.

From a functional perspective, the *Geospatial Indexing Module* was designed to address the following operational requirements:

1. Provide an interface for the specification of a reference Gazetter upon which all georeferencing would be done.
    - By default the global gazetteer will be selected for all geocoding activities[2].
    - For specialized content, a local gazetteer could be added to the system; this is not part of the core product. A localized gazetteer would allow for geocoding of geographical references for regional content.
2. Provide a programmatic interface to encapsulate the detection and geocoding algorithms to correspond to the data model.

In order to address these functional requirements, the following design constraints were applied to the *GeoSpatial Indexing Module*.

- Implementation as a standard COM object with a well-defined interface.
- All data access from the plugin via a new server database interface.
- Georeferencing to include all locations in the video transcript as well as any user defined fields.
- Identification of segment within which a location is identified.
- Determination of frequency with which a particular item is identified in a segment.
- Timestamps will be associated with each occurrence of a location within the video (as a time offset from the beginning of the clip)
- Two gazetteers, one global and (possibly) one local will be available for use. In case of location conflict, the local gazetteer will have precedence over the global.

Once capture was complete, modifications were also made to our web based video search and retrieval application, *WebFinder*, to allow for the display of maps to enable the navigation of geospatial content.

From a functional perspective, the web interface was modified to address the following operational requirements:

- Provide a S*how Map* button as part of the search results to identify content for which a map was available[3]; the existence of a map implies that that geospatial data has been captured via the *GeoSpatial* indexing module.

    - The map will be annotated to show the places identified in a query result.
    - The map will have zoom in/out capability.
    - The map will have panning capability.
    - The user will be able to reset the map to its initial view (reset pan/zoom).
    - The user will have the capability of performing a spatial search for the zoomed in area.
    - The user will have the capability to view the results for the searched area or any area within the searched area in the *WebFinder*.

[2] This Global Gazetteer is available as part of the standard module.

[3] The map display is an image rendered on an ESRI, MapExtreme, Map Server; the Map Server runs on a separate dedicated server machine.

- The user will be able to update the map display on demand to reflect the latest spatial query parameters.
- The UI will provide an indication of the extent of the query rectangle.

- Provide a *Map Search* button to allow the user to search video content based on geographic region or proximity to a given location.

- The locations mentioned in each clip will be highlighted on the map dynamically as the segment is played using the start time and end time of the occurrence of each location in the clip.

In order to address these functional requirements, the following design constraints were applied to the *WebFinder application.*

- The web client will be implemented as a server-side OCX/DLL and thus the client design will minimize the server round trips to retrieve data thereby expediting the generation of search results and associated maps.

- The Map Server code will be written using the ESRI MapObjects OCX.

- The Map Server(s) will respond to all user commands via the input query string.

Please see Appendix B for further detailed descriptions of the Process Flow & Data Flow for Geospatial Referencing.

GPS Referencing

With the framework of GPS as a Retrieval Index in place, the GPS referencing capabilities were integrated into the infrastructure. Because our initiative encompassed the embedding of GPS information in the videotape, we chose to integrate the extraction capability into our core *Indexer* product instead of creating an independent *Advanced Indexing Module*.

During the indexing process, a list of all GPS values and time-codes is created. When the segments are created all the distinct time-code and GPS values that correspond to the segment's time-span will be associated with the segment. These values will be entered in the database when the segment is submitted to the database.

From a functional perspective, the *GPS Indexing* module was designed to address the following operational requirements:

- Provide the ability to parse GPS data from a prerecorded video stream[4].
- Provide the end user with the ability to turn the capture of GPS data on and off.
- Provide the end user with the ability to tune the sensitivity of the GPS filter.
- Provide the end user with the ability to visualize the GPS data as it changes during indexing.

In order to address these functional requirements, the following design constraints were applied to the *Indexer application.*

[4] For the purpose of the IADVL prototype, the GPS indexing capability was limited to be available only when indexing from tape.

- A new DirectShow filter was created to provide the functionality of capturing the GPS data and making it available to the *Indexer*. This filter monitors the COM port on which the GPS data is available and extracts the latitudes and longitudes of the camera from the data stream; this information is captured as part of the audio track of the videotape. All GPS data in the stream, when available, will be sampled.

- The GPS data is made available by the GPS hardware equipment, a RedHen box, on the COM port as the videotape is playing. This data is available as standard NMEA GPGGA and GPRMC strings.

- The new database table for holding GPS data is updated by the *Indexer* application on a per segment database. The required information is as follows:

    - *TapeId*: The Tape Id assigned to the videotape.
    - *SegmentId*: The Segment Id generated for the segment being submitted.
    - *Latitude*: The latitude value for the GPS coordinates in decimal seconds
    - *Longitude*: The longitude value for the GPS coordinates in decimal seconds.
    - *StartTime*: The time value when the GPS coordinates values changed to the current one. If this is the first GPS value for the segment then it is the same as start time of the segment.
    - *EndTime*: The time value when the next GPS coordinate change occurs. If there are no more GPS changes till the end of the segment then it is same as end time of the segment.

Once capture was complete, modifications were also made to our web based video search and retrieval application, *WebFinder*, to allow for the display of a geographic map with an outline representing the linear path taken by the GPS camera as it was moving.

From a functional perspective, the web interface was modified to address the following operational requirements:

- Provide the end user with the ability to Zoom to a particular region of a map and click a *FindIt!* button to initiate a search and retrieval of the start latitudes and longitudes within the particular range.
- Provide a *Show Track* button to allow the user to see the linear path taken by the GPS camera as it was moving.
- In conjunction with user invocation via the *Show Track* button, the video associated with the path of movement will be played.

In order to address these functional requirements, the following design constraints were applied to the *WebFinder* application.

- The web client will be implemented as a server-side OCX/DLL and thus the client design will minimize the server round trips to retrieve data thereby expediting the generation of search results and associated maps.

- The Map Server(s) will respond to all user commands via the input query string.

Please see Appendix C for further detailed descriptions of the High Level Workflow for GPS Referencing.

Analyst Annotation of Library Content

Once a media repository has been created, a new challenge presents itself. How does one keep the information relevant, up-to-date and as robust as possible? An end-user, i.e. an analyst, may want to add annotation to the content that will enrich the search for a subsequent user of the same corpus. This capability enables the true domain experts to easily add their input (via annotations) to the library at their convenience, rather than requiring the domain experts to communicate their requests to those who submit new content into the library. As more is learned of the content, the information can be added. As a result, the library becomes more knowledgeable. By approaching this from a hierarchical perspective, various levels of annotations, administrators could choose to make available (or not) annotations to groups or individuals.

The technical approaches breaks into two distinct tasks:

1. Designing a hierarchical scheme for combining and managing access to multiple annotation sets in ISLIP's existing library[5].
2. Designing and developing the necessary applications for entering analyst/end user annotations.

From a functional perspective, the following operational requirements for User Access had to be met:

- An interface must access exist to allow to log on to the Analyst Annotation system in order to identify themselves uniquely, and in order to acquire privileges to annotate certain content.
- An administrative interface must exist to enable a System Administration to assign individual or groups of users to have the ability to review or update content.
- An interface must exist so that end users can view any existing annotations that they have privileges to see.
  - Each annotation should be accompanied by the name of the user who entered it.
  - End users should be able to define the set of users whose annotations they wish to search.

In order to address these functional requirements, the following design constraints were applied to the *Analyst Annotation* application.

[5] Access control mechanism would entitle certain privileges to the users and allow the administrator to create a hierarchy of users. This gives a finer and effective control of the privileges for the user, which may include permission to create an Access profile, permissions to view/edit the annotation of other users, etc.

- The *MediaSite JavaFinder 3.2* product would serve as the foundation for this development. *Analyst Annotation* capabilities would be built into this system.
- In addition to the basic IIS functionality required by the *JavaFinder 3.2* application, *Analyst Annotation* would incorporate functionality requiring the use of Jrun Pro 2.3.
- In order to accommodate the additional data requirements for security and annotation, database schema would be expanded to include the following tables.

ISL_USER

COLUMN NAME	TYPE	SIZE	NULLABLE
USERNAME	VARCHAR2	20	NOT NULL
PASSWORD	VARCHAR2	20	
GROUPID	NUMBER		NOT NULL

ISL_GROUP

COLUMN NAME	TYPE	SIZE	NULLABLE
GROUPID	NUMBER		NOT NULL
DESCRIPTION	VARCHAR2	50	

ISL_PERMISSION

COLUMN NAME	TYPE	SIZE	NULLABLE
S_GROUPID	NUMBER		NOT NULL
T_GROUPID	NUMBER		NOT NULL
PERMISSION	VARCHAR2	50	NOT NULL

ISL_COLLECTION_PERM

COLUMN NAME	TYPE	SIZE	NULLABLE
GROUP_ID	NUMBER		NOT NULL
COLLECTION_ID	NUMBER		NOT NULL

Table 1: Expanded Database Schema

Please see Appendix D for further detailed descriptions of the Workflow for Analyst Annotation.
Please see Appendix E for a representative of the Analyst Annotation Security Hierarchy.
Please see Appendix F for screen representations of the Analyst Annotation System.

Moving Object Detection

Tracking moving objects in video is important for military and commercial applications. Research in motion understanding has given Sonic the ability to track and segment moving objects in video. Sonic proposes the use of motion understanding for tracking moving objects within a video. The major advantage to Sonic's approach is that moving object detection is independent of the video corpus, delineating the occurrence of motion, rather than specific object parameters such as velocity or direction. Any video may be processed for significant camera and object motion, regardless of image quality or camera parameters.

The goal of Sonic's activity in regard to this attribute is to ameliorate the tedious task of searching through hours and hours of video in order to locate objects in motion. We believe it is useful to query for all moving objects in a video or library of videos. In addition, it may be desired to automate the marking of these moving objects so that they may be searched individually, or as part of a category of objects.

In order to address this issue in the timeframe allocated, with the resources, available, Sonic chose to narrow the scope of a very complex problem to provide some basic motion detection features: detecting "beginning" and "ending" of motion for relatively simple scenarios, such as "pedestrians on the street", with relatively static background (no camera motion). These features are salient to the IADVL prototype project because they are are common for many security systems, tracking systems, etc.

From a functional perspective, the following operational requirements for Objects-in-Motion had to met:

- The Motion detection plug-in will allow the user to tune up parameters influencing the sensitivity of algorithm.
- The Motion detection plug-in will have the capability to run in an automatic mode; beginning and ending of motion will correspond to begin/end segment points.
- In the scenario where closed caption information is available for the footage, *Indexer* 4.0, with the Motion detection plug-in, will provide "automatic" description of the segment, based on closed caption content.

In order to address these functional requirements, the following design constraints were applied to the *Motion detection* plug-in to the *MediaSite Indexer* application.

- The Motion detection algorithm will be a plug-in to the existing *Indexer 4.0* application.

- A new DirectShow filter was created to provide the functionality of capturing motion data and making it available to the Indexer. To distinguish camera and object motion, we examine individual regions of the image for localized motion. When a sufficient region of connected motion is detected with an adjacent region of static video, the scene is characterized as having a moving object. Certain conditions must be measured in order to declare "motion" as significant; user configurable parameters are available to allow the user to tune parameters identifying what will and will not be recognized as motion. These parameters are listed below and Appendix G contains screen representations of how these parameters are presented to the user in the *Indexer* application.
    - Noise level: The changes below this level are regarded as "noise" and are ignored. This parameter allows filtering out chaotic changes caused, for example, by camera trembling; the range: 32-128
    - Sensitivity level: When the change above "noise level" is detected, the motion detection filter "notices" it but postpones the decision about "detection motion" until a predefined amount of movement detected. The segmentation based on motion begins only when the specified "sensitivity level" is achieved. The higher this value is, the less sensitive, the algorithm; the range: 1500-3500.
    - Frequency: There is no need to scan every frame in order to successfully detect motion. Skipping some frames is actually quite useful from the point of view of efficiency, as "motion detection" is quite calculation-intensive. This parameter defines, how many frames will be skipped before a frame will be analyzed; the range: 10-30.
    - End of motion: Once segmentation based on motion begins, the Motion filter polls continuously to identify when the "amount of motion" falls below the "noise" level. At this point, the Motion filter suggests that the end of a segment should be flagged. During a predefined interval, however, this variable should be ignored. This parameter determines how frequently *Indexer* should analyze two consequent movements, as one long movement or 2 short ones; the range: 250-5000 milliseconds.

Mixed-Media Search Techniques

Providing the ability to search on different media types and have the query cross media domain boundaries provides clear advantages over restricting search requests and results to a single media type. Sonic's digital libraries will provide a cross mapping between the narrative (speech and text) and imagery domains. Text based search and navigation have been discussed at length in conjunction with the *Publisher* system overview and both the Global Positioning Systems as Retrieval Index and Analyst Annotations components of the IADVL prototype.

The *Image Indexing* component of multi-modal search offers the end-user the ability to combine the powerful text based search capabilities native to the *MediaSite* system with the ability to match similar video images to one another. An example of the utility of this application is the ability to search for the word 'forest'; the text-based search would return all clips in the database containing the term 'forest'. When the basic content index has been augmented using the advanced indexing, *Image* module, end-users can further refine their search criteria to include just certain forest(s) including certain color patterns. For example, with the addition of Image Match capabilities, it becomes possible to identify images of forests in one particular season or another.

From a functional perspective, the Mixed-Media Search Techniques requirement, was addressed by focusing on the following operational requirements:

- Provide the ability to incorporate visual properties of data into the search index created via *MediaSite Publisher*.
- Incorporate Image Matching capabilities into the core text-search engine to support the use of both text and image search in concert.
- Provide an Image Match button as part of the *WebFinder* search results for a segment when Image Indexing has been processed accordingly. Please refer to Appendix H for a visual representation of the Image Match interface.

In order to address these functional requirements, the following design constraints and implementation decisions were applied to the *Image Indexing Module* and the corresponding *Image Match* capability.

- Image indexing is based upon the extraction of histogram data from an existing database of information. Therefore, the *Image Indexing Module* was created to augment the data index created via the *Indexer* application.
- The Image Indexing algorithm extracts histogram information and stores that information in a table in the database.
- The Image Match link is available when image data exists for a particular segment. When present, selecting the link will instantiate an image based search.
- During the Image Matching process, the derivative of the histogram differences for the top and bottom portions of each image are compared.

Appendix A: MediaSite Database Schema Overview

This document describes the Database Schema, called the Content Schema, accessed by MediaSite Products namely Indexer, Module Manager, Editor and WebFinder. In addition to this, the WebFinder also uses a separate schema for presentation purposes and user preferences. This schema, called the User Schema, does not fall into the scope of this document.

This document describes the high-level entities along with their attributes present in the schema. The schema is transformed into database table design. During this transformation, de-normalization to first or second normal form may be required. Certain entities may be decomposed to still smaller entities risking redundancy and duplication. This needed to be done to achieve database vendor independent implementation (same code base for different DBMSs like Microsoft Access, Microsoft SQL Server and Oracle).

Content Schema Entities

❖ ISL_Tape

The ISL_Tape entity represents a video program or a video asset. This is the basic entity in the schema. *(Video program and video asset are used interchangeably in this document.)*

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Tape_Id	a unique Identifier for this program	**M**
Tape_Type	determines if the video asset is a physical asset or a virtual asset. A *virtual asset* means that it is a program created on the fly by aligning the time-codes of two or more physical assets.	**D**
Video_Format	the format of the original video program, for example, BetaSP, 8mm, VHS, etc.	**D**
Video_Standard	represents the video standard, for example, NTSC, PAL, SECAM, etc.	**D**
Tape_Title	title of the video program.	**D**
Tape_Description	a short description about the video program.	**D**
Tape_Log_Date	the date of logging the video program in the system. Can also be the date the video program was produced.	**D**
Tape_KeyFrame	An image to represent this video asset.	**D**
Collection_Name	the name(s) of the collection(s) which this video program is classified under. Every video asset belongs to at least one collection.	**M**

Table 2: ISL Tape Attributes

❖ ISL_Media_Data

This ISL_Media_Data entity stores the physical media storage attributes pertaining to a video asset. For each ISL_Tape entry there may one or more ISL_Media_Data entries referring to the various storage forms to the video program.

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Media_Data_Id	a unique identifier for this media	**M**
Digital_Format	represents the digital format of the video asset, for example video/mpeg, audio/wav, etc	**M**
Capture_Rate	represents the rate in frames-per-second at which the frames for this instance of video program were captured.	**D**
Bitrate	represents the bit rate in bits-per-second of this video component file.	**D**
Derived_From	the identifier of the media data of the video program from which this media instance was derived. NULL if this media instance is the source.	**D**
Duration	Duration on the timeline of the video program	**D**
Offset	Offset from the source timeline indicating the beginning of the media data instance.	**D**
Description	a short description of the media data instance.	**D**
When	the date on which this media data instance was created/logged.	**D**
FWidth	frame width, in pixels, of this video media file.	**D**
Fheight	frame height, in pixels, of this video media file.	**D**
Size	size of this video component media data file.	**D**

Table 3: ISL Media Data Attributes

- **ISL_MEDIA_FILE**

  This entity represents the physical file for the video media instance. In the instance of ISL_Media_Data, there is at least one instance of this entity.

  This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Media_File_Id	a unique identifier for this file	**M**
Repository_Id *(described below)*	represents the location where the file will be residing in the digital format. Can be a directory location or a remote internet/ftp location too with userid, password authentication mechanisms.	**M**
File_Name	the name of the file.	**M**

Table 4: ISL Media File Attributes

- **ISL_VIRTUAL_REPOSITORY**

  This entity represents the repository information of the physical locations of the digital media files for a program
  This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Repository_id	a unique identifier for this entry	**M**
user_id	the user-id for authentication purposes, if any	**D**
password	a password for authentication purposes, if any	**D**
ip_address	the internet protocol address, if any	**D**
directory	the directory location	**M**

Table 5: ISL Virtual Repository Attributes

- **ISL_COLLECTION**

  This entity determines the classification scheme of the video programs. Every video program belongs to at least one collection.

  This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Collection_Id	a unique identifier for this collection	**M**
Collection_Name	Name of the collection	**M**

Table 6: ISL Collection Attributes

❖ **ISL_SCENE**

The video program or a video asset (ISL_Tape) is composed of a series of frames. The point in the video where a frame varies considerably from a series of the previous frames based on some known parameter (like color histogram) is called scene or shot break in MediaSite terms. This entity represents such a frame where a scene/shot break has occurred. MediaSite products allow forcible scene breaks too.
(*Note: Scene is used in the above context henceforth in this document).*

This entity is composed of the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Scene_Id	a unique identifier for this scene	**M**
Tape_Id	the identifier of the program to which this scene belongs	**M**
Start_Time	represents the time in milliseconds, relative to the beginning of the tape, at the starting point of the scene.	**M**
End_Time	represents the time, relative to the end of the tape, at the end point of the scene	**D**
VTR_Time_Code	the time code burned into the raw video tape for that particular scene. It is represented as HH:MM:SS:FF where HH: hour; MM: minutes, SS: seconds from the beginning of the tape and FF: denotes the frame number from the time HH:MM:SS	**M**
Image_Type	the digital type of the image stored in the database for this frame, for example, JPEG, BMP, etc	**M**
Is_Auto_Capture	a flag representing if the scene break was captured automatically or manually forced	**M**
Frame_Data	the actual frame stored as an image in the database	**M**

Table 7: ISL Scene Attributes

❖ **ISL_SEGMENT**

This entity represents a collection of contiguous scenes. It is marked by an start scene and an end scene.

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esired
Segment_Id	a unique identifier for this segment	**M**
Tape_Id	the identifier of a the program to which this segment belongs	**M**
Collection_Id	the identifier of the collection to which this segment belongs	**M**
Seg_Start_Time	denotes the start time in milliseconds of the start scene for this segment.	**M**
Seg_End_Time	denotes the end time in milliseconds of the end scene for this segment	**M**
VTR_Seg_Start_Time_Code	represents in HH:MM:SS:FF the start time for the start scene. HH, MM, SS and FF carry the same meaning as in ISL_Scene.	**M**
VTR_Seg_End_Time_ Code	represents in HH:MM:SS:FF the end time for the start scene. HH, MM, SS and FF carry the same meaning as in ISL_Scene.	**M**
Start_Char	Used to mark the start of closed caption/transcript corresponding to this segment of video. Represented as the numerical offset of the character from the beginning of the closed caption/transcript for the complete video program.	**M**
End_Char	Used to mark the end of closed caption/transcript corresponding to this segment of video. Represented as the numerical offset of the character from the beginning of the closed caption/transcript for the complete video program.	**M**
Description	a short description of the segment	**M**
KeyFrame	a representative image for this segment	**D**

Table 8: ISL Segment Attributes

❖ ISL_GEOSPATIAL_DATA

This entity is used to represent the geospatial information, if any, for a video program. This information uses an external source, the geospatial gazette, to get the geographical information of latitude and longitude.

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Geospatial_Gazette	the external source representing the geographic information	**M**
Geospatial_Data_ Keyword	represents the keyword in the transcript with respect to which the geospatial information is recorded	**M**
Tape_Id	the identifier for the program for which this information is recorded	**M**
Segment_Id	the segment in the program in which the reference to the *geospatial_data_keyword* is made	**M**
Geospatial_Data_Start _Time	the time in the program where reference to the *geospatial_data_keyword* is first made	**M**
Geospatial_Data_End _Time	the time in the program where reference to the *geospatial_data_keyword* ends	**M**
Geospatial_Data_ Latitude	the latitude for the referenced *geospatial_data_keyword* from the *geospatial_gazette*	**M**
Geospatial_Data_ Longitude	the latitude for the referenced *geospatial_data_keyword* from the *geospatial_gazette*	**M**

Table 9: ISL Geospatial Data Attributes

❖ ISL_TAPE_TRANSCRIPT

This entity represents the transcript associated with the video program. Three types of transcripts are envisaged: the closed caption text which is stored along with the broadcast program, a manually written transcript and a transcript generated by he speech recognition process. The character offsets in the transcript are used to associate the part of the transcript to a segment of the video program, this process is called *aligning the transcript* to the segment data.

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Tape_Id	the identifier of the tape to which this transcript belongs	**M**
Transcript_Id	a unique identifier for this transcript	**M**
Transcript_Type	the type of transcript : closed caption text, manually generated or speech recognition output	**M**
Transcript_Chunk_Text	the actual transcript text	**M**

Table 10: ISL Tape Transcript Attributes

❖ ISL_TAPE_ALIGNMENT

This entity represent the alignment data, as discussed earlier, for the transcript of a video program. This is primarily used for speech recognition generated and manually generated transcripts.

This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Transcript_Id	the identifier of the transcript for which this data is recorded	**M**
Tape_Alignment_ Start_Time	the time from the beginning of the program for which the text to video alignment begins	**M**
Tape_Alignment_ Start_Char	the offset of the character from the start of the transcript for which the alignment starts	**M**
Tape_Alignment_ End_Time	the time from the beginning of the program for which the text to video alignment ends	**M**
Tape_Alignment_ End_Char	the offset of the character from the start of the transcript for which the alignment ends	**M**

Table 11: ISL Tape Alignment Attributes

❖ ISL_OWNER, ISL_PRICE

These two entities represent the Owner and Price, if any, information of the video program. The **ISL_OWNER** entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Owner_Id	a unique identifier for the owner	**M**
Owner_Name	the name of the owner	**M**
owner_address	address of the owner	**M**
owner_phone_number	phone number for the owner	**D**
owner_email_id	email_id for the owner	**D**
owner_comment	a comment for the owner	**D**

Table 12: ISL Owner Attributes

The **ISL_PRICE** entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Owner_Id	a unique identifier for the owner	**M**
Price	the price for the complete tape	**D**
Tape_Id	the identifier for the tape for which the price is recorded	**M**
Segment_Id	a segment to can have a price	**M**

Table 13: ISL Price Attributes

The data *may* be collected and recorded for these two entities. The system provides a facility to record this data.

The entities described above store the mandatory information for a video program, exceptions being of **ISL_OWNER** and **ISL_PRICE.**

The system also facilitates the users to generate meta-data of their own for a video program where they can capture and store any information useful for them. The examples of a such an information are meta-data like name of the producer, date of broadcast, etc. The following entities are used by the system to store this user specific and user generated information.

- **ISL_CATEGORY**

  This entity represents the classification of the user-defined fields into various levels like the fields at the program level, fields at segment level and fields at scene level. The examples are:
  a. Video Program Level: Producer Name, Date of Broadcast.
  b. Segment Level: Reporter Name, Actor Name.
  c. Scene Level: Scene annotation, Key Frame description.

  A user is free to define his own classification item (*category*) and manually enter the data for his/her classification item.

  This entity consists of the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Category_Id	unique identifier for this category	**M**
Category_Name	name for this category	**M**

Table 14: ISL Category Attributes

- **ISL_FIELD**

  This entity represents the fields belonging to a category. For example, the Video Program Category has fields Producer Name. The values for the fields can be pre-defined or overwritten by the user. This entity has attributes which are used by GUI for display purposes.

  This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Field_Id	a unique identifier for this field	**M**
Field_Type	the datatype of the value stored in this field	**M**
Category_Id	the identifier of the category to which this belongs	**M**
Field_name	the name of this field	**M**
Field_display_Name	the name of the field to the displayed in the GUI	**M**
Field_Length	the maximum number of bytes of a value to be stored in this field	**M**
Field_overridable	a flag to denote if the pre-defined value is overridable or not	**M**
Field_Required	a flag to denote if it is a required	**M**

	field or not, i.e., whether an application needs to provide a data value for this field or not	
Field_is_viewable	a flag to denote if the field is displayed in the GUI or not	**M**
Field_Search_type	a flag to denote if this field is to be included in the fielded search in GUI or not	**D**
Field_MultiSelect	a flag to denote if multiple data values can be assigned for this field	**D**

Table 15: ISL Field Attributes

- **ISL_FIELD_VALUE**
  This entity represents the pre-defined values stored for a particular field.
  The attributes for this entity are:

Attribute Name	Description	**M**andatory Or **D**esirable
Field_Value_Id	a unique identifier for this field value	**M**
Field_Id	the identifier of the field to which this value belongs	**M**
Field_Value	the actual data value stored for this field	**M**
Field_value_position	represents the position of the pre-defined field value in a pull-down list	**D**

Table 16: ISL Field Value Attributes

- **ISL_USER_DEFINED_FIELD**
  This entity stores the user-defined values for the fields.
  The attributes for this entity are:

Attribute Name	Description	**M**andatory Or **D**esirable
Udf_Field_Id	the identifier of the field to which this value belongs	**M**
Udf_tape_id	the identifier of the program with which this field value is associated	**M**
udf_Segment_id	the identifier of the segment of the program with which this field value is associated	**M**
udf_Field_Value	the actual data value stored for this field	**M**

Table 17: ISL User Defined Field Attributes

❖ **ISL_IMAGE_HIST_CLUSTER**

This entity holds the image histogram data for a perceptual color clustering. This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Cluster_Id	the unique identifier for this cluster	**M**
bottom_left_coeffx (x = 0 t o 11 -> 12 attributes	coordinates of the region	**M**
upper_right_coeffx (x = 0 to 11 -> 12 attributes	coordinates of the region	**M**

Table 18: Image Histogram Cluster Attributes

❖ **ISL_IMAGE_HIST**

This entity holds the region data for a perceptual color clustering. This entity has the following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Region_Id	a unique identifier for this region	**M**
Cluster_Id	the identifier for the cluster	**M**
image_hist_coeffx (x = 0 t o 11 -> 12 attributes	coordinates of the region	**M**

Table 19: ISL Image Histogram Attributes

❖ **CONT_SEGMENT_TEXT**

The following entity is created for the purpose of indexing the transcript and field data because of the current limitations of the text indexing engines. It duplicates the data broken into chunks of 2000 bytes and stored in this entity to extract a better performance from the text retrieval engine.This entity has following attributes:

Attribute Name	Description	**M**andatory Or **D**esirable
Tab_pkcol	the unique identifier for this row	**M**
ISL_Tape_Id	the identifier of the program with which this value is associated	**M**
ISL_Segment_Id	the identifier of the segment of the program with which this value is associated	**M**
Field_name	the name of the field to which the text belongs	**M**
TXT	the actual data text value stored	**M**

Table 20: Cont Segment Text Attributes

Appendix B: Process Flow & Data Flow for GeoSpatial Referencing

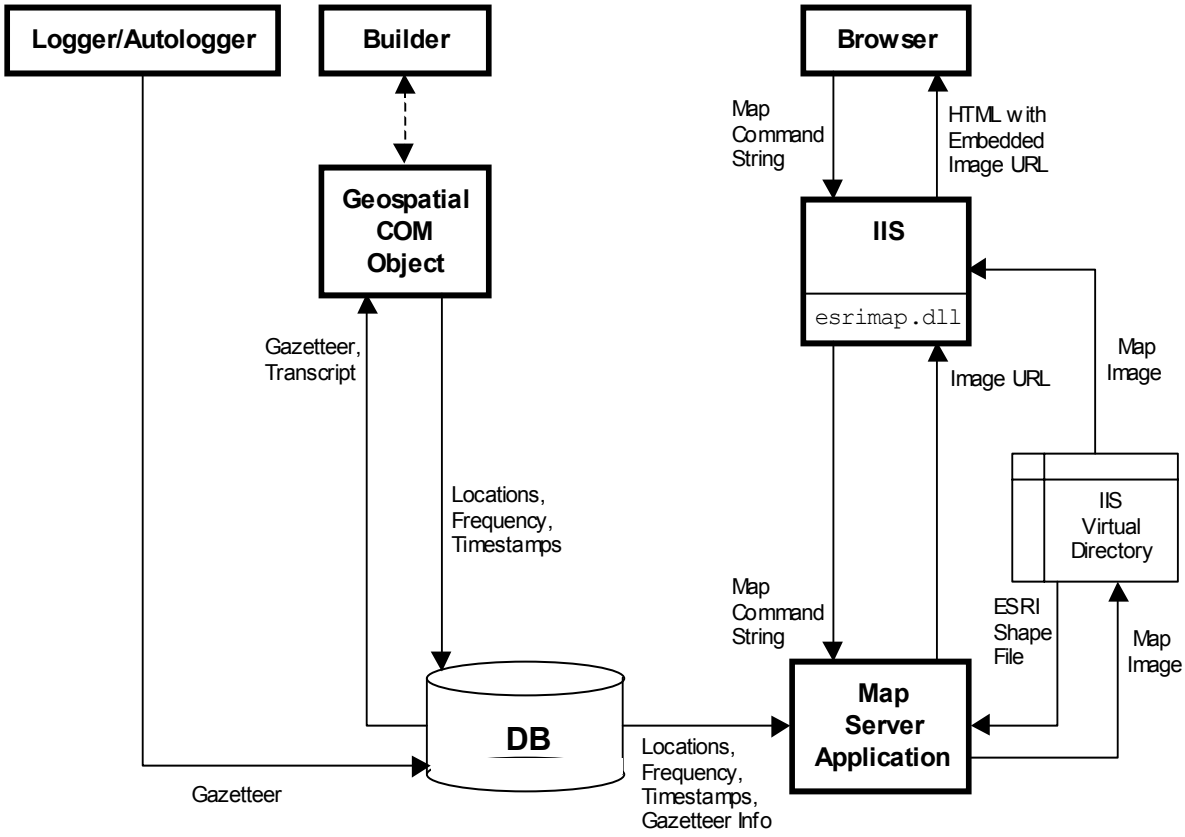

Figure 1: Process and Data Flow for Geospatial Referencing

Appendix C: High Level Workflow for GPS Referencing

- *Record with GPS data*: GPS receiver and encoding equipment will be used to capture the camera's GPS coordinates or physical location while shooting and embed it into the videotape.
- *Extract GPS coordinates*: The decoding equipment will be used to extract the GPS coordinates from the videotape during playback and made available to the Indexer.
- *Update Database*: Indexer will update the database with GPS enhanced data.
- *Use GPS enhanced data*: The user-interface will use the GPS data to enhance the search and display.

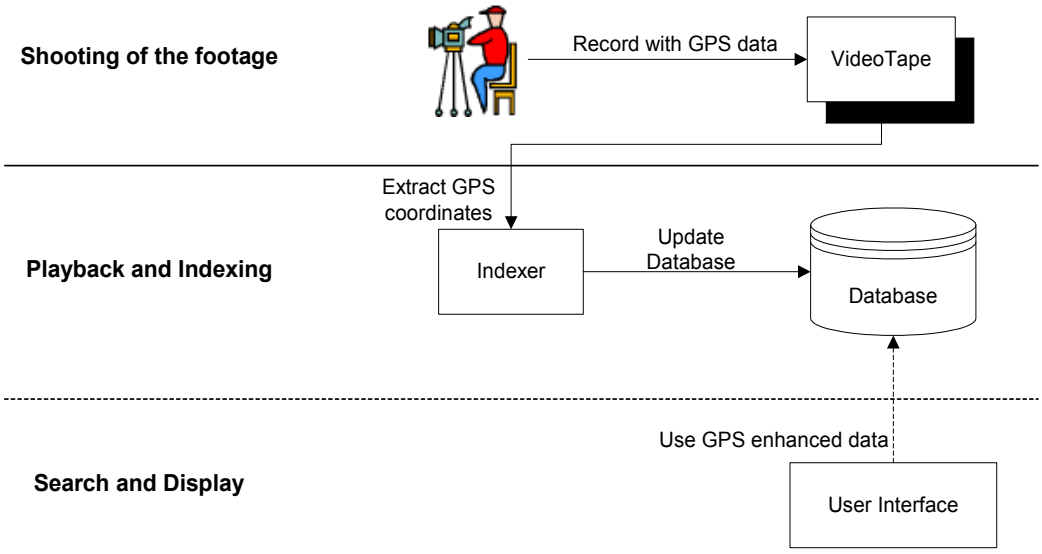

Figure 2: High Level Workflow for GPS Referencing

Appendix D: Analyst Annotation Workflow

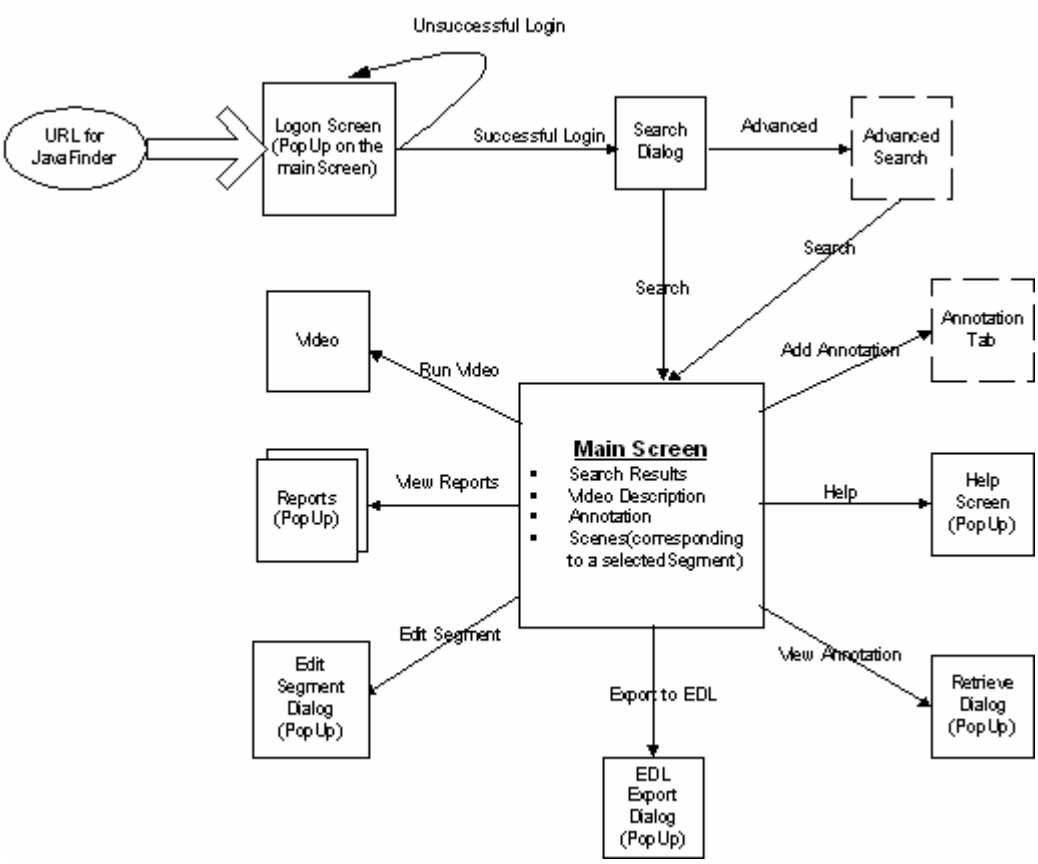

Figure 3: Analyst Annotation Workflow

Appendix E: Analyst Annotation Security Hierarchy

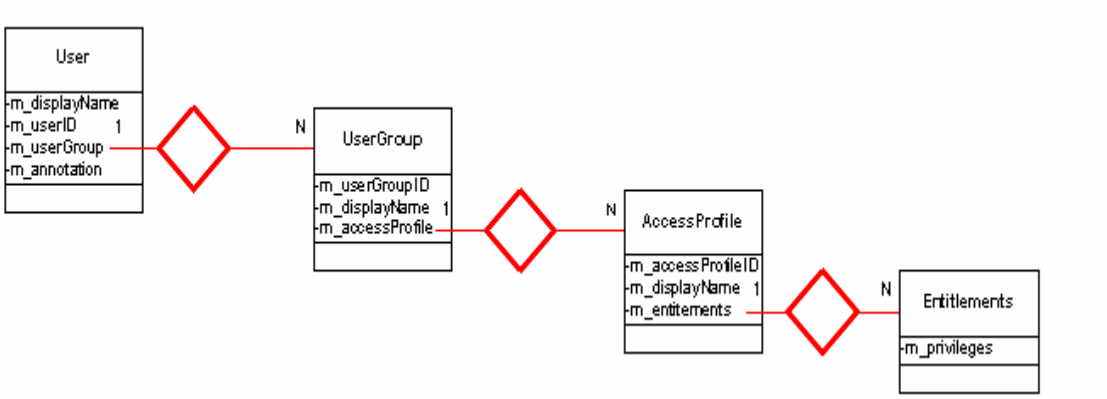

Figure 4: Analyst Annotation Security Hierarchy

Appendix F: Analyst Annotation Screen Representations

Login Screen

Figure 5: Analyst Annotation Login Screen

Search Dialog

(Pops up after a successful Login)

Simple Search Dialog (with simple/Advanced Toggle button)

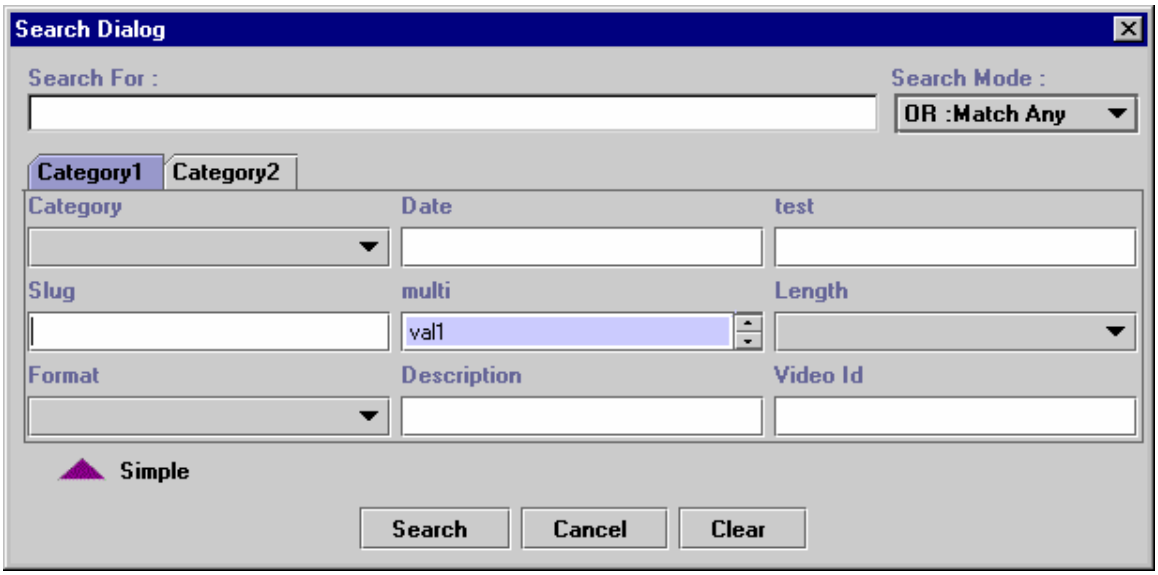

Figure 6: Analyst Annotation Search Dialog

Advanced Search Dialog (After Simple-toggle button is clicked on the above screen)

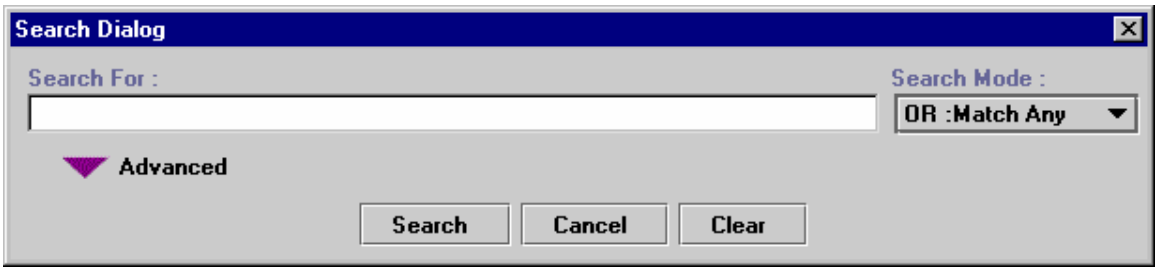

Figure 7: Analyst Annotation Advanced Search Dialog

The Main Screen

(The first shows the Video details tab and the second shows the Annotation tab)

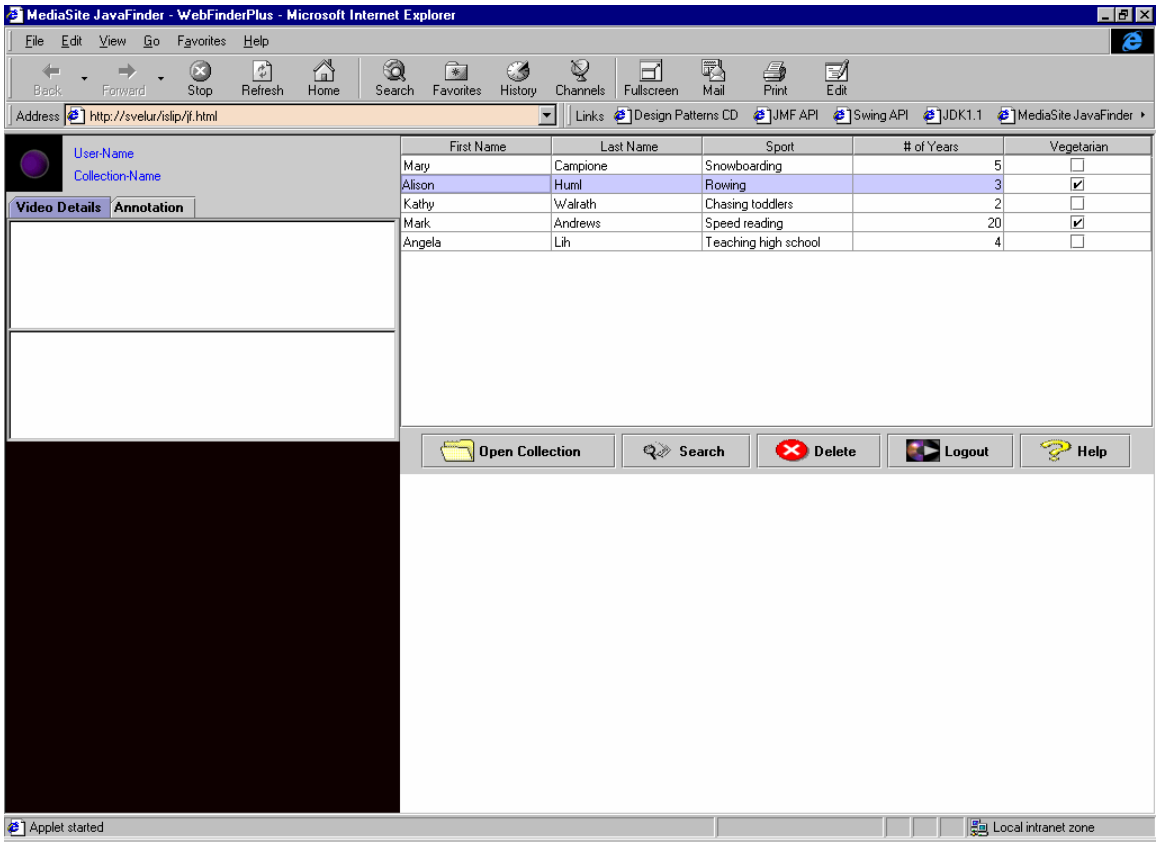

Figure 8: Analyst Annotation Main Screen

Retrieve Dialog

(A Dialog that pops up when the user clicks on the Retrieve button in the Annotation tab)

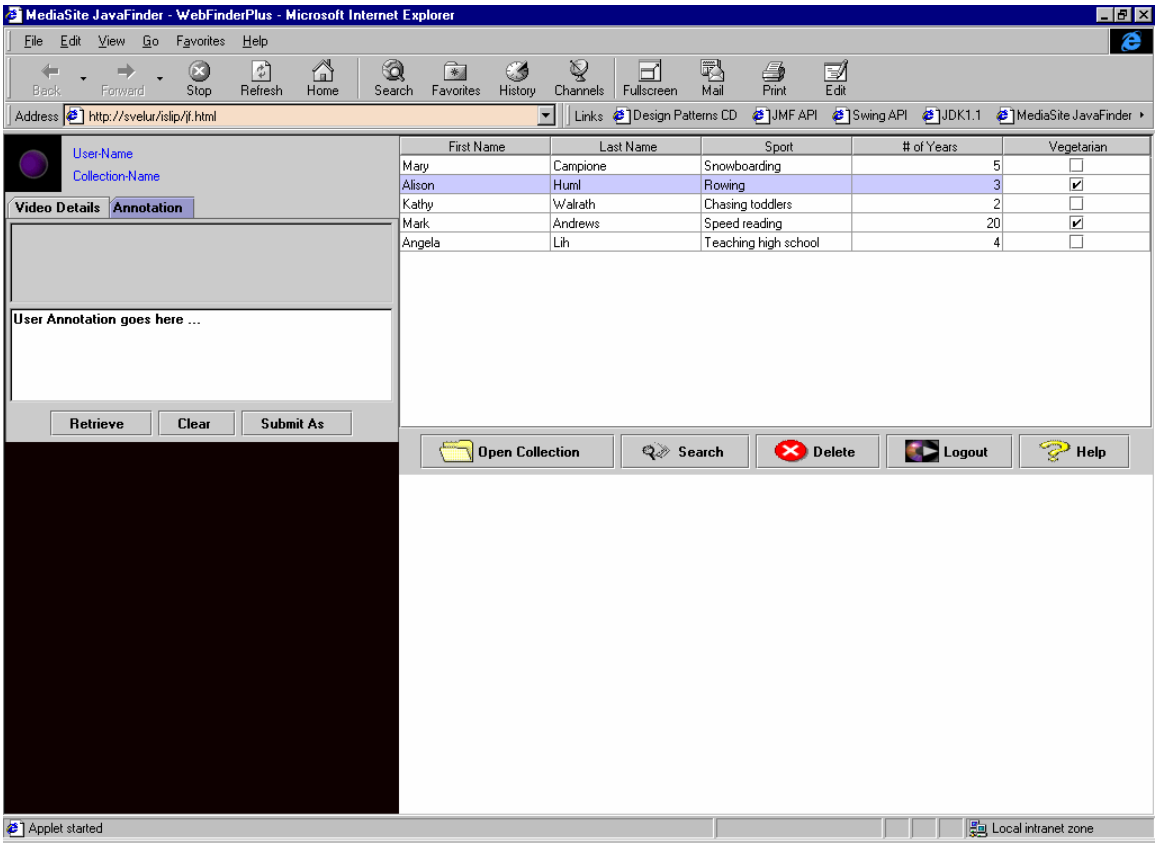

Figure 9: Analyst Annotation Retrieve Dialog

Appendix G: Motion Detection Parameters Screen Representations

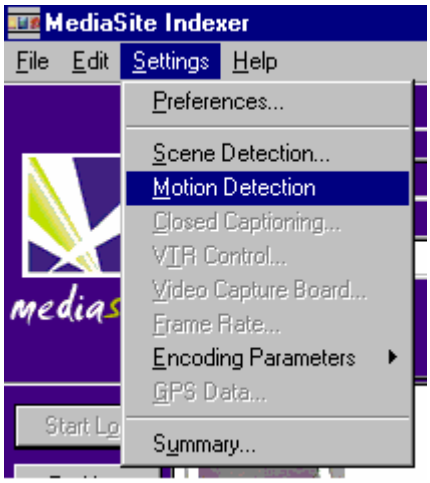

Figure 10: Menu, modified in order to make "motion detection" page available.

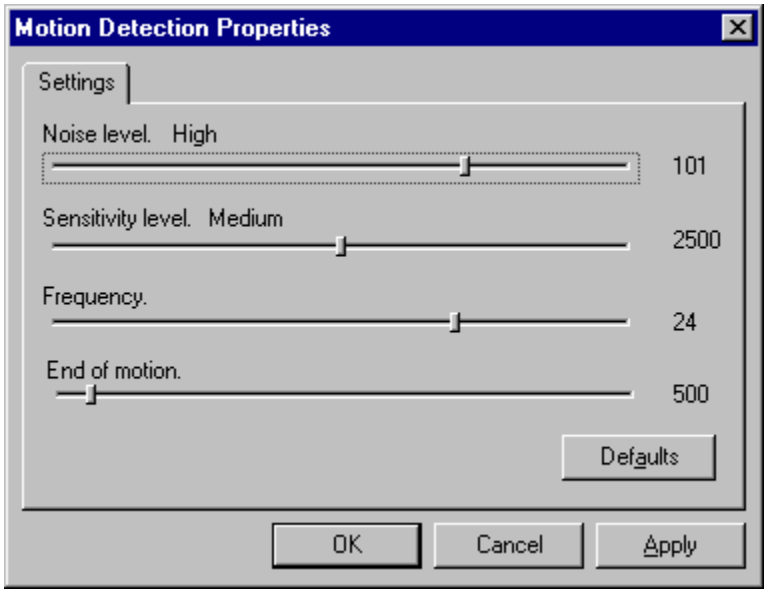

Figure 11: Motion Detection Property Page

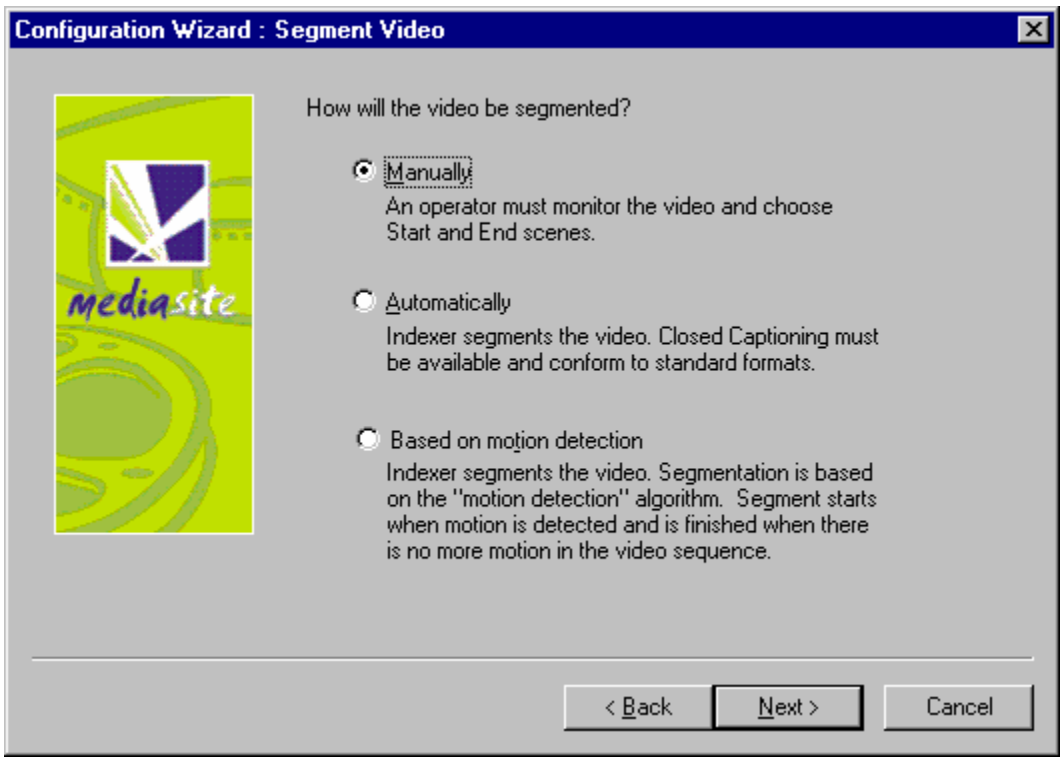

Figure 12: Segmentation Properties – Motion Detection option

Appendix H: Image Match Screen Representation

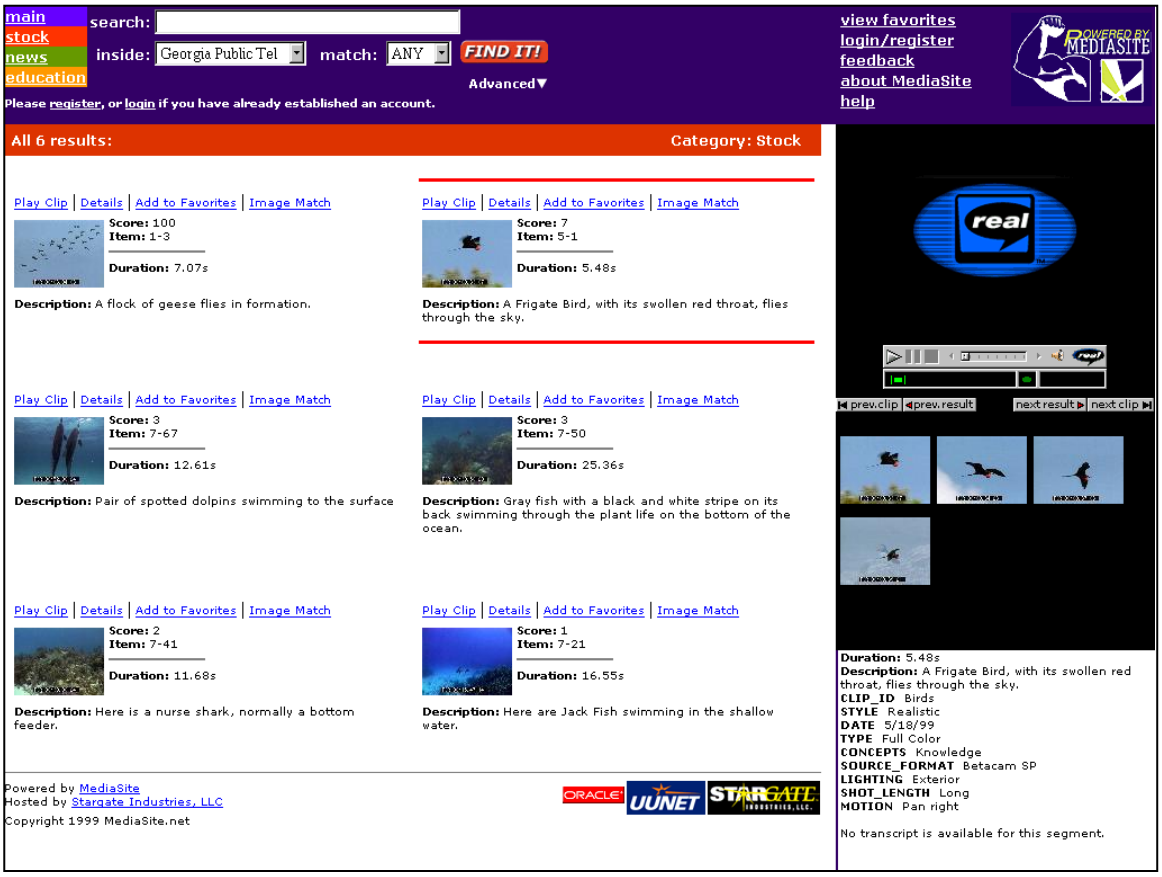

Figure 13: Image Match Screen