M.I.T. Media Laboratory Perceptual Computing Technical Report No. 283, June 1994

Short Version Appeared: IEEE Multimedia Magazine, Vol. 1, No. 2, pp. 73-75, 1994

Video And Image Semantics:

Advanced Tools For Telecommunications

Alex Pentland, Rosalind Picard, Glorianna Davenport, Ken Haase
Room E15-387, The Media Laboratory
Massachusetts Institute of Technology
20 Ames St., Cambridge MA 02139
Email: {sandy,picard,gid,haase}media.mit.edu
Phone: (617) 253-0648, FAX: (617) 253-8874

January 11, 1995

ABSTRACT

Within the next decade, the majority of data carried over telecommunications links is likely to be visual material. The biggest problem in delivering video and image services is that the technology for organizing, searching, and presenting images is still in its infancy. Consequently we are developing tools for building and browsing multimedia databases, and for using these databases to automatically create multimedia presentations. This paper describes our demonstration system, which gathers and presents video over standard ISDN telephone lines.

Keywords: Image and Video Database, Multimedia Presentation, Image Compression, Database Annotation, Image and Video Semantics.

1 Introduction

Within the next decade, the majority of data carried over telecommunications links is likely to be visual material. The biggest problem in delivering video and image services is that the technology for organizing, searching, and presenting images is still in its infancy. Consequently, the process of assembling a good multimedia presentation is extremely laborious and expensive.

If multimedia services are to become practical, we must be able to build multimedia databases quickly and cheaply. We must be able to extract and represent the content of the video clips and images sufficiently well so that the computer can automatically select material that fulfills the needs of wide range of users and purposes. And finally, the computer must be able to automatically assemble this material into a coherent presentation.

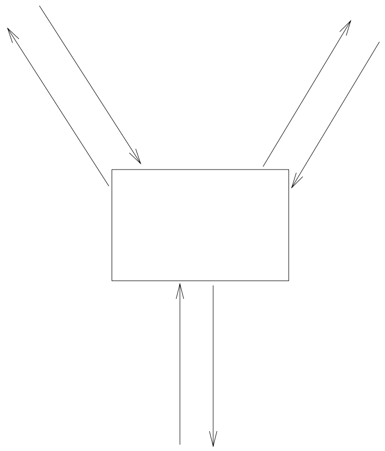


Figure 1: Overview of the system we are building: video and images come via ISDN lines, are subjected to semantics-preserving compression, and stored in an analogical database. Textual annotations are then added by the database builder. When a user query is received, the stored semantics are used to automatically create an appropriate presentation, which is sent out via ISDN lines using h.261 compression.

Consequently the goal of the M.I.T. Media Laboratory's Advanced Tools for Telecommunications Project is to develop tools for automatically understanding and using the semantics of video and image materials. The organization of this paper will be to first present an overview of the system, and then to describe the representations and important interfaces in more detail. Additional information about this system, referenced papers, and some of the computer code is available by anonymous FTP from whitechapel.media.mit.edu.

2 System Overview

Usually it is impossible to completely annotate a multimedia database. It is simply too expensive to have people type in text annotations for each property of an image — an image *really is* worth 1,000 words! Equally distressing is that each person's judgement of similarity is different, due to different weighting of the various image features, so that even for simple object-to-object comparisons it is difficult to obtain consistent, repeatable annotations.

For such comparison questions, it would be much better if the computer could "see" what is in the images, so that it could answer our questions by looking through the pictures. One problem with this approach is that images are just too large to efficiently store and search thousands of them. A more profound problem is that computers today do not have any way of knowing the semantic content of an image; there is no equivalent of computer-readable text or semantically-meaningful chunks like words for images or sound.

To effectively search through images and video, therefore, you need to be able to express the *content* of the image in a very compact way. In the image compression literature the process

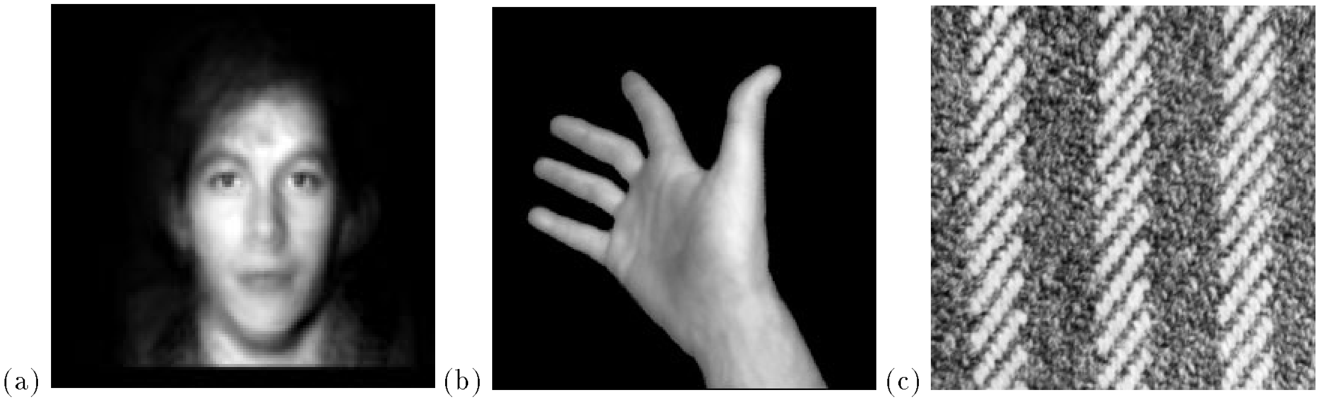


Figure 2: Semantics-preserving compression. Shown here are three examples of images reconstructed from the coefficients used for database search. (a) 30 coefficients, (b) 100 coefficients, (c) 60 coefficients

of compressing an image based on it's *semantic content* is is often called *semantic bandwidth compression*. We have extended this idea to that of *semantics-preserving compression*, and applied it to multimedia databases.

Figure 1 shows the outlines of the system we are building around this representation of video semantics. It consists of three major modules — input, annotation, and presentation — each of which are connected to a central database store and can be accessed via ISDN telephone lines. The system functions by taking in video and image material over ISDN lines, where it is parsed it into keyframes and subjected to *semantics-preserving image compression*, and then stored in an analogical database. This material can then be further annotated off-line, using the existing annotations to provide a "power assist" to the annotation process. Finally, when users ask a question the stored semantics and on-line similarity judgements are used to automatically assemble a multimedia presentation that can be sent back out over the telecommunications network.

2.1 The Input Module: Semantics-Preserving Compression

Our system functions by taking measurements of image features — brightness, edges, texture measures, etc. — and then using either the Karhunen-Loeve or Wold transforms to obtain a compact description of the set of images in terms of their most salient characteristics [6, 10, 12, 8]. The Karhunen-Loeve transform is used when the detailed relations between things are important, such as when describing the geometry of a scene or a human face. The Wold transform is used when describing more textural properties, such as orientation, randomness, or periodicity.

In both cases the resulting representation of the image content can be searched directly, *without* decompression, to find objects and compare textures. This new representation technique, which we call *semantics-preserving compression*, can also provide an extremely compact code for image compression purposes. Some examples of semantics-preserving compression are shown in Figure 2.

An important example of semantics-preserving compression applied to video is *keyframe extraction*. Editors and artists have long known that the semantic content of video can be accurately summarized by a series of appropriately-selected frames (images) taken from the video stream. These still-frame images are called *keyframes* and a sequence of them is called a *storyboard*.

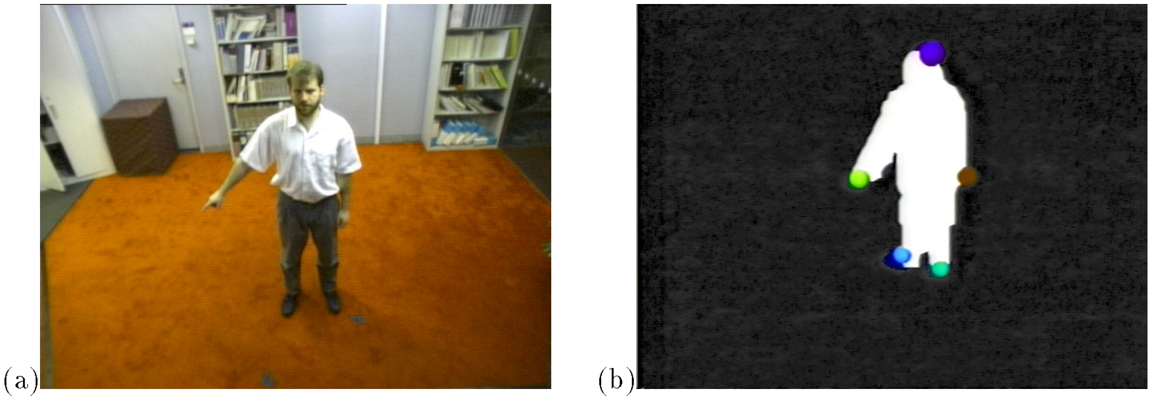


Figure 3: Using motion and color information, we can separate foreground objects from background. This figure shows a system that extracts the outlines of people in view; a geometric analysis of the outline is then used to label position of head, hands, and feet. This system runs at 20 frames/second without special hardware.

Keyframes are images that are "characteristic" or "typical" of the video clip's content; we have found that good keyframes can be found by analysis of the camera and scene motion. For instance, good keyframes often occur in the middle of no-motion segments, and in the middle of segments where the camera is tracking a foreground object, as well as at the beginning and end of clips.

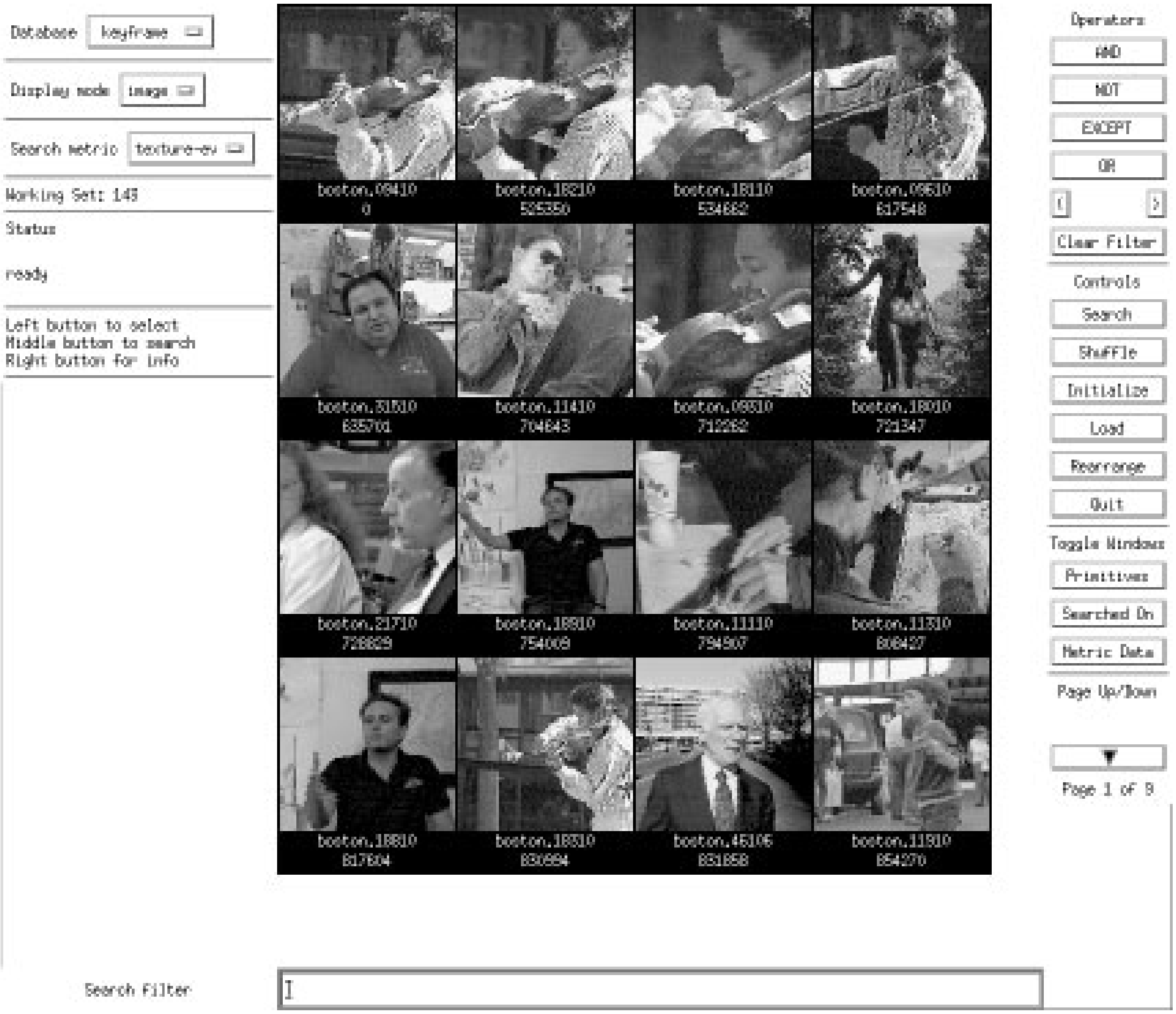
We can automatically extract such keyframes by computer analysis of the image motion and color in the video clip. By finding coherent subregions of motion in the video clip, we can automatically segment the scene into foreground, midground, and background, as illustrated in Figure 3. Then by comparison of foreground and background motions, we can automatically select useful keyframes [6].

2.2 The Annotation Module

We have used this automatically-produced representation of image content to create a browsing and database search tool called PHOTOBOOK [6]. This tool allows the user to browse large image databases quickly and efficiently, using both textual annotation information and by having the computer search the images using the descriptions resulting from the semantics-preserving compression process. This allows the user to search in a flexible and intuitive manner, using either analogies, e.g., "show me this type of image," or visual similarities, e.g., "show me images that look like this." Figure 4 shows using PHOTOBOOK to find similar keyframes from a video database.

These visual similarity relations can, of course, be augmented by more traditional text annotations. One method of accomplishing this is to use the visual similarities to give a "power assist" to the annotation process: you annotate one image, then use PHOTOBOOK to find all the visually similar images, and then simply propagate the annotations for the original image to the visually similar images. In small-scale tests, this power-assisted annotation process can cut the cost of annotating a image database by more than 80% [9].

Even with such an efficiency gain, the process of annotating images can still be quite expensive. We have therefore created the MEDIA STREAMS interface to make the annotation easier. MEDIA STREAMS uses an *icon language* for annotation, rather than having the user type in text strings.



Database keyframe

Display mode image

Search metric texture-ev

Working Set: 148

Status

ready

Left button to select
Middle button to search
Right button for info

Search Filter

Figure 4: An example of a content-based image query: Are there any images similar to the image of the violin player shown at the top left? After searching a database of several hundred keyframes, the result is the series of images shown here. The images are ranked by similarity to the query image in terms of their visual content. Currently the system does surprisingly well...although usually there are some cases where it is difficult to understand the computer's similarity judgement.

In small scale expriments we have found that this sort of iconic interface is not only more efficient, but also produces annotations that are more consistent across different users and different sessions.

2.3 The Presentation Module

Providing multimedia information is not like providing the latest cost figures from accounting. Each multimedia item shows only a small scene or action, so to provide information one has to string a series of images and video clips together so that they *tell a story*. Thus rather than treating multimedia database queries in a manner similar to traditional database queries, we must try to respond to a user query by creating a *presention*.

Because the material available for each query will be different, the machine must use similarity judgements (based on descriptions generated by semantics-preserving compression) together with analogical reasoning to decide what shots and stories best match the query. In our system this is accomplished using FRAMER, a persistent knowledge representation that uses analogical and similarity reasoning in addition to logical and set operations [2].

This allows the system to know which video clips are "right" for telling a particular story in the current context. However this is not the whole story, because a presentation requires sequencing the relevant video clips together into a full presentation. We have therefore created a story telling interface called HOMER that uses "semantic templates" first to guide the search for entries that are relevant to answering the user's query, and then to assemble these clips into a video presentation that answers the user's question [3, 4].

3 Algorithms for Semantics-Preserving Image Compression

The input module of our system takes in video and still images over ISDN lines, performs an initial visual content analysis, and enters the data and derived descriptions into a central database. The purpose of these automatically-produced descriptions is to allow us to efficiently search and browse the database based on visual similarity.

The ability to search at query-time for instances of the same (or visually similar) image events depends on two conditions:

- There must be a similarity metric for comparing objects or image properties (*e.g.,* shape, texture, color, object relationships, *etc.*) that matches human judgments of similarity. This is *not* to say that the computation must somehow mimic the human visual system; but rather that computer and human judgments of similarity must be generally correlated. Without this, the images the computer finds will not be those desired by the human user.

- The search must be efficient enough to be interactive. A search that requires minutes per image is simply not useful in a database with millions of images. Furthermore, interactive search speed makes it possible for users to recursively refine a search by selecting examples from the currently retrieved images and using these to initiate a new select-sort-display cycle. Thus users can iterate a search to quickly "zero in on" what they are looking for.

Consequently, we believe that the key to solving the image database problem is *semantics-preserving image compression*: compact representations that preserve essential image similarities. This concept is related to some of the "semantic bandwidth compression" ideas put forth in the context of image compression [7]. Image coding has utilized semantics primarily through efforts to compute a compact image representation by exploiting knowledge about the *content* of the image. A simple example of semantic bandwidth compression is coding the people in a scene using a model specialized for people, and then using a different model to code the background.

In the image database application, compression is no longer the singular goal. Instead, it is important that the coding representation 1) be "perceptually complete" and 2) be "semantically meaningful." The first criterion will typically require a measure of perceptual similarity. Measures of similarity on the coefficients of the coded representation should correlate with human judgments of similarity on the original images.

The definition of "semantically meaningful" is that the representation gives the user direct access to the parts of the image content that are important for their application. That is, it should be easy to map the coefficients that represent the image to "control knobs" that the user finds important. For instance, if the user wishes to search among faces, it should be easy to provide control knobs that allow selection of facial expressions or selection of features such as moustaches or glasses. If the user wishes to search among textures, then it should be easy to select features such as periodicity, orientation, or roughness.

Having a semantics-preserving image compression method allows you to quickly search through a large number of images because the representations are compact. It also allows you to find those images that have perceptually similar content by simply comparing the coefficients of the compressed image code. Thus in our view the image database problem requires development of semantics-preserving image compression methods.

3.1 Developing Specific Representations

How can we design "semantics-preserving image compression" algorithms? Our general idea is to first transform portions of the image into a canonical coordinate system that preserves perceptual similarities, and then to use a lossy compression method to extract and code the most important parts of that representation. By careful choice of transform and coding methods this approach can produce an optimally-compact, semantics-preserving code suitable for image database operations.

Note that because different parts of the image have different characteristics, we must use a variety of representations, each tuned for a specific type of image content. This is the same requirement as for semantic bandwidth compression. In the examples below we will describe representations for faces [12], textures [8], hand tools [10] and video keyframes [6].

Moreover, we must take care to distinguish between two basic classes of image description — texture-like descriptions of "stuff" and object-like descriptions of "things" — because they seem to play fundamentally different roles in human perception and cognition, corresponding roughly to the distinction between mass nouns and count nouns in language. While both refer to constellations of image features, "stuff" descriptions pool the features without regard to detailed local geometry, while the "things" descriptions preserve local geometry.

The necessity for multiple content-specific representations means that we must also have an efficient, automatic method for developing "basis functions" specific to either objects or textures. For representing object classes, which requires preservation of detailed geometric relations, we use an approach derived from the Karhunen-Loève transform [12, 10]. The Karhunen-Loève transform is known to provide an optimally-compact linear basis (with respect to RMS error) for a given class of signal. For characterization of texture classes, we use an approach based on the Wold decomposition

[8]. This transform separates "structured" and "random" texture components, allowing extremely efficient encoding of textured regions while preserving their perceptual qualities. For mathematical details see references [6, 12, 10, 8]. Detailed technical descriptions and computer code for these algorithms can be obtained by anonymous FTP from whitechapel.media.mit.edu.

4 Annotation Interfaces

Once the image data and automatically-produced descriptions have been entered into the central database, we now need to be able to browse the database and to be able to add additional annotations. Our image database browsing tool is called PHOTOBOOK. It uses the automatically-produced image descriptions to allow users to search for images by using either the shape or appearance of objects, or by using their textural properties.

The PHOTOBOOK interface can also be used to provide a "power-assist" to the text annotation process, by grouping visually similar images together so that we can annotate the entire group at once [9]. This can result in significant improvements in annotation efficiency.

In addition, the annotation process itself can be improved by using representations appropriate for images and video, and by using an iconic annotation language rather than a textual language. These ideas are the basis for the MEDIA STREAMS interface described below.

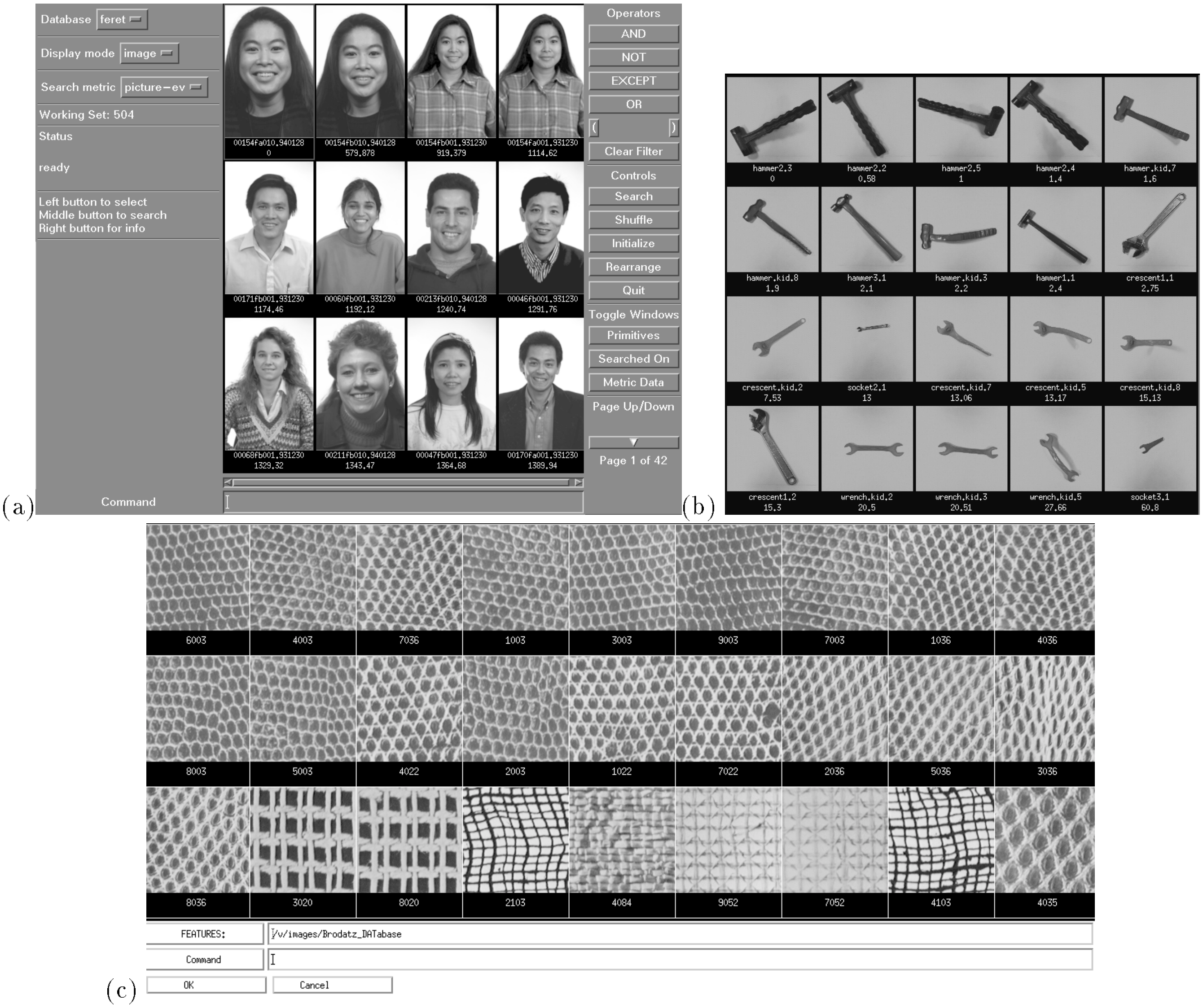
4.1 PHOTOBOOK

PHOTOBOOK is a computer system that allows the user to browse large image databases quickly and efficiently, both by using text annotation information in an AI database and by having the computer search the images directly based on their content [6]. This allows people to search in a flexible and intuitive manner, using semantic categories and analogies, *e.g.*, "show me images with text annotations similar to those of this image but shot in Boston," or visual similarities, *e.g.*, "show me images that have the same general appearance as this one."

Interactive image browsing is accomplished using a Motif interface. This interface allows the user to first select the category of images they wish to examine; *e.g.,* pictures of white males over 40 years of age, or images of mechanic's tools, or cloth samples for curtains. This subset selection is accomplished by searching text annotations using an object-oriented, memory-based AI database called FRAMER [2], described in more detail below. PHOTOBOOK then presents the user with the first screenful of these images (see Figure 4); the rest of the images can be viewed by "paging" through them one screen at a time.

Users most frequently employ PHOTOBOOK by selecting one (or several) of the currently-displayed images, and asking PHOTOBOOK to sort the entire set of images in terms of their similarity to the selected image (or set of images). PHOTOBOOK then re-presents the images to the user, now sorted by similarity to the selected images. The select-sort-redisplay cycle typically takes less than one second. When searching for a particular item, users quickly scan the newly-displayed images, and initiate a new select-sort-redisplay cycle every two or three seconds.

Photobook can have many different types of image descriptions available to it. Figure 5 illustrates searches on the basis of image appearance, shape, and textural properties. In each of these searches, the image at the upper left is the query image submitted by the user. The remainder of the images are those PHOTOBOOK thinks are most similar to the query image, ordered by similarity from top to bottom and left to right. A typical search takes under one second.



Database feret

Display mode image

Search metric picture−ev

Working Set: 504

Status

ready

FEATURES: /v/images/Brodatz_DATabase

Figure 5: In each of these three cases, the image at the upper left was selected by the user, and PHOTOBOOK returned the remaining images sorted by facial, shape, or texture similarity. Search accuracy over the database of 504 face images is 99.4%, over the 60 hand tool images is 100%, and over the 1008 texture images is 90%. Note that in each case the matching can be made position, orientation and scale invariant (modulo limits imposed by pixel resolution) if such invariance is desired by the user.

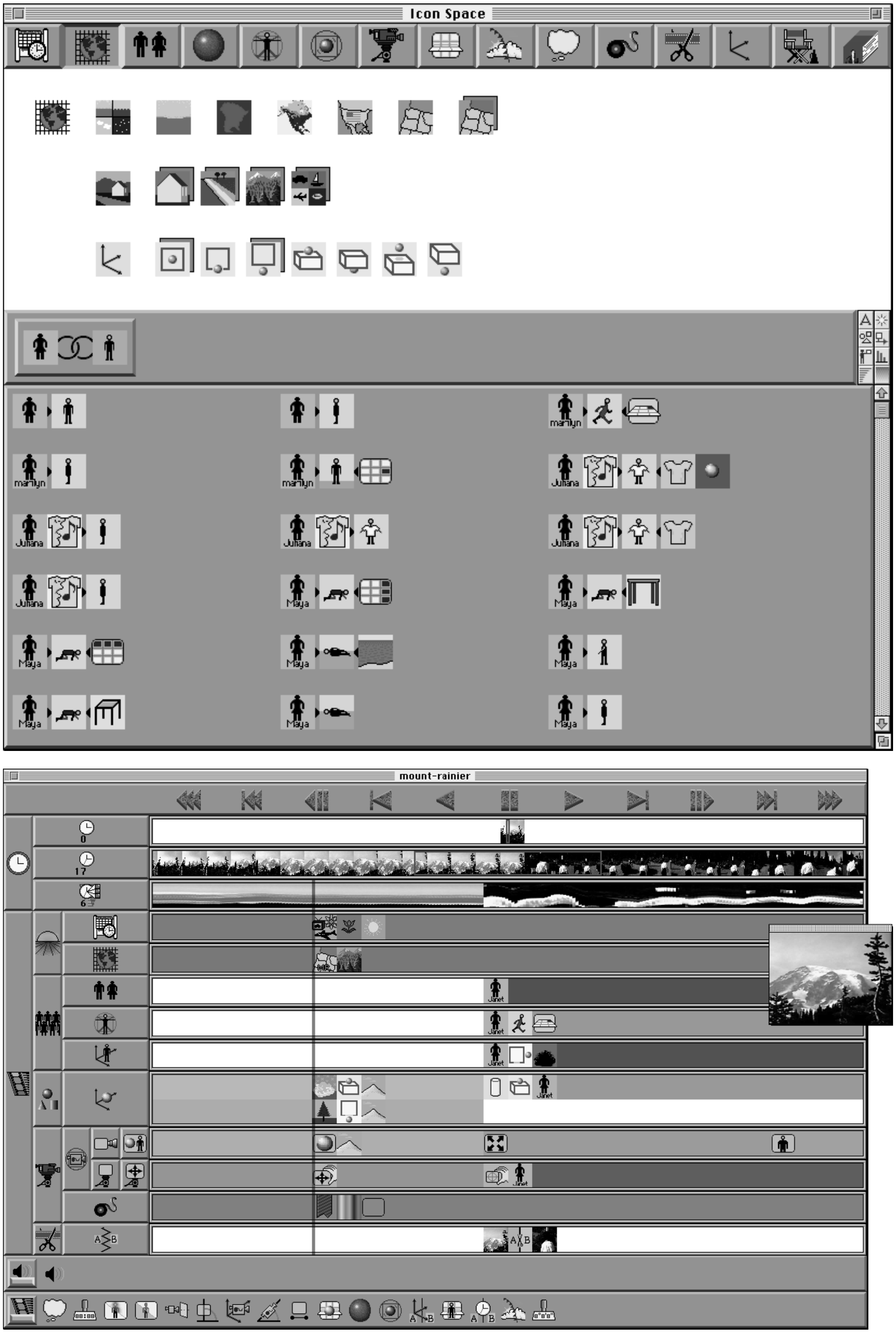


Figure 6: The upper image shows the icon palatte used by MEDIA STREAMS for annotation; this illustrates some of the icon language's spatial relationships. The bottom image shows how the icons are used to produce a layered annotation of a video stream.

PHOTOBOOK can also handle combinations of these descriptors, *e.g.*, shape and appearance, which we will illustrate using 3-D data of human brain ventricles. It can also handle complex functions of text annotations via functionality of the Framer knowledge representation language [2]. In tests on face image databases PHOTOBOOK has demonstrated a recognition accuracy that is competitive with that achieved by using single fingerprints. Similarly, PHOTOBOOK has shown itself to be very effective at finding perceptually-similar images in clip-art texture databases.

The ability to determine visual similarity can be used to aid the process of traditional text annotation. This is accomplished by using the visual similarities to give a "power assist" to the annotation process: first the user annotates a particular image, then they use PHOTOBOOK to find all the visually similar images, and then propagate the annotations for the original image to the visually similar images. In this way the user avoids having to reannotate visually similar images. In small-scale tests, this power-assisted annotation process can cut the cost of annotating a image database by more than 80% [9].

4.2 MEDIA STREAMS

Even with the "power-assist" provided by PHOTOBOOK, the process of annotating images is still quite expensive. The MEDIA STREAMS interface, created by Marc Davis working with Ken Haase, serves to make the annotation process faster and to alleviate problems caused by idiosyncratic terminology and divergence of description [1].

MEDIA STREAMS addresses these problems with four innovative design ideas:

- *Stream based annotation.* Video (and audio) are treated as streams of information upon which annotation is layered rather than as collections of pre-segmented clips to which annotations are attached. This allows the construction of new clips or segments out of the overlaps or unions of independent annotations.

- *Iconic description for physical appearance and action.* Physical appearances and actions are described by visual icons, providing a base-level language that is both easy to read and unambiguous.

- *A generative controlled vocabulary.* Iconic primitives constitute a controlled vocabulary that can be extended by a variety of means of combination. These include compounding icons into sentences indicating case-frame like relations, specializing icons with text strings providing additional detail, and movement along a specialization/generalization hierarchy.

- *Descriptor search enables convergence* . The same search mechanism used for video material can be used for the descriptors themselves, making it easier to annotate descriptions in ways in which they have been annotated before. This provision supports the convergence of descriptions; in small-scale formal tests, we have demonstrated that the use of descriptor search in generative iconic vocabulary leads different individuals to describe similar footage similarly to one another and different footage distinctly.

Figure 6 shows the MEDIA STREAMS interface. In small scale expriments we have confirmed that this sort of interface is not only more efficient, but also produces annotations that are more consistent across different users and different sessions.

5 The Query and Output Interface: Power-Assisted Presentation

The goal of our system is to respond to user queries by by creating a *presention* of the requested information. That is, we want our system to string together a series of images and video clips that *tell a story*, one that appropriately informs the user.

To accomplish this we must use image similarity judgements together with analogical reasoning to decide what shots and stories best match the query. This can be accomplished interactively, using our browsing/database visualization tool called STRATAGRAPH. Alternatively, we can automatically select video clips using our analogical database called FRAMER, described in more detail below [2]. Finally, we can string video clips together into a full presentation by use of our story-telling interface called HOMER, creating a video presentation that answers the user's question [5, 4].

5.1 FRAMER

FRAMER is a knowledge representation system being used as a common database for a variety of projects around the Media Laboratory [2]. Developed to support work in content-aware media systems, FRAMER is being actively used in over a dozen projects around the Media Laboratory. FRAMER combines the persistent structure of a database, the inferential mechanisms of a knowledge representation, and the annotation facilities of a hypertext.

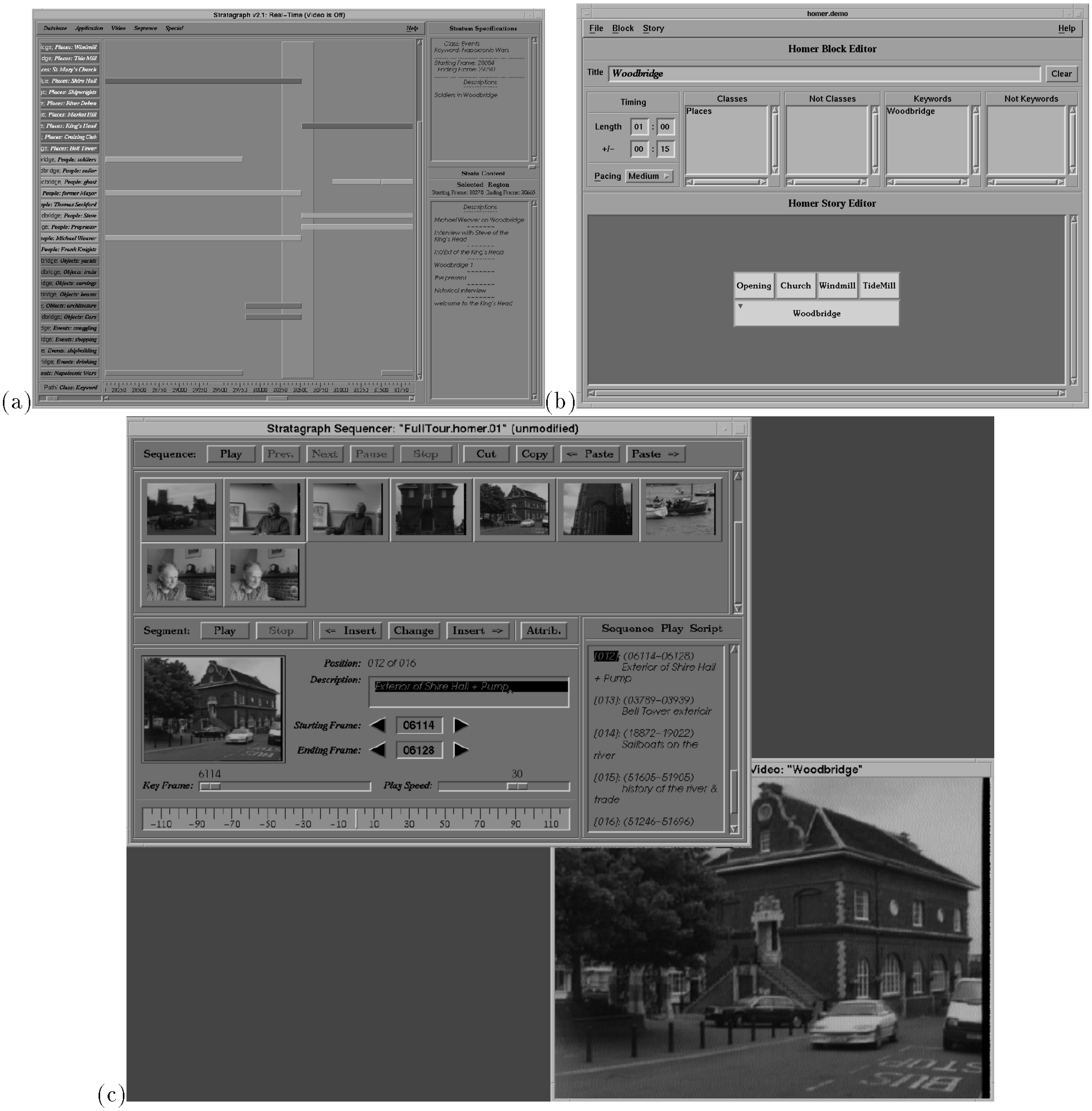
FRAMER supports the description of richly structured objects ("frames") by combining three kinds of relations: *annotation* relations connecting frames and their components; *prototype* relations for inheriting structural information between frames; and *ground* relations determining connections between frames in a structure and between frames and certain literal values (numbers, strings, etc).

Users of FRAMER can either access FRAMER structures directly (from LISP or C) or use FRAXL (FRAmer eXtension Language) to write programs that operate over and extend the FRAMER structure. FRAXL is a dialect of SCHEME extended with special facilities supporting search, inference, and indexing over FRAMER structures. FRAMER structures and FRAXL programs transfer across a variety of platforms: Unix based workstations, Apple computers, and DOS, Windows, and OS/2 based PCs.

5.2 STRATAGRAPH

STRATAGRAPH is both a representation for video and a tool for visualizing and browsing video data [3, 11]. The STRATAGRAPH system treats the video as an uninterrupted stream of frames, allowing descriptions to be attached to any group of contiguous frames. Such descriptions can be *layered* on the video stream, so that any group of frames may have a number of descriptions associated with it. This allows the video to be described at different granularities and in different contexts.

STRATAGRAPH is a tool that allows a user to familiarize themselves with a database of annotated video. As shown in Figure 7(a), the STRATAGRAPH provides a graphical representation of the annotations in the video database. Along the y-axis of the display is a list of all the unique descriptions in the database. Along the x-axis is a timeline depicting frame numbers in the video. In the main region of the display are several bars that represent actual descriptions attached to video. These descriptions are called Strata Lines or Stratum. Each Stratum has an in frame and



icge; Places: Windmill
dge; Places: Tide Mill
ces: St. Mary's Church
lge; Places: Shire Hall
gc; Places: Shipwrights
e; Places: River Deben
ie; Places: Market Hill
e; Places: King's Head
; Places: Cruising Club
ge; Places: Bell Tower
oridge; People: soldiers
dbr dge; People: sailor
icbridge; People: ghost
People: former Mayor
ople: Thomas Seckford
dbridge; People: Steve
ge; People: Proprietor
eople: Michael Weaver
People: Frank Knights
bridge; Objects: yachts
dbricge; Objects: train
idge; Objects: carvings
bridge; Objects: beams
e; Objects: architecture
dbridgc; Objects: Cars
dge; Events: smuggling
idge; Events: shopping
e; Events: shipbuilding
ridge; Events: drinking
ents: Napoleonic Wars

Class: Events
Keyword: Napoleonic Wars

Starting Frame: 28084
Ending Frame: 29790

Descriptions

Soldiers in Woodbridge

Starting Frame: 30270 Ending Frame: 30665

Descriptions

Michael Weaver on Woodbridge

Interview with Steve of the King's Head

Int/Ext of the King's Head

Woodbrdige 1

the present

historical interview

welcome to the King's Head

Title: Woodbridge

Length 01 : 00

+/– 00 : 15

Pacing Medium

Classes: Places

Keywords: Woodbridge

Position: 012 of 016

Description: Exterior of Shire Hall + Pump

Starting Frame: 06114

Ending Frame: 06128

Key Frame: 6114

Play Speed: 30

[012]: (06114–06128) Exterior of Shire Hall + Pump

[013]: (03789–03939) Bell Tower exterior

[014]: (18872–19022) Sailboats on the river

[015]: (51605–51905) history of the river & trade

[016]: (51246–51696)

Figure 7: (a) STRATAGRAPH, (b) HOMER, (c) the SEQUENCER interfaces

an out frame that relate to the video timeline. The in and out points determine the duration of the description, which is reflected in the length of the Stratum in the display.

5.3 HOMER

HOMER, created by Lee Morgenroth working with Glorianna Davenport, is the tool that allows the user to build stories from the database [5, 4]. Homer was designed as a graphical workspace in which editors could build structures that are accurate models of the stories they wish to tell in video.

Figure 7(b) shows the Homer interface and a story model. Stories are built in Homer using abstract story chunks, called Blocks. Each Block has a size, which is proportional to the length of story time that the Block covers. Block sizes can range from one second to several hours. Each Block also has a number of descriptions that determine story content. The maker can design a story by creating a progression of Blocks. Blocks can also be layered to create sequence structures.

Each Block specifies a set of constraints on the type and order of material to be presented. These constraints can be applied to traditional text annotations, or to the iconic descriptions produced by semantics-preserving compression. The structure of the story model and the constraints specified by each Block serve as a semantic filter that allows semantically-appropriate clips to be retrieved in a sequence that tells a coherent story.

Although the Block model gives an accurate description of the story to be told, what is generated by Homer would be considered a "rough cut" by a professional video producer. The rendered edit has most of the footage necessary to tell the story, but there are still some poor cuts between clips and some of the clip choices may not be ideal. Thus we provide an interface call the SEQUENCER, shown in Figure 7(c), to fine tune the rough edit. Using the SEQUENCER, shots can be reordered or replaced, and cuts can be trimmed to provide better transitions.

6 Conclusion

We have described a prototype system that is built on the idea of parsing video into semantically-meaningful chunks, and then encoding those chunks into a compact, easily-searchable representation that preserves the visual similarity relations. This *semantics-preserving compression* process can then be augmented with textual and analogical annotations. The result is a representation of the visual material that can be used to automatically assemble and efficiently edit multimedia presentations in response to user's needs.

Acknowledgment: The work described here was funded by BT (British Telecom). We also wish to thank Fang Liu for her work on the Wold implementation, Baback Moghaddam for his work on the eigenface implementation, Stan Sclaroff for his work in shape matching, Tom Minka for his work on the PHOTOBOOK user interface, Lee Morgenroth for his work on the HOMER interface, and Marc Davis for his work on the Media Streams interface.

References

[1] Davis, Marc. "Media Streams: An Iconic Visual Language for Video Annotation." *Telektronikk* 4.93 (1993): 59-71. (Available on the WorldWideWeb at: http://www.nta.no/telektronikk/4.93.dir/Davis_M.html)

[2] Haase, K. (1993) “Framer: A Persistent Portable Representation Library.” *Proceedings of the American Asso. for AI (AAAI-93).*

[3] MacKay, W. and Davenport, G. (1989) “Virtual Video Editing in Interactive Multimedia Applications.” *Communication of the ACM* 32 (7 1989): 802-810.

[4] Morgenroth, L. (1992) “Homer: A Story Model Generator.” B.S. Thesis, MIT, 1992.

[5] Morgenroth, L., Davenport, G. (1994) “Let’s See That Again: A Multiuse Video Database Project.” *ACM Multimedia 1994*, San Francisco, CA

[6] Pentland, A., Picard. R., Sclaroff, S., (1994). “Photobook: Tools for Content-Based Manipulation of Image Databases.” *SPIE Conf. Storage and Retrieval of Image and Video Databases II*, No. 2185 . San Jose, CA:

[7] Picard, R., (1992) “Random Field Texture Coding,” *Society for Information Display International Symposium Digest*, Vol XXIII, May 1992, pages 685–688.

[8] Picard, R., and Liu, F., (1994) “A new Wold ordering for image similarity,” *International Conference on Acoustic Signals and Signal Processing* March 1994, Adalaide, Austrailia. vol. 5, page 129.

[9] Picard, R., and Minka, T., (1994), “Vision Texture for Annotation” *ACM/Springer-Verlag Journal of Multimedia Systems, in press.*

[10] Sclaroff, S., and Pentland, A., (1994) “Modal Matching,” *IEEE Trans. Pattern Analysis and Machine Vision, in press*

[11] Smith, A., Thomas, G. and Davenport, G. (1994). “The Stratification System: A Design Environment for Random Access Video.” *ACM Workshop on Networking and Operating System Support for Digital Audio and Video* San Diego, CA.

[12] Turk, M., and Pentland, A., (1991) “ Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86.