Hyperlinked television research at the MIT Media Laboratory

by V. M. Bove, Jr.
J. Dakss
E. Chalom
S. Agamanolis

In hyperlinked video, objects are selectable—resulting in an associated action—akin to linked words or graphics on Web pages. Possible venues for hyperlinked video include broadcast television, streaming video (e.g., video on the Internet or other forms of video-on-demand systems), and published media such as DVD (digital versatile disc). Hyperlinked video offers new interaction possibilities with streaming media and the opportunity to move electronic commerce from the personal computer to the television receiver while making it more dynamic and engaging. In this paper we examine some opportunities and challenges of hyperlinked video, describe an object tracking and identification algorithm and an authoring tool we have developed, and discuss two prototype hyperlinked television programs—one an augmented broadcast and one a video-on-demand application.

Extending the concept of the hyperlink from regions of text on a page to defined regions of still images is a relatively small conceptual jump, and if video is considered as a sequence of still images, it would seem that hyperlinking of video is conceptually similar. But the temporal nature of video poses technical and design problems: first, the fact that there are at least 25 or 30 frames per second in which the link regions must be defined requires some degree of automation in the authoring process; second, the connection between the concept of a hyperlink and the navigation issues common to all forms of nonlinear media must be considered.

An early example of the use of hyperlinks in video was the 1978 *Aspen Movie Map*,[1] in which nonlinear playback from analog videodiscs combined with computer-generated overlays permitted a user to take an arbitrary virtual drive through Aspen, Colorado, and to find additional information about buildings and locations. Two main hyperlink opportunities were presented. The simpler of these involved icons that allowed such actions as turning at an intersection; the more complex enabled touching buildings and accessing information about them. The latter spatial link locations were calculated based on a model of the town generated initially from aerial photography. Since every building was linked, there was no need to provide indications of the link opportunities, a design consideration we will revisit in a later section of this paper.

The 1989 *Elastic Charles* system[2] indicated link opportunities through "Micons," small looping video clips that appear as icons atop relevant objects on a still or video screen. The location and duration of the Micons had to be specified manually by the author. *HyperPlant*,[3] like the *Aspen Movie Map*, relied upon having a known mapping between the video and a pre-existing model. This system allowed factory machinery to be remotely monitored and controlled and was able to overlay controls and information atop live video captured by a camera whose panning and zooming were controlled by the user.

0018-8670/00/$5.00 © 2000 IBM

Requirements

Until recently, the biggest barrier to widespread application of hyperlinked video was a conceptual one on the part of the user: why would viewers think about "clicking" on linked items? But the Web has accustomed many people to the point-and-click scenario, so at this time the major unresolved issues are: standardization, authoring tools, and business models.

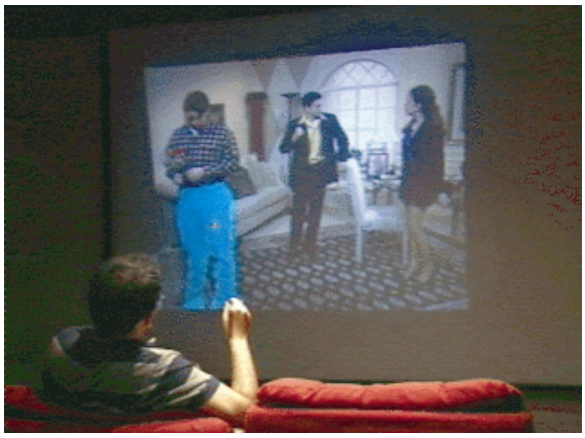
Technical issues relating to standardization are outside the scope of this paper, but a likely scenario is use of the standard set by the Moving Picture Experts Group, MPEG-2 video (possibly MPEG-4), with related interaction software written in the Java** language and a back channel employing the HyperText Transfer Protocol (HTTP). In North America, subcommittees of the ATSC (Advanced Television Systems Committee) are developing technical standards relevant to the delivery of hyperlinked television content, in particular T3/S16 (Interactive Services), T3/S17 (DASE-Digital Television Applications Software Environment), and the Ad Hoc Data Implementation Working Group (DIWG) of the Implementation Subcommittee.[4] Another relevant group is the Advanced TV Enhancement Forum (ATVEF), which is defining protocols for the use of HyperText Markup Language (HTML) in television applications,[5] as is the World Wide Web Consortium (W3C).[6] The Internet Engineering Task Force[7] is developing protocols for video and audio transport over Internet links that may also be relevant to these applications.

Viewing and interacting with hyperlinked video (Figure 1) requires several elements:

- A digital video decoder
- A decoder or processor for a secondary data stream ("metadata"). In general this would be a microcontroller associated with the digital video decoder.
- Graphical overlay ability (often included in digital video decoder chips)
- A pointing device (e.g., a mouse, or a remote control with a touchpad, joystick, light pen, inertial sensor, special buttons)
- For some applications, a back channel

Current personal computers and digital television set top boxes (STBs) or receivers, as well as DVD (digital versatile disc) players, and some game devices, already have sufficient hardware processing power to provide a playback platform, given appropriate programming.

Figure 1 Hyperlinked television allows "point-and-click" interactions with moving video, given a remote control with a pointing device



System architectures

System architectures for hyperlinked television can be divided into five types. We discuss examples of each of the following:

- Stand-alone, local video storage
- Stand-alone, local video storage with back channel
- Broadcast video
- Broadcast video with back channel
- Streaming video-on-demand with back channel

The simplest system, and the easiest to implement at the current time, is a stand-alone system in which video is stored locally. Examples of this type of system might include a touch-screen kiosk in a store that directs shoppers to particular items of merchandise featured in a video, or an interactive training DVD viewed on a personal computer.

The addition of a low-bandwidth back channel such as a telephone modem to the above system allows the system to pull in externally stored information for use in augmenting the local video material or to send the results of the interactions to a remote location. An example might be a clothing catalog stored on DVD, where the ordering could be done through the Internet and the presentation could be altered to include updates, special offers, and the like.

If the video content (and the hyperlink information) are not stored locally, then two options exist: broadcast or video-on-demand. In a broadcast system without a back channel, the hyperlinks are useful for clickable augmentations—such as player identification and statistics in a sporting event or a clickable weather map—or perhaps for switching to a different program stream in a multiplexed broadcast.

Adding a back channel to a broadcast setup allows analogous applications to those discussed above for the case of local storage with back channel. We will look at *HyperSoap*, an example of such a system, in a later section.

Given sufficient bandwidth to the home, and video file servers able to handle all the simultaneous requests (or caching appropriately distributed throughout the network), one could build a hyperlinked video-on-demand system. Such a system, perhaps not yet economically feasible for wide deployment, enables dynamic, personalized assembly of content from almost endless source libraries. Our *Interactive Dinner at Julia's* demonstration (described later) provides a small glimpse of the possibilities of such a system.

Business opportunities

One of the most interesting facets of hyperlinked television is the variety of commercial opportunities presented, some obvious or analogous to other commercial situations, and others unique. A single company could provide the entire system, could provide whatever portions were not already possessed by a client (for instance, a mail-order company already has a fulfillment service), or could specialize in just one area like placement consolidation. We suggest the following segments as relevant business opportunities:

- Authoring service
- Placement sales
- Transaction fees
- Credit authorization or identity verification
- Fulfillment
- Back-channel provision
- Server operation for video-on-demand
- Data mining

Authoring problems

As noted above, a significant concern in hyperlinked video has been the difficulty of authoring the material in a labor-efficient way. For still images containing hyperlinks, Web authors have been satisfied with outlining "hot" regions manually, but for television it is necessary to generate 30 (or in Europe 25) "pixmaps" a second, and some degree of automation is essential.

Several companies have announced products for authoring hyperlinked video, most notably *VisualSHOCK MOVIE* from the New Business Development Group of Mitsubishi Electric America, Inc., *HotVideo* (now *HotMedia**) from IBM's Internet Media Group, and Veon's *V-Active*.[8–10] Aimed at inserting one or at most a small number of links into an image, these tools typically require a user to outline a clickable region in key frames with a bounding box or other geometrical shape, and then the tool automatically tracks the region through intermediate frames.

Advantages of using a simple geometrical description of the clickable "hot spots" include the relatively small overhead incurred in transmitting them and the ability to describe them using obvious extensions to standard HTML practice. Such a scheme appears to us to be more than sufficient for simple scenes or for low-resolution Web video content, but in situations where large numbers of objects overlap or interact in complex ways, or do both, the tracking approach used by the previously mentioned tools can be too limiting. A system that would automatically and dynamically fit the approximate contours of moving objects would enable the use of more visually complex scenes and as a bonus would generate segmentation masks of high-enough quality that they might be applied in graphical overlays to indicate selectable regions.

We have developed a novel segmentation system that can classify every pixel in every frame of a video sequence as a member of an object or group of objects.[11] Unlike image segmentation methods that rely on one or two features such as color or motion, our system works with multiple features. The use of multiple features provides additional constraints that can help in segmenting material for which just one feature would produce an ambiguous result, and can also compensate for the inaccuracies and failure modes that feature estimation algorithms occasionally exhibit. In the most general form, our algorithm is a multidimensional, multimodal Gaussian classifier that can operate on any numerical feature vectors that can be computed for a pixel or a neighborhood of pixels. For hyperlinked video authoring, we

use features such as color, motion, texture, and position. A discussion of each of these features follows.

Video typically will be provided in a standard set of three color components such as RGB (red, green, blue) or YUV (a color encoding scheme in which the luminance [Y] and the chrominance [UV] color components are separate). The system can work with the input format directly, or can convert from one color space to another, turning, for example, RGB into YUV or YIQ (color components in the color space of the National Television System Committee standards) via vector-matrix multiplication or any of the representations into LAB (a device-independent color space defined by the Commission Internationale de l'Eclairage [CIE]) by an intermediate transformation through the CIE XYZ color space.[12] In tests against a "ground truth" of pure manual segmentation of a variety of video sequences, we have found that luminance-chrominance spaces provide more reliable segmentations than RGB, but one pair of chrominance components does not seem to have a significant advantage over another.[13]

Estimating motion for every pixel in the intensity component of an image provides a dense set of two-vectors. We have used several different algorithms to generate these vectors: overlapping block-matching and a variety of brightness-constraint optical flow algorithms, the most reliable generally being a least-squares polynomial fit algorithm developed by Krause and Martinez.[14]

Numerous methods exist for modeling texture in images. What is important from our perspective is that we have available one or more numerical parameters describing the texture in the neighborhood of each pixel in the intensity image. In our experiments we have used simple local estimates of mean and variance (the latter perhaps at multiple scales), oriented filters at various scales,[15] and a simultaneous autoregressive (SAR) model that provides the parameters for a linear predictor that estimates a pixel based on its neighbors.[16] The oriented filters method was later discarded because of the high computational cost and the large number of informational dimensions it generated, which complicated the segmentation process. In tests, the two other methods performed about equally well.[13]

A final "feature" that we have found improves the performance enough to merit inclusion is the (x, y) positions of the pixels. The position information allows better discrimination in cases where multiple similar-appearing objects are visible simultaneously.

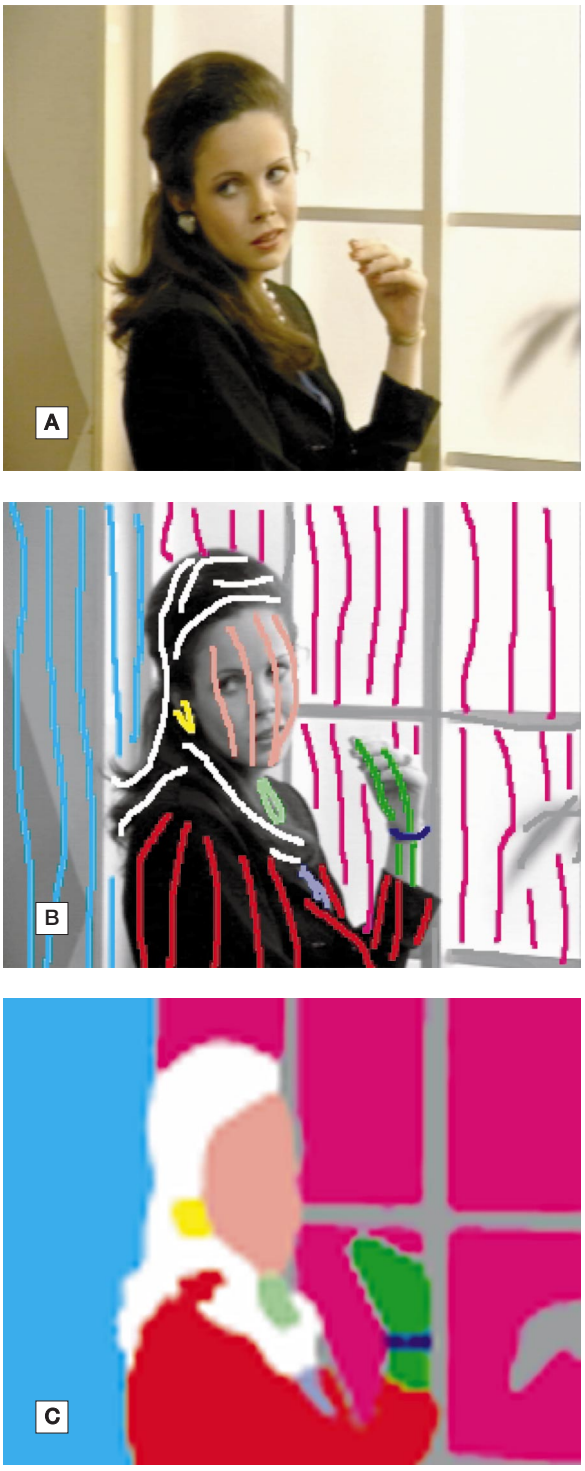
From the perspective of the annotator of the video, the operation of the authoring system is straightforward. Our tool uses standard scene-change detection to break up the video into shots. In each shot, the author chooses one or more frames on which he or she defines what the desired objects are by providing the algorithm with "training data" in the form of rough scribbles on each (Figure 2). The system creates a parametric (multimodal Gaussian) model for the probability density function (PDF) for each feature in each region. Assuming in turn every number of possible Gaussian modes from one to some maximum, we use an Expectation Maximization (EM) algorithm to estimate "optimal" parameters (mean, variance, weight) for each Gaussian mode in each mixture. Parameter estimation is improved by the use of deterministic annealing, which prevents settling into solutions that represent local minima. Clearly, more modes will always better fit the training points than would fewer, so we need to choose the number of modes beyond which diminishing returns begin to set in; the method of cross validation is used as part of this process. The overall segmentation system is described in detail by Chalom.[13]

For hyperlinked video authoring, we use features such as color, motion, texture, and position.

While some features such as color may be expected to be relatively unchanged through time, the estimate for certain parameters (such as motion and position) is recomputed for other frames in the shot by tracking the training points forward and backward and recalculating the models.

The final step involves classifying all the unlabeled pixels in the training frame and in other frames of the shot by performing what is often called "hypothesis testing" between the feature parameters of the unclassified pixels and the statistical models for the regions. At the end, we have what amounts to an object map for each frame and a table or database associating each object region with some action to be

Figure 2 (A) Key frame from a video shot; (B) training data for segmentation and tracking tool; (C) region segmentation mask for this frame



performed when selected by the viewer. The classification process also generates a certainty measurement associated with the classification of each pixel, which can be used as part of an automatic clean-up step if desired (i.e., pixels with low certainty are candidates for reclassification based on regional morphological operators or higher-level processing).

The above-described system functions well enough that we were able to use it with no manual reworking of object maps to produce the *HyperSoap* video described in a later section. However, we quickly learned that in video with a large number of linked objects and a large number of shots, the author spends a great deal of time manually making associations between an object appearance in one shot and the same object in another shot. Since the same objects will appear in more than one shot in most video sequences, it would be beneficial to an author for the tool to compare each object in a newly processed sequence of images with entries in a catalog of previously identified objects, associating the description and link action automatically if a match is found. This potential benefit was the underlying motivation for creating a "properties management" component of a hyperlinked video authoring tool. Our HyperActive system attempts to equate the objects in a newly segmented frame sequence with objects from previous shots. If it guesses the objects correctly, the author merely needs to confirm the associations.

The properties manager retains the multimodal Gaussian probability density functions for color and perhaps texture distribution of each object generated by the authoring process. All of the PDFs computed for an object in each shot are kept, and a method described by Vasconcelos and Lippman[17] is used to create a composite PDF from these PDFs to use in searches. This method applies an Expectation Maximization algorithm for both matching a query PDF to a set of PDFs and creating a PDF "hierarchy" for effectively representing multiple views of the same object in an object database. When an object in a later shot is tagged by the author, the properties manager provides a ranked list of the most likely matches among already-known objects from previous shots, and the author selects among these. If no match is found or if the author indicates that the object in question is new, the system creates a new entry for the object. If the author confirms the guess made by HyperActive, the PDFs from the new appearance of the object are combined with the already-known PDF

hierarchy for the object—the tool "learns" about different ways in which the object can appear.

Since the purpose of the automatic object identification is to increase authoring speed, we decided to implement a "presearch" step to reduce the number of PDF comparisons needed. One way to help limit the time spent during a search is to eliminate objects whose color information differs by a significant amount. To accomplish this elimination, HyperActive creates a separate object database consisting of three sets of "bins," one set for each color channel. Each bin represents a single integer value of a color channel. When an object is identified for the first time, a number is computed by summing the product of the mean and weight of each Gaussian for the PDF of each channel. This number is an index for the bin of the channel in which a marker for that object will be stored. Thus, a marker is stored in three sets of bins.

When a search is conducted for a match for a particular object, the first step consists of computing three of these weighted-mean values for that object. In the next step, HyperActive uses these three numbers to create a list of potential matches by looking into bins that are within a certain range around those three values (a range of eight increments for luminance and four increments for each chrominance channel). It is these values that will ultimately be used in the PDF-based search. The present system does not eliminate objects that did not have markers in all three color channel bins within the range of a given search object; thus this step may generate false positives but is less likely to reject potential matches.

This additional presearch step has the benefit of not only reducing the search space, but also expediting cases where a new object looks vastly different from all previously identified objects. If no objects have markers in the set of bins of each color channel within a search range around the weighted-mean values of a new object, then it is quite likely that the object is new to the database, and therefore the PDF comparison step is not necessary. Similarly, if there are only a very small number of objects whose weighted-mean values are within range of the values of a new object, these objects can be displayed to the user as potential candidates without having to perform a PDF comparison. In trials conducted to gauge the accuracy of the properties manager, the presearch step commonly eliminated more than half of the objects in the database before applying the more resource-intensive EM query.[18]

We further improve the performance of this method by generating an "object occurrence matrix" that keeps track of which objects appear in frames with other objects, and in how many frames each object appears. After the first two shots of a program have been segmented, this information is used to help choose among candidates with close likelihoods so as to favor objects that have appeared in conjunction with the already-classified objects. Testing showed an accuracy (object correctly identified automatically) of 85 percent, up from 67 percent when the object occurrence matrix was not used.[18]

Production design problems

Besides the mechanical issues of generating a region database, associating actions with objects, and so forth, there remain a set of visual design issues associated with hyperlinked video. Although the evolution of the Web has provided us with the beginnings of a visual language for hyperlinked media, the extrapolation of these norms to dynamic pictorial content is not yet settled. Some remaining questions include:

- How do we indicate links unambiguously but also unobtrusively?
- How do we indicate that a link has been selected, if the selection might not have immediate consequences (e.g., a video might continue playing until the end of a scene)?
- If the program thread must be interrupted, when is it appropriate to do so?
- How do we indicate the types of material (text, video) to which a link might lead and whether or not an interruption of the main program thread will result?
- In video with branch points, how do we indicate the viewer's location and history?

Typical existing hyperlinked video playback systems indicate the presence of hyperlinks by changing the shape of the cursor when it is positioned over a hyperlinked object. When this happens, a name or description for the link may be displayed in an area of the computer screen. For example, IBM's *HotVideo* software changes the cursor to an icon that corresponds to the media type of the information contained in the link; the playback software can also display a wireframe shape around the object and change the brightness or tint of the image region inside the wireframe. We used a method similar to the last of these in our prototypes below. It is also possible to implement a button (e.g., on a remote control) that

Figure 3 *HyperSoap* allows viewers to select clothing, accessories, and furnishings from a soap opera episode

Figure 4 A frame from *Interactive Dinner at Julia's* showing "back" button in upper left; viewer here is one link away from the main program. Note that items may link to text overlays as well as video clips.

Image copyright WGBH, used with permission.

illuminates all the possible hyperlinks simultaneously.

Two example programs

In order to exercise our authoring software and to explore production and post-production methods for hyperlinked television, we created two prototype programs. The first, *HyperSoap*,[19] is an example of an augmented broadcast with a low-bandwidth back channel. The second, *Interactive Dinner at Julia's*, is a nonlinear presentation such as might be enabled by a video-on-demand system or a DVD-ROM. Because we have a very efficient implementation of our Isis scripting language under the UNIX** system for the Alpha processor,[20] we developed the playback application for that platform, but the same feature set should be implementable with relative ease using the Java language or as an application for various set-top-box operating systems.

HyperSoap is a hyperlinked video drama closely patterned after daytime television soap operas (Figure 3), produced with assistance from retailer J. C. Penney Company. The viewer knows that everything in the scene is for sale, including clothes, props, and furnishings. The pointing device highlights selectable objects (we generate the highlight from the actual segmentation mask), and keeping the pointer fixed on one object for a short period selects it (indicated by a change in color of the highlight). In an augmented-broadcast scenario, selecting an item provides an information box on the screen. The playback system remembers all objects that have been selected, and at the end of the program can provide more interactions based on the selections (such as using the back channel to take the viewer to an on-line shopping page). If local storage such as a disk recorder is available, the program can be interrupted, and purchasing or other interactions can be inserted into the program while the incoming stream is being recorded for playback after the break. In this case we found it most effective for the system not to interrupt until an author-specified appropriate point, typically when an actor has finished speaking a set of lines.

In the absence of viewer interactions, *HyperSoap* proceeds as a seemingly normal soap opera episode, but the idea that everything would be for sale and that viewers might be interacting was part of the design of the program from the earliest stages of planning. For example, shots were designed to emphasize the products for sale while not diverting too much attention from the flow of the story. Certain actions are written into the script in order to provide natural opportunities for close-ups of products that would have otherwise been difficult to see or select with our interaction device. Similarly, the scene is edited so that shots are either long enough or returned to often enough for viewers to spot products and decide to select them. The story itself involves the products in ways that might increase their value

to the consumer. Integrating the story with the products makes the interruptions to show product information less jarring overall. Knowing that everything is for sale and that the program was designed for the dual purpose of entertainment and shopping motivates the viewer to "stay tuned" and to interact with the products.

HyperSoap was basically a linear presentation with augmentations. To prototype a nonlinear hyperlinked video application such as might be available from a video-on-demand system or a DVD, we created a "web" of video clips with the help of television station WGBH and Julia Child. In *Interactive Dinner at Julia's*, Julia Child introduces the viewer to a dinner party at which a succession of courses is served (Figure 4).[18] The viewer simply interested in menu ideas can watch the program with no interactions, but selecting food, drink, or accessory items leads not only to explanatory text overlays (as in *HyperSoap*) but also to illustrated preparation instructions. These may in turn contain further links. For example, selecting the appetizer leads to a lesson in preparing it; as a sharp chef's knife is necessary, the viewer can go from that point to a video clip on knife sharpening. At the end of a video clip, the viewer is returned not to the exact jumping-off point but to a slightly earlier shot to re-establish context, since the viewer may have been to a variety of places in the meantime. In accordance with the "web" idea, we also created a "back button" that appears on the screen when a video link is pursued, showing a small image of the scene to which the viewer will return at the end of the clip; the viewer can return at any time by selecting the button. Additional stacked back buttons appear when following nested video links, so that the viewer can immediately return to any of the previous clips.

Although the links in the cooking application are instructional rather than immediately commerce-related as in the soap opera, many commercial opportunities still exist. For example, video or text links could be sponsored, and the ingredient list for a selected recipe could automatically be added to a viewer's shopping list for on-line grocery shopping.

Conclusion

We have developed a practical authoring tool for tracking objects and embedding hyperlinks in video. Several developments bode well for the success of hyperlinked television. First, of course, is the ongoing interest in new forms of e-commerce, which may prove to be a driver for development. Second is the deployment of digital television delivery channels—wired, terrestrial broadcast, satellite, and recorded—into homes, bringing with them the needed electronics. Finally comes the human element; millions of ordinary people every year learn what a "hyperlink" is by means of the Web, and many more are already accustomed to pushing buttons associated with a TV receiver, whether from channel-surfing or video games.

Acknowledgments

The authors wish to thank Nuno Vasconcelos for explaining his algorithms, Michael Ponder at J. C. Penney Company, Inc., for his assistance with *HyperSoap*, and WGBH-TV, Annie Valva, David Scutvick, William Truslow, Jeff Garmel, and Julia Child for their support of *Interactive Dinner at Julia's*.

This work has been supported by the Digital Life consortium and the Broadercasting special interest group at the MIT Media Laboratory.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., or The Open Group.

Cited references

1. W. Bender, *Animation Via Video Disk*, S.M. thesis, MIT, Visual Studies, Cambridge, MA (1980).
2. H. P. Brondmo and G. Davenport, "Creating and Viewing the Elastic Charles—A Hypermedia Journal," *Hypertext: State of the Art*, R. Aleese and C. Green, Editors, Intellect, Oxford, U.K. (1991), pp. 43–51.
3. M. Tani et al., "Object-Oriented Video: Interaction with Real-World Objects Through Live Video," *Proceedings of CHI* (1992), pp. 593–598.
4. Advanced Television Systems Committee, http://www.atsc.org.
5. Advanced Television Enhancement Forum, http://www.atvef.com.
6. World Wide Web Consortium, http://www.w3.org.
7. The Internet Engineering Task Force, http://www.ietf.org.
8. VisualSHOCK MOVIE, Mitsubishi Electric, http://www.visualshock.com.
9. HotMedia, IBM Corporation, http://www.software.ibm.com/net.media.
10. V-Active, formerly from Ephyx Technologies, now from Veon Inc., http://www.veon.com.
11. E. Chalom and V. M. Bove, Jr., "Segmentation of an Image Sequence Using Multi-Dimensional Image Attributes," *Proceedings of IEEE ICIP-96* (1996), pp. II-525–II-528.
12. R. Jackson, L. W. MacDonald, and K. Freeman, *Computer Generated Color: A Practical Guide to Presentation and Display*, John Wiley & Sons, New York (1992).

13. E. Chalom, *Statistical Image Segmentation Using Multidimensional Attributes*, Ph.D. thesis, MIT, Electrical Engineering and Computer Science, Cambridge, MA (January 1998).
14. E. A. Krause, *Motion Estimation for Frame-Rate Conversion*, Ph.D. thesis, MIT, Electrical Engineering and Computer Science, Cambridge, MA (June 1987).
15. J. R. Bergen and M. S. Landy, "Computational Modeling of Visual Texture Segregation," *Computational Models of Visual Processing*, M. S. Landy and J. A. Movshon, Editors, MIT Press, Cambridge, MA (1994).
16. A. K. Jain and J. Mao, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognition* **25**, No. 2, 173–188 (February 1992).
17. N. Vasconcelos and A. Lippman, "Learning Mixture Hierarchies," *Proceedings of NIPS '98* (1998), pp. 606–612.
18. J. Dakss, *HyperActive: An Automated Tool for Creating Hyperlinked Video*, S.M. thesis, MIT, Media Arts and Sciences, Cambridge, MA (September 1999).
19. V. M. Bove, Jr., J. Dakss, S. Agamanolis, and E. Chalom, "Adding Hyperlinks to Digital Television," *SMPTE Journal* **108**, No. 11, 795–801 (November 1999).
20. S. Agamanolis and V. M. Bove, Jr., "Multilevel Scripting for Responsive Multimedia," *IEEE Multimedia* **4**, No. 4, 40–50 (October–December 1997).

Accepted for publication May 15, 2000.

V. Michael Bove, Jr. *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: vmb@media.mit.edu).* Dr. Bove holds an S.B.E.E., an S.M. in visual studies, and a Ph.D. in media technology, all from the Massachusetts Institute of Technology, where he is currently head of the Media Laboratory's Object-Based Media Group. He is a cofounder of, and technical advisor to, WatchPoint Media, Inc.

Jonathan Dakss *WatchPoint Media, Inc., 129 South Street, Boston, Massachusetts 02111 (electronic mail: dakss@watchpointmedia.com).* Mr. Dakss received a B.A. in computer science from Columbia University in 1997. He worked at Columbia's Center for Telecommunications Research and aided in the development of algorithms for a wavelet-based image and video compression and segmentation system now in use by the United States Air Force, Rome Laboratories. He received a master's degree in media technology at the MIT Media Laboratory in 1999. He cofounded WatchPoint Media, Inc. in 1999.

Edmond Chalom *Sarnoff Corporation, 201 Washington Road, Princeton, New Jersey 08540 (electronic mail: echalom@sarnoff.com).* Dr. Chalom received his B.S., M.S., E.E., and Ph.D. degrees all from the Massachusetts Institute of Technology. His Ph.D. research was performed at the MIT Media Laboratory. He is currently a member of the technical staff at Sarnoff Corporation in the Multimedia Technology Laboratory, specializing in image compression, image analysis, and pattern recognition. He is a member of Eta Kappa Nu since 1987.

Stefan Agamanolis *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: stefan@media.mit.edu).* Mr. Agamanolis is a Ph.D. degree candidate at the MIT Media Laboratory and a research assistant in the Object-Based Media Group. He has worked on a wide variety of research projects involving telepresence and telecollaboration, hyperlinked video, interactive storytelling, responsive environments, ambient media, and intelligent authoring tools for multimedia applications. His experimental Isis programming language serves as the foundation for the hyperlinked video prototypes described in this paper. Before coming to MIT, Mr. Agamanolis studied computer science, philosophy, and film at Oberlin College. Additional information is available at http://www.media.mit.edu/~stefan.