Printed Embedded Data Graphical User Interfaces



Printed embedded data graphical user interfaces generalize the interaction domain to images and objects to enable interaction with passive objects and active displays throughout user environments.

David L. Hecht Xerox Palo Alto Research Center raphical user interfaces (GUIs)¹ have become the dominant computer user interface paradigm during the past three decades. A typical system includes a computer with a bitmap display and a mouse or stylus for pointing and acting. Interaction is based upon selection and action at positions in the graphical display, guided by graphical content and the user's intent. For example, the user can select a document icon on a PC screen desktop and drag it to a printer icon or choose a highlighted text element and open a hyperlink.

Printed embedded data graphical user interfaces (PEDGUIs) generalize the interaction domain to images and objects throughout user environments such as printed documents (forms, memos, books, catalogs, and so on), projections on a wall, and active computer displays. The interface can be almost any object. The selection and control tool is an image capture device such as a camera with control buttons. The embedded data helps establish the selection location and gesture, as well as the identity or context of the chosen substrate and links to other resources. Even in the case of multiple active and passive display substrates, the GUI system captures and interprets user interactions throughout the workspace.

EMBEDDED DATA TECHNOLOGIES

A key element for PEDGUI systems is imprinting appropriate machine-readable encoding and electronic capture of images containing the embedded data for machine interpretation of user interaction. The encoding provides a variety of information: object identification, context, action instruction, digital pointers, network paths, position and orientation in real and virtual spaces, and content information generally characterized as portable data files. Data marking should be well integrated graphically for functionality.

DataGlyph technology^{2,3} (http://www.dataglyphs. com), based on self-clocking glyph code, is a particularly effective and flexible encoding technology for embedded data GUIs. The "Embedded Data Technologies" sidebar describes emerging developments using other technologies.

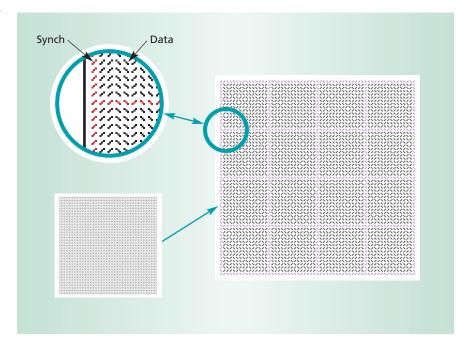
DATAGLYPH TECHNOLOGY

Our work at Xerox has focused on developing embedded data technology that provides substantial data capacity and local and global spatial address capability, with good aesthetics and graphical integration with line art, text, and pictorial documents in color or black and white.

What's in a DataGlyph?

Figure 1 illustrates a large-scale bitmap portion of glyph code. This self-clocking code consists of two types of glyphs representing digital ones and zeros. The term *self-clocking* denotes a mark encoded to represent data at each position: The presence of a mark is a clocking mechanism, where one or more of the mark's features—such as orientation—encodes data. In general, each glyph mark can have 2^m distinguishable states, encoding *m* bits.

The bitmap pattern for each mark of one bit consists of three black pixels along a 45-degree line, tilted to the left to represent a one and tilted to the right to represent a zero. The glyphs are arranged in a pattern; Figure 1. Self-clocking glyph code bitmap. The bitmap pattern for each mark consists of three black pixels along a 45-degree line, tilted to the left to represent a data value one and tilted to the right to represent a data value zero. The rectangular frames contain glyph code for synchronization and other functions. DataGlyph pattern by Glen Petrie and Jeff Breidenbach.



Embedded Data Technologies

Although this article focuses on Data-Glyph technology, other relevant developments and applications are available for recording and retrieving embedded data.

Consumers have become very familiar with data marking technologies for recording and retrieving data from print, such as conventional one- and two-dimensional barcodes and matrix or "checkerboard" codes.¹ We can use these technologies in GUI applications where their aesthetic qualities are acceptable, but they have not been optimized for graphical aesthetics or imperceptibility; moreover, their user interaction mode is to capture the code rather than interact with a graphic object. Steganographic encoding is also imperceptible so that interaction with the object data becomes implicit. In general, these technologies were primarily designed to record and retrieve discrete messages, not to retrieve addresses at selected positions in the image.

The explosive growth of the World Wide Web has provided an infrastructure for linking and manipulating printed and electronic documents via embedded data, and major corporations are looking for ways to capitalize on this new opportunity. AirClic, Motorola, Connect Things, and barcode giant Symbol Technologies have partnered in a half-billion-dollar venture to introduce scanning technology into wireless devices. Digital Convergence (http://www.digitalconvergence.com) is investing heavily in conventional barcode technology as a means of linking advertisements and marked products to vendor Web pages. Digimarc (http://www.digimarc.com/index.shtml) has developed imperceptible embedded data for robust pictorial watermarking (ownership/management) and is using this technology for magazine ad linking via camera. Anoto (http://www.anoto.com) has developed an address-code embedded data implementation using invisible ink and a compact wireless camera pen-processor targeted at capture and transmission of handwriting for various applications.

Jun Rekimoto's research at the Sony Computer Science Laboratory focuses on embedded data marking for identification and registration in three-dimensional spaces. The applications for augmented and augmentable reality include:^{2,3}

 InfoPoint, a small handheld device that acts as a "universal commander" for various kinds of information appliances including VCR decks, TVs, and computers, as well as physical objects such as printed documents and paper cards. InfoPoint recognizes affixed IDs (visual markers) and alters its functionality according to the object in front of it. InfoPoint also extends the concept of direct manipulations such as "pointand-click" or "drag-and-drop" into the physical space to support information exchange among these objects.

- *NaviCam*, a video device that detects color-coded IDs in real-world environments and displays situation-sensitive information on its video screen.
- *Pick-and-drop*, a direct-manipulation technique for multiple-computer environments that extends the drag-and-drop concept. For example, a user can select or create text on a PDA and pick-and-drop it at the desired location on a whiteboard. This technique transfers data through the network and allows a user to pick up digital data as if it were a physical object.

References

- T. Pavlidis, J. Swartz, and Y.P. Wang, "Information Encoding with Two-Dimensional Barcodes," *Computer*, June 1992, pp. 18-28.
- J. Rekimoto and K. Nagao, "The World through the Computer: Computer Augmented Interaction with Real-World Environments," *Proc. Symp. User Interface Software and Technology* (UIST 95), ACM Press, New York, 1995, pp. 29-36.
- 3. J. Rekimoto, "NaviCam: A Magnifying Glass Approach to Augmented Reality Systems," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, 1997, pp. 399-412.

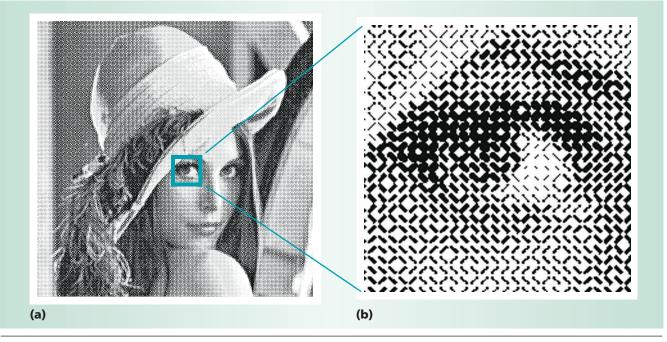


Figure 2. Pictorial halftone using elliptical dots on a rectangular lattice. (a) At an appropriate viewing distance, the visual effect is pictorial, and the individual cells are imperceptible. (b) A blowup detail of the cells. Glyphtone by Noah Flores. Figure 2a has some artifacts due to rescreening in the publishing process.

in this example, the pattern is a square lattice with center-to-center, five-by-five pixel cells. When printed, the code is composed of a rectangular lattice of linear or elliptical marks oriented in one of two orthogonal directions for recording one bit per mark.

Block code structures. Mapping logical encoding structures onto the glyph lattice pattern offers considerable flexibility. Figure 1 also shows an example of some of the structures a self-clocking glyph code can include to provide robustness in a block message coding application. The code layout format reserves rectangular framing of glyph code for synchronization codes. This assures the proper logical ordering of bits when interpreting a scanned image of the glyph-coded print, even in the presence of substantial image damage and distortion.

The synchronization code can be composed of exactly the same type of glyph marks as the rest of the glyph code with the synchronization information in the framing code. Maximal-length, shift-register sequences—pseudonoise or PN codes—are particularly useful synchronization codes that provide global addressing from a local portion of code.

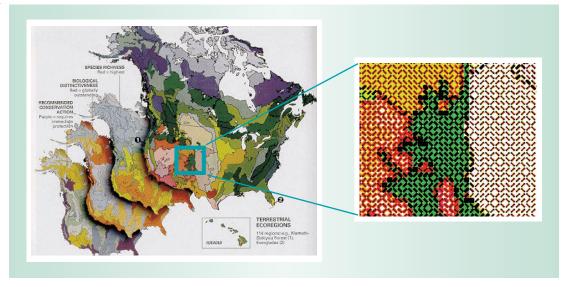
The framing code also globally distributes block parameter information implicitly by referencing the local address from both ends of the block and explicitly by interleaving parameter information and flag bits. We encode variable data glyphs in logical arrangement relative to the synchronization codes. Local subarrays of glyphs can correspond to bytes of data. The data bytes can be error-correction protected via supplementary parity bytes using, for example, Reed-Solomon codes. The message can be encrypted when appropriate. Thus, we can use the glyph pattern to realize effective implementations of key elements of data storage and communication systems. **Information capacity.** The example code in Figure 1 provides m = 1 raw bit per glyph and, with five-by-five pixel glyph cells at 300 dpi, contains 3,600 raw bits per square inch. The synchronization frame uses one of each 15 row and column patterns, leaving about 400 bytes per square inch for data and error correction. Naturally, smaller-scale printing increases glyph code data density. Glyph codes generally have data density comparable to high-performance two-dimensional barcodes at similar printing and scanning resolutions.

DataGlyph aesthetics

A key aspect of DataGlyph technology is its aesthetics and graphic character. When the structural elements are arranged together on the uniform glyph code lattice, the entire code generally appears as homogeneous texture, as Figure 1 shows. The marks for all data states have the same number of dark (colored) pixels or print coverage; thus, the visual appearance is substantially independent of the data states at typical user viewing distance. Data states are machinereadable when the image-capture system scans the pattern with sufficient resolution.

Glyphtones for graphics and pictorials. We can generalize glyph states to distinguishable halftone cell patterns of equal gray or color value, especially rotations of patterns without circular symmetry as Robert Tow proposed.⁴ Figure 2a shows a pictorial halftone, using elliptical dots on a rectangular lattice. At an appropriate viewing distance, the visual effect is pictorial, and the individual cells are imperceptible. The blowup in Figure 2b shows detail of the cells.

We can also use dot-on-dot halftones with the same data state in each color separation layer to make color pictorial glyphtones, as Figure 3 shows. Doug Curry Figure 3. Illustration with blowup showing color pictorial glyphtone. Glyphtones by Noah Flores.



has developed serpentine cell patterns that provide enhanced image quality on high-addressability printers as well as security features like copy resistance.⁵ With these methods, DataGlyph encoding is not only compatible with graphics but also can compose the graphics. The key enabler is that data modulation is substantially independent of image luminance and color modulation for practical imaging purposes. Thus, DataGlyphs encode digital data communication and analog visual imaging channels simultaneously.

Addressability

Address codes play a key role in embedded data GUI applications; they enable finely resolved user selection of objects and point-of-action in general graphic substrates, analogous to point-and-click action in electronic GUI systems. Glyph codes generally provide an address to a precision of one glyph and finer addressing by interpolation. A typical glyph lattice is 75 glyphs per inch with 600 dpi printing, which is comparable to good quality screen pixel resolution—for example, 72 dpi with gray scale. The codes also establish the orientation of the lattice. Glyph codes enable logical orientation and a precision angular orientation (approximately one degree) of the action tool with the substrate.

Carpet codes. GUI images typically are graphically intensive, particularly in the neighborhood of selection objects. This can include substantial foreground or background graphics or light/dark saturation regions in glyphtones, which make some glyphs unreadable. Figure 4 schematically illustrates a form of address code designed for such applications. Two sets of parallel one-dimensional address codes—for example, PN sequence of length $2^{12} - 1$ —are inter-

laced with successive code lines in each set progressively offset, shifting odd rows two glyphs forward and shifting even rows two glyphs backward.

This results in diagonal loci of constant phase, which intersect to define a unique 2D address location. The image-capture system can use the diagonal patterns of redundant glyph marks on the alternating rows to determine correct bit values in the presence of severe graphic clutter, including overprinted text and line art or even other glyph codes. We have termed these codes *address carpets*. We can think of objects as sitting on a location in the address carpet that defines their logical position in abstract space or a physical position in real space. Other interleaved or even superimposed codes can augment address carpets.

Periodic tiled codes. Using periodic tiled glyphs is an effective technique for distributing small messages, such as substrate identifiers, throughout an interface. As long as the image-capture footprint spans at least one tile of the pattern, image-capture tools can reconstruct tiled code from tile segments even if the capture is not aligned with a whole tile. This is important because the user does not have to visually identify the tiled code. The tiled glyph pattern can be graphically homogeneous with the rest of the glyph code without perceptible tile borders. Tiling is particularly effective for limited capture fields such as pen cameras.

EMBEDDED DATA GUIs

Embedded data systems have two main interactioncapture modes. In one mode, the system scans the entire substrate for data, markups, and objects the user places. The data embedded throughout the captured document image facilitates processing. Capture mechanisms include fax machines, page input scan-

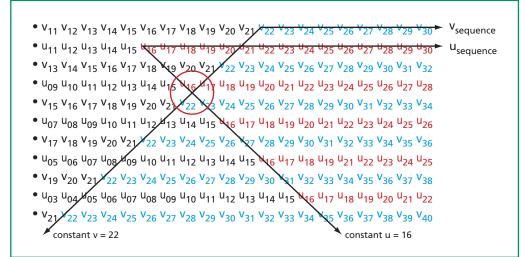


Figure 4. Schematic illustration of address code. Two sets of parallel one-dimensional address codes are interlaced, with successive code lines in each set progressively offset, shifting odd rows two glyphs forward and shifting even rows two glyphs backward. This results in diagonal loci of constant phase, which intersect to define a unique 2D address location. Design by David Hecht and David Jared.

ners, and multifunction scan-copy-fax-print document processing machines. Daniel Bobrow and colleagues' "Paper Intermedium: Using Paper to Support Interactive Educational Experiences" sidebar describes an application of DataGlyph technology in this mode.

In an extended 3D environment, a room-scanning camera image system can support multiuser interactions by capturing embedded data distributed over a large user environment such as a meeting room whiteboard.⁶ In one implementation, magnetic wall tag icons with appropriate graphics and custom-designed DataGlyph patterns for zoom camera capture incorporate the embedded data.

In the second mode, a hand capture tool selects graphical objects or positions of action, analogous to "point-and-click" electronic screen GUIs. Buttons or other control mechanisms or gestures can invoke action. The capture tools we have implemented include a pen that acts as a stylus, a camera-mouse that is used on the graphic substrate to capture and act on objects, and a physical magic lens that superimposes computer-generated synthetic displays when placed over a substrate with enabling embedded data.

User interface desktops

Graphic icons with text labels using DataGlyph carpet code can provide precision selection addressing. The user places an image-capture tool, such as a camera-pen or camera-mouse, on the desktop, selects an icon, and activates (opens) the icon or drags it to another selected icon such as a printer. Selection is based on capture on or close to an icon using screen address rules similar to those of an electronic display GUI system. The response can occur on another physical system such as a computer screen or a printer. An electronic display can show PEDGUI desktop segments for interaction via the image capture tool. Users can manipulate multiple pages of the printed desktop to augment interaction with a limited display computer or a system without a display such as a networked printer.

We can adapt the printed desktop selection paradigm to different application spaces. For example, the user could select individual objects in a pictorial catalog page or an advertisement printed with an address carpet and drag them for purchase via a pocket wireless computer.

Video access from printed documents

NoteLook⁷ lets users take notes on a pen-computer while viewing video that the system digitizes and stores on a server. The system time stamps the notes and synchronizes them with the video. In addition, the user can snap still images from the video into the notes, either as thumbnails in the margin or as fullsize images in the note page. The system also time stamps these images and synchronizes them with the video, so that they too serve as indexes.

After a note-taking session, the user can play back the recorded video with NoteLook. Standard VCR controls such as play forward or reverse control the playback position. In addition, the user can access specific parts of the video with the indexes the pen strokes and still images in the notes created. The user selects the desired index object and presses the play button. An annotated timeline is also available to position the time of video playback.

NoteLook is particularly useful in a conference room setting, where a wired or wireless network can transmit video from room cameras and projectors to the NoteLook computer. While it is easy to imagine such note-taking devices being available in a conference room, it is less likely that users will have their own devices for use outside the room. Thus, giving

Paper Intermedium: Using Paper to Support Interactive Educational Experiences

Daniel G. Bobrow, John O. Everett, and Andrew Vyrros Xerox Palo Alto Research Center

Computers are common in today's classrooms but are available to most students only for special projects. Paper and pencil are available at all times but do not have the processing flexibility we associate with computers. A new class of printing and scanning software blurs the boundary between online and offline information, giving rise to the term *paper intermedium*, a communication tool that has characteristics of both digital and print media.

The printing and scanning software that enables this interaction has only a few simple primitives. DataGlyphs encode machine-readable instructions telling response software what to do with the text it scans from a sheet of paper. This process recognizes specific interaction areas such as multiple-choice check boxes or blank areas for essay questions. The response software extracts and processes this data.

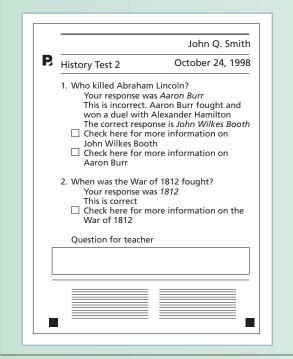


Figure A. Computer-generated feedback to an incorrect response on a multiple-choice test. Students can check a box on the annotated exam to request more information on a particular topic, and the teacher receives a summary of what all the students have done. The computer can scan in free-text interaction, such as an essay test, and sort it by question rather than by student to facilitate grading. The computer can similarly capture teacher comments along with encoded check boxes—theme well developed, full credit, bookmark for parents—that facilitate individual grading and distribution of results. For example, the computer can save the essay in a portfolio to show to the student's parents.

Given the simple affordability of this printing-scanningresponse paradigm, we can create many useful scenarios to support education in the schoolroom.

Scenarios

For each of these scenarios, we give the student pages containing learning materials and activities. The student reads the pages, does these activities, marks the pages as instructed, and returns them to the scanner or the teacher. The computer processes the information in a context-dependent way—for example, storing the page's image, interpreting the marks, or forwarding the image to an interested party. This allows instant feedback, as the computer can generate an immediate response to the student's work in the form of a new set of pages, as Figure A shows.

- Offline preparation. Each student team receives background reading for an ecology experiment involving relationships between wolf, elk, and cattle populations in Yellowstone Park. The written materials contain short exercises to check comprehension, including several that ask the students to predict how the animal populations will vary under different conditions by drawing graphs and writing comments to explain their reasoning. When the computer scans in the pages, the predictions become part of the team's permanent, online notebook. Next, students actually run the experiment with an ecology simulator, viewing both the simulated population curves and their earlier predictions. Online, they discuss with each other and with adult mentors significant features of the differences between what they predicted and what happened.
- Publisher collaterals. The background reading in the ecology example contains references to more detailed articles stored at the publisher's Web site. The computer scans a form in a published workbook to order additional materials, then it fetches the material from the publisher and prints it locally. The references in the workbook are pointers to the kind of additional information desired, as the actual articles returned can change over time. In this example, the paper intermedium offers a way of encapsulating a Web locator for additional resources so that teachers and students don't need to deal directly with the publisher's Web resources.
- *Field-data collection*. The teacher prepares a form for collecting data about a nearby stream's ecology, including check boxes for the plant species expected, blank space for additional species found, line charts (to which ticks can be added for measurements) for recording water and temperature levels, a map of the stream to indicate where the sample was recorded, and check boxes to identify the student and the date. The teacher makes copies of this form for the students to take home. The computer scans the returned forms, aggregates the data, and displays the results in the classroom to give a growing view of the collective research data. This example stresses the teacher's ability to create customized, portable documents and replicate them easily and inexpensively.

In addition to these customized learning materials, there are myriad possibilities for automating a teacher's back-office duties. The system can easily generate tests with simple interfaces and customized libraries, shown in Figure B, and automatically score and collate the results, with particular responses flagged by the software. Thus, the system can direct a student who responds favorably to a topic toward further reading on that topic or it can pair the student with others with similar interests. The system also enables automated collection of students' work in portfolios that the teacher can distribute to parents in hard- or softcopy form and to other students in newsletter form, enabling a rich student-teacher-parent dialogue.

System architecture

The architecture connects three principal components: application clients, the Paper Intermedium Interaction Server, and rendering/reading engines. Specialized teaching application clients provide a way for educators to prepare new instructional materials, make requests to distribute previously prepared materials, and examine online the results of educational interactions. The PI interaction server's Java objects preserve the output from these application clients. Part of an object is an XML description of the educational materials. Object types include recipient, distribution lists, type of test, questions, and links-pointers to additional textual information. The XML descriptions represent the element's semantic content; nothing in the description constrains the medium in which the server can render it. When an application client sends a distribute-on-paper request to the interaction server, the server creates for each recipient a paper-specific description of the form to be printed.

The return trip for paper media consists of scanning filled-in forms into the FlowPort server (http://www.xerox.com/flowport). In what we call a completion, the server recognizes the form, does appropriate image processing, and returns an XML description of the filled-in form to the interaction server. An action engine implemented as a Java applet)—decides on the basis of the completion and the original input whether to create a new form to send back to the respondent or to anyone else. Because our XMLbased Mform language contains translations to other media, students can access materials via the Web with a different rendering engine. But it will be some time before this dream of equal opportunity access to computation is achieved, and using paper still plays an important role in helping teachers provide flexible computer-guided instruction to students.

Daniel G. Bobrow is a research fellow at the Xerox Palo Alto Research Center, where he also manages the Scientific Engineering and Reasoning Area. His research interests include community knowledge systems, knowledge representation, and programming languages and systems. He received a PhD in artificial intelligence from MIT. Contact him at bobrow@parc. xerox.com.

John O. Everett is manager of the Reasoning about Document Collections Area in the Systems and Practices Laboratory of the Xerox Palo Alto Research Center. His research interests include knowledge representation and qualitative and commonsense reasoning. He received a PhD in artificial intelligence from Northwestern University. Contact him at jeverett@parc. xerox.com.

Andrew Vyrros is chief scientist of Zircus Research, a software innovation lab in San Francisco, where he leads projects in user interfaces and distributed systems. Contact him at av@zircus. com.

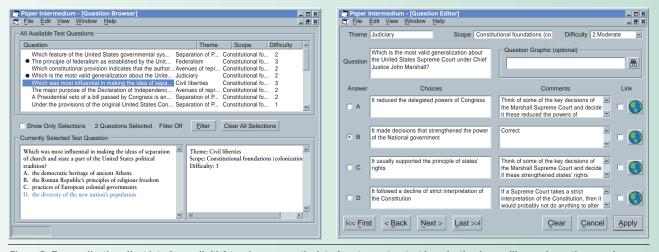


Figure B. Two application client interfaces. (left) A teacher can use the interface to create a test by selecting from a library of questions, perhaps prepared by an outside publisher. This provides local choice for the questions, but helps to ensure validation of the set of questions through more extensive interaction with students than any one teacher could provide. (right) A teacher can customize the exam and provide feedback in the form of comments and additional information.

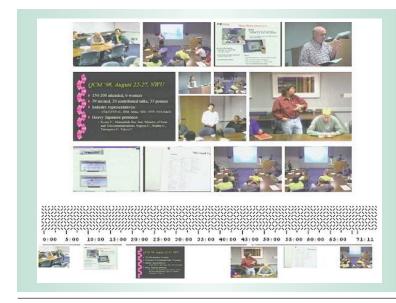


Figure 5. NoteLook lets users take notes on a pen-computer while viewing video. The system time stamps the notes and synchronizes them with the video. The user can snap still images from the video into the notes, either as thumbnails in the margin or as fullsize images in the note page. The images are keyed to the DataGlyph address carpet in the form of a horizontal timeline strip underneath the printed note page or summary.

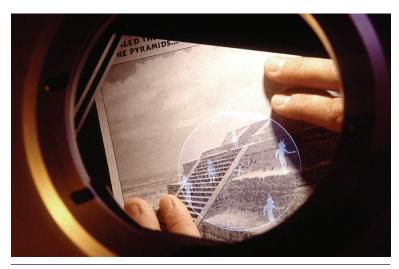


Figure 6. Glyph-O-Scope. After placing a document printed with embedded data under the Glyph-O-Scope's view-port, the user views a computer-generated image overlaid on, and physically registered with, the document.

users an alternate means to access their notes and the recorded video outside the conference room is an important consideration. One of the easiest ways to do this is to provide the user with a printout of the notes.

Embedding data in the notes provides a GUI that enables video access. One way to embed data in the notes is to use DataGlyph address carpet in the form of a horizontal timeline strip underneath the printed note page or summary. The timeline maps horizontal position to time so that the user can access any point in the video. Figure 5 shows how displaying objects at appropriate positions on the timeline aids navigation. Annotations or coloring on the timeline indicate a linear or nonlinear time scale. The timeline can use multiple parallel address carpet strips to access multiple video channels with identical or different time scales. Because of their 2D nature, address carpets can also implement coarse/fine-timing resolution in one address code strip.

Physical magic lens

Magic Lens is a novel computer interface software tool that combines a viewing region with an operator that changes the view of objects through that region.^{8,9} Users can interactively position these tools on screen applications as when you move a magnifying glass over a newspaper. However, this abstract magic lens was applicable only in the computer display image, a software analog of a hardware device. With embedded data technology it is possible to realize physical magic lenses—that is, hardware devices acting on a printed document and yielding an augmented or transformed image.

Glyph-O-Scope. Xerox PARC developed the Glyph-O-Scope shown in Figure 6 as an interactive museum exhibit.¹⁰ The device resembles a large magnifying glass mounted over a table surface. After placing a document printed with embedded data under the Glyph-O-Scope's view-port, the user views a computer-generated image overlaid on, and physically registered with, the document. The physical registration means that, as the user moves the document beneath the lens, the information presented moves with it so that it remains aligned with the physical page the user is manipulating. Thus, display information that is visible only when the user views the page through the apparatus can augment the pages. Viewing the augmentation requires the user's intuitive action, as it is similar to looking closely at a document through a magnifying lens.

The Glyph-O-Scope uses a camera to capture a magnified region of the document under the view-port. The apparatus sends the captured image to a computer, which uses the information embedded in the captured DataGlyphs to retrieve or produce an appropriate overlay for the document. The computer generates the overlay on an LCD screen concealed in the support arm of the apparatus, so that the user views the overlay through a semitransparent mirror image combiner located under the view-port.

The decoded DataGlyphs convey the identity of the document and the position and orientation of the document with respect to the camera via address carpet code. Thus, using DataGlyphs generates an accurately registered and appropriate image for overlay. As the user moves the document beneath the view-port, the apparatus updates the display to keep the augmentation aligned with the physical document.

Physical documents augmented with overlaid data have many uses, ranging from providing dynamically changing technical drawings for repair manuals to graphically filtering complex information like maps or blueprints. For example, an architect using the Glyph-O-Scope to view an augmented blueprint could choose to overlay plumbing, electrical wiring, or structural supports on the physical blueprint. In entertainment contexts, the augmented documents could consist of interactive story pages with illustrations that are animated when viewed through the Glyph-O-Scope or texts augmented with translation.

Integrated magic lens hand tool. The Glyph-O-Scope is not a portable device, but state-of-the-art electronics make an integrated magic lens hand tool feasible. Implementing image combination, augmentation, or transformation electronically and placing the physical magic lens over a substrate with an embedded address would cause it to take on the qualitative character of a hand magnifying glass that acts as a computer user interface display registered to the underlying substrate. Buttons or other signaling and gesture capturing mechanisms could provide additional input control to the interaction and display.

Developments in user interface technologies include image capture of objects explicitly or implicitly encoded with embedded data for identification, spatial registration, context setting, and other system functions. These developments provide substantial data capacity, local and global spatial address capability with good aesthetics, and graphical integration with line art, text, and pictorial documents in color or black and white. The embedded data technology, user interface research, and emerging commercial development highlighted here provide a substantial foundation for attractive and novel applications inviting widespread practical use. *****

Acknowledgments

It is impossible to mention all the colleagues at Xerox who have contributed to this work. However, I want to thank Shilajeet Banerjee, Dan Bloomberg, Dan Bobrow, Jeff Breidenbach, Ken Chang, Patrick Chiu, Doug Curry, Dan Davies, Noah Flores, Matt Gorbet, David Jared, Sven Karlsson, Leigh Klotz, Jing Lin, Ranjit Makkuni, Jun Miyazaki, Glen Petrie, Makoto Sasaoka, Eric Saund, Rick Stearns, Rob Tow, Michael Plass, Tom Webster, and Lynn Wilcox. I also thank the Xerox PARC MARS group, particularly Brian Tramontana, Deanna Horvath, and Dan Murphy, for graphics contributions documenting our work.

References

- S.K. Card, T.P. Moran, and A. Newell, *The Psychology* of *Human-Computer Interaction*, Erlbaum, Hillsdale, N.J., 1983.
- D.L. Hecht, "Embedded Data Glyph Technology for Hardcopy Digital Documents," *Proc. Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts III*, SPIE—The Int'l Soc. Optical Engineering, Bellingham, Wash., vol. 2171, 1994, pp. 341-352.
- D.S. Bloomberg et al., *Self-Clocking Glyph Shape Codes*, US patent 6,076,738, Patent and Trademark Office, Washington, D.C., 2000.
- R.F. Tow, Methods and Means for Embedding Machine-Readable Digital Data in Halftone Images, US patent 5,315,098, Patent and Trademark Office, Washington, D.C., 1994.
- D.N. Curry, "Color Pictorial Serpentine Halftone for Secure Embedded Data," *Proc. Optical Security and Counterfeit Deterrence Techniques II*, SPIE—The Int'l Soc. Optical Engineering, Bellingham, Wash., vol. 3314, 1998, pp. 309-317.
- T.P. Moran et al., "Design and Technology for Collaborage: Collaborative Collages of Information on Physical Walls," *Proc. Symp. User Interface Software and Technology* (UIST 99), ACM Press, New York, 1999, pp. 197-206.
- P. Chiu et al., "NoteLook: Taking Notes in Meetings with Digital Video and Ink," *Proc. Multimedia*, ACM Press, New York, 1999, pp. 149-158.
- E.A. Bier et al., "Toolglass and Magic Lenses: The See-Through User Interface," *Proc. Siggraph*, ACM Press, New York, 1993, pp. 73-80.
- M.C. Stone, K. Fishkin, and E.A. Bier, "The Moveable Filter as a User Interface Tool," *Proc. Computer-Human Interaction* (CHI 94), ACM Press, New York, 1994, pp. 306-312.
- M. Back et al., "Designing Innovative Reading Experiences for a Museum Exhibition," *Computer*, Jan. 2001, pp. 80-87.

David L. Hecht is a Principal Scientist at Xerox PARC. His research interests include image/signal processing and ultrasonic/optical devices and systems: acoustooptics, electrooptics, and lasers. He received a PhD in electrical engineering from Stanford University. He is a member of the IEEE, OSA, SPIE, and the Steering Committee of the IEEE/OSA Journal of Lightwave Technology. Contact him at hecht@parc. xerox.com.