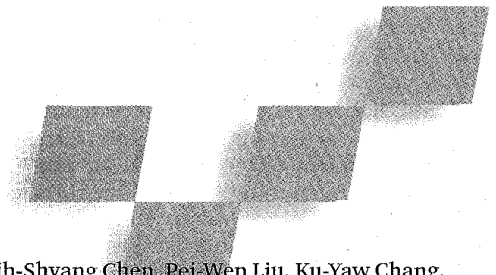


Using Hypermedia in Computer-Aided Instruction

Lih-Shyang Chen, Pei-Wen Liu, Ku-Yaw Chang, Jong-Ping Chen, Su-Chou Chen
National Cheng-Kung University, Taiwan

Hong-Chow Hong
National Chiao-Tung University, Taiwan

Jain Liu
National Cheng-Kung University, Taiwan

This article is a companion to the previous article, which describes the Discover system's architecture and functions.

The rapid advances of computer and communication technologies in the last few years offer unique opportunities for people in all disciplines and walks of life to use computer systems. The medical community is one area in which these technologies are beginning to play a vital role. As the number of medical discoveries increases, and medical and clinical cases become more complex, computers and computer networks are becoming essential diagnostic and treatment-planning tools.

Physicians use this hypermedia network system to learn the processes of the Discover visualization system and to help them standardize the way they examine images.

Visualization is also playing a larger role in diagnostic procedures. Medical modalities such as magnetic resonance imaging, computed tomography, positron emission tomography, single photon emission computed tomography, and ultrasound continue to create huge amounts of data for clinical diagnoses. The result is that image-processing workstations must get more complex to support image analysis and generation in medical applications.

These advances come at a cost to usability, however. Often, users must have extensive knowledge in a variety of disciplines. A sophisticated medical-image workstation, for example, may require physicians, who generally work in a film-based environment, to face unfamiliar commands and procedures. To learn to use the system effectively, physicians must learn the details of image processing and computer graphics. This creates a problem because most physicians naturally prefer to spend as much time as possible studying complex clinical cases. Yet, without intensive training, it is virtually impossible for them to master an advanced medical-image system.

And even if they could overcome the system's complexity, they might still use the system in ways that could produce incorrect results. At the very least, they would not use the system to its full advantage. For these reasons, we believe it is important to standardize the protocol for examining 2D and 3D computer images. The benefits are higher quality examinations and a greater likelihood of consistent performance.

The previous article in this issue describes the Discover distributed interactive visualization system, which has been running at National Cheng-Kung University Hospital since 1993. Shortly after Discover was implemented, we observed that physicians were having trouble understanding and navigating through Discover processes for image analysis and generation. To address this problem, we developed a computer-aided instruction mechanism based on hypermedia technology—multimedia with hypertext navigation. Hints (Hypermedia Information Network System) is built around the concept of *hypercontrol*—the use of a multimedia document to control hardware or software systems. It is currently being used with Discover at the university hospital.

As its name implies, Hints gives suggestions to Discover users in specific stages of image analysis and helps standardize image-examination protocols. We designed Hints to be an independent application. That is, users can run it in conjunction with Discover or use Discover alone. This arrangement satisfies the varying need for help as Discover users gain more knowledge about system processes.

Although our emphasis here is on medical applications, the Hints architecture is suitable for other types of applications—such as control systems for a nuclear power plant—in which complexity endangers usability and where the standardization of protocols is vital.

Moving beyond hypermedia help

Hypermedia is a simple and natural extension of multimedia and hypertext. Multimedia provides rich data types that make it easy to express information, while hypertext provides a control structure that elegantly supports navigation through the information. With the recent multimedia explosion and the attendant

0272-17-16/96/$5.00 © 1996 IEEE

advances in multimedia technology and computer networks, hypermedia systems have become quite popular for a range of applications, including legal research and medical imaging. The World Wide Web hypermedia system is an example of that popularity.

In a hypermedia system, information representation and management are built around a network of multimedia nodes connected by typed links.[1] Users navigate through a network following links from node to node until they gain some information of interest or get answers to their questions.

In the context of computer-aided instruction, a hypermedia system is a kind of middle ground between online help and intelligent CAI, two endpoints in the spectrum of help systems. Online help systems generally provide user assistance (tips, or a status bar), task-oriented help (cue card or wizards), and references to help (reference help window),[2] but do not provide for end-user programming.[3] ICAI systems, on the other hand, are very sophisticated and powerful in solving problems in a particular application. Unfortunately, many systems are prototypes, not practical tools.[4]

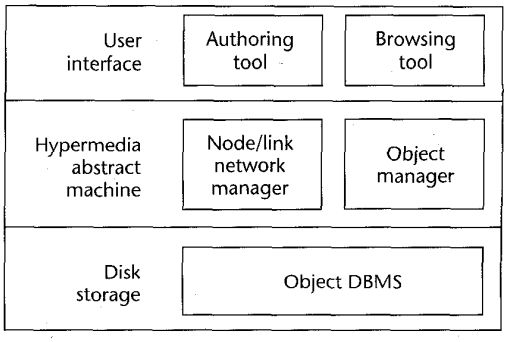
A hypermedia system is certainly more complex than online help, but it is generally a passive storage and retrieval system, not a problem solver like an ICAI system. This inability to actively direct the creation or modification of the network or the information it contains is becoming more of a disadvantage as hypermedia systems gain widespread use: The simple node/link model is just not rich and complete enough to support information representation, management, processing, and presentation—tasks many applications require.

In designing Hints, our goal was not only to provide passive help information in the problem domain but also to allow the storage of some commands of associated systems that can actively create and process the information in question.

Hints architecture

Figure 1 shows the architecture of Hints, which has the following levels:[5]

- *Representation.* This level, which is essentially the user interface, includes browsing and authoring tools. The authoring tools let the user author, manipulate, and update the contents of the node/link network. The browsing tools provide a user-friendly way to look through the hypermedia network. Users can follow the links in any node to activate new nodes.
- *Hypermedia abstract machine.* This level comprises the Node/Link Network Manager and Object Manager. Each node in the hypermedia system represents a concept like a disease, a procedure like a Discover control procedure, or an object like a spine. Each node contains at least one multimedia object, such as text, images, video, voice, or graphical object, and is displayed within a window. Hints organizes nodes in a hierarchical directory similar to the hierarchical file system in Unix. Related nodes are in the same directory. We have chosen to relate nodes by topic, but the hypermedia document author can choose other meanings for "related."



1 Architecture of the Hints hypermedia-based help system.

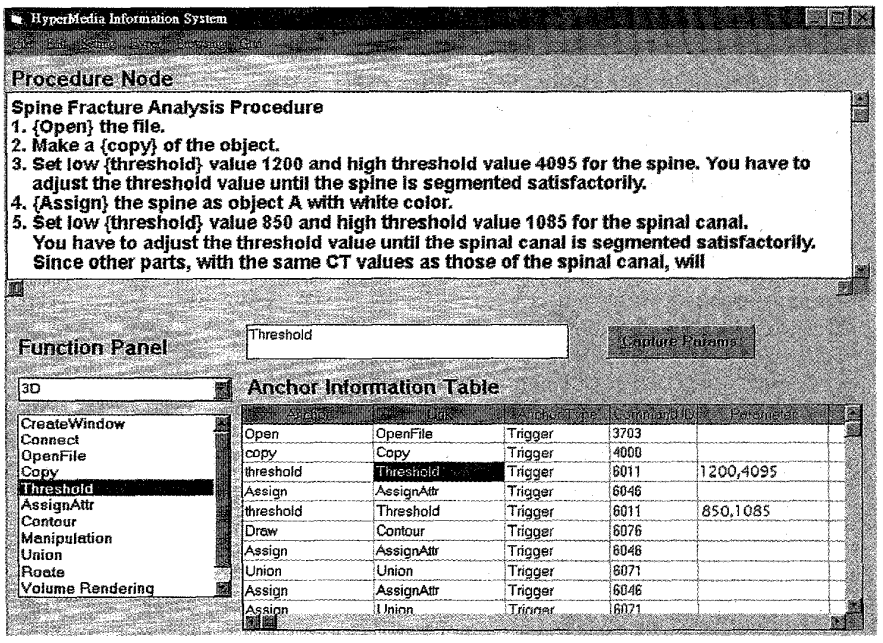
The Node/Link Network Manager is responsible for managing and maintaining the nodes and links. The Object Manager accesses and manipulates the objects in the object DBMS, including voice, video, image, graphical, and animation objects. Animation objects let users play objects or nodes sequentially or play nodes, voice, and a moving cursor simultaneously so that the author can explain the contents of the nodes and move the cursor around to indicate where in the nodes he is referring.

- *Disk storage.* This level contains the object DBMS, which resides on a network file server shared by all workstations on the local area network.

How hypercontrol works

The basic idea of hypercontrol is to use a multimedia document (written by nonprogrammers working with engineers) to control other computer hardware or software systems. The control commands are installed in anchors, called *hypercontrol anchors*. The following procedure shows how hypercontrol works:

1. For each clinical case (in this example, a spinal fracture), physicians collect enough medical images to make a diagnosis.
2. Physicians use Discover to process the image data. At this time, physicians work with Discover engineers to understand Discover's various image-analysis and image-generation processes. We expect that eventually we will have enough "seed" physicians (those who are familiar with Discover processes and how to write such a document) that engineers will rarely be needed. Physicians record each Discover function they use to process the images and try to come up with a procedure that would apply to a similar data set. If necessary, they work with Discover engineers to develop new functions to cope with the problem. The new functions become either part of the kernel system or an AP, depending on how specific they are to the particular application.
3. Physicians apply the same procedure to many data sets to make sure the procedure consistently produces reasonable results. If so, the procedure becomes a diagnostic protocol for the particular case.
4. Physicians or engineers use the authoring tool to edit the procedure into a *procedure node*, which then becomes a regular node in the hypermedia network.



2 The authoring mode in Hints. The author of the procedure node can link any of the PN's anchors (words in braces) to one or more Discover commands by dragging the command from the function panel (lower left) to the anchor-information table (lower right). In this case, the author has linked the Threshold anchor in step 3 to the Threshold command. Hints will now display the threshold dialog box initialized to 1,200 low and 4,095 high when the user clicks on the anchor.

A single Discover function or sequence of functions used in the procedure corresponds to one step in the PN. Within each step, authors can create several anchors. Each anchor can be a single word, a series of words, or a particular notation like [ex1]. Authors can then link the anchors to particular Discover commands or to help information.

5. Figure 2 shows how the author links the "Threshold" anchor in the PN for setting the threshold on a 3D image. The function panel (lower left box) contains all the commands (or functions) Discover provides. The anchor-information table (lower right box) contains the names of all the anchors in the PN in the order they occur. In the figure, the author has linked the {Threshold} anchor in step 3 (in authoring mode, anchors are in braces) to the Threshold command by dragging and dropping Threshold from the function panel into the Command column of the anchor-information table.
6. Now when the user activates the anchor during browsing, Hints activates the Threshold button in the global command menu, which results in the display of the Threshold dialog box containing two scroll bars initialized at 1,200 and 4,095, respectively. The user can then adjust the threshold values until the region of interest is segmented from the scene. Without Hints, the user would have to know what procedure must be done and where and which pulldown menus to use. With Hints, the user simply follows through each instruction in the PN, clicking on each anchor, and interacting with Discover as prompted.

PNs and anchors are useful for training or for gathering information to aid similar cases. As we mentioned earlier, Hints has a browsing and authoring mode. To browse the hypermedia network, users double-click on the Hints icon in Discover and select browsing mode. Hints asks them to open a multimedia document, which can contain one PN or a series of PNs, as well as the object types in the the object DBMS. Users can either find a particular PN by indexing the network or browse through and pick PNs at random. Once a user finds a PN, he can exercise the procedure within a PN, clicking on the various anchors to activate each step.

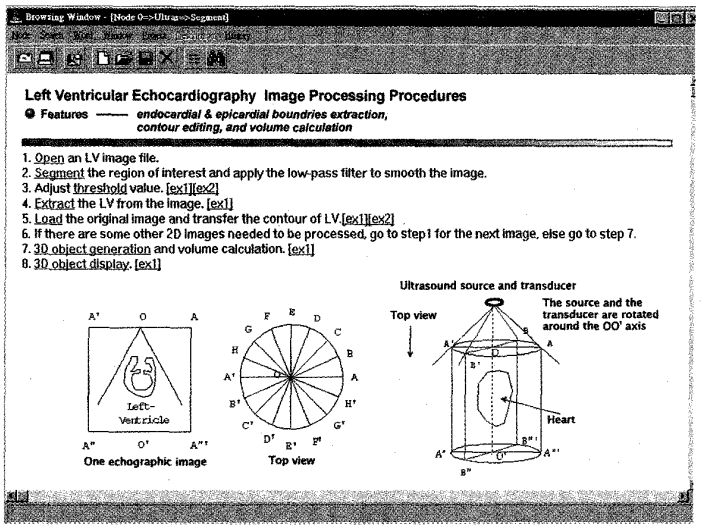
If users select the authoring mode, they can create a PN or series of PNs with linked anchors that can partially or fully automate a Discover process. If the author links several Discover functions (commands) to one anchor, Discover can process data without user intervention (similar to the use of macros in many commercial software packages). If there is a need for user intervention for a particular function, the author avoids linking it with other functions to the same anchor, and the user will be able to take control over the Discover process at that moment. (In many image-processing applications, user intervention is mandatory to achieve correct results.) Because the text in the PN briefly describes the purpose of each step, users are able to see what they are doing and can decide if they want to automate or manually control the process.

Users can also link Discover processes to an anchor by a process similar to recording a macro in word processing. While in the Hints authoring mode, the user simply starts up Discover, selects Record in the Hints window menu, completes a series of activities using Discover, and selects Stop in the Hints window. That series of activities will now automatically occur when the anchor is activated.

The procedure in the PN can also have an If statement so that the user can decide if the condition in the If statement is satisfied or not and click different anchors accordingly. For example, the user might see "If the boundary of the tumor in the image is blurred, apply {high-pass filter} to emphasize the edges, else apply

{low-pass filter} to reduce the noise." Whether or not the image is blurred is a highly subjective issue, so user intervention is necessary. Yet the user has some degree of automation once the decision is made.

The anchor links can be structured to reflect all types of system knowledge. For example, if users are very new to the system, they often do not know how to deal with various dialog boxes. For these users, Hints provides sample uses of the unfamiliar dialog box along with explanations and related help information in the form of example anchors ([ex1], [ex2], and so on). If users have enough knowledge about Discover processes, image processing, and computer graphics, they can use Discover without Hints, accessing the pulldown menus and command buttons directly. However, Hints is still a useful reminder of what protocols have been tested and found correct.



3 The procedure node for extracting a series of 2D left ventricle images and reconstructing a 3D left ventricle image. The node also includes additional information in the form of diagrams.

Case studies

To illustrate how Hints works, we present two case studies. Regrettably, we cannot illustrate how it works with video and voice, so we omit them from the examples. However, because a PN is a regular hypermedia document in Hints, it can include all these kinds of media.

Case study 1

Echocardiography has been widely used as a real-time clinical tool for the diagnosis of cardiac diseases. The extraction of left ventricular endocardial and epicardial boundaries from digital 2D echocardiography and the estimation of the ejection ratio of the left ventricle are essential in the quantitative analysis of cardiac function.[6] We have developed an AP to aid in solving this problem. The AP can process a series of 2D LV images and eventually reconstruct the 3D LV images. (In the example, we ignore the technical details of the procedure, giving just enough to explain how Hints can help users through the processing procedure.)

The physician uses the AP to extract the LV from the original images. After extracting the LV, she can edit the LV contours. Contour editing is a basic image-processing operation, so it is transferred to the 2D image window of the kernel system for further editing. After Discover processes each image and generates a series of contours, the physician can reconstruct the 3D LV and display it.

Figure 3 shows the PN for the extraction procedure. [ex1] in step 3 is an example anchor linked to a node for further explanation. [ex2] is linked to another node for a specific example. All anchors are highlighted. The anchors without brackets are the hypercontrol anchors. When the user places the cursor on an anchor, it changes appearance to indicate that it is indeed an anchor.

The following explanations correspond to the PN steps in Figure 3:

1. Load one ultrasound LV image into the window.
2. Segment the region of interest and apply the low-pass filter to the segmented image to smooth the image.
3. Adjust threshold value. Discover automatically calculates the threshold value. Users can adjust the value so that the left ventricle is thresholded properly.
4. Apply cutting, expansion, morphology, edge-detection, and contour-finding operations to the image. All the image-processing functions are linked to the same anchor.
5. Load the original image into a 2D image window of the Discover kernel system and transfer the LV contour into the 2D image window for editing.
6. If some other 2D images need to be processed, go to step 1 for the next image, or else go to step 7.
7. Generate the 3D object and calculate its volume.
8. Display the 3D object.

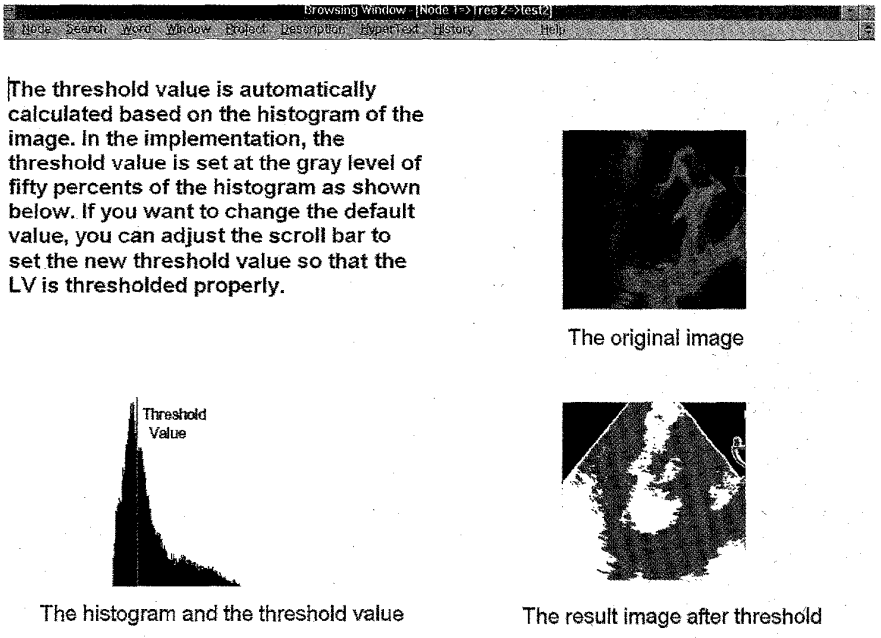
If the user clicks on [ex1] in step 3 of Figure 3's procedure, Hints will display the node shown in Figure 4, as any other hypermedia system would do. If the user clicks on [ex2], Hints will provide examples of a threshold value that is set correctly, too high, and too low.

Case study 2

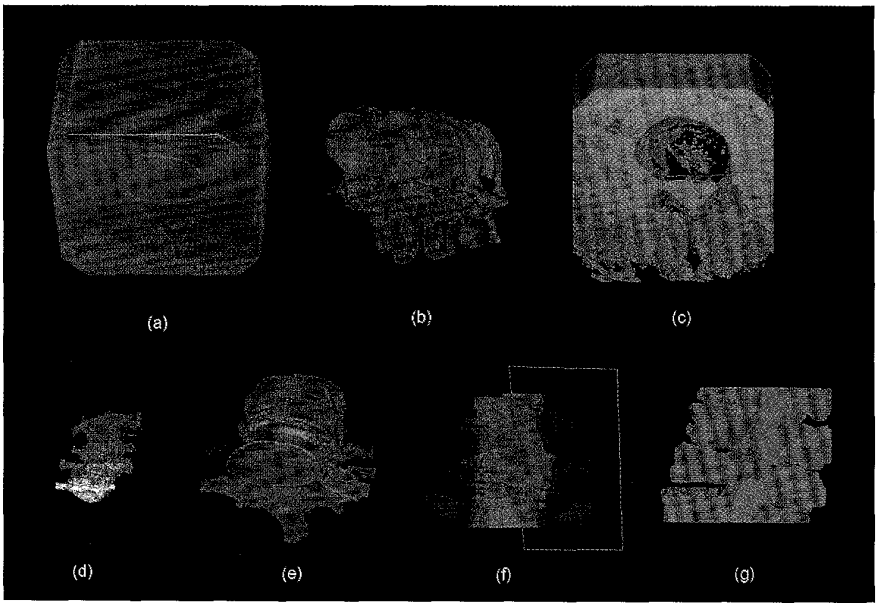
In this case, the patient has a spine fracture. The surgeon would like to determine if the fracture is with or without dura compression and if the fracture pattern is a burst fracture or compression fracture. Both of these are hard to determine with 2D images. Therefore, the surgeon would like to reconstruct the 3D images using 2D image slices and to manipulate the 3D objects. These are functions in the Discover kernel system, so no APs are needed in this example.

The surgeon uses the following PN (anchors are in italics, although they are usually underlined):

1. *Open* the file.
2. Make a *copy* of the object.



4 The explanation of how a threshold value is calculated, which appears when the user clicks on the [ex1] example anchor in step 3 of Figure 3.



5 Results corresponding to the steps of a PN that aids in diagnosing a spine fracture. (a) Result of step 2. (b) Result of step 3. (c) Result of steps 5 and 6. (d) Result after the manipulation of step 6. (e) Result of steps 8 and 9. (f) Result of steps 10 and 11. (g) Result of steps 12 and 13.

3. Set low *threshold* value 1,200 and high threshold value 4,095 for the spine. You have to adjust the threshold value until the spine is segmented satisfactorily.
4. *Assign* the spine as object A with white color.
5. Set low *threshold* value 850 and high threshold value 1,085 for the spinal canal. You have to adjust the threshold value until the spinal canal is segmented satisfactorily. Since other parts, with the same CT values as those of the spinal canal, will also be selected, they will be eliminated in step six.
6. *Draw* a contour to select the spinal canal.
7. *Assign* the spinal canal as object B with yellow color.
8. *Union* object A and object B and assign the resultant object as object C.
9. *Assign* different colors to object A and object B.
10. *Assign* component A (object A) of object C as a transparent object.
11. *Draw* a contour to cut the spine for revealing the dural compression.
12. *Assign* component A (object A) of object C as an opaque object.
13. *Display* the results.

Figure 5 shows the results of the various steps. In this

case, the fracture pattern is a burst fracture with a posteriorly displaced fragment. It also has severe dura compression.

Conclusions

We believe Hints makes two valuable contributions in its current application environment: It promotes the use of 2D and 3D computerized images, and it helps standardize the protocols for examining those images. Both of these benefits can lead to more accurate and faster diagnoses of medical and clinical cases and a resultant decrease in health care costs.

Physicians report that Hints makes Discover easier to use. Even Discover's developers (graduate students) benefit from Hints because they find it easier to demonstrate system capabilities to others. (Each developer created a small part of the system and may not be familiar with other parts or the entire system.) Finally, we have found Hints useful for training novices without getting into too much detail about Discover system operations. This is helpful because Discover is getting more complicated every year.

Hypercontrol is a powerful and flexible mechanism for commercial and industrial applications (like the control of a nuclear power plant) in which fully automatic control or processing is impossible. When there is a need for periodic user intervention, hypercontrol can shorten training time, ensure that users apply the system consistently and at a uniform quality level, and make the user interface more friendly.

Although we have described a medical application of Hints, we believe it is suitable for many other kinds of window systems. The implementation of hypercontrol is based on Microsoft Windows' event-driven and message-passing mechanisms, which other popular window systems also use. Thus, hypercontrol can be used as part of the user interface for many other window-based applications. Moreover, the popularity of the World Wide Web has brought forth implementation technologies like Java and Visual Basic Script that can make a hypermedia document "come alive"—pieces of executable code can be added to the document text. Such technologies are highly suitable for implementing our concept of hypercontrol.

Our immediate plans for enhancing Hints include incorporating access to the World Wide Web and making it compatible with Java or VBS technologies. In this way, we hope to provide intelligent help for medical imaging and other applications. ■

Acknowledgments

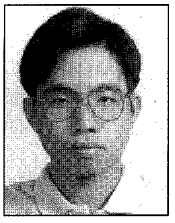
This research was supported in part by National Science Council of Taiwan, Republic of China.

References

1. F.G. Halasz, "Reflections on NoteCards: Seven Issues for the Next Generation of Multimedia System," *Proc. Hypertext 87*, ACM Press, New York, 1987, pp. 345-365.
2. *The Windows Interface Guidelines for Software Design*, Chapt. 12, "User Assistance," Microsoft Press, Redmond, Wash., 1995.
3. *Watch What I Do: Programming by Demonstration*, A. Cypher et al., eds., MIT Press, Cambridge, Mass., 1993.
4. F. Roberts and O. Park, "Intelligent Computer-Assisted Instruction: An Explanation and Overview," *Expert Systems and Intelligent Computer-Aided Instruction*, Vol. 2, Educational Technology Publications, Englewood Cliffs, N.J., 1991, pp. 131-136.
5. B. Campbell and J.M. Goodman, "HAM: A General-Purpose Hypertext Abstract Machine," *Comm. ACM*, July 1988, pp. 856-861.
6. J. Feng, W.-C. Lin, and C.-T. Chen, "Epicardial Boundary Detection Using Fuzzy Reasoning," *IEEE Trans. Medical Imaging*, June 1991, pp. 187-199.

Lih-Shyang Chen *is an author of the previous article in this issue. His biography and photo appear on p. 51.*

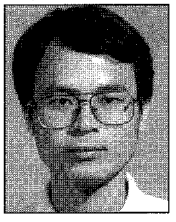
Pei-Wen Liu *is an author of the previous article in this issue. His biography and photo appear on p. 51.*



Ku-Yaw Chang *is a PhD student in electrical engineering at National Cheng-Kung University, Taiwan. His current research interests are in image processing and pattern recognition. Chang received a BS and an MS in electrical engineering from National Cheng-Kung University, Taiwan.*

Jong-Ping Chen *is an author of the previous article in this issue. His biography and photo appear on p. 51.*

Su-Chou Chen *is an author of the previous article in this issue. His biography and photo appear on p. 51.*



H. C. Hong *is a PhD student in computer science and information engineering at National Chiao-Tung University. His research interests include multimedia, hypermedia, computer-supported cooperative work, and image processing. Hong received a BS in computer science from the Chinese Culture University and an MS in electrical engineering from National Cheng-Kung University, Taiwan.*

Jain Liu *(deceased) was a lecturer in orthepedics at National Taiwan University. He was a well-known expert in diagnosing and treating shoulder injuries and traumas. He was also a research fellow at the Mayo Clinic and University of Pittsburg. Liu held a masters in biomedical engineering from National Cheng Kung University.*