IDB: An image database system

by A. Turtur
F. Prampolini
M. Fantini
R. Guarda
M. A. Imperato

Specialized software and hardware tools are needed to work on digital color images; the usability of an image system implies the availability of such resources inside a coherent environment and a friendly user interface. Furthermore, a large volume of data must be efficiently stored and retrieved. To cope with these problems, the prototype of an image database system, named IDB, has been developed to manage image data in an integrated way. The important features of the system are distributed functions, a multi-user environment, interactivity, and modularity.

Introduction

Most information management systems are designed to handle traditional alphanumeric data. Most hardware and software instruments, such as monitors (alphanumeric displays), printers, memory devices, and input devices (from card readers to data entry devices), have been designed to manage information of this kind.

Today technology makes available resources that allow the management of new classes of information, such as image and voice. Compared with digital alphanumeric data, these are much more complex data types, but in terms of human perception they are both rich in information and easy to understand. They therefore represent a more natural approach to man–machine communication, a perspective which is shared by many of the research groups involved in the "fifth-generation" computer systems project [1].

When dealing with images, we must generalize the input, processing, and output phases that characterize the management of traditional data types. Furthermore, all these activities require suitable hardware and software instruments. From the end-user point of view, the interface between user and system would be much more attractive if it were possible to use images to manage images as well as we use words to manage traditional information. In other words, to the same extent that we use words to search alphanumeric data, we can use the concept of image to help our navigation inside the image world. For instance, using image could be a simple and powerful aid in performing operations on an image system [2].

Two kinds of data must be managed by an image system: image files (or image), and their descriptions. The former are characterized by large size and unstructured form, while the latter have small size and structured form. Images and descriptions are stored on different kinds of devices. The main requirement for image data is the availability of a large memory at low cost. Optical disks can meet this requirement [3, 4], and image data are now increasingly stored on those special devices, while descriptive information continues to be stored on magnetic disks. In recent years the use of image systems has been growing in many fields [5], such as medicine [6], publishing [7], land use [8], and museum management

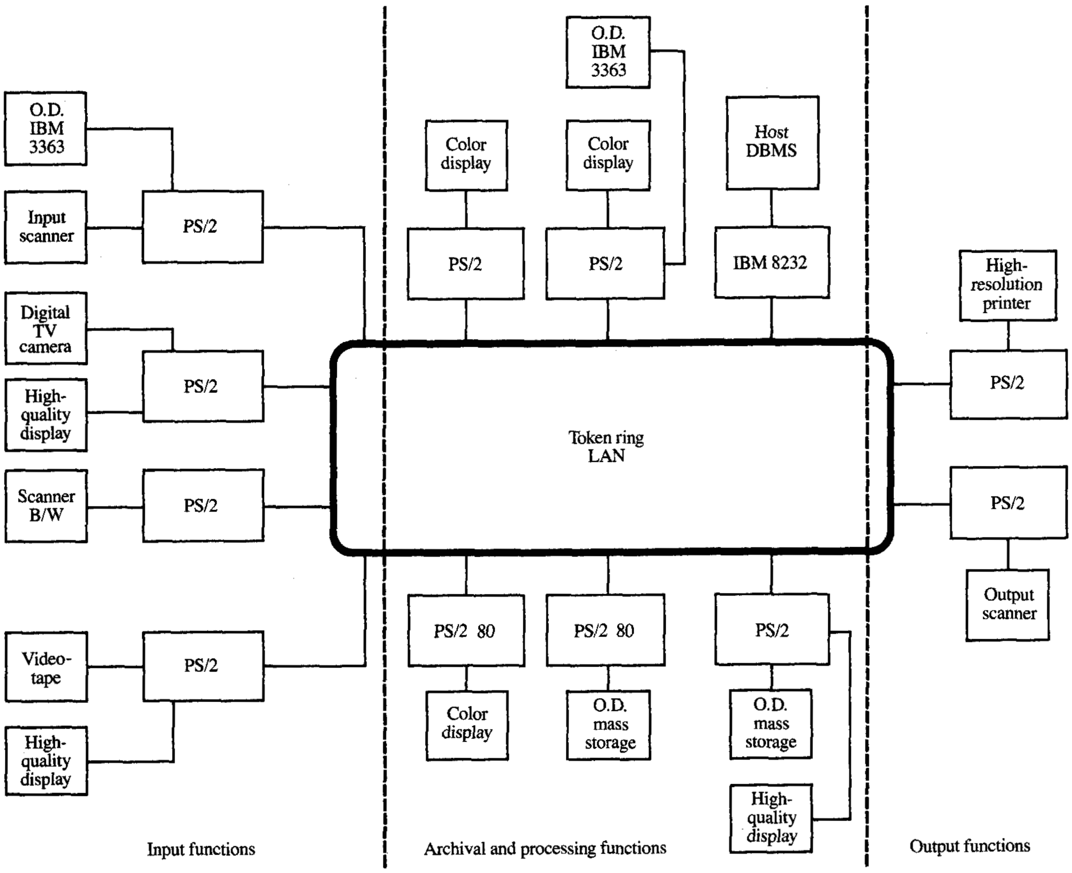

Figure 1

Image management system: Hardware architecture.

[9]. Many applications have been developed from solutions oriented to particular environments.

In this paper, we propose a general-purpose system called IDB, which is not application-specific but can easily be customized according to specific user requirements [10].

System architecture

The image database system developed at the Rome Scientific Center includes image input, processing, and output functions to provide full image management capability. It satisfies three main requirements:

- Large volume (tens of thousands) of color images.
- Integration of several resources
- Multi-user environment.

The architecture implements the concept of resource distribution using a local area network (LAN). **Figure 1** shows a typical system on a token ring network, connected to a mainframe through a channel attachment. Image data can be entered into the system from many sources through digitization, which is performed at specialized workstations where various input devices are connected to personal computers. The many input devices cover a variety of user needs in terms of picture support (paper or film) and color and space resolution, and the modularity of the IDB allows the inclusion of new devices just by adding low-level drivers.

The image database comprises a number of image datasets and a relational SQL database of alphanumeric information. The database management system (DBMS) runs on the mainframe. Each user workstation interacts

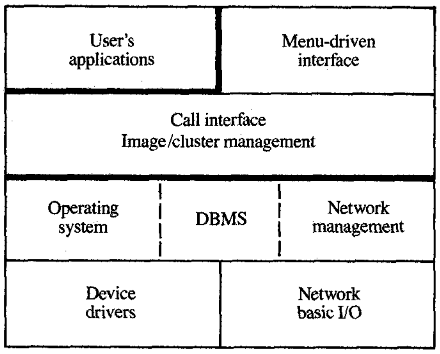

Figure 2

Image management system: Software architecture.

with the host system to file and retrieve images and the associated textual information.

Images require a large amount of storage space, generally much more than that required by traditional alphanumeric applications. To cope with this problem, storage has been organized into three hierarchical levels. Magnetic disks (both personal computer and mainframe devices) comprise the first level, whose peculiarities are low access time and the capability of being rewritten. The second level contains low-capacity (hundreds of MB) optical disks. The main image archive is stored in the third level of the hierarchy, which consists of high-capacity (thousands of MB) optical disks [3, 4].

The memory hierarchy is established on a logical rather than a physical basis, because what is put onto the faster devices depends on the query conditions defined by the user and not on the physical localization of data (as is the case for systems that use paging or caching mechanisms). This logical approach to hierarchy management allows prescriptive searching; before searching the archive the user can decide *a priori*, and then select with queries, just that subset of images he needs. There is no need to retrieve all the images of a given device. Further, the user need not know which storage devices contain the original images, and can simply identify the device on which he wants to put his own copies, usually one of his workstation disks.

Several processing functions are available for work on images retrieved from the database or entered from the external world. To display and process images, the personal computers are connected to several types of display devices (IBM 8514, RGB monitors, etc.); the available chromatic resolutions are 8 or 24 bits per pixel (RGB images). Each function can be performed from any workstation that has the required modalities. System output is spooled to network nodes, where images can be plotted on film by scanners or printed in black and white.

Software design

The user handles *images*, *clusters*, and *folders*. Images are elementary objects, and each operation on these objects corresponds to a primitive function of the system. Clusters are sets of images associated with the same textual description. The images that belong to a given cluster can represent the same subject at different resolutions. Folders are collections of clusters created on the basis of classification criteria defined by the user.

The user can interact with the system in two ways (**Figure 2**)—through an end-user-oriented *menu-driven interface*, or through a *call interface* that is used for writing application programs. These interfaces issue commands to the *image/cluster manager*, a second layer of software which includes four modules:

1. *Image I/O* for image acquisition and restitution.
2. An *image editor*, which incorporates a set of image processing functions for color and geometric handling.
3. An *image librarian* for image archiving and retrieval.
4. *Image display* for displaying images on different kinds of monitors.

A third layer of software includes the *operating system* and the *network management* software, which makes the LAN transparent for higher-level programs.

The data managed by the system are organized according to the relational model and are accessed by the DBMS, conceptually at the same level as the operating system. The hardware resources are directly addressed by the fourth and last level of software, *the network basic I/O* and the *device drivers*. Software modularity helps to achieve device independence by isolating those parts of programs that address physical devices from inside applications. Further, the modular architecture ensures that the software can be expanded to meet future needs.

Logical data organization

The entities managed by the system and their functional dependencies are logically organized according to an extended relational model [11] representing the data scheme in terms of tables or relations. **Figure 3** shows a simplified representation of the logical scheme; the boxes stand for relations and the lines stand for links among relations. The cluster is the entity at the highest hierarchical level; it is described in the table CLUSTER. This relation reports for every cluster its unique identifier, its name, and an "abstract" describing the contents of the cluster. At an intermediate hierarchical level are the images, listed in the table IMAGES, where the identifiers, the names, and other main attributes of the image files (such as image size, chromatic resolution, creation date, and time) are reported. The relationship between images and clusters is represented in the table IMGCLS, which reports for each image identifier the corresponding identifier of the cluster to which the image belongs.

Three types of images are managed by the system; for each of them a relation is defined:

- IMGRGB for images made up of three files corresponding to the red, green, and blue bands (standard RGB images at 24 bits per pixel).
- IMGBAND for multispectral images represented by several band files.
- IMGCOMP for images represented by one file per image in which the three bands are stored in a compact way using 8 bits per pixel.

Information about the physical allocation of images is supplied by the relation FILES. For each image file this table reports the identifier and type for the storage device to be used. The storage devices are divided into three classes according to the device type: IMGMAGN for magnetic disks, IMGOPT for low-capacity optical disks, and IMGMASS for high-capacity optical disks. The FILES table identifies each image with a specific device in one of these classes.

The relation FILES also points to other tables used to supervise the configuration of LAN devices and to manage the public and private copies of images extracted from the archive.

IDB prototype

To validate the architecture just described, a prototype of a general-purpose image database system has been developed (**Figure 4**). The prototype implements four main categories of basic functions on images (**Figure 5**): *acquisition*, *archive management*, *processing*, and *restitution*. The user starts a function by selecting from a window a line associated with a command, or by entering the required parameters on an input panel.

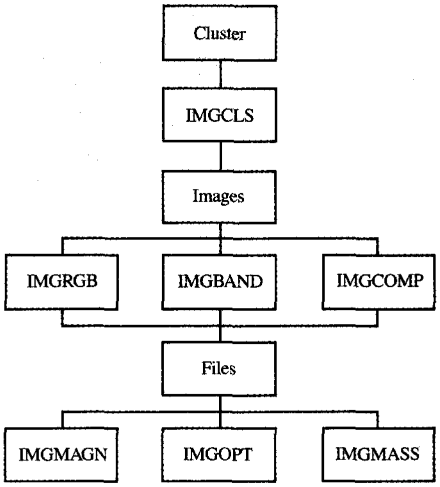

Figure 3

Logical data organization.

Figure 4

User stations of the prototype.

Acquisition

Image digitization is performed by using specialized devices, such as scanners and TV cameras (**Figures 6** and 7). The user starts and controls the process at a specialized workstation where an input device is connected to a personal computer. According to specified parameters, the system produces a three-band RGB image and the necessary information about image characteristics such as horizontal and vertical size and scan mode.

Archive management

Every image generated by the input phase can be placed in an archive, from which users can retrieve both the image and any associated alphanumeric information. Five subcategories of functions are provided to work with

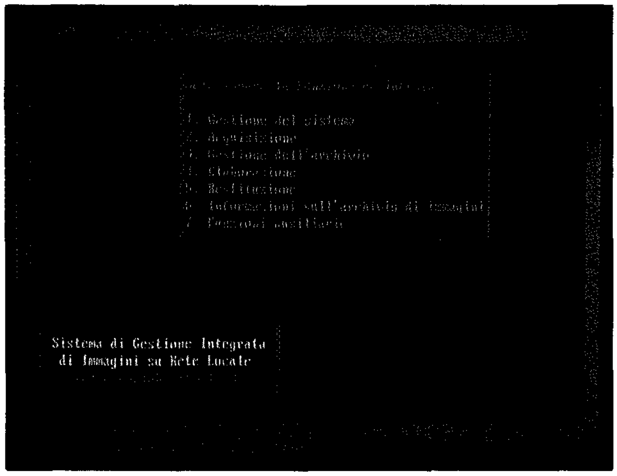

Figure 5

Main menu of the system.

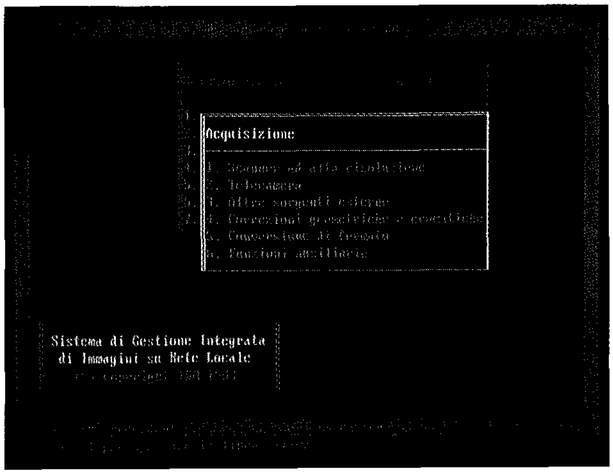

Figure 7

Acquisition menu.

Figure 6

Workstation for image acquisition by TV camera.

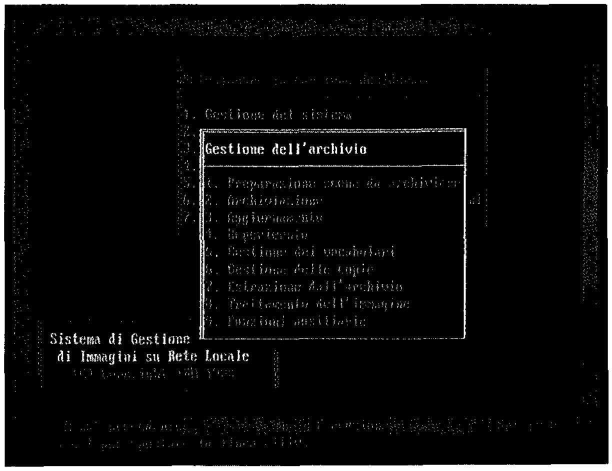

Figure 8

Archive management menu.

the archive (**Figure 8**): *image preparation, filing, retrieval, updating*, and *extraction.*

Image preparation Corrections can be made on the digitized images to increase their fidelity or to fit specific user requirements. Geometric and chromatic functions are available for this purpose. The former operate on spatial parameters (cutting, mirroring, rotation, and zooming); the latter change the chromatic levels. Having obtained the best version of the source image, the system automatically generates two other versions at a predetermined resolution, the *index image* (120 × 120 pixels) and the full-screen image (512 × 420 pixels, which can be displayed completely by many display devices). Each of the three images is then converted to 8 bits per pixel. The availability of both 24- and 8-bit images allows the system to be used for many different applications.

To complete the cluster structure, the system prompts the user to insert the following image-descriptive information:

- A *document* associated with the images.
- An *abstract* of the document.
- *Filing information* (name, owner, date, . . .).

Document and abstract lengths are not limited, and the texts may be written in current natural language.

Filing The images and the associated descriptive information are entered into the archive. When the filing operation starts, after a number of checks on the database and selected storage devices, the images are stored on the chosen devices through the LAN, while the descriptive information is added to the relational database. Image data are usually stored on third-level devices unless the user specifies otherwise. In any case, their location is transparent to the user during the retrieval phase. For the bulk input of image data, the system provides a batch filing modality. In this case, the system does not require interactive entry of descriptive information by the user; the descriptive data are taken from a specific file. Using this information, the system carries out the preparation and filing of one or more sets of digitized images.

Retrieval This environment allows the user to retrieve information from the archive without knowing the exact location of images or data and without needing to enter descriptive information directly. Textual information (abstracts and documents) can be accessed in the relational database by a "free-text search" mechanism; a thesaurus is activated at the user's request. Queries are formulated by defining conditions on the three classes of text information (documents, abstracts, and filing information). For retrieval based on filing information, the system selects the cluster for which the filing-information attributes satisfy values specified by the user.

For documents and abstracts, a free-text search is performed on the basis of the words or portions of words entered by the user. Logical operators (and, or, not, etc.) can be used to connect items in queries, and relational operators (=, >, <, etc.) can be used for numeric data (**Figure 9**). From this user-generated query, the system derives a definition of the correct sequence of words and operators. The query result is a set of images, or descriptive data, or both, for the selected clusters.

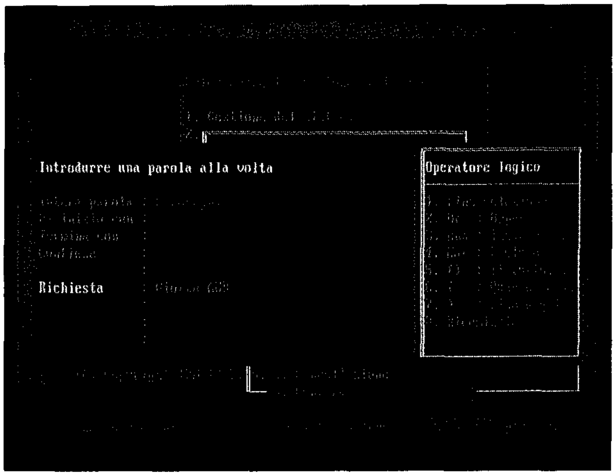

Figure 9

Query on text.

If images have been requested, a 4×4 board menu is displayed on the alphanumeric screen with the names of the selected clusters; a complete set of scroll functions is also available (**Figure 10**). Since names generally are not enough to identify a cluster, the system displays on the color screen the sixteen index images associated with the cluster names on the alphanumeric menu (**Figure 11**). The index images exhibit a much richer content for human perception than names do; in this way the user can use visual perception and his awareness of the appearance of the desired images to navigate the selection process.

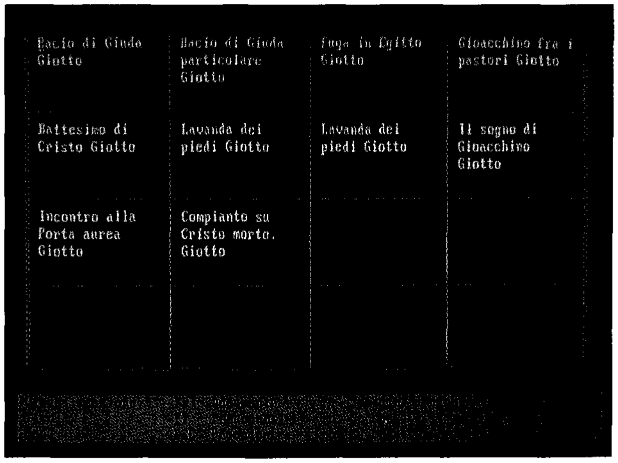

Figure 10

Menu on the alphanumeric screen.

At this point user interaction can follow several paths. By browsing the index images, one can select a cluster and display the associated full-screen image with its related texts and attributes (**Figure 12**). Alternatively, another menu allows the user to select a portion of the high-resolution image of the cluster, so that details of the subject can be inspected. A simultaneous display of the cluster-index image in the bottom left corner of the screen gives a reference for the position of the details inside the image as a whole. Further, additional portions of the same image or parts of different images can be displayed at the same time, so that comparisons can be made.

Updating If descriptive data and images themselves are entered incorrectly or become obsolete, updating functions are available to modify information without

Figure 11

Related index images.

knowing the database structures. The user searches for information he wants to modify in the same way as for retrieval; then he uses the updating functions to make the necessary changes. The information in the database is updated and becomes immediately available.

Folder management The system provides functions to manage folders, which are collections of clusters. The folder is described by some textual information. This structure allows an alternative way of access to the images, which can also be selected by retrieval conditions formulated on the information associated with the folders.

Extraction It is possible to create public or private copies of the images. Public copies are used when a subset of images in the archive must be frequently accessed. These copies are visible to all users, so that when an image is reidentified by a query, the system reports the existence of the copy to the user workstation and provides all information needed for the access. Private copies can be created by a single user so that the system will make them visible only to their owner. The database contains all the information needed by the system to perform these operations, so that storage management is completely transparent to the user.

Third-level storage is recommended for public copies, because it permits the fastest access to images extracted from the main archive. Private copies are best stored on the first two levels; in this case faster access to the images is achieved by using local devices and thereby reducing traffic through the LAN. To select images to be copied, the user makes a query very similar to that used for retrieval and display, but specifies the secondary storage device. During retrieval, the system produces the fastest available copies, based on the current allocation of resource to users and the current storage configuration of the archive.

Data migration functions are also available to transfer all the images stored on one or more storage devices to one or more other devices.

Processing The system supplies two classes of basic image processing functions—geometric handling (mirroring, zooming, cutting, rotating, etc.), and chromatic handling (modification of brightness, contrast, chromatic components, conversions, etc.). The execution of each processing operation is considered to be a working phase, and its results are stored in a list structure using coding techniques.

Restitution Both scanners and printers are used as system output devices. A dedicated personal computer drives an output scanner, which can plot on film the data taken from system storage. Remote spooling is available, and image hard copies can be produced on high-resolution printers. Furthermore, since all image files are standard binary files, they can be used by any other application.

The prototype is a general-purpose system which, because of the modularity of its architecture, can easily be customized for specific applications.

Remarks and conclusions

The IDB system exploits images as a vehicle of interaction with the user; index images play a fundamental role in completing the selection of images from the archive. The architecture is characterized by modularity and flexibility; each single module is related to a specific task to be performed during the image management process. Functions have been integrated by distributing resources among the nodes of a LAN; each node corresponds to a workstation, and many users can work with the system. The user interface makes transparent the management of communication and storage resources—at any moment it shows the set of available functions and how to work with them.

The prototype was the subject of a joint study between IBM and a major Italian publisher. The joint study implied some specific extensions to the database to manage further image attributes, and some minor changes to the user interface. Since the joint study was successfully completed with end-user satisfaction, the system was announced as a Country Program Offering (Program Number 5788-GHY) for Italy in December 1990.

Future extensions will involve the integration of new kinds of information such as audio data and image animation. A hypermedia approach is also being evaluated.

Figure 12

Full-screen image.

References

1. G. Gardarin and E. Gelembe, *New Application of Data Bases*, Academic Press, Inc., New York, 1984.
2. S. K. Chang, "An Intelligent Image Database System," *IEEE Trans. Software Eng.* **14,** No. 5, 681–688 (1988).
3. R. A. Bartolini, "Optical Recording: High-Density Information Storage and Retrieval," *Proc. IEEE* **70,** No. 6, 589–597 (1982).
4. A. E. Bell, "Optical Data Storage: A Status Report," *Proceedings of Conference on Mass Storage Systems*, Vail, CO, June 1984, pp. 93–99.
5. S. K. Chang, "Image Information Systems," *Proc. IEEE* **73,** No. 4, 757–764 (1986).
6. A. J. Duerinckx, "Digital Picture Archiving and Communications Systems in Medicine," *Computer* **8,** No. 3 (1983).
7. S. Cavaliere, M. Fantini, and A. Turtur, "An Integrated System for Printing and Publishing Applications," *Computed Generated Images*, Springer-Verlag New York, 1986.
8. G. Nagy, *Image Data Base*, Vol. 3, No. 3, Butterworth and Co., Boston, 1985.
9. J. F. Amus, "Digital Image Processing in Art Conservation," *Byte* **12,** No. 3, 151–165 (1987).
10. A. Turtur, F. Prampolini, and M. Fantini, "Integrated Image Management on a Local Area Network," *Proceedings of RIAO 88 Conference*, March 1988, pp. 1072–1084.
11. E. F. Codd, "Extending the Data Base Relational Model to Capture More Meaning," *ACM Trans. Data Base Syst.* **4,** No. 4, 397–434 (1979).

Received November 17, 1989; accepted for publication December 18, 1990

Antonio Turtur *IBM Rome Scientific Center, via Giorgione 159, 00147 Rome, Italy.* Mr. Turtur is a Senior Research Staff Member at the Rome Scientific Center. In 1974 he received his Master's degree in computer science from Pisa University and joined the IBM Bari Scientific Center, where he worked on the design and development of relational database and formal query languages. In 1979 he joined the IBM Rome Scientific Center and was involved in extending a relational database management system for territorial applications. Since 1982 Mr. Turtur has worked on digital image applications, first on designing and developing an integrated system for a publishing environment, then on designing an integrated image database system. He is currently involved in designing a multimedia database system. Mr. Turtur is a member of the Association for Computing Machinery and the Institute of Electrical and Electronics Engineers.

Franco Prampolini *IBM Rome Scientific Center, via Giorgione 159, 00147 Rome, Italy.* Mr. Prampolini received his Master's degree in electronic engineering from the University of Rome "La Sapienza" in 1984. Since then he has worked on systems for digital image management at the IBM Rome Scientific Center. At present Mr. Prampolini is designing and developing a multimedia database; he is also interested in medical image database systems. He is a member of the Association for Computing Machinery and the Institute of Electrical and Electronics Engineers.

Mario Fantini *IBM Rome Scientific Center, via Giorgione 159, 00147 Rome, Italy.* Mr. Fantini received his Bachelor's degree in 1970. Since 1979 he has worked at the IBM Rome Scientific Center in the area of image processing, where he took part in the development of the IBM 7350 Image Processing System. Since 1983 Mr. Fantini has worked on digital image applications, first on developing an integrated system for a publishing environment, then on developing a system for computer art. He is currently working on image database systems.

Roberto Guarda *IBM Rome Scientific Center, Via Giorgione 159, 00147 Rome, Italy.* Mr. Guarda received his Master's degree in electronic engineering from the University of Rome "La Sapienza" in 1987. Since 1988 his work at the IBM Rome Scientific Center has involved the design and development of an integrated image database system.

Mario A. Imperato *IBM Rome Scientific Center, via Giorgione 159, 00147 Rome, Italy.* Mr. Imperato received his Master's degree in electronic engineering from the University of Rome "La Sapienza" in 1989. Since then he has worked on image processing at the IBM Rome Scientific Center. Mr. Imperato is currently involved in the development of medical image database systems.