MAGMA (1994) **2**, 551–558

Use of 24-bit false-color imagery to enhance visualization of multiparameter MR images

Francesco Beltrame,[1,*] Luisa Cotta,[1] Marilena De Ceglia,[1] Marco Fato,[1] Giampiero Marcenaro,[1]
Davide Caramella,[2] Massimo Del Sarto,[2] and Irwin Sobel[3]

[1] *Department of Communication, Computer and System Sciences, University of Genoa, Genoa, Italy*
[2] *Department of Diagnostic Radiology, University of Pisa, Pisa, Italy*
[3] *Hewlett Packard Laboratories, Palo Alto, USA*

Biomedical image analysis workstations can be linked to 3D data-oriented devices for a new approach to image manipulation in biology and medicine. Stereo monitors allow an intuitive approach to medical diagnosis. The use of 3D head-tracking devices allows a more compelling 3D illusion to be generated. A stylus can be used as an electronic knife for dissecting a 3D data set; furthermore, other 3D sensors are available for tracking operator arm movements. The overall character of this work is firmly application oriented, in order to provide concrete operational tools to the medical user. Such tools range from diagnostic up to therapeutic and robotized use of bioimages.

Keywords: bioengineering, bioimages, 3D reconstruction, stereo vision, false-color composition in MR, clustering.

BIOIMAGE-ORIENTED INTERFACES: A CASE STUDY FOR STEREO VISUALIZATION IN MAGNETIC RESONANCE

The reduction of costs for computing and the increase of workstation performance allows for the exploration of new application fields for advanced informatic technologies, including medical applications. Bioimage processors, typically based on workstations, can be interfaced with devices for pointing at 3D data and for their visualization using stereo monitors, thus allowing a new approach to medical diagnosis [1, 2]. The operator works with data representing the whole body structure [3], overcoming the classical drawback of radiology, that is, the two-dimensional nature of data. This environment allows the creation of a spatial manipulation system in a 3D object space and allows for the interactive navigation of the operator through this space. Sometimes, such environments are named "virtual reality" or "augmented reality" systems [4–6].

As an example, consider MR (magnetic resonance) images. In normal clinical practice, MR images are generally used for qualitative examination. The aim of our study has been the quantitative spatial analysis of MR data using the three fundamental weighted images: PD (proton density), T_1 (longitudinal relaxation time, spin–reticulum), and T_2 (transverse relaxation time, spin–spin).

Starting from three 2D MR images, of 256 × 256 pixels (each pixel represented by 12 bits), at the same slice location in a given patient, a new single image representation of all three parameters has been generated by using the false-color technique on a HP 9000/730 workstation in a standard UNIX and X-11 environment. A transformation linking together the MR parameters and the RGB (red, green, blue) color components has been used. In particular, in our study, PD corresponds to green, T_1 to blue, and T_2 to red, respectively. The operator has several interactive controls for modifying the false-color compositing process. He/she interacts with the display shown in Fig. 1. The image in the center is the composite result of the mapping of the three parameters by means of false colors. It is displayed using a resolution of 24 bits per

---

* *Address for correspondence: Department of Communication, Computer and System Sciences, University of Genoa, Viale Causa 13, 16145 Genoa, Italy.*

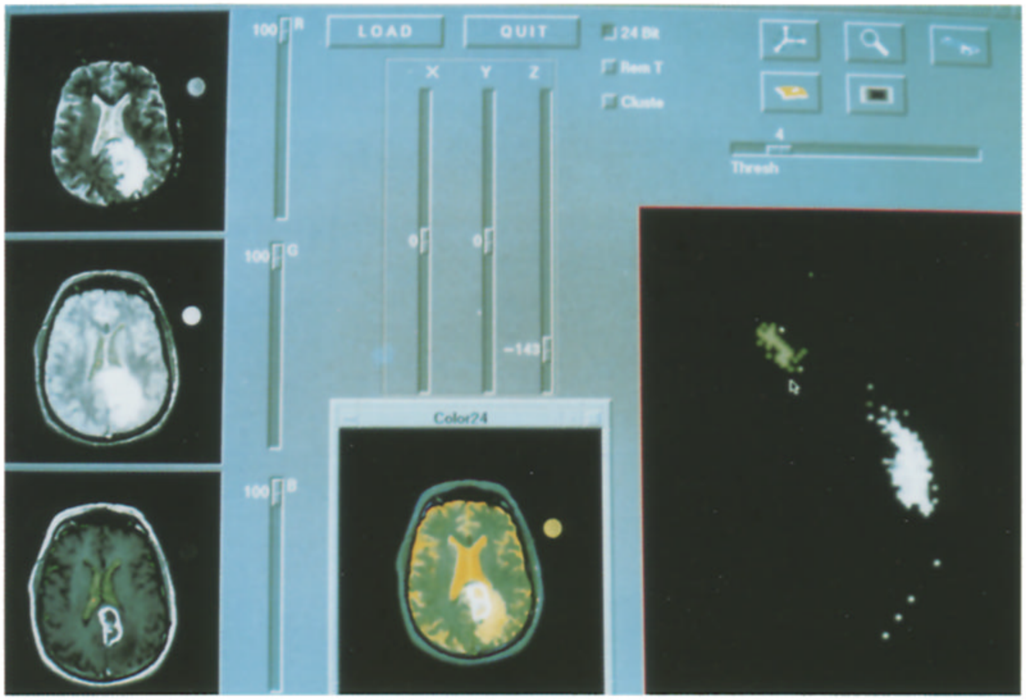

Fig. 1. Layout of working environment for multimodal MR image processing. On the left (top to bottom) are T_2, PD, T_1 (gadolinium-enhanced) MR original images. Center bottom is the color composite image. On the bottom right is a view into the 3D histogram (PD, T_1, T_2).

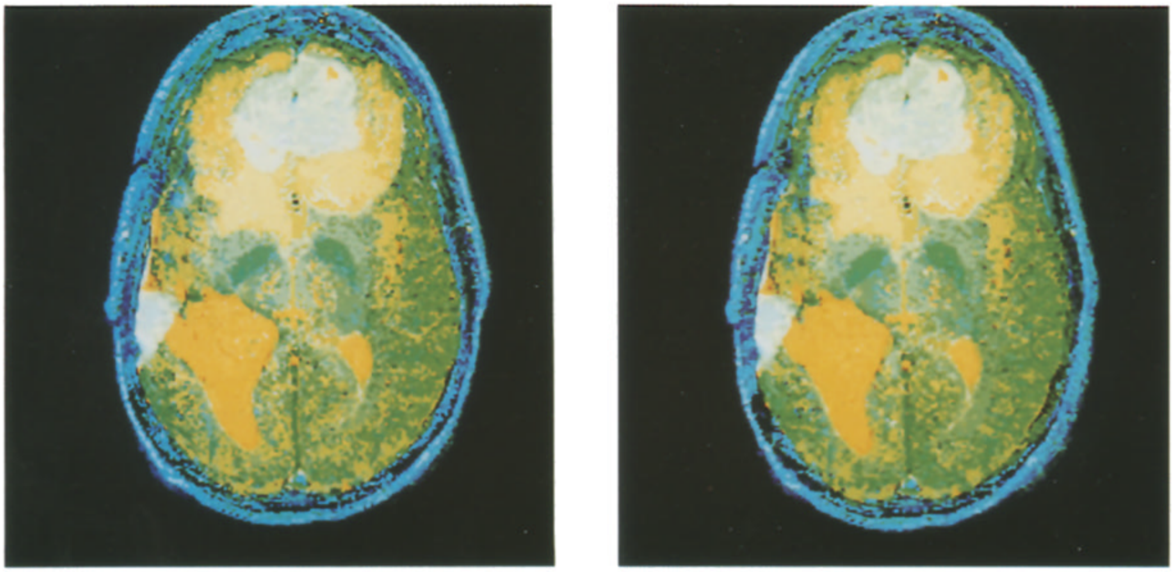

Fig. 2. Stereo pair made using 3D direct volume rendering of a false-colored brain volume starting from a sequence of 27 2D MR slices, 3 parameters per slice, composited into false color. Certain tissues have been made transparent, allowing a stereo view into cavities created. (Data set courtesy of Dr L.P. Clarke and Robert Velthuizen of University of South Florida and the H. Lee Moffitt Cancer Center and Research Institute.)

pixel via an HP CRX24 graphics board. This results in an image with high color definition which is desireable in order to better identify regions of interest for diagnostic purposes in radiology.

The operator may independently vary the mix of each of the three components in the false-color composite via three (R,G,B) mixing sliders. Each slider specifies a percentage of its associated component to be included in the composite. The operator can also do proportional "black-clipping" of the low-intensity parts of the three components via the horizontal "Black Clip Percent" slider above the composite image. Furthermore, by using the pixel data in the PD, T_1, T_2 component images, a three-dimensional space containing the density distribution of these three parameters has been constructed.

This 3D histogram space is shown in Fig. 1 (lower right side). It is constructed as follows: on the three 2D images (Fig. 1, left part), each coordinate pair (x, y) identifies a specific data triplet: PD, T_1, T_2. These data values are the coordinate indices of a 3D histogram. The values of the histogram identify the number of pixels with the same PD, T_1, and T_2. Memory requirements are critical in the realization of such a data structure. In fact, because each pixel is represented by 12 bits, in theory each coordinate of the histogram may range between 0 and 2^{12}. In a 3D space, the maximum dimension of the scatter diagram will be $2^{12} \times 2^{12} \times 2^{12}$ pixels. As each pixel has been defined as a "short int" (two bytes), the image would occupy up to $2^{36} \times 2 = 2^{37}$ bytes. In order to reduce memory requirements, only six bits of each R, G, and B channel have been considered. These are generated in three steps for each component: (a) the 12-bit data is (1D) histogrammed and mapped into an 8-bit range after clipping 1% off the high-intensity (white) tail of the distribution; (b) this 8-bit scaled component data may then be interactively attenuated by the mixing sliders and/or black-clipped as described above; (c) the high-order 6 bits of each processed component are extracted to yield an 18-bit address into a histogram array of 4-byte 'unsigned int' values. In this case, the scatter diagram requires a memory space equal to

$$2^6 \times 2^6 \times 2^6 \times 2^2 \text{ bytes} = 2^{20} \text{ bytes} = 1\text{MB}.$$

The histogram (scatter diagram) has been displayed in stereo: the stereo monitor allows the three-dimensional rendering of visual data through LCD (liquid crystal display) shuttered glasses (Crystal Eyes by StereoGraphics, Inc., USA). Each eye shutter alternately opens and closes for half of a 72-Hz stereo cycle while the graphics board synchonously displays a 144-Hz sequence of left- and right-eye views. Using the mouse in this 3D space in a interactive way, it is possible to define regions of interest (ROI) in the tissue space by highlighting the anatomical zones corresponding to a given data cluster (i.e., color range). In addition, a sort of inverse operation can be performed, that is, to define some point of interest in one of the component parameter images, and see the data cluster containing its PD, T_1, and T_2 values highlighted. This latter operation is more easily understandable to a clinician; that is, it can be explained as equivalent to asking the request "show me all pixels having similar tissue parameters as the one I am pointing at" the criterion for similarity being membership in a given data cluster.

We have what we think is a novel technique (patent applied for) for defining a data cluster; namely, we treat the histogram function as a data density function and apply a threshold T to it to define a set of histogram locations where $H(\text{PD}, T_1, T_2) \geq T$. This set, in general, will consist of several distinct connected components. These components are our operational definition of data clusters, that is, all points in a given component are considered "similar." Moreover, the user of the system can interactively vary the threshold T and view its effect on the histogram clusters. Raising T will cause the clusters to shrink, in a manner similar to peeling layers off an onion. Conversely, lowering T will cause the clusters to expand.

This kind of approach offers an easier interpretation of MR data and a clearer distinction between normal and pathologic tissues, allowing an immediate visual evaluation of the parameters of MR acquisition systems. The user can display in the same application the scatter diagram, the component parameter images, and the false-colored composite tissue image. In such a way, it is possible to interact with all the available images, simultaneously reasoning about both qualitative and quantitative aspects.

Figure 2 shows a stereo pair which is an extension of this technique for a stack of brain slices making up a complete 3D tissue volume. This image is from the Group of University of Southern Florida and the H. Lee Moffit Cancer Center and Research Institute. It is from a 28-year-old white male with three inoperable dural-based tumors. Nine months prior to the MR scan, a ventricular meningioma had been resected. The site of the resection can be seen as a dilation of the left lateral ventricle. Radiation therapy was completed 3 months before the MR scan and was followed by chemotherapy. The growth of the frontal lesion suggests this is a meningeal sarcoma. The slice thickness of the MR scans is 5 mm. The original data was three volumes of 27 slices, 256 × 256 × 12 bits each. These were processed as follows:

- each 12-bit component volume was histogrammed and scaled into 8 bits by
  - white-clip set by clipping 1% of the picture area of histogram tail
  - black-clip set to 30% of the resulting range
  - the reduced range (black-clip, white-clip) linearly mapped into 8 bits;

- the resulting volume at 24 bits per voxel was reduced to an 8-bit per voxel volume and specific color palette by
  - choosing an initial palette consisting of the 256 most popular buckets in a histogram of the high-order 12 bits of the raw voxels (4R, 4G, 4B)
  - using the mean value of voxels in each bucket to represent each bucket
  - assigning voxels not lying in one of the chosen buckets to the nearest one (in color space) and updating the resulting bucket mean accordingly;
- the resulting 8-bit paletted volume was then fed to the new ISG/IAP paletted color volume renderer, no *z*-interpolation was done, scaling was assumed to be 4 mm per slice contiguous;
- a default opacity function was used—a ramp (0,0) → (255,255). However the ramp is applied to the raw palette indices which have no intuitive color order (see description above of paletting). The result of this is that arbitrarily chosen tissues have been made transparent, creating cavities in the stereo image into which the user can peer.

The intent is to add the 3D stereo histogram display to this application. This would allow simultaneous 3D visualization of false-colored tissue space and histogram space. Moreover, in contrast to Fig. 2, the histogram cluster-selection technique can be used to rationally set the opacity of a cluster of interest to opaque (and everything else transparent) or alternatively transparent (with everything else opaque). This allows us to create a very powerful interactive virtual dissection tool.

Figure 3 shows a single stereo pair from a volume-rendered cine-stereo sequence for flow-enhanced heart imaging. This is a monochrome data set acquired by a group at Picker, Inc. consisting of 10 volumes spaced in time over a complete heart-cycle. Each volume consists of a sequence of 49 2D MR slices of a normal human thorax. For each given user azimuth angle (i.e., about the vertical axis), all 10 volumes were rendered into a cine loop of 10 stereo pairs. The resulting cine-stereo display was quite compelling and informative—several radiologists including at least one cardiac angiographer seemed interested enough to examine at it quite closely for about 10 min.

BIOIMAGE INTERACTION TOOLS

Using a stereo visual system [7], it is necessary to choose a highly dependable and accurate 3D locating tool. For this purpose, a 3D control system based on low-frequency magnetic fields (Polhemus Fastrak, by Polhemus, Inc.) has been used. Figure 4 shows the experimental assembly.

The Polhemus is used for recording the movements of the head of the operator with respect to the stereo

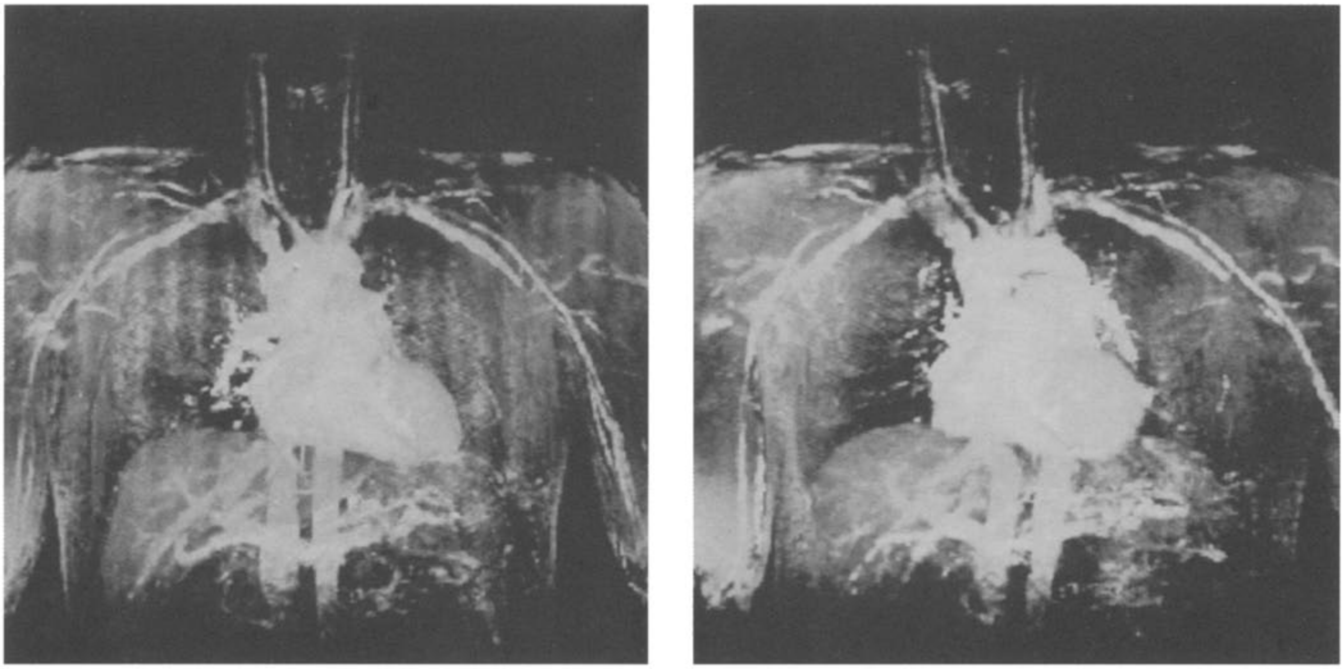

Fig. 3. Three-dimensional rendering of one stereo pair from a 10-frame stereo cine-loop of a reconstructed heart starting from sequences of 2D MR slices. (Data courtesy of Dr. Paul Margosian, Picker Inc, Cleveland, Ohio.)

Fig. 4. Augmented reality environment for the user.

display, by means of a head-tracker, permitting input of the user's head position to affect the video image. The head position is used to map the user space into the object space. In this way, a head motion of the user is associated with a corresponding rotation of the 3D object on the display. If the renderer used is fast enough, the result is a compelling illusion of 'looking around' a 3D object suspended in front of the user.

A hand-held Polhemus stylus can represent an "electronic knife" with which it is possible to simulate a surgical operation. The operator, using the stylus, can simulate a surgical operation by moving clipping planes in the volumetric data displayed by the computer. The union of head-tracking and stylus manipulation (together with stereo vision) offers a promising environment for planning and simulation of surgical interventions, useful for training purposes or for presurgical analysis.

APPENDIX

IAP—The Application Development Environment

The software interface, in our case, is based on the IAP (image application platform) (ISG Technologies, Toronto, Canada). It is possible to make the following operations on each image:

- Zooming for a better resolution of details
- Cutting for deleting tissues
- Filling for adding tissues.

All these operations and others can be performed interactively.

The IAP is designed to satisfy a wide range of applications. However, it should be noted that development has been driven by experience in medical imaging and, to some extent, imaging problems in the life sciences. At the most general level, IAP supports imaging applications in the three areas of database, processing, and hardcopy. These three functions are embodied in three separate servers which form the core of the IAP implementation. These three pieces are separated to allow application writers the freedom to use one piece without the others. Because IAP is a client/server architecture, the use of any component requires compatible versions of servers and libraries.

The IAP server processes have specific names: "prserver," "hcserver," and "dbserver." The packages of client libraries and servers are given other names:

- PrS is the processing server (prserver) plus its client interface libraries.

- HcS is the hardcopy server (hcserver) plus its client interface libraries.
- DbS is the database server (dbserver) plus its client interface libraries.

In addition to these three pieces, IAP includes some support tools (demo applications) and some libraries which are common to all servers. Some of the important requirements that led design include the following:

- Applications built on IAP should be able to coexist on the same screen with other applications. Due to the emergence of X-11 as the standard windowing system of UNIX, IAP should be able to work with an X-11 server.
- The architecture should maximize application independence from the hardware resources. This refers both to the task-porting applications to different hosts and to the transparent use of remote and local computational resources (e.g., a fast workstation or an attached accelerator). The use of such resources should be configurable at run time.
- The architecture should allow a range of processing operations on 2D and 3D images, with supports for a fourth (e.g., time) dimension as well. These tasks include 2D operations such as intensity remapping, zoom, pan and rotation, and filtering. The 3D operations should include volume rendering, surface rendering, reformats, and measurements.
- The processing algorithms should provide run-time trade-off of speed for quality under application control.
- The implementation should provide mechanisms for common tasks in medical applications such as annotation, regions of interest, and coordinate transformations.
- Although the obvious choices for user interfaces are all constructed on X-11, IAP should remain independent of the user interface, so that it does not rely on any specific UI toolkit, and even the use of X-11 should not be mandatory.
- The database should support the ACR–NEMA naming conventions and make it easy to import such data, including the ability to handle arrays and unknown attributes.

IAP is essentially an object-oriented programming system. From the programmer's point of view, an IAP client process creates objects, sends messages, receives events, and destroys objects. The PrS part of the system is programmed in a dataflow-like language wherein the objects transform input data types (called "protocols") into output data types. The objects and their dataflow interfaces are instantiated and syntactically checked via remote procedure calls (RPCs) to the processing server (prserver).

These concepts can be used, for example, to implement a simple image display pipeline, consisting of three major objects: a 2D raster object, an image viewing object, and a "screen" display object, along with 2D Transform object for positioning and a Pixval object to specify an intensity look-up table (LUT). The programming can be schematized by the dataflow diagram in Fig. 5.

This "program," when supplied with image input, positioning transform, and look-up table, will display the image in the specified position with the specified intensity transformation in the Xwindow or Motif Widget to which it is programmatically "connected" to. Typically, inputs to the Trans2 and PixVal objects will be interactively changed by the user, making even this simply specified application a useful tool.

Figure 6 shows the set of fundamental objects and their possible interconnections (from the PrS user's manual).

The Polhemus Tracking System—Hardware

The Fastrak Polhemus tracking system uses electromagnetic fields to determine the position and the orientation of a remote object. The technology is based on generating near-field, low-frequency, magnetic field vectors from an antenna called the transmitter (Xmtr) and detecting the field vectors with a single assembly of three colocated, remote sensing antennas called the receiver (Rcvr). The sensed signals are input to a mathematical algorithm that computes the receiver's positions and orientations relative to the transmitter.

The Fastrak consists of a System Electronics Unit (SEU), one to four receivers, and a single transmitter. The system is capable of operating at any of four discrete carrier frequencies. Different carrier frequencies allow operation of up to four Fastraks simultaneously and in close proximity one to another. The Fastrak has two possible interfaces to the host computer: RS-232 and IEEE-488. Any single receiver may be operated at the fastest update rate (120 Hz), any two receivers at one-half of this rate, any three at

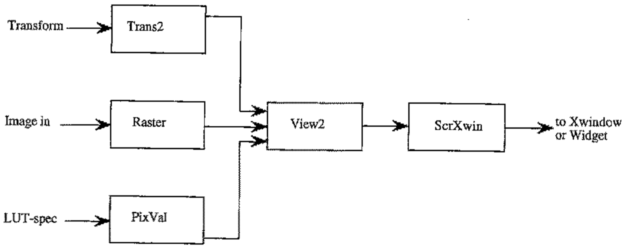

Fig. 5. Dataflow diagram.

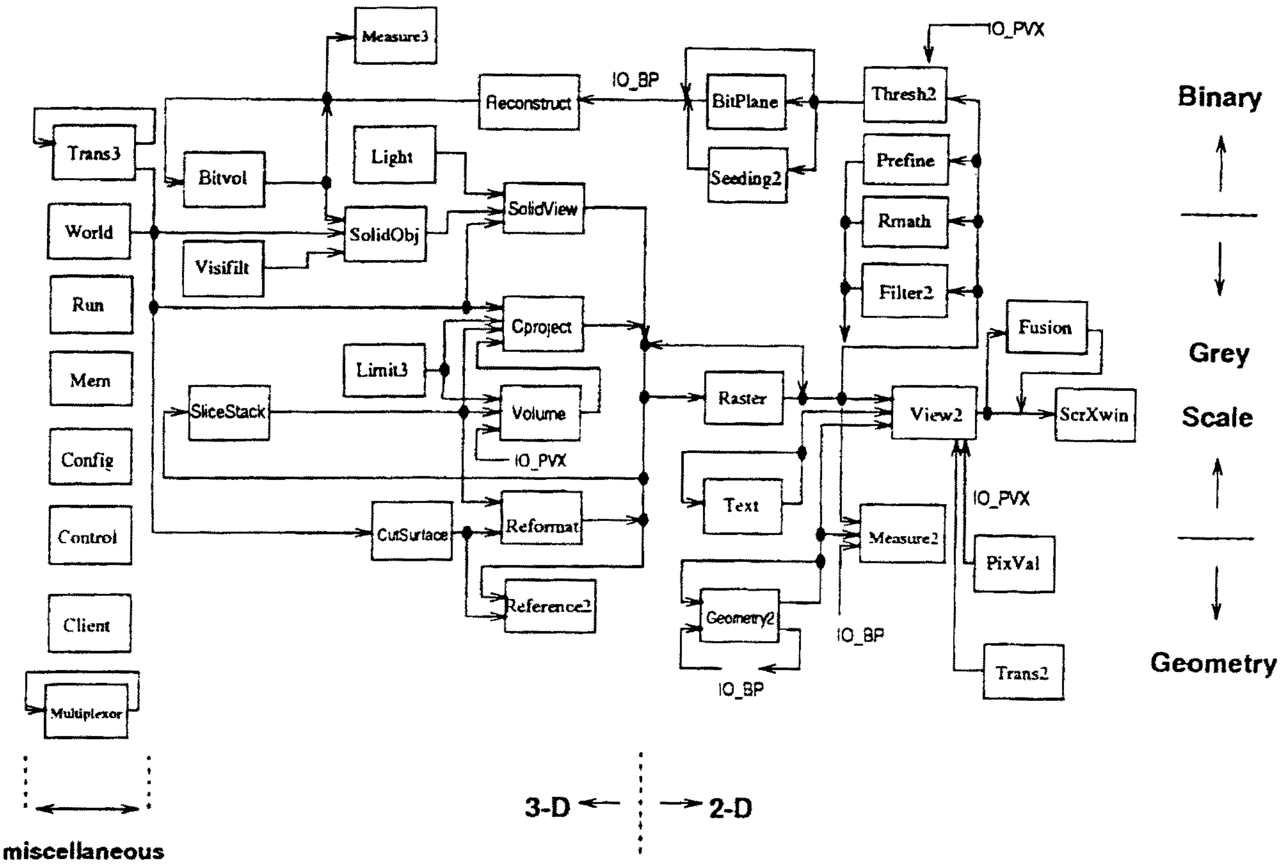

Fig. 6. Object Classes Diagram (IAP).

one-third of this rate, or all four at one-fourth the fastest rate. Mixed rates are not permitted, meaning that all receivers operate at the same update rate: one cannot be operated faster than another. Active receivers are selected by a combination of software configuration commands and receiver selector switch settings.

The instrument will provide the specified accuracy when the receivers are located within 30 in. (76 cm) from the transmitter. Operation with separations up to 120 in. (305 cm) is possible with reduced accuracy. The static accuracy is 0.03 in. (0.08 cm) RMS for the X, Y, Z receiver position, and 0.15 in. RMS, while the resolution is 0.0002 in./in. of range (0.0005 cm cm^{-1} of range), and 0.025°.

Additionally, the Fastrak may be used with a stylus as a receiver. Tip offsets are automatically calculated and no special commands are required for this mode of operation. However, button functionality is not provided and must be simulated via a keyboard. The stylus can be used as a "virtual knife" for interacting with the biomedical images. For example, it allows for choosing suitable cutting surfaces for surgical planning.

The Polhemus Tracking System—Software

A C language program provides the serial communication interface in the UNIX/X-11 environment.

Beside the hardware tools, software tools for the visualization of the movements of the stylus in a 3D stereo space have been developed. With the aim of providing a more realistic representation of the stylus, two graphic objects have been defined: a line with a square-based pyramid to represent the tip. These are part of an ad hoc graphic library.

The software package contains a set of primitives and functions written in C language for perspective and orthographic projection. Objects may be chosen from an available set (stored in a single file) or defined *ex novo* according to user's needs. The graphic objects are defined as a set of surfaces in coordinates relative to the center of the image. The primitives of translation and rotation operate on these coordinates to position the displayed object. The coordinates can be updated quite rapidly by reading data from the Polhemus. Subsequent calls to the procedures for computing and displaying the object allow for smooth interactive manipulation.

A second application has been realized in the IAP environment. In real time (as described above), the computer tracks the position of the head of the user with a sensor fixed on a helmet or on the stereo glasses. Using this kind of software and hardware architecture, the stylus can be used for lighting a distinctive area of the image and the head-tracker input is used to update the view of the object displayed. This application allows the user to choose a part of the image utilizing the stylus—the stylus, in this application, works as a 3D mouse. The user positions the stylus in the interesting area and clicks the "light" button in the list of tools (available in IAP) making the IAP "light" object available. The light object is connected via software with the stylus, which represents a virtual brightness source and lights the interesting part of the image.

Because the stylus work space is the whole surrounding environment and not the image space, a calibration procedure is necessary to map onto the display space. The algorithm is related to the size and location of the window on the screen as follows. When the window is displayed, the user aims the stylus at two diagonally opposite corners of the rectangle to find the center of the window via a simple calculation. Thus, the starting coordinates of the axes of the sensor are virtually moved from a point outside the screen to a central point of the window area. In this way, it is possible to obtain a central light source. The Polhemus is also used in stereo visualization. The software provides graphical computation to update the point of view at a fixed rate. This algorithm allows for a change of 35–40° in the view angle with which the image is displayed. The process makes possible the visualization of otherwise hidden parts. It works as follows. The system gets the position and the angles of the observer with respect to the visualized image. These data are processed and drive the primitives of the visualization program, continuously updating viewing transform. The apparent rotation of the image yields a more realistic 3D rendering. The sampling rate needs to be chosen in order to satisfy two fundamental requirements:

- the image update interval should be short enough to maintain the illusion of an object floating in space in front of the viewer (e.g., ≤ 100 ms);
- on the other hand, the image update interval should be long enough to permit complete calculation of high-definition images.

Thus, there is a trade-off between image update speed and image quality. This gives a strong incentive for obtaining a fast, high-quality, volume-rendering engine.

Because the renderer is the bottleneck in the overall system, there is a related problem of how to filter the Polhemus head-tracking data in selecting the next viewpoint to render. Part of this problem is in synchronizing two separate UNIX processes: one for Polhemus data acquisition and one for image rendering. Two possible solutions have been envisaged. The first possibility is to process all the Polhemus data at whatever rate it comes in. In general, this will flood the system and not allow for real-time image updates; that is, storing up the data at a fixed frequency and processing when possible may cause nonacceptable delays between the real movement and image update. The second possibility is the creation of a handshaking protocol between the rendering and the data acquisition processes. In this case, reading of new data takes place only when rendering stops, thanks to a special signal that alerts the availability of the rendering process to new data. When rendering routines are still working, they do not query for new data and, therefore, none is read. This second solution, nevertheless, leads to a potential pitfall. The reading rate may be insufficient to track fast movements of the sensor. Skipping over many intermediate readings, the next visualization would lack continuity with respect to the previous one. Thus we are presented with the dilemma of either having good tracking but not in real time, or intermittent tracking in real time. The ultimate decision depends on the particular task requirements. Some compromise is possible by carefully filtering the data to be rendered in the second case.

REFERENCES

1. Delp SL, Loan P, Hoy M, Zajac F, Topp E, Rosen J (1990) An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Trans Biomed Eng* **BE-37:** 8.
2. Pieper S, Delp S, Rosen J, Fisher S (1991) A virtual environment system for simulation of leg surgery. *SPIE: Stereoscopic Displays Appl II* **1457:** 188–197.
3. Galyean TA, Hughes JF (1989) Sculpting: an interactive volumetric modeling technique. *Comput Graphics* **25:** 267–274.
4. Bajura M, Fuchs H, Ohbuchi R (1992) Merging virtual objects with the real word. *Comput Graphics* **26:** 203–209.
5. Blanchard C, Burgess S, Harvill Y, Lanier J, Lasko A, Oberman M, Teitel M (1990) Reality built for two: a virtual reality tool. *Comput Graphics* **24:** 35–36.
6. Brooks FP, Ouh-Young M, Batter JJ, Kilpatrick PJ (1990) Project GROPE: haptic displays for scientific visualization. *Comput Graphics* **24:** 177–185.
7. Fisher S (1982) Viewpoint dependent imaging: an interactive stereoscopic display. *Proc SPIE* 367.