IMEX: a tool for image display and contour management in a windowing environment

Peter H. Mills*, Henry Fuchs, Stephen M. Pizer, Julian G. Rosenman

Department of Computer Science, University of North Carolina
Chapel Hill, N.C. 27599-3175 USA

Abstract

Medical workstations of the future will support the real-time display and interactive manipulation of 3-D objects derived from CAT, MRI and other imaging modalities. As part of such an integrated system for visualizing 3-D volumes, we have developed an highly interactive, flexible, and portable program for the 2D display of image slices and contours outlining anatomical objects. The editing of these contours as well as their automatic creation by thresholding and edge-tracking is supported. The contours may later be used to generate 3D surfaces for shaded-graphics rendering, or to mask out regions of interest in the image for volume rendering. This "image executive" program, or Imex, is designed to run in a windowing environment (i.e., the X Window System). The user-interface model, which may be described as a "Macintosh [1] for images", associates one movable and resizable window with each displayed view of a 2D image slice or of an indexed array of slices. Any number of views into one slice or into a subset of an array of slices may be present. Natural interaction is achieved by providing immediate response and by using the mouse to effect navigation and viewing functions, for example grabbing and dragging a slice onto another to copy its field-of-view or other attributes.

1. Introduction

1.1. Visualizing 3D Volumes

The advent of 3-dimensional imaging modalities, such as X-ray CT (computed tomography) and MRI (magnetic resonance imaging), have heralded the acquisition of 3-D information about human anatomy. These technologies create a 3-dimensional array, or *volume*, of digital data by stacking a series of 2D images of cross-sectional slices covering part of the body. The problem is how to process and display the data so that their information content is made available to human perception in a form closely matched to everyday human perceptual experience. In medical applications, comprehension of the 3-D structure and interrelationships among internal organs and their surroundings is critical to diagnosis and therapy. Thus the issue becomes how to extract such information from the 3D scene and present it to the human observer so that he apprehends the 3D structure. Various techniques specific to medical imaging, to be discussed later, have been developed to extract object definitions from a 3D scene and to render realistic pictures of the objects' surfaces which preserve a sense of depth and three-dimensionality; methods for direct imaging without surfaces also exist. The importance of this 3D display of medical objects in medical research and clinical diagnosis, surgical planning, and radiation therapy is well established [Herm83,Gold85,Felli86,Udup86]. In this paper we will discuss *Imex*, one tool in the pipeline for interactive 3D display of medical objects which exploits the windowing environment of a conventional workstation.

1.2. Surface rendering

The simplest method of visualization is that of viewing massaged sets of the 2D cross-sectional slices which comprise the image volume [Budi84]. PACS (Picture Archiving and Communications System) research focuses on this issue in attempting to duplicate the capabilities of film-based radiology [Huan87,Temp85]; indeed, 2D viewing and navigation techniques thru a set of 3D slices comprise one function of the *medical image workstation* [Grew85,Pize88,Wend84]. Cross-sectional views from any oblique angle can also be formed and displayed [Glen75,Budi84].

*This research was supported in part by the National Institute of Health under grant number R01 CA39060.

[1] Macintosh is a trademark of Apple Computer, Inc.

However, since we are attempting to perceive 3-D objects in a 3-D (or 4-D) scene using 2-D display screens, computer graphics renditions of the 3-D anatomical or physiological structures of interest would seem to better present the information we wish to extract. The techniques that render a picture of the 3-D scene fall into two categories: *surface-shaded graphics* and *volume rendering*. In the former, the technique for presenting 3D structures is to use depth-shaded graphics on the surfaces of objects. However, this first requires the extraction of a surface description from the 3D scene. This two-step process of using image processing to reduce the scene into surfaces of objects, and then using computer graphics to project a $2\frac{1}{2}$D picture, involves the subtasks of:

- *object identification*: segmenting the image into regions of interest, and then identifying those regions that together form the object.
- *surface detection or formation*: fitting geometric primitives which approximate the surface of the detected object, and
- *rendering*: creating a picture from a specified viewpoint using the depth-queues of hidden-surface elimination, realistic depth-shading, and transparency.

Various mathematical representations of objects and surfaces are used: these may be partitioned into *surface patching schemes* or *solid modeling approaches* [Udup86].

In the surface patching scheme, the 2D borders of the region of interest are determined on a slice-by-slice basis, and approximated by contours. These contours may be manaully traced [Mazz76,Sung78] or automatically detected using any of a variety of segmentation techniques [Sung78,Vann83], for example by threshholding on intensity values to separate the desired object from the surrounding material [Dev84,Chen84]. In most cases both manual editing and automatic boundary-finding are supported. Semi-automatic boundary detection methods requiring user interaction also exist: the operator may trace an approximate outline around the region of interest, after which edge-detectors take over [Heff85]. Once the contours have been derived, they must then be connected to handle ambiguous branching situations when contours split or merge between slices [Pize86]. This may be done via user interaction [Chri78] or by a heuristic approach [Cook83]. A skin is then fit over the connected contours, either by *tiling* using triangular surface elements [Kepp75,Fuch77,Chri78,Gana82], or by using higher order surface patches such as cardinal splines [Sung78]. The surface may then be rendered using conventional graphics algorithms [Udup83].

Alternatively, in the *cuberille* approach, one regards the 3D volume of space scanned as being subdivided by three mutually orthogonal sets of parallel planes, forming a a 3D array of identical rectangular parallelpiped cells or *voxels* (short for *volume element*), each having a density value [Herm79]. Objects are defined as connected sets of voxels in a binary volume produced by segmentation. Boundary detection [Artz81] then yields a set of directed faces of voxels which approximates the real object's surface. Each solid voxel is thus treated as a small cube, and the polygonal faces of the cubes forming the boundary are input to the surface display algorithms, a number of which have been developed [Chen85]. The cuberille object may also be encoded using the *directed contour* representation, which describes the boundary on a slice-by-slice basis and provides greater manipulative flexibility [Udup82]. Another compact representation is that of the *octree*, which merges adjacent cubes [Meag82]. In yet another variation, the *marching cubes* approach estimates where the surface intersects the cube by looking at the sampled densities at cube vertices [Lore87].

1.3. Volume rendering

As an alternative to surface rendering methods, one may directly image the volume of data using a variety of algorithms. In these *volume rendering* techniques, there is no explicit use of a boundary representation of the object, so no data reduction or surface formation is prerequisite [Talt87]. Instead, all voxel samples participate in the shading and projection onto the picture plane. Segmentation into objects may be based on probabilistic rather than binary classification decisions: this allows display of weak or fuzzy surfaces [Dreb88,Levo88,Upso88].

1.4. Integrated medical image workstations

In addition to *visualization* using realistic pictures, the interdependent capabilities of *quantitative measurements* analyzing an object and *interactive manipulation* with immediate response are also needed in tools, such as those for surgery simulation [Udup86]. The aforementioned display functions, together with other manipulation and analysis techniques (e.g. volume measurment for surgical planning) may be integrated into an *medical image workstation*.

A variety of these workstations have been developed; some of the more notable are ANALYZE [Robb87], the Voxel Processor approach [Gold85], the MIPG projects [Herm84], and that of L. Fellingham et al. [Felli86].

1.5. Imex: an overview

In rest of this paper we focus on an interface model for the 2D display of sets of image slices and contours outlining anatomical objects. The editing of these contours and their automatic creation by edge-tracking is also supported. The image slices are typically derived from CAT or MRI data; each slice is a two-dimensional array of intensities, and each study is a set of slices. The contours may be later used to generate 3D polygonal surfaces for rendering on various graphics engines, or to delimit the regions of interest in the image for volume rendering. *Imex* forms one part of an integrated system being developed at UNC for object definition and interactive display of medical images using surface-shaded graphics and volume rendering [Pize86].

During the design the goals of portability, natural interaction using windows, and immediate response were deemed crucial. The program is built on top of X Windows software [Sche86], so it runs on almost any color workstation with a mouse. Interaction is based upon the Macintosh (actually Xerox) style: resizable and movable windows are provided, and dialog with the user is through panels of gadgets including push-buttons, scrollbars, and menus. Response and feedback is immediate; e.g., you may manipulate a scrollbar and see effects, with the program completing any outstanding display operations whenever it can.

Briefly, you may display an image (a set of slices) and edit sets of contours which are grouped into *anatomical objects*, each group being a different color. To do this you are allowed two types of views: a *panel of slice miniatures* and separate *views of individual slices*. You may have several panel views as well as several individual views of the same slice: every view is in a resizable and movable window. You may edit contours in each view: actions affect all views of the image. *Intensity-windowing*, linear remapping of the intensity values, may be performed on each view. Within each slice-view you may *zoom* in and out or alter the center point of display. Creation of individual views of a slice is done by merely grabbing the desired slice within a panel and dragging it onto an empty area of the screen; alternatively, dragging onto an existing individual view will replace it while preserving the target's zoom factor and center position, while dragging onto a panel miniature will copy the zoom factor and center position without altering which slice is displayed. The abstraction supporting these viewing and navigation functions is the *view-world paradigm*, which will be seen later to potentially encompass other navigation strategies.

Autocontouring, the automatic generation of contours, is done by edge-tracking on a binary-mask derived from lower and upper threshold intensity values on the image. These threshold intensities are set interactively with a scrollbar: those parts of the image within the threshholds will appear tinged with red instead of being displayed with the regular greyscale. Intensity windowing may be performed simultaneously with this manipulation. *Partial autocontouring* from existing open contours and *hand painting* of the binary mask is allowed. These actions may be combined orthogonally in any order with manual contour editing, yielding a powerful tool for defining anatomical objects through contours.

2. Image Display

2.1. Worlds, Views, and Windows

Before beginning a tutorial explanation of the navigation and viewing functions, a brief description of the underlying paradigm is in order. It is useful to think of a *study*, or one set of 2D image slices, as a *world* into which we are given several *views*. Each view will be displayed on its own *window* on the workstation. Changes made thru one view will affect all other views of that world: any change in contours in the study will be reflected in all views into it. This is a slight simplification of the definition of a world, but it will suffice for tutorial purposes.

2.2. Panel and individual views

Figure 1 shows *Imex* running on a Sun 3/60 color workstation, displaying one study consisting of a sequence of 2D slices derived from a CAT scan of a patient with cancer of the esophagus. In the upper right of Figure 1 appears a **panel view** of the study, an image index which consists of a rectangular display of a range of slices. You may scroll thru the miniature image slices using the scrollbar at the top of the panel view. Note that the miniature slices wrap around onto the second row in row-major order.

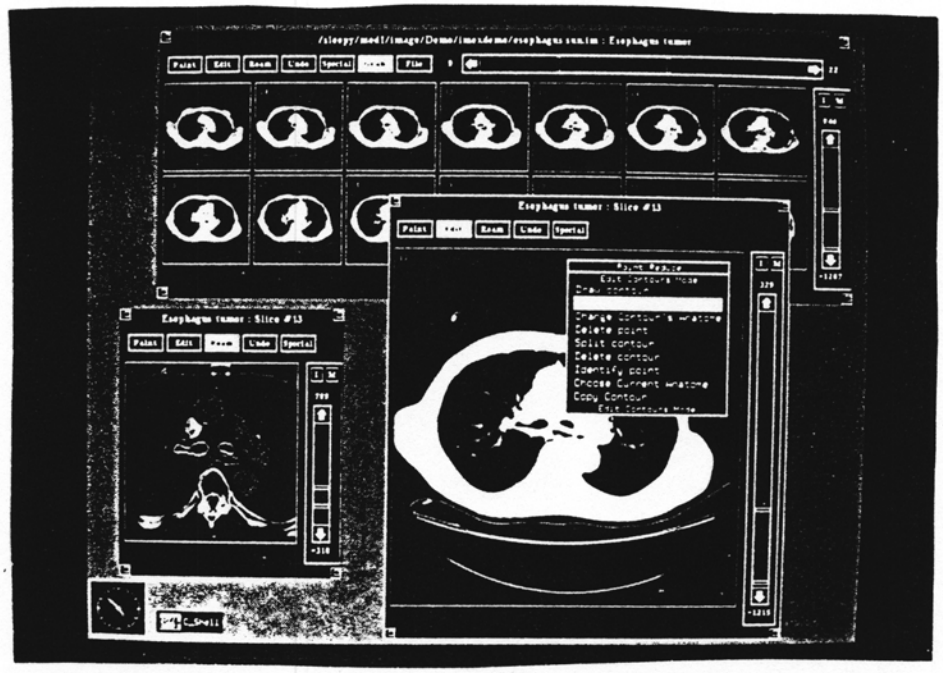

Figure 1: Imex running under X on a color Sun 3/60 workstation

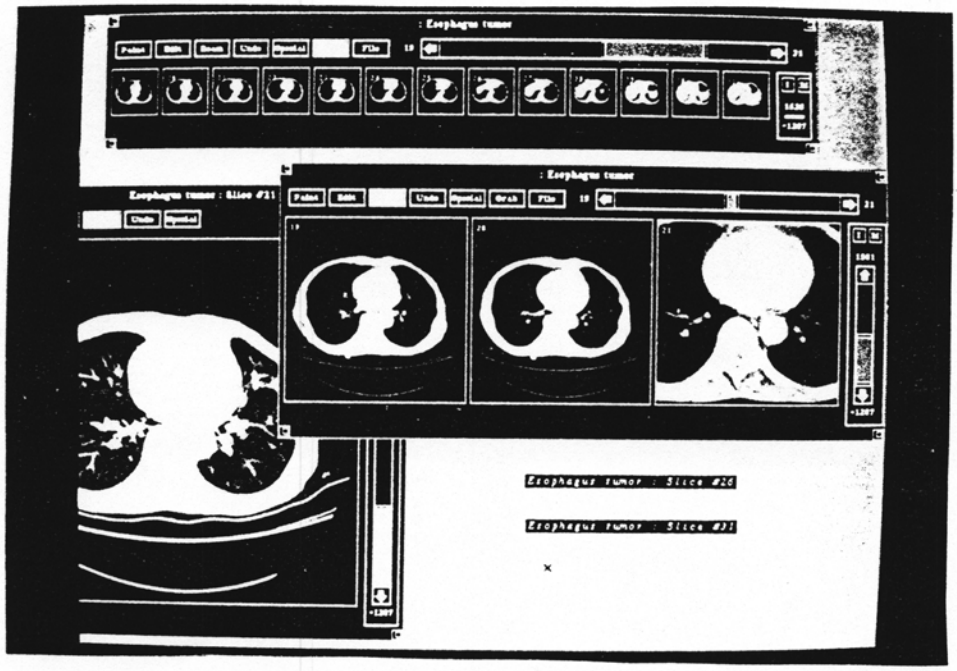

Figure 2: Multiple panels and iconified slice views

To do detailed work, any given slice of an image may be grabbed and blown up into a separate view (in a separate window): these are termed **individual views**. You may have several panel views of one study, and several individual views of one slice open at once. In figure 1 two views of slice 13 appear in the lower half of the screen. This was accomplished by twice placing the cursor over the desired miniature, dragging the cursor and releasing it over an uncovered area of the screen. To create a new panel with arbitrarily sized miniatures, a *special* menu entry is provided. Figure 2 evidences two panels of different size looking into the same image.

2.3. Altering the field-of-view

Notice that in figure 1 we have zoomed in on the leftmost individual view; we have thus altered the *field-of-view*, that rectangular portion of the image-slice appearing in the window as specified by the zoom factor and center point for display. An explanation of this requires a brief digression. When we spawned the individual views above, we assumed we had already put the panel into *grab* mode. Each window has a series of *mode* buttons along the top. You can be in only one mode at any time. The effects of using the cursor are determined by the mode within the window falling under it, as follows:

grab – create slice windows or copy fields of view.
roam – zoom or change the center of the field of view.
edit – draw and change contours. Submodes are selected by a menu.
paint – edit the binary mask image for autocontouring.

When *roam* mode is enabled on a window, the left button indicates we are to zoom out by a factor of 2; the right button corresponds to zooming in by a factor of 2; and the center button indicates we are to center our view around this cursor position, or *pan*.

Within the panel, roaming affects each miniature separately. Future work will allow you to explicitly set the field-of-view relative to the displayed window by reshaping a box outlining the boundary of the field-of-view. This technique can be applied to panels as well.

2.4. Resizing Windows

Since *Imex* runs in a windowing environment, we can exploit the typical real-estate window manager functions, including moving windows (even off the screen), iconifying and deiconifying, and resizing. If you were to grab a corner of the individual window and resize it, it would sample or replicate pixels as necessary using original slice data while leaving unchanged the field-of-view. When you resize a panel, the size of the miniatures remains unchanged: instead we re-fill the panel while keeping the same starting slice number.

2.5. Intensity Mapping

Lastly, on the right of each window appears a vertical scrollbar to effect intensity-windowing. This linear remapping of intensity values may be individually performed on each view. Copying of intensity values from one window to another is effected merely by grabbing the top of one scrollbar and dragging onto another windowing scrollbar.

3. Contour Management

Techniques for the manual editing and automatic derivation of contours representing the boundaries of objects have existed for some time [Sung78]. The distinctive features of our system are its ease of use, the orthogonality of the component operations, and the inclusion of a painting function to update the binary segmented image prior to edge-detection.

3.1. Contours and Objects

Contours are represented as a piecewise-linear curve defined by a chain of boundary points: they may be closed or open. Sets of contours may be grouped into *anatomes* (an abbreviation for *anatomical objects*), each group identified by a unique name and color. Each group may be made visibile or invisible, and be saved to or loaded from a file. Support is provided for creating and deleting anatomes, changing an anatome's color, and transfering the ownership of a contour between anatomes.

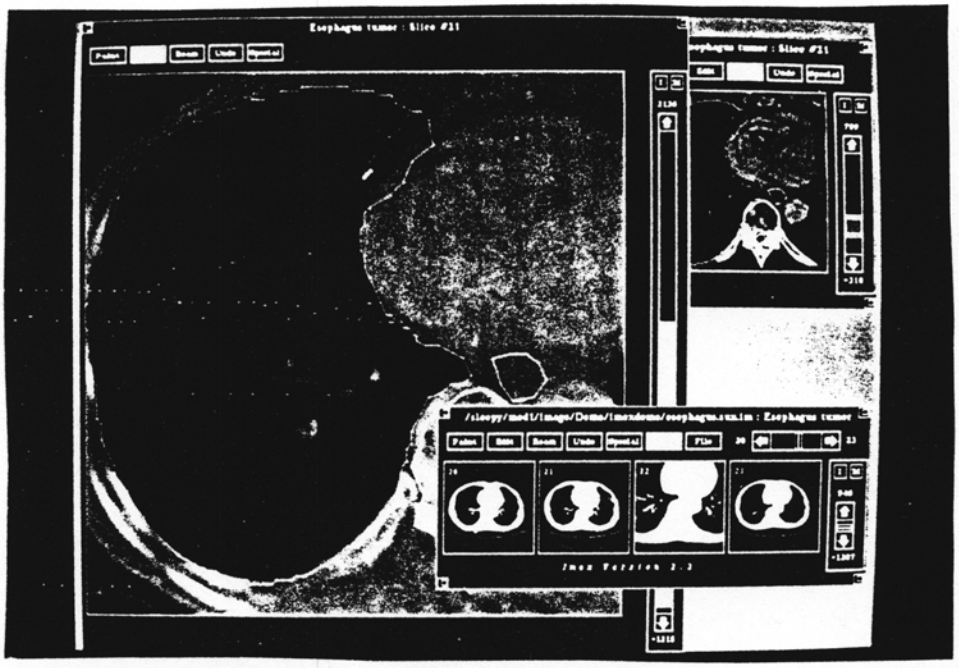

Figure 3: Manually editing a contour

3.2. Manual Contour Editing

The features for drawing and updating a contour by hand are similar to state-of-the-art drawing packages that are in turn based on principles dating back to the early 60's [Suth63]. Figures 3 and 4 show contour editing under *Imex*. You may edit contours in any view: actions are reflected all views of the image. The user is allowed to draw any contour segment he wishes; if either of the endpoints of the new segment are attached to an existing contour point, the existing target contour is updated with the new segment. Figure 5 shows how a newly drawn contour segment is joined to those contours it touches: the ambiguity of which portion of the target contour to replace is resolved by retaining that branch of greater length or which has been previously selected. Drawing of the new segment may be continuous: holding down the mouse button key lets the system approximate the drawn curve by a piecewise-linear arc. Otherwise, standard rubber-banding techniques to draw polygonal objects are used. To reduce the complexity of tiling and the number of polygons in the fitted surface, functions are provided which approximate a contour by a subset of its points.

3.3. Automatic contour generation

Automatic object definition within *Imex* is supported only by means of edge-tracking on an image segmented by threshholding, as described in the introduction. A separate control panel is provided for auto-contouring, as shown in figure 6. Moreover, a simple paint program is being developed to allow modification of the segmentation. This paint program has three brush functions, *Paint-inside, Paint-outside*, and *Erase-paint*, which allow the user to force pixels onto either side of the binary classification. Interruption and resumption of auto-contouring from open contours is also included in the design. These features combine to form a user-friendly tool for object definition.

4. View-world paradigm

At the heart of the system is the *view-world paradigm*, a simple but powerful recursive abstraction. Here, a *world* is a 2D array of views, and a *view* is a 2D subregion of a world. At the lowest level a world corresponds to one slice of an image; the views into it are mapped into X windows on the display, each thus holding a presentation of an image slice. A panel of miniatures, or index, is itself modeled as a world of image-slice views, one for each slice in the image; that subset being displayed is a view into this linear world. Thus, when we resize the panel, or use a scrollbar to select which slices are to be displayed within the panel, those miniatures which newly appear will remember their previous zoom-factor and center. An example is shown in figure 7.

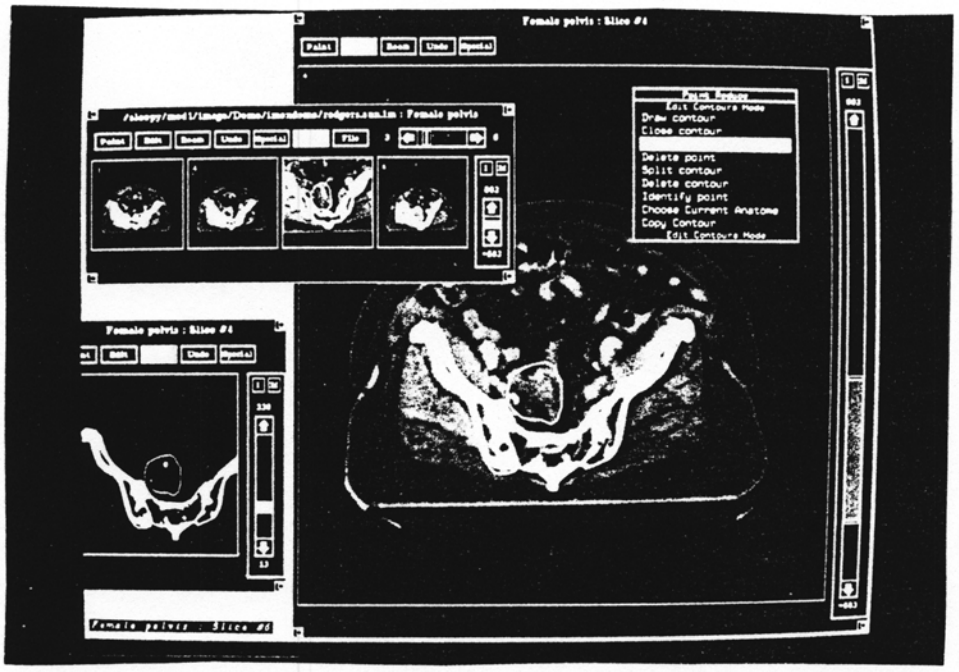

Figure 4: Another view of contour editing

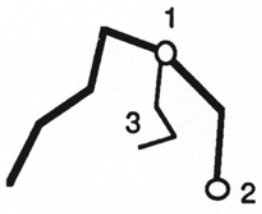

Figure 5: Editing an open contour:
Select vertex (1);
Add points (1-->2);
Old segment is deleted (3).

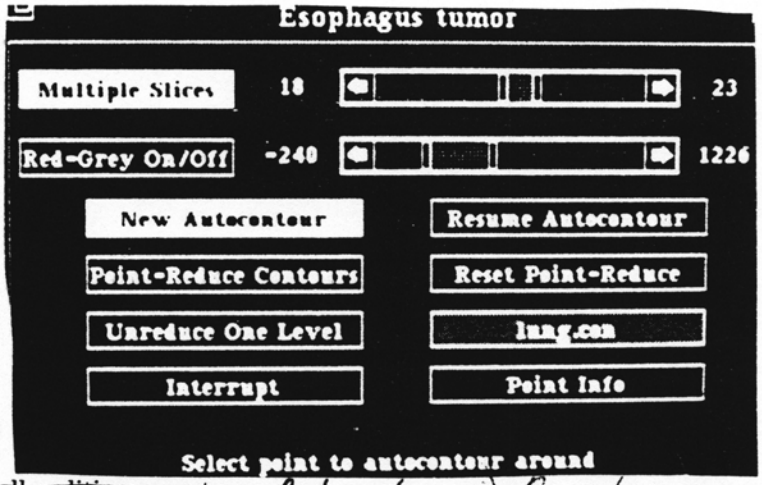

Figure 6: ~~Manually editing a contour~~ Autocontouring Panel

World of views into slices

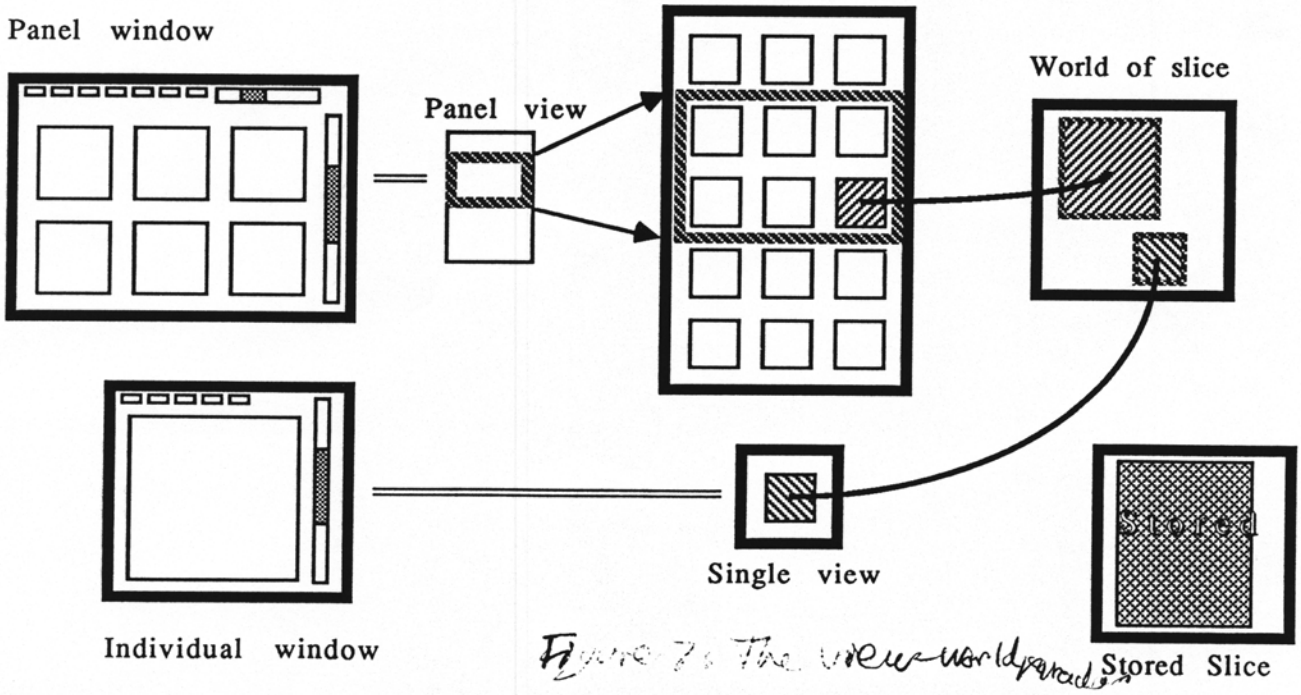

Figure 7: The view-world paradigm

One issue not yet made explicit is how we map the 2D view to the window on display screen. For an image slice, we magnify or reduce the chosen region of the image as required to fit into it own X window, while clipping to preserve the aspect ratio. For an index panel, however, we typically intend to display a subsequence of a linear world of stacked slices, yet the displayed panel may be made up multiple rows. We choose in this case to wrap around in row-major order.

This paradigm supports hierarchies of panels (i.e., panels of panels), as well as panels whose arrays of views are not linear. The latter application would help in the navigation thru multiple studies [Bear87,Bear89]: we could have a 2D panel which samples a 2D array of image studies, each study appearing as one column of image slices. This is an easy extension to the system, since it currently supports the ability to view several images at once. In this case the issue of reshaping the chosen view for the 2D panel into a different array for display (e.g., one row) poses an interesting problem which is still being investigated. The easist solution for mapping the view onto the window on the display screen is again to force preservation of aspect ratio.

Storage management is a real headache, since this system potentially allows many views at different magnifications into different areas of each image, exacerbated by the fact that indices typically sample many images. We perform storage optimization by attempting to retain in memory only the referenced portions of the slice, at roughly the minimally required resolution.

5. Conclusions

Imex was designed to provide a flexible interface for managing the 2D display and analysis of images in a windowing environment. As one piece of an integrated system at UNC it can be expected to run concurrently with other programs having graphical output, so giving the user the ability to manage screen space is another important benefit of adopting a windowing outlook. Multiple-console systems are not precluded, since X supports multiple screens and is network transparent.

The utility of the program is not restricted to merely preparing contours for tiling and eventual shaded-graphics rendering. The display functions of *Imex* subsume many of the suggested 2-D viewing and navigation strageties suggested by PACS and medical image workstation literature [Grew85,Wend84,Pize88]. Indeed, any interactive segmentation technique operating on a slice-by-slice basis could use this model.

The ultimate design criteria of this program is that it ***must be able to be used by a physician in a clinical setting***, perhaps by a radiation oncologist as one step towards visualizing radiation treatment planning using 3D graphics. To this end we have obeyed the dicta of natural interaction in a mouse-based windowing environment, orthogonality of function, and immediate response.

References

[Artz81] Artzy, E., Frieder, G. and Herman, G.T., "The theory, design, implementation and evaluation of a three-dimensional surface dettection algorithm", *Computer Graphics and Image Processing*, Vol.15, No.1 (Jan.1981) pp. 1-24.

[Bear87] D.V. Beard, S.M. Pizer, D. Rogers, R. Cromartie, S. Desirazu, S. Ramanathan and R. Rubin, "A Prototype Single-Screen PACs Console Development using Human Computer Interaction Techniques", *SPIE Proc. of Conference on Medical Imaging*, Vol.767 (1987) pp.646-653.

[Bear89] D.V. Beard, I.B. Bell, R. Cromartie, J. Walker, "Evolved design of a radiology workstation using time-motion analysis and the keystroke model", to appear in: *Proceedings of the SPIE Conference on Medical Imaging III* (Newport Beach, California, Jan 29-Feb 3, 1989), Vol.1091 (1989).

[Budi84] Budinger, T.F., "An analysis of 3-D display strategies", *Proceedings of the SPIE Conference on Processing and Display of Three-Dimensional Data II*, Vol.507 (1984) pp. 2-8.

[Chen84] Chen, L-S., Herman, G.T., Hung, H-M., Levkowitz, H., Trivedi, S.S. and Udupa, J.K., "Interactive manipulation of 3D data via a 2D display device", *Proceedings of the SPIE Conference on Processing and Display of Three-Dimensional Data II*, Vol.507 (1984) pp. 25-37.

[Chen85] Chen, L., Herman, G., Reynolds, R. and Udupa, J. "Surface Shading in the Cuberille Environment", *IEEE Computer Graphics & Applications*, Vol.5, No.12 (Dec. 1985) pp. 33-43.

[Chri78] Christianson, H.N. and Sederberg. T.W., "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics", *Computer Graphics* (SIGGRAPH'78 Proceedings), Vol.12, No.3 (1978) pp. 187-192.

[Cook83] Cook,L., Dwyer, S., Batnitzky, S. and Lee, K., "A Three-Dimensional Display System for Diagnostic Imaging Applications", *IEEE Computer Graphics & Applications*, Vol.3, No.5 (Aug. 1983) pp. 13-19.

[Dev84] Dev, P., Fellingham, L.L. and Vassiliadis, A., "3D graphics for interactive surgical simulation and implant design", *Proceedings of the SPIE Conference on Processing and Display of Three-Dimensional Data II*, Vol.507 (1984) pp. 52-57.

[Dreb88] Drebin, R.A., Carpenter, L. and Hanrahan, P., "Volume Rendering", *Computer Graphics*, SIGGRAPH'88 Conference Proceedings, Vol.22, No.4 (August 1988) pp. 65-74.

[Felli86] Fellingham, L.L., Vogel, J.H., Lau, C. and Dev, P., "Interactive graphics and 3-D modeling for surgical planning and prothesis and implant design", *Proc. NCGA '86*, III (1986) pp. 132-142.

[Fuch77] Fuchs, H., Kedem, Z.M. and Uselton, S.P., "Optimal Surface Reconstruction from Planar Contours", *CACM*, Vol.20, No.10 (October 1977) pp. 693-702.

[Gana82] Ganapathy, S. and Dennehy, T.G., "A New General Triangulation Method for Planar Contours", *Computer Graphics* (SIGGRAPH'82 Proceedings), Vol.16 (1982) pp. 69-75.

[Glen75] Glenn, W.V., Johnston, R.J., Morton, P.E. and Dwyer, S.J., "Image Generation and Display Techniques for CT Scan Data", *Investigative Radiology*, Vol.10, No.5 (Sept. 1975) pp. 403-416.

[Gold85] Goldwasser, S.M., Reynolds, R.A., Bapty, T., Baraff, D., Summers, J., Talton, D.A. and Walsh, E., "Physician's Workstation with Real-Time Performance", *IEEE Computer Graphics & Applications*, Vol.5, No.12 (Dec. 1985) pp. 44-57.

[Grew85] Grewer, R., Monnich, K.J., Schmidt, J., Svensson, H., Wendler, Th. "Design of Interactive Workstations for the Interpretation of Medical Images in Pictorial Information Systems", *Computer Assisted Radiology*, Proceedings of the International Symposium CAR'85, H.U. Lemke et al., eds. (Springer-Verlag, Berlin, 1985) pp. 679-686.

[Heff85] Heffernan, P.B. and Robb, R.A., "Display and Analysis of 4-D Medical Images", *Computer Assisted Radiology*, Proceedings of the International Symposium CAR'85, H.U. Lemke et al., eds. (Springer-Verlag, Berlin, 1985) pp. 583-592.

[Herm79] Herman, G.T. and Liu, H.K., "Three-dimensional Display of Organs from Computed Tomograms", *Computer Graphics and Image Processing*, Vol.9, No.1 (Jan.1979) pp. 1-21.

[Herm83] Herman, G.T. and Udupa, J.K., "Display of 3-D Digital Images: Computational Foundations and Medical Applications", *IEEE Computer Graphics & Applications*, Vol.3, No.8 (Aug. 1983) pp. 39-46.

[Herm84] Herman, Gabor T., "Three-Dimensional Computer Graphic Display in Medicine: the MIPG Perspective", *Pictorial Information Systems in Medicine*, K.H. Hoehne, ed. (Springer-Verlag, Berlin, 1986) pp. 181-210.

[Huan87] Huang, H.K., Mankovich, N.J., Taira, R., Cho, P., Stewart, B., Ho, B.K., Kangarloo, H., Boechat, M.I. and Dietrich, R.B., "Picture archiving and communication systems for Radiology", *Computer Assisted Radiology*, Proceedings of the International Symposium CAR'87, H.U. Lemke et al., eds. (Springer-Verlag, Berlin, 1987) pp. 487-492.

[Kepp75] Keppel, E., "Approximation of Complex Surfaces by Triangulation of Contour Lines", *IBM Journal of Research and Development*, Vol.19 (1975) pp.2-11.

[Levo88] Levoy, Marc, "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications*, Vol.8, No.5 (May 1988) pp. 29-37.

[Lore87] Lorensen, W. and Cline, H., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics* (SIGGRAPH'87 Proceedings), Vol.21, No.4 (1987) pp. 163-169.

[Mazz76] Mazziotta, J.C. and Huang, K.H., "THREAD (Three-Dimensional Reconstruction and Display) with Biomedical Applications in Neuron Ultrastructure and Display", *American Federation of Information Processing Society*, Vol.45, (1976) pp. 241-250.

[Meag82] Meagher, Donald J., "Geometric modeling using octree encoding", *Computer Graphics and Image Processing*, Vol.19 (1982) pp. 129-147.

[Pize86] S.M. Pizer, H. Fuchs, C. Mosher, L. Lifshitz, G.D. Abram, S. Ramanathan, B.T. Whitney, J.G. Rosenman, E.V. Staab, E.L. Chaney, G. Sherouse, "3D Shaded Graphics in Radiotherapy and Diagnostic Imaging", in: *Proc. Computer Graphics 1986*, National Computer Graphics Association, III (1986) pp. 107-113.

[Pize88] Pizer, S.M. and Beard, D.V., "Medical Image Workstations: State of Science and Technology", Technical Report TR88-016, University of North Carolina, Chapel Hill (March, 1988), also presented at *ACR Workshop* (June, 1988).

[Robb87] Robb, R.A., "A workstation for interactive display and analysis of multi-dimensional biomedical images", in: *Computer Assisted Radiology*, Proceedings of the International Symposium CAR'87, H.U. Lemke et al., eds. (Springer-Verlag, Berlin, 1987) pp. 642-656.

[Sche86] Scheifler, R.W and Gettys, Jim, "The X Window System", *ACM Transactions on Graphics*, Vol.5, No.2 (April 1986) pp. 79-109.

[Sung78] Sunguroff, A. and Greenberg, D., "Computer Generated Images for Medical Applications", *Computer Graphics* (SIGGRAPH'78 Proceedings), Vol.12 (1978) pp. 196-202.

[Suth63] Sutherland, I., *Sketchpad*, Ph.D. Thesis, MIT Laboratory for Computer Science, Cambridge, Massachusetts (1963).

[Talt87] Talton, D.A., Goldwasser, S.M., Reynolds, R.A. and Walsh, E.S., "Volume rendering algorithms for the presentation of 3D medical data", *Proc. NCGA'87*, III (1987), pp. 119-128.

[Temp85] Templeton, A.W., Dwyer, S.J., Cox, G.G., Johnson, J.A., Anderson, W.H. and Cook, L.T, "Picture Archiving and Communication Systems (PACS) for Radiology", *Computer Assisted Radiology*, Proceedings of the International Symposium CAR'85, H.U. Lemke et al., eds. (Springer-Verlag, Berlin, 1985) pp. 697-715.

[Udup82] Udupa, Jayaram K., "Interactive segmentation and boundary surface formation for 3D digital images", *Computer Graphics and Image Processing*, Vol.18 (1982) pp. 213-235.

[Udup83] Udupa, Jayaram K., "Display of 3D Information in Discrete 3D Scenes Produced by Computerized Tomography", *Proceedings of the IEEE*, Vol.71, No.3 (March 1983) pp. 420-431.

[Udup86] Udupa, J. K., "Simulation of Surgical Procedures Using Computer Graphics", *Proc. NCGA '86*, III (1986) pp. 80-91.

[Upso88] Upson, C. and Keeler, M., "VBUFFER: Visible Volume Rendering", *Computer Graphics*, SIGGRAPH'88 Conference Proceedings, Vol.22, No.4 (August 1988) pp. 59-64.

[Vann83] Vannier, M.W., Marsh, J.L. and Warren, J.O., "Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation", *Computer Graphics*, Vol.17, No.3 (1983) pp. 263-273.

[Wend84] Wendler, Th., Grewer, R., Monnich, K.J., Svensson, H., "Design Considerations for Multi Modality Medical Image Workstations", *Pictorial Information Systems in Medicine*, K.H. Hoehne, ed. (Springer-Verlag, Berlin, 1986) pp. 401-420.