INTERACTION TECHNIQUES FOR A VIRTUAL WORKSPACE

LUIS SERRA
TIMOTHY POSTON
NG HERN
CHUA BENG CHOON
Institute of Systems Science, National University of Singapore
{luis, tim, nghern, bcchua}@iss.nus.sg
and
JOHN A. WATERWORTH
Department of Informatics, University of Umeå, Sweden
jwworth@informatik.umu.se

Abstract

Precise, dextrous work on spatial data requires a natural interface paradigm, and interaction techniques that allow for efficient work for hours on end. We present the interaction scheme we use for medical work, based on the reach-in hand immersion environment of the Virtual Workbench. The scheme uses an straightforward 'reach in with a hand-held stylus' metaphor and combines tool selectors which become active when approached, and tools which act on objects they approach, to make an easily learned, powerful interface in which the user can achieve productive work. Predictable behaviour by selectors and manipulators, at locations real to both hand and eye, produces a robust environment in which substantive work can be comfortably performed.

CR Categories and Subject Descriptors: I.3.2 [**Computer Graphics**]: Graphics Systems; I.3.6 [**Computer Graphics**]: Methodology and Techniques - Interaction techniques.

General terms: Human Factors

Additional Key Words and Phrases: VR interaction and navigation techniques, dexterity, hand-eye coordination, medical applications.

1 Motivation

Much work in VR has focused on the 'sense of immersion' as the gold standard of quality. Most visits to such realities leave them unchanged. We aim to provide an interaction environment with virtual tools controlled through material devices, in which to perform precise, accurate 3D work (resolution in millimetres) for work sessions of serious length (3 to 4 hours at a time). This is necessary for the medical applications we are developing[1], and applicable to such diverse fields as CAD, welding planning and training, and 3D sketching. The medical applications allow one, for example, to edit multiple computer estimates of the moving heart boundary, for strain analysis that reveals ischæmic muscle, or to define central curves for arteries, nerves, etc., detectable in volume data and in 3D space. We use the Virtual Workbench [15], a robust environment for general 3D manipulation tasks (see Section 4). Most 3D interfaces are based either

(a) on the head mounted 'immersive' paradigm, (e.g.,[3][22]) where low resolution, long drawing delays for complex objects (and consequent simulator sickness), and calibration problems, combine to make accurate work extremely hard;

(b) on a directly viewed stereo display [2][7], in which the hand and its position sensor cannot go behind any part of the display without obscuring the screen and destroying stereo. When

1. In the Centre for Information-enhanced Medicine (CI_EMED) established jointly by Johns Hopkins University and the Institute of Systems Science of the National University of Singapore.

such systems use negative parallax to bring the display 'in reach' in front of the screen, they create conflict between the stereo and lens-accommodation depth cues.

Calibration of the Virtual Workbench is straightforward, and cues from the single-eye focus and two-eye stereo fusion are in better accord. Here, we concentrate on interface design, leaving calibration, precision, predictive filtering etc. to papers already presented [15], [11].

To date, the VR community has paid limited attention to how the user interacts with application aspects beyond the simulated 3D reality, such as access to system functions and tools, status, help and other information, all vital for professional uses of VR in for instance medicine, architecture and collaborative design. In [23] we refer to the necessary means as **VR management tools.**

There are two classes of these: **Manipulators,** an integral part to the VR, include such 'virtual tools' as scalpel, calipers, hammer, pen, and so on. **Selectors** such as menus and sliders show a range of options and/or provide status information. Selectors tend to be used in combination with manipulators; for example, selecting from a menu with a button-pusher manipulator.

2 Design issues

The user should be free to focus on problems of engineering or artistic design, diagnosis, etc., with a minimum of conscious attention to details of selection, rotation, and so on. The actions required should thus echo natural ones as far as possible (reach for something rather than recall a button sequence), allow both visual and motor memory to assist, and lead to error as rarely as possible. A Workbench tool always has a handle, always located where it is seen and felt to be, through the eyes and through the user's grip on a material sensor, matching human need for hand-eye coordinated location (emphasized by the finger marks on every monitor) becomes much more acute in 3D.

Similarly, we do not select or manipulate objects by remote pointing such as 'ray casting' [5][12]. This is partly because we do not need to, since we do not use the work volume to display objects that are out of reach, and partly because we need a tool steadier than a laser pointer. Moreover, pressing a button usually disturbs the hand a little; a tool tip near the fingers moves much less than the end of a ray. Selecting 'delete' when 'move' is intended, or selecting the wrong small item for deletion, can waste a great deal of previous work. Choosing one out of (for instance) fifty standard landmarks on a skull requires little concentration if one thinks 'tweezers', a great deal if one is pointing from twenty centimetres away—particularly with a device like the DataGlove. (A virtual hand allows a very literal 'finger and thumb' grasping, but displaying it is costly and obscures the workpiece; it is better to model a handle than the hand.)

The 3D motions possible for a menu scheme—spinning, unfolding, retreating,...—are seductive to the designer, but all dangerously interesting. If one needs to choose as quickly and unthinkingly as the left little finger finds 'a' on a keyboard, unstable locations are a menace.

Although the Workbench immerses the workpiece in your space, rather than you in a larger volume, the tools in your hands are an integral part of the virtual world. This gives the sense of co-presence characteristic of immersion and reproduces most of their constraints: no access to the keyboard or other 'external world' devices than the material handles of the virtual tools.

The display must be used efficiently, in both space and time; it must appear uncluttered, and respond fast to minimize time-lag problems, though medical applications make great demands on rendering speed. Both gain by concentrating display effort on the focus of interaction. While the user interacts with one object, others can be displayed in a simplified form. While the user is dealing with menu buttons, data such as an MRI volume may be rendered at lower resolution; conversely, when dealing with the volume, the menu is rendered as a simplified box.

Effective work demands ease in transitions. In "pick 3D point, rotate volume, pick next 3D point," the implied tool changes from selector to rotator to selector should need a minimum of time and conscious attention. Feedback is vital in the user's sense of the state of the interaction.

We currently combine visual feedback via the display with the user's neuromuscular sense of the tool's location, and will shortly add auditory and haptic feedback.

We limit interactions to a natural volume within which reach and stereo fusion are comfortable. This places objects within 15 to 50 cm from the eyes, in a cone perhaps 50° wide. Objects too close to the eye are hard to fuse, and conflict with the eyes' focus on the display. Those too far to reach can be retrieved by shrinking the scene towards one. This control of scale is a key to precise control; one should move easily between designing a cathedral and detailing its pews.

3 Related work

Many VR management schemes depend on gesture. Jacoby and Ellis [8][9] use gloved hand gestures to invoke 'virtual menus' to highlight an item, to select an item, and to move the menu, with ray casting as selection feedback. Fairchild et al. [4] implemented and evaluated a scheme know as Gesture Sequence Navigation, a major problem for users of which was the amount of gesticulation required. In many environments [5][1][19], a circular or radial menu of symbols serves as the selector from which the user chooses. Although this method has been shown to work well in 2D interfaces (Kurtenbach and Buxton's 'marking menus' [10]), the amount of physical effort required of 3D users in repeated selection led us to look for non-gestural approaches to VR interaction design.

We here address mainly approaches that involve precise, calibrated, 3D spatial work similar to our own task domain. We pass over 3D systems like [17] that do not use stereo; depth cues such as controllable level planes must then be added to let the user navigate, and the design problems are quite different from those of a shared space environment.

The closest (and earliest) analogous work to the Virtual WorkBench described below is the half-silvered mirror device of Schmandt [18], which pioneered the use of menus and icons in space shared by the user and the display. Textured icons of buildings could be moved around, and it allowed curve sketching on a plane coinciding with a 45° surface and a tablet. Ergonomic problems, and the complexity limitations of real-time graphics in the early 1980s, prevented any immediate move to uses such as the hoped-for application to VLSI design.

To date, the environment most seriously aimed at precise, calibrated 3D work is the directly viewed stereo HoloSketch of Deering [1]. The goal, supported by impressive precision engineering, is accurate 3D work (for instance, in CAD), but the examples offered mix simple stored forms like planks with shaky freehand 3D forms, so it is hard to judge the precision a user can actually achieve. (On a standard Sun workstation, some form of elbow support must clearly be added before most users could hope for adequate control, or use it in long sessions.) The elaborate menus (daisy wheels within wheels) are supplemented by keyboard input.

Shaw and Green [19] present the user with rotating menus, 'ray cast' for selection, memorize multi-button sequences for both hands, and still require keyboard and mouse input.

Hinckley et al. [6] avoid widgets, and attach virtual objects to real world props, moving parallel to the graphics but elsewhere. A new physical prop is needed for each new object type in the display; for example, a doll's head in the user's hand can represent a CT head volume better than a heart. A finite flat prop can represent an infinite cutting plane, to control a sectional view, but placing virtual therapeutic objects inside something represented by an impenetrable prop is difficult. Scaling—often vital for precision—breaks the metaphor.

The 'virtual tricorder' of [22] hangs from the user's virtual belt in a fully immersive HMD environment, where the technology cannot yet make normal text readable at a normal distance. When the calibration, latency, resolution and field of view problems are solved, such a tool will become a natural widget in such worlds. It is not applicable to precision work in the shorter term.

4 The Virtual Workbench

The Virtual Workbench [15] is a VR workstation modelled more on the binocular microscope familiar to medical workers than on the immersion paradigm, but is 'hands on' in a way that a microscope is not. Our display configuration (Figure 1), using stereo achieved by time-splitting the display with Crystal Eyes™ glasses, allows the real hands to move in the positions at which the eyes seem to look. Fine control depends on exploiting this.

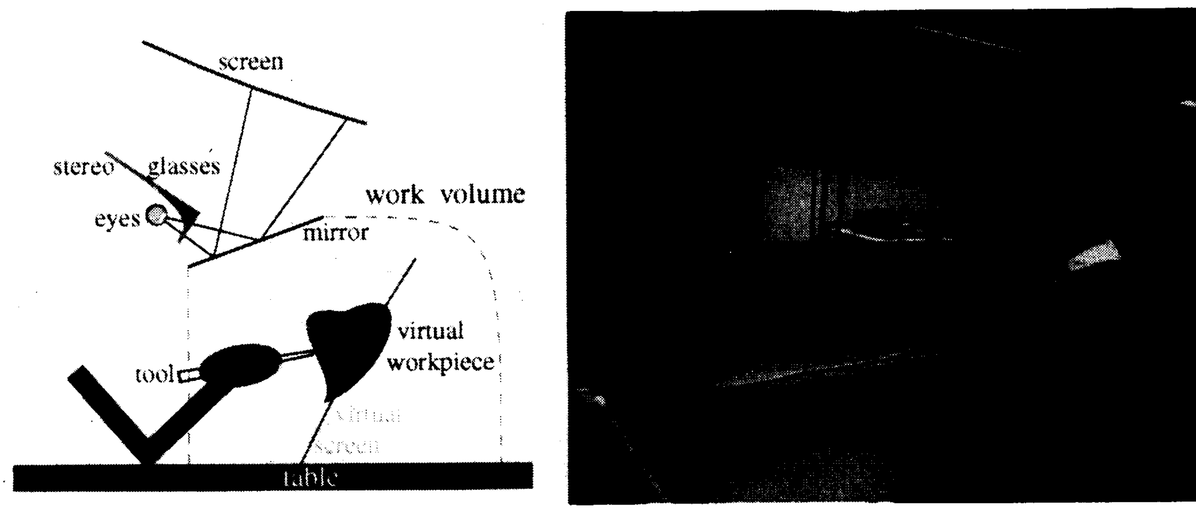


Figure 1. The Virtual Workbench. The left view shows a schematic, the right a photograph of the hardware implementation. The user sees a virtual tool, in the stereo display in the mirror, exactly where the hand holds the real sensor-equipped handle: the hand is invisible.

The 6DOF sensor currently installed is a Polhemus FASTRAK™, with a single button. The mechanical Immersion Probe™ (with somewhat superior noise resistance and precision but awkward at certain angles) is normally supplied without buttons. The two custom-added for our order were uncomfortable, and fell below the robust engineering standards of the basic unit.

5 The metaphor of interaction

In the Workbench, interactions follow the principle of "reach in for the object of interest and press the switch on the stylus to interact with it," using a simple bounding-box detection algorithm. Thus, all relevant widgets must be within view (in full, or simplified/iconized, to save rendering time). There is no special double click on the single stylus switch, nor magic key to pop menus up. Whenever the stylus tip enters the volume of influence surrounding an object, the object shows it has become active by a change of colour or other highlighting trick.

At any one time, one operating mode controls the function of the stylus (as signalled by its appearance), which in different modes becomes a slice-selector, a point-mover, a rotator, etc.

There are *application-specific* objects, and *control-objects* or widgets (Figure 2). Application-specific objects are mode-sensitive, and respond to the current mode of operation whenever the stylus enters their volume of influence. Thus, for polyline curves and rotation mode, the user can rotate any curve by reaching near it, and activating the switch. In curve-editing, the selected curve is highlighted and selection descends to a lower level, where the user picks a control or node point and moves it. In contrast, widgets follow a mode-insensitive paradigm: when the stylus enters a widget's volume of influence, the stylus transforms to the manipulator associated with the widget, that knows how to interact with it. For example, when the stylus enters the slider widget, it will change to the slider-dragger from, say, the rotator, indicating to the user that it will not rotate the slider widget but interact with its sliding head. On leaving the slider's volume, the stylus returns to its former function. This difference between application object and widget keeps interactions simple. If widgets were mode-sensitive, it would be hard to change modes: a rotator approaching

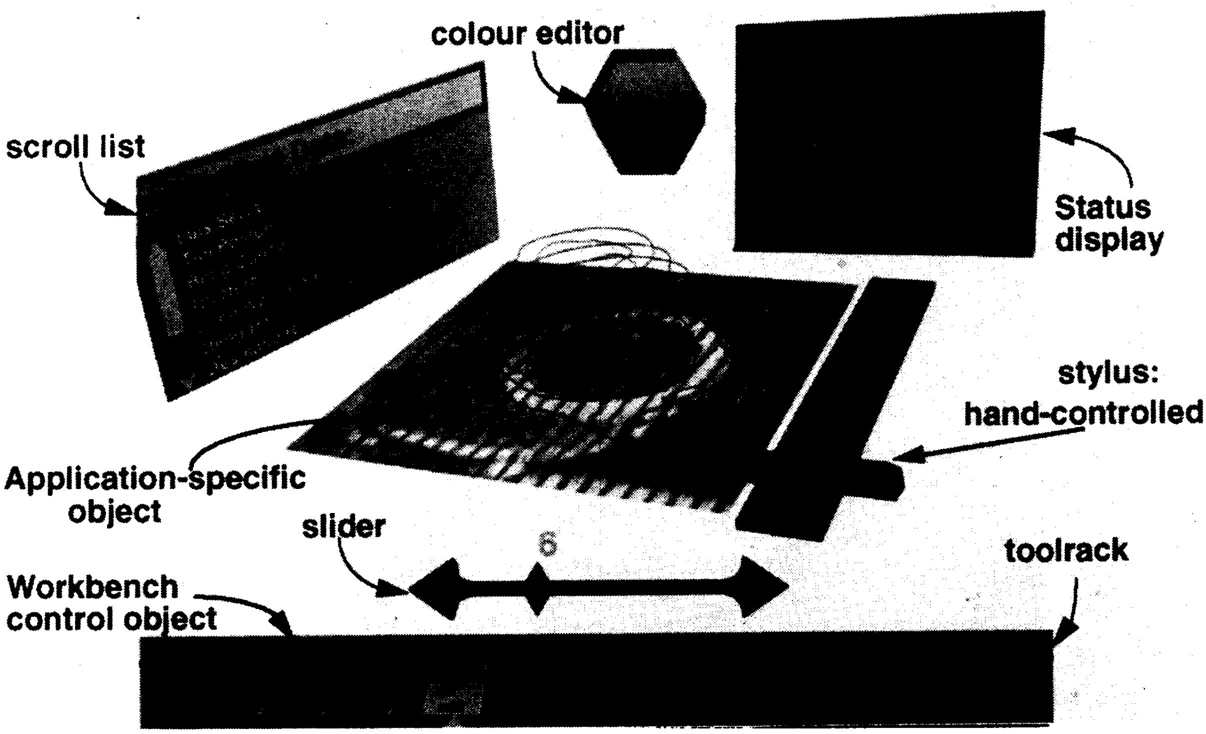


Figure 2. The basic elements of a Virtual Workbench application. Examples of selectors

a menu, for instance, could rotate it but not select from it. If a menu is to be movable, it can be given mode-sensitive handles.

Mode transitions do not need millimetre precision, and can thus be relegated to the non-dominant hand. We do not yet have hardware support for an absolute-position slider, which the user can move to a remembered location without looking down at it, but buttons can be selected with the mouse. With the customized two-button Immersion Probe we reserved the thumb button to the 'rotate' mode, always active when this was pressed; active selection in returning to (say) the 'edit' function is more obtrusive, since rotation during a sequence of similar operations is so frequent. (Comfort with head tracking imposes more stringent latency constraints than we can meet in rendering medical volume data, so even small changes of viewing angle require rotation.)

6 The Workbench objects

6.1 Manipulators: tools

The most generic tools, provided with the Workbench toolkit (Figure 3), can be included in an application by a line of two of code. One example is the rotator, which controls the orientation of the objects by a 'rubber-band' interaction scheme. While the button is down, the selected object rotates to follow a 'rubber' string between its the centre C and the stylus tip, each step being a rotation in the plane formed by C and two successive stylus positions. Intuitively, the feel is 'reach in and turn it about its centre'. An alternate version adds scaling ('turn it and pull it larger'), which is very useful when seeking an improved view of a detail. Equally general are the volume of interest box (allowing examination and zooming of a limited region, unobscured by surrounding material), the 6DOF mover ('stick a fork in it'), the tools to specify/modify/delete points, curves, or surfaces, and measurement tools. A general purpose memorizing tool that can record user input for later autonomous replay (convenient when making monoscopic videos of interactions for which the user's stereo perception is crucial).

Other tools apply to narrower classes of application, such as those that manipulate the display of volume data (such as MR and CT scans) by changing transfer function and opacity, or govern

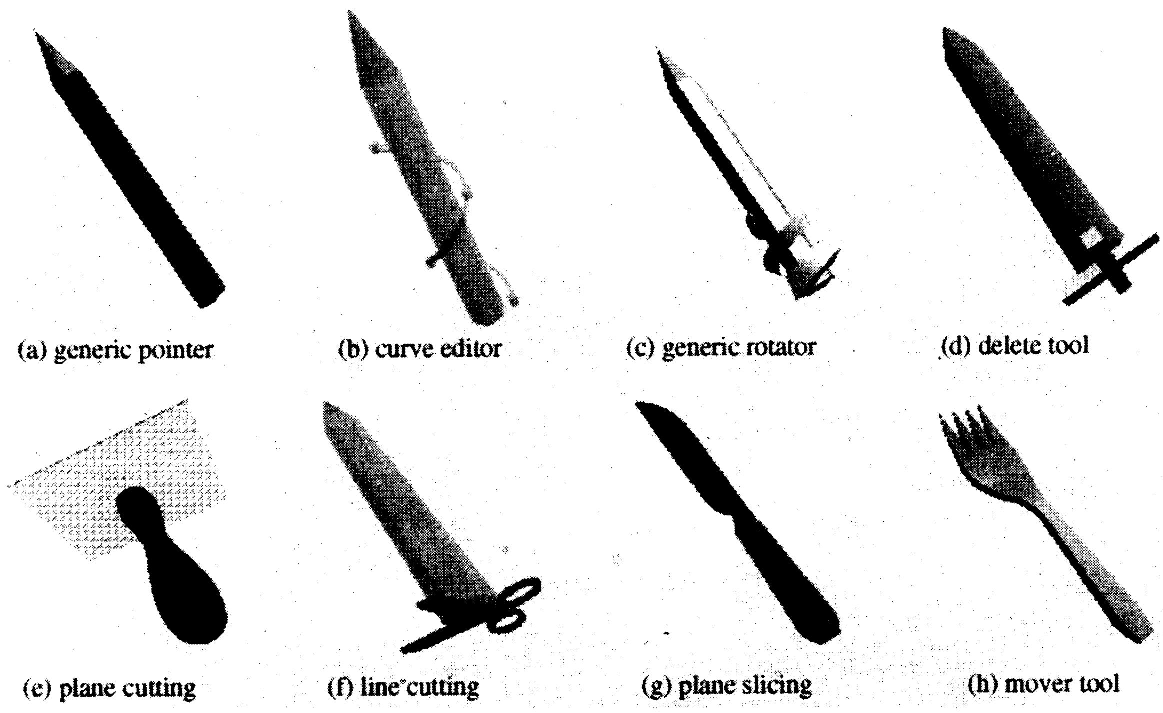


Figure 3. Examples of manipulators

segmentation, surface extraction, and so on: a fully comprehensive toolkit would cover all volume data formats, but at present we support only the one most commonly used by CI$_E$MED.

Finally, some applications require custom tools. For example, it is helpful [13] to confine a display of tagged cardiac MR to an ellipsoidal volume of interest: choosing a particular ellipsoid involves specifying up to nine degrees of freedom, requiring a somewhat specialized tool. The toolkit facilitates the development of application-specific tools, allowing the programmer to attach selection and object-object communication easily to objects with new behaviours ('bend when pushed', 'be cut by a Scalpel tool', 'visualize tumour destruction by radiation focused here') that require simulation computations within the objects.

6.2 Selectors: Widgets

The Workbench uses the following selectors: the toolrack, the scroll list, the slider, the colour editor, and the status display (Figure 2). The most frequently used is the virtual 'toolrack' menu, holding the buttons that enable tools and activate different modes of operation.

6.2.1 The toolrack. The toolrack becomes active when the stylus moves into its bounding box.

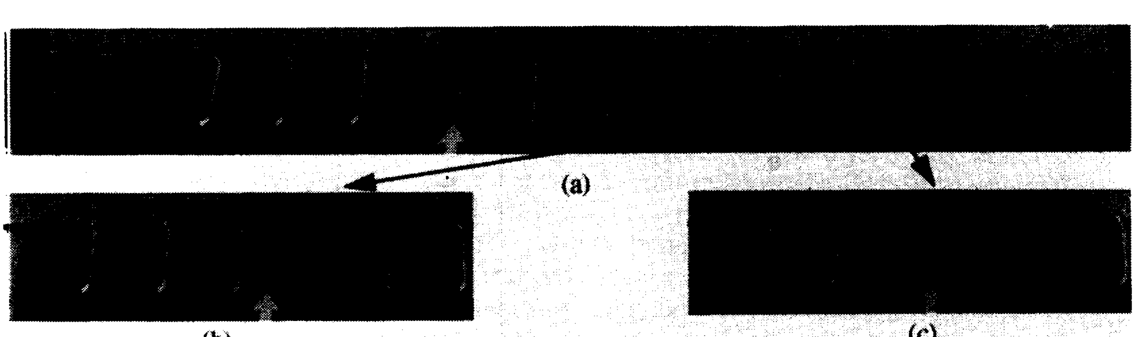


Figure 4. The Tool Rack. Hierarchical

As the user moves the stylus over it, the button nearest the tip is highlighted, and the rack displays above it an expanded text about its function (Figure 4a), enabled by clicking. Placing the toolrack in virtual space to coincide with the material Workbench surface makes it easier to hit the buttons quickly, by preventing overshoot. However, since the toolrack is a 1D row of buttons, a constrained slider device (or the mouse) in the non-dominant hand is often more convenient.

A subordinate set of buttons can take over the rack (among them a 'return' button to restore the status quo ante). In Figure 4a the third from the right enables the heartbeat control toolrack of Figure 4b, while the button on its right enables the recording buttons in Figure 4c.

6.2.2 The slider. The user reaches into the slider and picks the bead, which then can be dragged to the desired scalar value, displayed nearby as text. The developer can attach the slider to any parameter of a simulation or display.

6.2.3 The scroll list. The list scroll is a simple but essential widget that lets one to scroll through a list of texts (in Figure 2, at the left-hand side). This is a straightforward extension of the standard X/Motif widget, optimized for 3D rendering, showing only ten items at a time for performance reasons, and operated by the stylus or (more commonly) attached to a slider device controller.

6.2.4 The colour editor. This widget has an hexagonal prismatic shape in which the base is a palette formed by six vertices (RGB and their complementary), with saturation controlled along the axis orthogonal to the base.

6.2.5 The status display. This widget lets the user monitor system or application variables via an alphanumeric display. One system variable is the frames/second rate the application is achieving; an application variable might be a score, an estimate of blood flow or bone density, etc.

7 Medical application examples

The manipulation of medical image data, which increasingly come in 3D form (or 4D, with the time dimension), as Magnetic Resonance, CATscan, ultrasound, etc., has been our main concern given our research collaboration with the Johns Hopkins University. It is frequently necessary to revolve the display, to specify a cutting plane exposing (for example) a particular cross-section of brain data, to specify a point within a suspected tumour and ask the system to display the connected region of similar data values it appears to belong to, and so on. Here we briefly describe four medical applications that have been integrated into the Virtual Workbench.

7.1 The 3D Contour Editor

In the 3D Contour Editor [21], the user makes fine adjustments in computer-estimated curves, using their neighbours in space or time as a guide to help distinguish fact from artefact. Figure 2 shows a snapshot of this environment, in which the editing tool is editing a control point.

The stylus interacts with the stack of MRI slices. Each slice is a 2D MRI density map, texture-mapped over a polygon. In each stack, there is always a 'working slice' or slice of interest, in which the contour editing takes place. Editing primitives include move or delete points, change of slice, play-back control, magnification, etc...

7.2 The tube finder editor

Medical images of such branching structures as blood vessels are clear to the human visual system, but easily confuse computer vision programs. The tube finder [16] provides an intuitive, hand-eye coordinated, reach-in interface that allows the user to sketch central curves for arteries, nerves, etc., detectable in volume data and in 3D space, and have this position/shape estimate refined by active contour methods (Figure 5). The editor enables the intuitive creation, modification and deletion of 3D curves by means of the interactive tool.

The user creates curves as 3D splines by sketching them with 3D stylus motions, matched to the different tubes visible in the stereo maximum intensity images. One can later edit these

Figure 5. The Virtual Workbench 3D curve editing tool (stereo pair, cross-eyed). Upon stereo fusion one sees the arteries as translucent, with the 'hand-held' stylus showing through them.

curves, viewed from the same or another direction, to achieve better matches. (Errors in depth perception can thus be corrected from a side view.) Editing primitives include sketch a curve, add/delete curve, move control points, cut, link, clone, move, mirror, rotate a curve.

7.3 Visualization tools for diagnostics

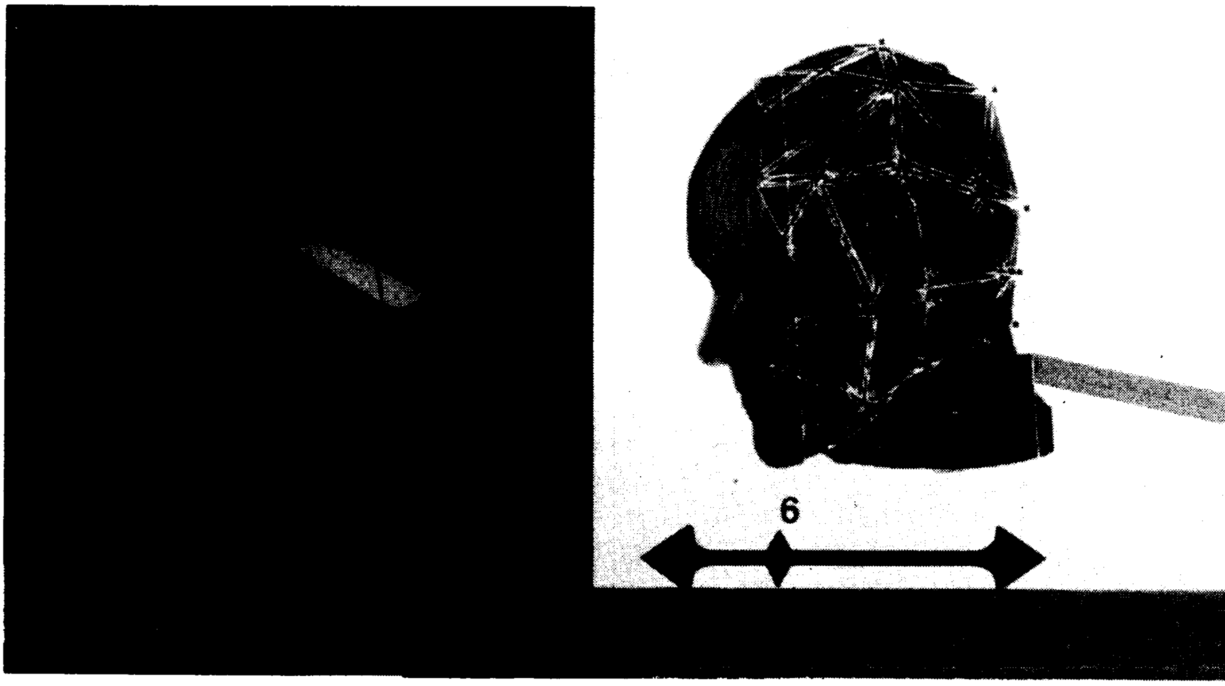
To facilitate the exploration of volume data sets we have a general interface which can deal with 3D and 4D medical dynamic data sets such as cine-loop MRI (Figure 6a). This lets one turn an object by reaching in and dragging it around, select an arbitrary cutting plane by means of the stylus, crop the volume to a region of interest and zoom this up, etc. Specific functions, such as tube finding or fitting an ellipsoid to the heart, can be added in particular applications.

7.4 A tetrahedron editor

A fast, simple way to morph a 2D image is to triangulate it, compute new vertex positions, and linearly interpolate within the triangles (indeed, this is the basis of most 2D texture mapping).

The analogue for 3D morphing is to divide a volume into tetrahedra. The choice of these can be automated, but it is hard to replace the human judgement that (for instance) a cranium can be morphed adequately with quite large tetrahedra, while the region of the eyes—aesthetically and thus surgically vital—requires more detail. Manual editing thus serves a useful function. Our tetrahedron editor (Figure 6b) combines the volume display tools above with other generic tools such as point create/delete/move, and a few newly created tools specific to the task.

This is the first Virtual Workbench application to be used for whole-day work sessions. There has been no experience of eyestrain or headache: elbow-on-the-table support proved sufficient for millimetric discriminatory precision (selecting between nearby points); the application does not depend on accuracy of absolute apparent precision (which indeed shifts when motions of an untracked head are possible), as long as the displayed tool remains close enough to the felt but unseen stylus for the user to accept that they coincide. Physical layout is important: an arrangement that had seemed satisfactory with short experiments produced neck strain in a tall user, until minor adjustments were made.



(a) (b)

Figure 6. (a) The cine-loop explorer. The oblique cutting plane follows the movements of the knife blade, slaved to the handle felt by the hand [14]. (b) The tetrahedron editor. Buttons control rotation (with and without scaling), free 6DOF motion, and the volume of interest; adding, deleting, moving vertices; adding a tetrahedron by selecting four vertices, undoing the most recent, or deleting a selected one; toggling the Z-buffer or the rendering mode to see tetrahedral surface vs. depth, skin vs. skull; toggle volume display; and read/write functions. The slider controls the trade-off between rendering speed and volume display quality.

8 Implementation

The current implementation of the Workbench runs on OpenGL on Silicon Graphics workstations. BrixMed, the software toolkit that has been developed from Bricks [20] to drive the Workbench, allows rapid creation of interactive 3D user interfaces.

9 Conclusions and Further work

A 3D interface for substantive work should be unobtrusive, and quick and accurate in use. This requires that the user have a reliable sense of where objects and widgets are, and easy use of that sense in choosing them. We have found the hand-eye coordination built into the Virtual Workbench to be very helpful in achieving this.

A straightforward 'reach in with a hand-held stylus' metaphor has directed our interface design, producing a toolkit which can be comfortably used for long sessions. Tool selectors which become active when approached, and tools which act on objects they approach, combine to make an easily learned, powerful interface in which the user can achieve productive work.

In future developments we aim to integrate the Virtual Workbench with more comprehensive medical image/simulation/planning software under development at CI$_{\text{E}}$MED, as well as developing tools to manage the acquisition (not merely the processing) of MR data volumes. This will require more extensive management of data external to the work volume, both in files and in databases of patients and clinical reference material such as the CI$_{\text{E}}$MED electronic brain atlas: these will inevitably require extensions to handle text input and provide help displays including stereo 'how tool X works' animations.

10 References

[1] Deering, M., "HoloSketch: a virtual reality sketching/animation tool". [1995] ACM Transactions on Computer-Human Interactions. September (to appear).

[2] Deering, M., "High Resolution Virtual Reality". [1992] Computer Graphics 26, pp. 195-201.

[3] Edwards, E.K., Rolland, J.P. and Keller, K.P., [1993] "Video see-through design for merging of real and virtual environments". In Proc. IEEE VRAIS'93, pp. 223-233.

[4] Fairchild, K.M., Serra, L., Ng, H., Lee, B.H. and Ang, T.L., [1992] "Dynamic fish-eye information visualizations". In Proc. British Computer Society International Conference on Virtual Reality Systems, London. Also in Virtual Reality Systems, 1993, R. Earnshaw, M. Gigante & H. Jones (eds), Academic Press, pp. 161-177.

[5] Halliday, S.,and Green, M., [1994] "A geometric modeling and animation system for Virtual Reality". In Proc. VRST'94 - Virtual Reality Software and Technology (Singapore, August 23-26, 1994). G. Singh, S.K. Feiner, D. Thalmann (eds.). World Scientific. Singapore, pp. 71-84.

[6] Hinckley, K., Paush, R., Goble, J.C., and Kassell, N.F., [1994] "Passive real-world interface props for neurological visualization". In SIGCHI'94, pp 452-458, New York: ACM

[7] Iwata, H., "Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator". [1990] Computer Graphics 24, pp. 165-170.

[8] Jacoby, R. H. and Ellis, S. R., "Virtual Menus". [1992] In Proc. SPIE, 1666.

[9] Jacoby, R. H. and Ellis, S. R., [1993] "Using virtual menus in a virtual environment". In Course Notes 43 - Implementing Virtual Reality - pages 7.1 to 7.8. SIGGRAPH'93, 20th International Conference on Computer Graphics and Interactive Techniques (Anaheim, August 1993). New York: ACM.

[10] Kurtenbach, G. and Buxton, W., [1993] "The limits of expert performance using hierarchic marking menus". In INTERCHI'93, pp 482-487, New York: ACM.

[11] Lawton, W., Poston, T. and Serra, L., [1994] "Time-lag reduction in a medical Virtual Workbench". In Proc. Applications of Virtual Reality, British Computer Society conference, Leeds.

[12] Liang, J. and Green, M., [1993] "JDCAD: a highly interactive 3D modeling system". In Proc. Third International Conference on CAD and Computer Graphics, Beijing, China, pp. 217-222.

[13] Meiyappan, S., Poston, T., Heng, P.A., McVeigh, E.R., and Guttman, M.A. and Zerhouni, E.A., [1996] "Interactive Visualization for Speedy Non-invasive Cardiac Assessment", to appear in *IEEE COMPUTER,* January 1996 Special Issue on Computer Applications in Surgery and Therapeutic Procedures.

[14] McVeigh, E. R., Guttman, M., Poon, E., Chandrasekhar, P., Moore, C., Zerhouni, E.A., Meiyappan, S., and Heng, P, A. [1994] "Visualization and analysis of functional cardiac MRI data". In Proc. Medical Imaging 94, SPIE. 2168.

[15] Poston, T. and Serra, L. [1994] "The Virtual Workbench: Dextrous VR". In Proc. ACM VRST'94 - Virtual Reality Software and Technology, G. Singh, S.K. Feiner, D. Thalmann (eds.). World Scientific. Singapore (1994), pp. 111-122.

[16] Poston, T., Serra, L., Lawton, W., Chua, B.C., [1995] "Interactive tube finding on a Virtual Workbench". to appear in Second International Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS 95), Baltimore, Maryland, November 5-7

[17] Sachs, E., Roberts, A., Stoops, D., [1991] "3D-draw: a tool for designing 3D shapes". IEEE Computer Graphics and Applications, 11 (6), pp. 18-26.

[18] Schmandt, C., [1983] "Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station". Computer Graphics 17, pp. 253-259.

[19] Shaw, C. and Green, M. [1994] "Two-handed polygonal surface design". In Proc. User Interface Software and Technology conference, pp. 205-212.

[20] Serra, L., Ng H. and Fairchild, K.M., [1993] "Building Virtual Realities with Bricks". In Proc. Informatique'93, 2nd International Conference Interface to Real and Virtual Worlds, Montpelier, France, pp 101-109.

[21] Serra, L., Poston, T., Ng, H., Heng P.A., and Chua B.C., [1995] "Virtual Space Editing of Tagged MRI Heart Data". In Proc. CVRMed'95 - First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine Nice, France, pp. 70-76.

[22] Sowizral, H.A., [1994] "Interacting with virtual environments using augmented virtual tools" In Proc. Stereoscopic Displays and Virtual Reality Systems 1994, SPIE (1994), 2177, pp. 409-416.

[23] Waterworth, J. A. and Serra, L. [1994] "VR Management Tools: Beyond Spatial Presence". In CHI'94 Conference Companion, SIGCHI'94, pp. 319-320, New York: ACM.