14. Liang, J., Green, M., “JDCAD: A Highly Interactive 3D Modeling System,” *3rd International Conference on CAD and Computer Graphics*, Beijing, China, Aug. 1993, pp. 217-222.

15. Polhemus Navigation Sciences, Inc., P. O. Box 560, Colchester, VT 05446. (802) 655-3159.

16. Sachs, E., Roberts, A., Stoops, D., “3-Draw: A Tool for Designing 3D Shapes,” *IEEE Computer Graphics and Applications*, Nov. 1991, pp. 18-26.

17. Sears, A., Plaisant, C., Shneiderman, B., “A New Era for High Precision Touchscreens,” in Hartson and Hix, eds., Advances in Human-Computer Interaction, Vol. 3, 1992, pp. 1-33.

18. Sears, A., Shneiderman, B., “High Precision Touchscreens: Design Strategies and Comparisons with a Mouse,” *International Journal of Man-Machine Studies*, 34 (4), 1991, pp. 593-613.

19. Sellers, M., “Designing for Demanding Users,” *ACM Interactions*, July 1994, pp. 55-64.

20. Shaw, C., Green, M., Two-Handed Polygonal Surface Design, *ACM UIST’94 Symposium on User Interface Software & Technology*, pp. 205-212.

21. Snell, J. W., et al, “Segmentation of the Brain from 3D MRI Using a Hierarchical Active Surface Template,” *Proc. of the SPIE Conference on Medical Imaging*, 1994.

22. Trident Systems Inc., 10201 Lee Highway, Suite 300, Fairfax, VA 22030. (703) 273-1012.

For questions, further information, or comments, please contact:

Ken Hinckley
kph2q@virginia.edu
(804) 243-0310

Neurosurgical Visualization Laboratory
University of Virginia
Department of Neurosurgery
853 West Main Street, Suite 102
Charlottesville, Virginia 22903

We also are looking at techniques to facilitate use of our surgical planning software in the operating room. Compared to other input devices, touchscreens have attractive features for use in an operating room:

- They take up no desk real estate. Workstations in the operating room are typically mounted on small carts which barely have enough desk surface for a mouse and keyboard, making the mouse very difficult to use.
- They can be used by an operator who must remain sterile. By using a sterilizable stylus in conjunction with the touchscreen, the user can remain sterile even though the surface of the touchscreen itself may be unsterile. Some touchscreens can be fit with a gasket, making the entire surface of the screen sterilizable.
- They have no moving parts. Touchscreens are not prone to the mechanical failures which can plague mice and trackballs.

We have developed an early prototype of a touchscreen-driven 2D image viewing system for use in the operating room (*fig. 2, right*), but we are not yet using the prototype for operations on real patients.

Based on implemented prototypes of the systems we have discussed, and informal observation of users of these systems, we suggest that touchscreens offer not only intuitive 2D input which is well accepted by physicians, but that touchscreens also offer fast and accurate input which blends well with 3D interaction techniques.

7. REFERENCES

1. Conner, D., Snibbe, S., Herndon, K., Robbins, D., Zeleznik, R., van Dam, A., “Three-Dimensional Widgets,” *1992 Symposium on Interactive 3D Graphics*, pp. 183-188.

2. Elographics, 105 Randolph Road, Oak Ridge, TN 37830. (615) 482-4100.

3. Feiner, S., MacIntyre, B., Haupt, M., Solomon, E., “Windows on the World: 2D Windows for 3D Augmented Reality,” *Proc. ACM Symposium on User Interface Software and Technology*, 1993, pp. 145-155.

4. Goble, J., Hinckley, K., Snell, J., Pausch, R., Kassell, N., “Two-handed Spatial Interface Tools for Neurosurgical Planning,” *IEEE Computer*, special issue on virtual reality (to appear), 1995.

5. Goble, J., Snell, J., Hinckley, K., Kassell, N., “A Real-Time System for 3D Neurosurgical Planning,” *Proc. Visualization in Biomedical Computing 1994*, published by SPIE.

6. Herndon, K., Zeleznik, R., Robbins, D., Conner, B., Snibbe, S., van Dam, A., “Interactive Shadows,” *Proc. ACM Symposium on User Interface Software and Technology*, 1992, pp. 1-6.

7. Herndon, K., Meyer, T., “3D Widget for Exploratory Scientific Visualization,” *Proc. ACM Symposium on User Interface Software and Technology*, 1994, pp. 69-70.

8. Hinckley, K., Pausch, R., Goble, J. C., Kassell, N. F., “A Three Dimensional User Interface for Neurosurgical Visualization,” *Proc. of the SPIE Conference on Medical Imaging*, 1994.

9. Hinckley, K., Pausch, R., Goble, J. C., Kassell, N. F., “Passive Real-World Interface Props for Neurosurgical Visualization,” *Proc. ACM CHI’94 Conference on Human Factors in Computing Systems*, 1994.

10. Hinckley, K., Pausch, R., Goble, J. C., Kassell, N. F., “A Survey of Design Issues in Spatial Input,” *Proc. ACM Symposium on User Interface Software and Technology*, 1994, pp. 213-222.

11. Karl, L., Pettey, M., Shneiderman, B., “Speech-Activated versus Mouse-Activated Commands for Word Processing Applications: An Empirical Evaluation,” *International Journal of Man-Machine Studies*, 1993.

12. Kelly, P. J., Kall, B. A., eds., Computers in Stereotactic Neurosurgery, Blackwell Scientific Publications Inc., Cambridge, MA, 1992.

13. Lasseter, J., “Principles of Traditional Animation Applied to 3D Computer Animation,” *Computer Graphics (Proc. ACM SIGGRAPH ‘87)*, 21 (4), July 1987, pp. 35-44.

The function *b* is defined as follows:

$$b(C_{x_{i-1}}, T_{x_i}) = C_{x_{i-1}} + \Delta_x \left[(\Gamma_{\mathrm{MAX}} - \Gamma)\, slowIO(u, \beta) + \Gamma \right]$$

The *slowIO* (“slow-in, slow-out”) function is a function traditionally used in animation [13] to make objects look as if they are moving realistically (*fig. 3*) The function just returns a value between 0 and 1 depending on the percentage of β that u represents. In the above equation, the slowIO function is being used to smoothly generate values between Γ and Γ_{MAX} as Δ moves over the B region. Γ is a gain factor that attenuates cursor movement in the B region.

On a 19” 1280x1024 pixel monitor, we have found empirically that values of $\alpha = 3.0$, $\beta = 30.0$, $\Gamma = 0$ and $\Gamma_{MAX} = 0.10$ yield good results.

```
float slowIO(float elapsedTime, float duration) {
  float A = 0.3; float B = 0.8; float x = elapsedTime / duration; float y;
  if (x < A) {
    float a1 = (B - 1)/(A *  (B * B - A * B + A - 1));
    y = a1 * x * x;
  }
  else if (x > B) {
    float a3 = 1 / (B * B - A * B +  A - 1);
    float b3 = -2 * a3;
    float c3 = 1 + a3;
    y = a3 * x * x + b3 * x + c3;
  }
  else {
    float m = 2 * (B - 1) / (B * B - A * B + A - 1);
    float b2 = -m * A / 2;
    y = m * x + b2;
  }
  return (y);
}
```

Figure 6: **C language code that we use to implement the slowIO function.**

Based on our informal user observations to date, our stabilization algorithm allows targets as small as 1.08 mm x 1.08 mm to be easily selected by novices, and makes possible selection of targets as small as 0.27 mm x 0.27 mm after some training. These observations were made on test users who touched the screen with their bare fingers (and not with a stylus or other implement).

6. CONCLUSION AND FUTURE WORK

We envision a variety of improvements and new capabilities to the hybrid props-based interface. The link between the props interface and our volume rendering software is still somewhat cumbersome, so we are working to further integrate these tools. One promising innovation is to volume-render a subset of the voxels on the surface of the brain, allowing use of the interface props at interactive frame rates on a much more detailed brain surface than our current polygonal approximation. We also plan to integrate our stereotactic planning software with the props-based interface and to improve the trajectory planning capabilities. Currently, these tasks must be performed externally to the interface by another program.

We also plan to further explore the potential design space of hybrid interface techniques. The interface props might serve as 3D input, constrained 3D input, and 2D input devices depending on how and when they have touched the screen. For example, to navigate through slices adjacent to a 2D slice shown on the screen, one might touch the slice with the cross-sectioning prop, and then move the prop back and forth relative to the screen to browse through the slices. The resulting behavior might also vary depending on which prop is used to touch the screen. To extend the previous example, if the trajectory prop were used to touch the slice instead, this could control selection of the trajectory target point instead of browsing through slices.

directly to the new touch location. The net effect is that small movements have a damping factor applied which allows more precise control, but rapid movements (such as dragging an object across the screen) follow the user's finger directly.

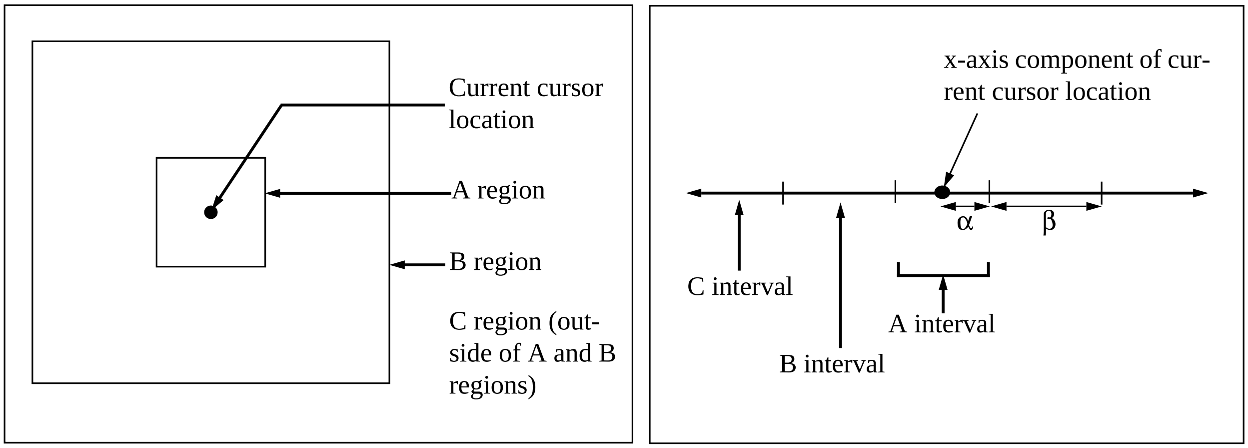


Figure 5: **(Left) Regions used by Sears's touchscreen stabilization algorithm [18]. (Right) We separate the stabilization along the x and y axes, and use intervals which correspond to Sears's A, B, and C regions.**

We found that for precise pointing tasks, it may be advantageous to uncouple the x-axis and y-axis completely (*fig. 5, right*), so that instead of having A, B, and C regions, we define A, B, and C intervals along each axis. (In the figure, only the B and C intervals on the left side of the A interval are labelled, but of course there are corresponding B and C intervals on the right side as well.) This has the effect of allowing the user to independently make adjustments to the x or y coordinate of a touch location during precise pointing.

Sears only describes his algorithm in general, so many of the implementation details are left unspecified. We believe the following formal description of our algorithm will prove helpful to persons wishing to experiment with touchscreen stabilization. The touchscreen stabilization function S specifies the new cursor position C_i as a function of the previous cursor position C_{i-1} and the current touch position T_i:

$$S(C_{i-1}, T_i) \rightarrow C_i$$

Note that T_i is the integral (x, y) pixel coordinate as returned by the touchscreen hardware, but that C_i should actually be computed as a floating point coordinate pair. Of course, the cursor can only be displayed at integral pixel coordinates, but retaining fractions of pixels in C_i is important as this allows for subtle effects caused by fractions of pixels adding up over time, helping to produce more controlled motion.

Below, we give the equations used for stabilization along the x-axis; the equations along the y-axis are identical. The parameter Δ_x is just the distance between T_i and C_{i-1} along the x axis:

$$\Delta_x = T_{x_i} - C_{x_{i-1}}$$

S_x, the stabilization function along the x axis, is then defined as:

$$S_x(C_{i-1}, T_i) = \begin{cases} \Delta_x < \alpha \rightarrow & C_{x_{i-1}} \\ \Delta_x < (\alpha + \beta) \rightarrow & b(C_{x_{i-1}}, T_{x_i}) \\ \Delta_x \geq (\alpha + \beta) \rightarrow & T_{x_i} \end{cases}$$

with the initial condition that $C_0 = T_0$ and with constants α and β as shown in fig. 5 (right).

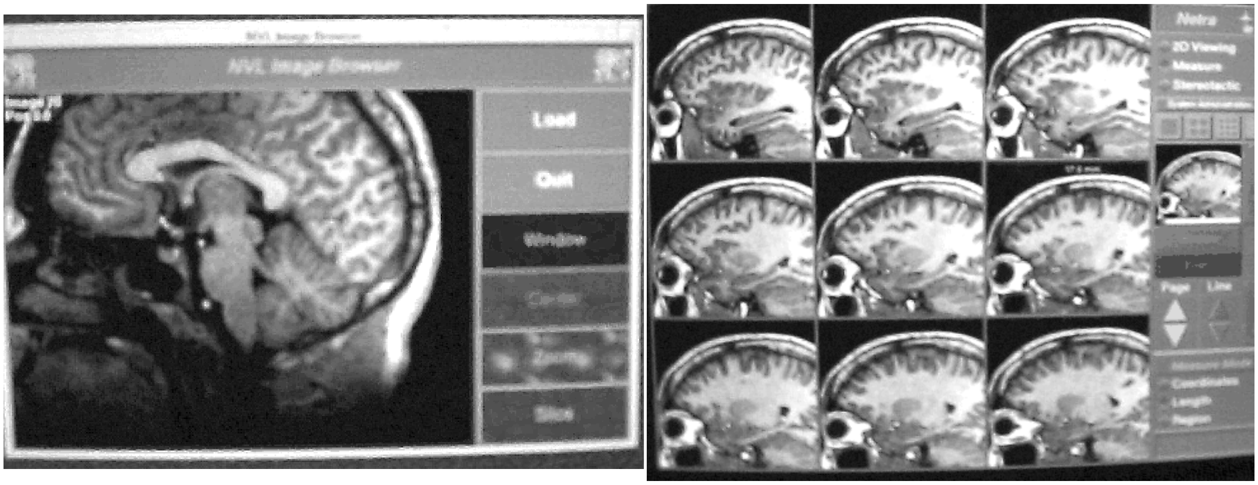


Figure 4: **(Left) Image browser utility designed to allow quick and simple previewing of data sets using the touchscreen. (Right) Our 2D image viewing tool, with touchscreen-driven graphical user interface. In ongoing research, we are looking to use these tools in the surgical operating room by using a sterilizable probe in conjunction with the touchscreen.**

4. SYSTEM COMPONENTS

We are currently using an Elographics [2] Surface Acoustic Wave (SAW) touchscreen with an X windows driver from Trident Systems, Inc. [22]. In addition to the usual (x, y) coordinate information, the SAW touchscreen also detects pressure information, which it approximates by determining the surface area of the touch on the screen. This allows the user to position the cursor with a light touch, and then activate (for example) buttons on the screen by pressing down. Advantages of this touchscreen include its high transmissivity (many other touchscreens significantly attenuate the light coming from the monitor) and its ability to detect pressure.

The major disadvantage we see results from parallax errors. The touchscreen surface is offset slightly from surface of the monitor, which can make the touchscreen difficult to use if the user is not directly in front of it. A second potential disadvantage of the SAW touchscreen is that only soft objects can be used to touch it. We have modified our interface props to work with the SAW touchscreen, but the materials used in the props do not work quite as well as using a bare finger. We are planning to evaluate our interfaces with resistive membrane touchscreens (also manufactured by Elographics), which have lower transmissivity than the SAW touchscreen but do not suffer from the above problems, to see which touchscreen technology best suits our needs.

The SAW touchscreen is fitted to a 19” 1280x1024 pixel cylindrical monitor (each pixel is approximately 0.27 mm^2). The system software runs on a Hewlett Packard 735 workstation with a CRX48Z polygonal graphics accelerator. We track the 3D interface props using the Polhemus FASTRAK six-degree-of-freedom magnetic tracker [15].

5. TOUCHSCREEN STABILIZATION

In the past it has often been assumed that targets smaller than the tip of one’s finger cannot be selected using a touchscreen, but previous work by Sears [17][18] has demonstrated that if the touchscreen returns visual feedback in the form of a small cursor which appears above the user’s finger, much smaller targets can be selected.

Sears uses a scheme which divides the screen into three regions which surround the current cursor location (*fig. 5, left*). If a new touch falls in the innermost, or ‘A’ region, surrounding the cursor, the cursor does not move. If the new touch falls into the ‘B’ region surrounding the A region, the cursor moves a fraction of the distance between the current cursor location and the new touch location. If the new touch falls into the outermost ‘C’ region, the cursor moves the touchscreen in conjunction with the 3D props, surgeons have all the capabilities of the advanced visualization software which we have developed at their fingertips.

From our preliminary observations of users of the hybrid interface, we find that there are a variety of strategies which can be used to touch the screen. Some users prefer to touch the screen with a knuckle or with a finger which is not currently being used to grasp the tools. Other users prefer to use the input prop as an extension of their hand, and touch the screen directly with the tools themselves. The key point is that users do not have to think about how to access the 2D interface controls, but rather can choose a gesture which comes naturally and focus on the task at hand.

3.2.Alternatives to the touchscreen for 2D input

We have not done side-by-side comparisons of the touchscreen with other candidates for hybrid input, such as mice or voice input, but we have experimented with both mice and voice input [8][9] as adjuncts to the props-based interface. While we believe it is clear that using a mouse in conjunction with the 3D interface props is unacceptable, it is less clear whether or not voice input might provide a more intuitive input medium for physicians. Based on our experience to date with voice input and touchscreens, we have three main arguments against voice input:

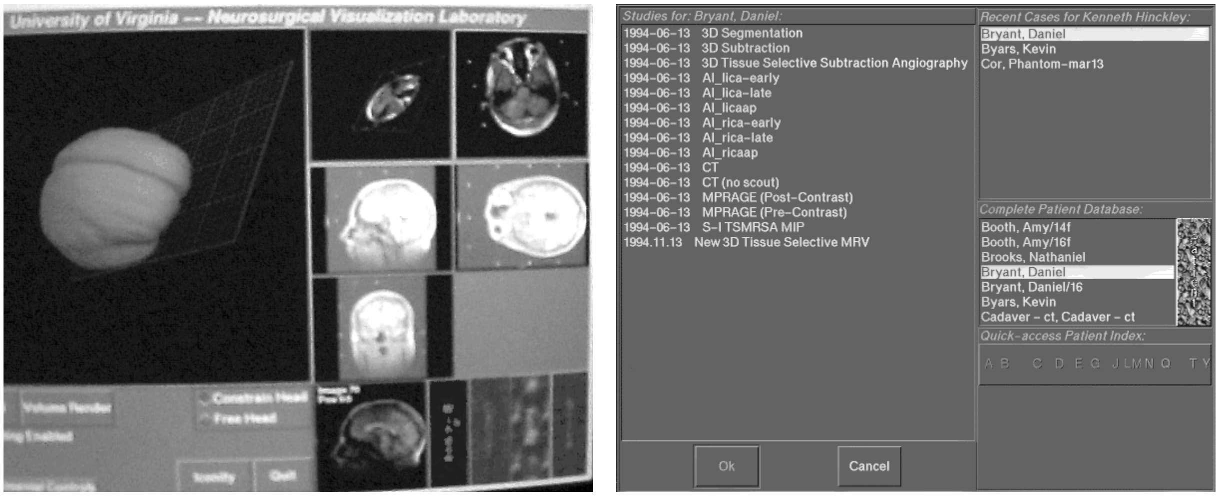
- *Range of hybrid tasks supported*. Voice input facilitates discrete command-based tasks, but it is more difficult to translate direct manipulation tasks which require motor/visual coordination to a voice interface. For example, consider the task of panning an image. Voice commands for “pan left,” “pan right,” “pan up,” and “pan down” could be used, but the interface must then pan the image in discrete jumps. Alternatively, the various “pan” voice commands could cause the image to start moving in the desired direction, but then another command must be spoken to stop the movement. Furthermore, the voice command must somehow indicate which image on the screen is to be panned. Already, the user must remember six voice commands just to pan images. When center/window, zooming, slice navigation, database manipulation, and other commands are added, the interface quickly becomes unwieldy. These same tasks translate readily to a touchscreen interface without imposing a cognitive burden.
- *Cost and reliability.* Speaker-independent and speaker-adaptive speech recognition systems are currently available, but such systems are still much more expensive than touchscreens. Speaker-dependent systems are less expensive, but physicians do not have time to train the system, even for a small vocabulary of commands. The recognition must also be reliable, as physicians find recognition errors extremely annoying.
- *Possible cognitive difficulties*. If users must verbalize their intentions to perform actions, there are indications that under some conditions this can interfere with short term memory [11]. Also, the manipulation of the props is a physical and gestural action; in our experience, this style of interaction does not seem to mix as well with verbal input as it does with touchscreen input.

An interesting question is whether voice input would be a useful adjunct to an interface which already incorporates both 3D props and a touchscreen. If the cost and reliability questions outlined above could be addressed, the ability for the physician to speak commands, in addition to the gestural 2D and 3D input, might prove useful. We plan to experiment with speaker-adaptive or speaker-independent voice recognition systems in this capacity in the future.

3.3.Touchscreen-driven 2D surgical planning and visualization tools

We have also been experimenting with a variety of 2D image viewing applications with touchscreen interfaces. The focus in these designs has been on simplicity, rather than on providing every possible image analysis algorithm. These are tools which are meant to be pulled up quickly by the surgeon to preview image data sets during surgical planning or to revise surgical plans in the operating room when the need arises.

For some simple types of procedures, the surgeon does not need 3D planning capabilities, but still wants the convenience of manipulating digital images. We have developed prototype touchscreen-driven slice-by-slice 2D surgical planning applications (*fig. 4*), and in ongoing research, we are looking to transition these tools to the surgical operating room by using a sterilizable probe in conjunction with the touchscreen (see section 6).



Studies for: Bryant, Daniel:

1994-06-13	3D Segmentation
1994-06-13	3D Subtraction
1994-06-13	3D Tissue Selective Subtraction Angiography
1994-06-13	AI_lica-early
1994-06-13	AI_lica-late
1994-06-13	AI_licaap
1994-06-13	AI_rica-early
1994-06-13	AI_rica-late
1994-06-13	AI_ricaap
1994-06-13	CT
1994-06-13	CT (no scout)
1994-06-13	MPRAGE (Post-Contrast)
1994-06-13	MPRAGE (Pre-Contrast)
1994-06-13	S-I TSMRSA MIP
1994.11.13	New 3D Tissue Selective MRV

Recent Cases for Kenneth Hinckley:

Bryant, Daniel
Byars, Kevin
Cor, Phantom-mar13

Complete Patient Database:

Booth, Amy/14f
Booth, Amy/16f
Brooks, Nathaniel
Bryant, Daniel
Bryant, Daniel/16
Byars, Kevin
Cadaver – ct, Cadaver – ct

Quick-access Patient Index:

A B C D E G J L M N Q T Y

Figure 2: **(Left) Current prototype of touchscreen graphical user interface for use with the 3D interface props. In addition to the 3D operations supported by the props, the user can examine cross-sections of the data set in the standard orthogonal planes by touching the interface controls in the lower right. (Right) Graphical user interface for the touchscreen to allow loading of patients from an image database. The database maintains a cache of recent cases, allowing the surgeon to quickly access patient data sets that he has been working with.**

The center/window, zoom, and slice navigation controls (*fig. 3, right*) are designed to behave like the thumb-wheels which are found on many 2D medical image viewing consoles. When the user touches these controls, the background textures slide up and down, giving immediate visual feedback. In addition to this feedback, the textures also add visual appeal, inviting the physician to explore the interface and try it out. In previous work with 2D medical imaging interfaces, Sellers [19] proposed a “three foot/five-minute” heuristic for initial user acceptance; the idea was that the interface had to be attractive enough to get a physician within three feet of the computer, and had to be simple enough to allow the physician to perform a meaningful task within five minutes. The novelty of the 3D props interface, and the visual appeal of the textures and other 2D interface components, lures physicians within the three foot zone and invites them to try out the interface.

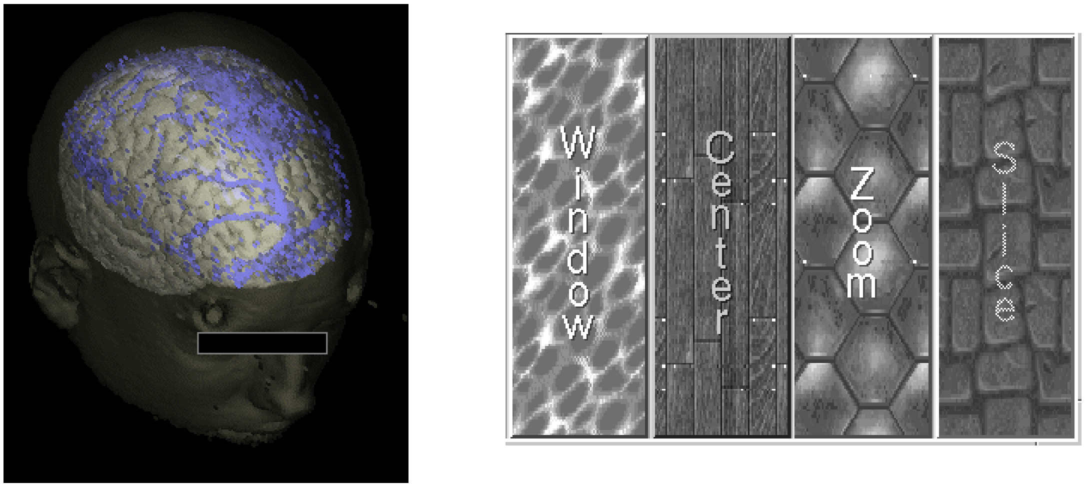


Figure 3: **(Left) Detailed volume rendering of a recent patient data set. (The patient’s identity has been intentionally concealed). (Right) Users can slide textured wheels on the touchscreen to adjust image attributes.**

The surgeon uses a combination of 3D and 2D manipulation facilities when planning a surgery. There is not a distinct “3D interaction phase” followed by a distinct “2D interaction phase.” Rather there is a continuous dialog which combines 2D and 3D visualization tools to accomplish the surgical plan. Using a mouse for 2D input does not facilitate this style of dialog. In our experience surgeons are hesitant to use mice in the first place; when manipulating the props in addition to the mouse, in practice the surgeon would rely on someone else to move the mouse for him. But using for surgeons, we believe that mouse-based 3D interaction techniques will not be well accepted, since surgeons are not accustomed to using mice or widgets of any sort, and resist using mice even for standard 2D input tasks.

Feiner [3] has constructed a system which combines a 3D augmented reality head-mounted display with a standard 2D desktop display. This allows the user to see 3D virtual objects arranged around his head, but does not block out the user's view of the real world, allowing standard mice, keyboards, and CRT displays to be used.

In work at the University of Alberta, Shaw [20] has also encountered the hybrid input problem, which he refers to as *device fusion*. Shaw's system for polygonal surface design and CAD tasks uses two small hand-held "bats" for input devices. Each "bat" is a six-degree-of-freedom magnetic tracker augmented with three small push-buttons, which can be pushed to indicate various commands.

Shaw reports that this solution works well for this specific interface, but we see two potential difficulties which could limit application of this solution to other 3D interfaces. First, in our experience, having fixed buttons on a 3D input device can sometimes make the device awkward and fatiguing to control, since the user must hold the device in a static posture in order to maintain contact with the button. Second, the solution does not scale well. There are only three buttons on each tracker, so only a few discrete commands can be accessed in this way, and the user cannot easily perform actions which require 2D direct manipulation or dragging.

Shaw's interface also incorporates a *ring menu* [14], which is a pop-up menu which can be accessed in 3D using a tracker. The menu appears as a ring surrounding the user's 3D cursor on the screen; by rotating the tracker, the user can select different items which are highlighted in the ring. Again, this solution has limited scale, as only a few items can be shown in the ring.

We believe that using a touchscreen could offer a more general solution to the hybrid interface problem than these techniques. The scale of the interface is limited only by available screen real-estate, and by the degree of complexity which users find comprehensible. Furthermore, the touchscreen can support a wider class of 2D interaction techniques, including not only the discrete command selection and menu selection supported by the above techniques, but also 2D point selection and other direct manipulation or dragging operations which require motor/visual coordination.

3. TOUCHSCREEN INTERFACES FOR SURGICAL PLANNING AND VISUALIZATION

Our current prototype of the hybrid user interface supports 3D manipulation using a set of interface props (*fig. 1*), including a *head prop*, a *cross-sectioning prop*, and a *trajectory prop* [4][8][9]. The head prop is a small doll's head which can be held comfortably in one hand. Rotating the prop causes a polygonal model of the patient's brain to rotate correspondingly on the screen. The user can also control the image zoom factor by moving the prop towards or away from his or her body.

The cross-sectioning prop is a rectangular plate used to specify the position and orientation of an arbitrary slice through the patient's anatomy. The user holds the plate against the head to indicate a cross section of the brain data, and can explore the patient MRI data by interactively sweeping the plate through the volume.

The trajectory prop is a stylus-shaped tool which allows the surgeon to specify 3D vectors and points. Moving the trajectory prop relative to the head prop specifies the position and orientation of a cylindrical virtual probe relative to the polygonal brain model.

3.1.Description of prototype hybrid 2D+3D interface

We have augmented this 3D interface with a touchscreen and a variety of 2D image viewing (*fig. 2, left*) and patient database browsing controls (*fig. 2, right*). Once the surgeon selects the desired cross-sections and trajectories in 3D, the surgeon often wishes to refer back to 2D image slices taken in the standard sagittal, axial, and coronal orientations. The current interface allows the surgeon to examine these cross-sections by using the touchscreen to pan and zoom the images, to adjust center/window controls, and to perform slice-by-slice navigation. At any time, the surgeon can also generate a high-quality volume rendering (*fig. 3, left*), which takes approximately five seconds to generate, by touching a button labelled "Volume Render" in the interface.

In previous work we have implemented a 3D user interface [4][8][9] based on the two-handed physical manipulation of hand-held tools or “props” in free space (*fig. 1, left*). From the surgeon’s perspective, this interface is analogous to holding a miniature head in one hand which can be “sliced open” or “pointed to” using a clear plastic cross-sectioning plane or a stylus tool, respectively, held in the other hand. In informal evaluation of the 3D props-based interface with over 50 neurosurgeons and hundreds of non-neurosurgeons, we find that users can understand and use the interface within one minute of touching the props.

The props-based interface excels for 3D manipulation, but when 2D sub-tasks such as loading a new patient from an image database arise, there is an awkward pause in the human-computer dialog while the user must put down the interface props to move the mouse or to use the keyboard. The problem lies in a 3D input versus 2D input dichotomy: some user tasks are best done in 3D, others are better suited to 2D, and there is no intuitive and consistent mechanism for switching between the different styles of input. Users are distracted from the underlying problem they are trying to solve because they must decide which device is best to use for a given input task.

We propose a new interface technique which augments the 3D props-based interface with a touchscreen. This *hybrid interface* intuitively and seamlessly combines 3D input with more traditional 2D input in the same user interface. Note the ergonomic facility with which a touchscreen can be used: the surgeon can move in 3D using the props, and then, without having to put the props down, the surgeon can reach out and touch the screen to perform 2D tasks (*fig. 1, right*). Furthermore, this strategy removes the cognitive load typically associated with switching input devices. One interacts gesturally with the props to perform 3D operations; one interacts gesturally with the touchscreen to perform 2D operations. From the user’s perspective, one is always interacting gesturally with objects in the real environment, and the user may not even be aware that 3D gestures are being digitized with a magnetic tracker while 2D gestures are being digitized with a touchscreen.

In the context of this hybrid interface we find that the touchscreen compares favorably to other techniques which could be used for 2D input, such as using a mouse or voice input. In the case of the mouse, it is annoying and laborious to put down the 3D interface props to move the mouse. Voice input provides a better solution than the mouse for discrete command-based tasks, but tasks which require motor/visual coordination, such as adjusting image contrast, are difficult to perform with voice recognition. These same motor/visual tasks translate easily to a touchscreen interface.

As part of this work, we have also been investigating ways of improving the speed and accuracy with which users may select small targets using a touchscreen. In the past it has often been assumed that targets smaller than the tip of one’s finger cannot be selected using a touchscreen, but as previous work by Sears [17][18] demonstrated, by providing users with visual feedback in the form of a small cursor which appears above the finger, much smaller targets can be selected. We have implemented a stabilization algorithm based on Sears’s results [18], and based on our informal user observations to date, our stabilization algorithm allows targets as small as 1.08 mm x 1.08 mm to be easily selected by novices, and makes possible selection of targets as small as 0.27 mm x 0.27 mm after some training.

We are also currently exploring use of the touchscreen for interaction with imaging software in the operating room. Touchscreens take up no desk space, can be used by an operator who must remain sterile, and contain no moving parts, making them reliable. These can all be problems for other 2D input devices.

2. PREVIOUS WORK

Most previous work in 3D interaction has either ignored the problem of adding 2D input or has integrated it in an ad hoc manner. Some work in the virtual reality and free-space 3D interaction literature [10] has mentioned the ergonomic and cognitive difficulties of fusing 3D and 2D input devices into a single coherent user interface, but no published papers we are aware of treat this topic in depth.

Work at Brown University [1][6][7] has looked at ways of using mouse-controlled “3D Widgets” for 3D interaction. Here the problem of combining the 3D and 2D interfaces is obviated, as the mouse is used for all input tasks. This advantage of using the mouse for 3D interaction should not be underestimated, but in the case of 3D user interfaces

To appear in: Proceedings of SPIE Medical Imaging '95:
Image Capture, Formatting, and Display, February, 1995

New Applications for the Touchscreen in 2D and 3D Medical Imaging Workstations

Ken Hinckley, John C. Goble, Randy Pausch, Neal F. Kassell
University of Virginia, Charlottesville, VA 22903
kph2q@virginia.edu, jcg7q@virginia.edu, pausch@virginia.edu

ABSTRACT

We present a new interface technique which augments a 3D user interface based on the physical manipulation of tools, or *props*, with a touchscreen. This *hybrid interface* intuitively and seamlessly combines 3D input with more traditional 2D input in the same user interface. Example 2D interface tasks of interest include selecting patient images from a database, browsing through axial, coronal, and sagittal image slices, or adjusting image center and window parameters. Note the facility with which a touchscreen can be used: the surgeon can move in 3D using the props, and then, without having to put the props down, the surgeon can reach out and touch the screen to perform 2D tasks.

Based on previous work by Sears, we provide touchscreen users with visual feedback in the form of a small cursor which appears above the finger, allowing targets much smaller than the finger itself to be selected. Based on our informal user observations to date, this touchscreen stabilization algorithm allows targets as small as 1.08 mm x 1.08 mm to be selected by novices, and makes possible selection of targets as small as 0.27 mm x 0.27 mm after some training.

Based on implemented prototype systems, we suggest that touchscreens offer not only intuitive 2D input which is well accepted by physicians, but that touchscreens also offer fast and accurate input which blends well with 3D interaction techniques.

1. INTRODUCTION

Software usability is crucial to getting neurosurgeons to actually use advanced visualization software in the clinical routine, and to use such software with reliable results. We therefore seek to design interaction techniques which facilitate use of the software by surgeons, without need for a technical assistant. The manipulative capabilities of common input devices such as mice and keyboards are poorly matched to the existing skill sets in neurosurgeons and to the volumetric manipulation and visualization tasks required for surgical planning. Trying to express the style of interface which he might find usable, one surgeon commented "I want a skull I can hold in my hand."

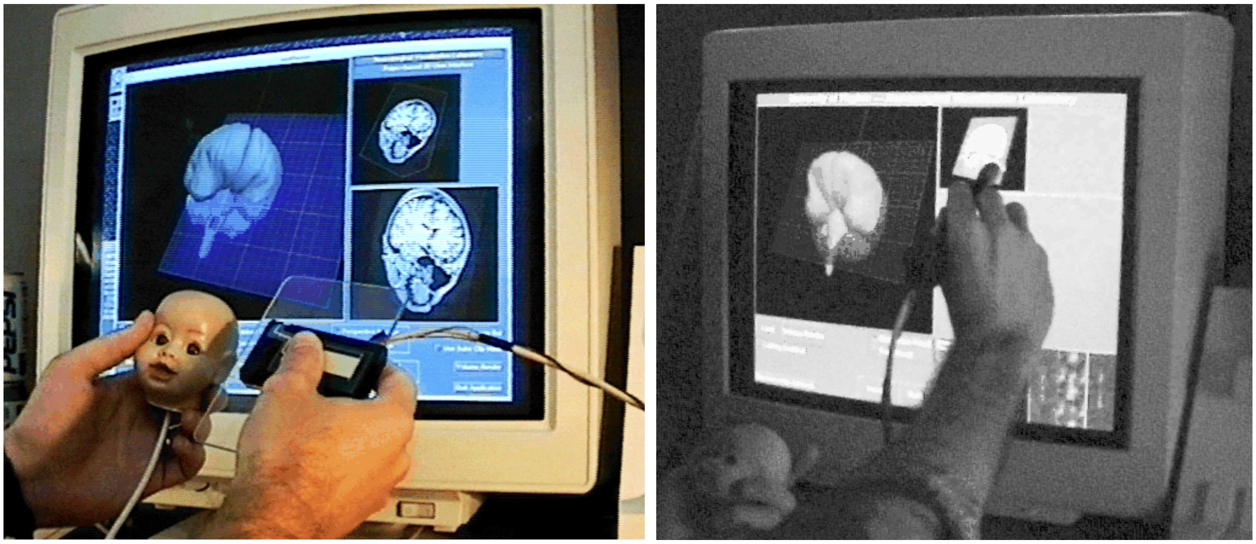


Figure 1: **(Left) User selecting a cutting plane using the 3D props-based interface. (Right) The monitor is equipped with a touchscreen, allowing the surgeon to easily reach out and touch the screen to control 2D image viewing tasks in addition to 3D manipulation. Here the operator uses the 3D trajectory stylus to touch the screen.**