A Hybrid User Interface for Manipulation of Volumetric Medical Data

Alexander Bornik[*] Reinhard Beichel Ernst Kruijff Bernhard Reitinger Dieter Schmalstieg

Institute for Computer Graphics and Vision
Graz University of Technology

Abstract

This paper presents a novel system for interactive visualization and manipulation of medical datasets for surgery planning based on a hybrid VR / Tablet PC user interface. The goal of the system is to facilitate efficient visual inspection and correction of surface models generated by automated segmentation algorithms based on x-ray computed tomography scans, needed for planning surgical resections of liver tumors. Factors like the quality of the visualization, nature of the dataset and interaction efficiency strongly influence system design decisions, in particular the design of the user interface, input devices and interaction techniques, leading to a hybrid setup. Finally, a user study is presented, which characterizes the system in terms of method efficiency and usability.

CR Categories: C.2.4 [Computer-Communication Networks]: Distributed Systems—Distributed Applications; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary Representations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; J.3.2 [Medical information systems]: Project and People Management—Life Cycle

1 Introduction

Imaging modalities like X-ray computed tomography (CT) or magnetic resonance tomography (MR) are important information sources for surgical planning. Proper planning requires physicians to understand the 3D relations within the dataset. For example, the resection of liver tumor requires understanding the arrangement of liver tissue, vasculature and tumor. Looking at individual 2D slices of CT data using conventional radiological workstation software makes this task difficult. Our aim was therefore to build a system for liver surgery planning using Virtual Reality (VR) techniques, capable of supporting radiologists and surgeons by providing visualization of 3D medical models and tools for computer-assisted planning of the surgical intervention.

Typically, the first step in liver surgery planning is segmentation of the individual structures, needed to plan the surgical intervention. This task can be done manually, but this is tedious and time consuming, since it involves drawing contours on several hundred slices.

However, a fully automated segmentation of the liver is difficult to achieve, because the shape of the human liver highly varies. This fact makes it almost impossible to use a priori shape knowledge for the design of a segmentation algorithm. In addition, the gray-value appearance can show large variations due to pathological changes of the liver, which can cause problems in distinguishing the liver from adjacent organs with similar gray-values (e.g. heart or colon). Furthermore, tumors located close to the liver boundary might be excluded from the segmentation.

In advanced automatic segmentation algorithms, the segmentation problems are usually limited to local errors, while most areas of the liver boundary can be correctly found using the automatic algorithms. A radiologist's task can therefore be simplified from manual contour specification to interactively correcting errors in segmented datasets. This *segmentation refinement* approach is expected to be much less time consuming in most cases.

At a first glance 3D segmentation refinement tools afford VR techniques: Stereoscopic visualization provides good 3D perception of the dataset, whereas tracked input devices allow for direct 3D interaction with the dataset. However, 2D screens have a much higher resolution than their 3D counterparts, and an inexpensive optical mouse easily outperforms high-end tracking devices in terms of accuracy when precision input in 2D is required. In the medical field, where imprecision may have dire consequences, the virtues of established 2D techniques should not be discarded lightly. Moreover, physicians are used to desktop interfaces, and in particular for system control, VR interfaces are not yet mature.

These considerations lead us to the design of a hybrid user interface that combines multiple display and interaction techniques, in order to match the work processes at hand. The objective of the hybrid user interface is to pair 3D perception and direct 3D interaction with 2D system control and precise 2D interaction. For such an interface, it is important that the flow of action of working between 2D and 3D visualization and interaction techniques is not disturbed. Both the different views and the interaction with the data need to be handled coherently.

To ease the transition between the interface modalities, a hybrid input device, which can be conveniently used in all 2D and 3D tasks, was designed and developed. A focus was put on analyzing the differences between action performance in the 2D and 3D domain, leading to a more extensive human factors study. This paper presents results on the complexity of tasks and their associated tools, and the duration of usage in reflection to ergonomics.

2 Related Work

The three-dimensional nature of surgery planning and surgery simulation has led researchers to the use of VR techniques. An overview of VR systems and human interface issues in the medical context can be found in [21] and [9]. Liver surgery planning in particular has been addressed by a number of groups, although VR aspects are rudimentary in most projects.

The German cancer research center (DKFZ) located in Heidelberg has developed a computer-aided planning system for liver surgery [20]. Research was focused on medical image processing tasks such as segmentation. There have also been attempts to deal with segmentation errors described in [16], although the VR aspects of the system are limited, since all image segmentation and planning procedures are performed on a normal desktop PC.

The Center for Medical Diagnostic Systems and Visualization (MeVis) in Bremen has developed a desktop-based liver surgery planning system. The main research focus was segmentation and modeling of liver structures [15].

Researchers at INRIA have addressing several aspects of liver surgery planning, such as the segmentation of the liver surface using

[*] e-mail:bornik@icg.tu-graz.ac.at

deformable surface models. Later the group worked on surgical simulation with realistic liver tissue models using force feedback input devices [8]. The virtual liver surgery planning system [3], which is the foundation of the work presented in this paper, has been developed at Graz University of Technology since 2000. An earlier version of the VR based segmentation refinement toolset was described in [4].

Segmentation refinement is a rather new field and there are hardly any publications dealing directly with it. 2D segmentation can be trivially implemented as a painting tool, but this approach is ineffective for large 3D datasets. Interactive segmentation techniques can be seen as closely related, although they address the segmentation problem and not the correction of erroneous segmentation. An interactive segmentation approach named *Live-Wire* was introduced in [1]. It reduces the amount of user interaction required to segment the object boundary. An extension to 3D data can be found in [10]. To our knowledge, the desktop based interactive approach described in [19] is the only method based on the segmentation refinement principle. An application in the context of data preparation for liver surgery was reported in [2].

3D interaction with medical volumetric datasets is a reoccurring topic in VR. Some input devices have been developed that specifically focus at exploring medical data, including Hinckley's prop-based system [18] or the Fakespace CubicMouse [13].

Hybrid user interfaces in general are an emerging research field [11]. Most developments focus at combining different visual displays, like in mixed reality setups [23]. Only a few have focused at truly hybrid input, such as the Virtual Tricorder [27], the Pick-and-Drop approach by Rekimoto [22], and some tangible user interfaces [26]. Handhelds and touch screens have been integrated into immersive environments for interaction purposes, like [12], [27], [14] and [6]. These handhelds are mostly used for GUI-style control elements (system control), only little work is done on direct manipulation of immersive data. Similar to handhelds, tablet interfaces have been designed for the display of 2D data and menus on a flat surface, for example [25]. Finally, a frequent approach to system control in immersive environments are pen-like devices, such as the Stylus products from Polhemus or Intersense. For an extensive overview of pen devices, please refer to [5].

3 Hybrid User Interface

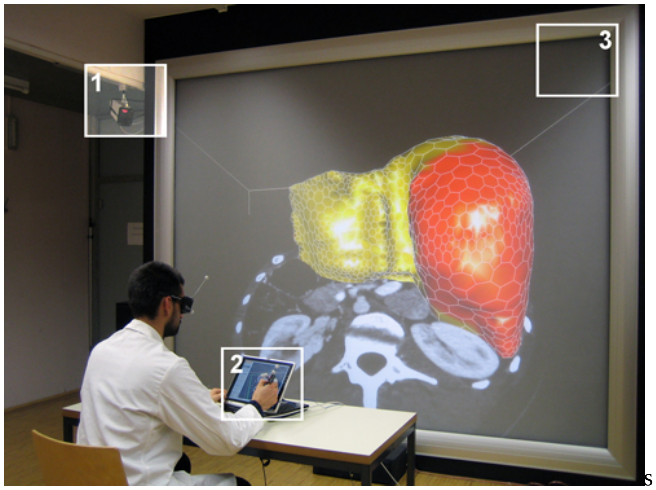

Figure 1: Hybrid Setup: camera of the optical tracking system (1), Tablet PC and *Eye of Ra* (2), stereoscopic large screen projection system (3).

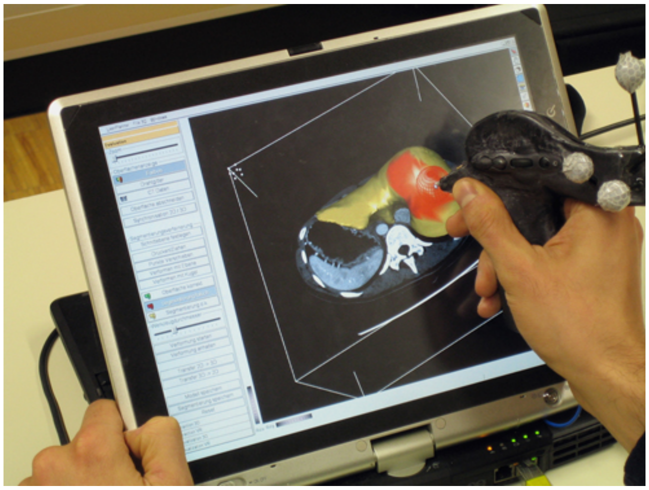

Figure 2: Desktop Setup: Tablet PC with conventional 2D User interface for system control; 2D view for viewing and interaction with the dataset using the *Eye of Ra* input device, which behaves similar to a conventional stylus in the desktop setup.

3.1 Hardware Setup

The hardware setup consists of two main parts, the VR system and the 2D system. The VR system's display is a large stereo wall (stereoscopic back projection screen, 375cm diameter, 1280x1024 pixels) viewed with shutter glasses. A Barco Galaxy 3-chip DLP projector provides high quality active stereo rendering with very good channel separation, which is important when displaying virtual objects close to the user. The stereo wall is driven by a PC workstation (dual 3GHz Xeon, NVidia Quadro FX 3400). Optical tracking of the user's head and the input device is done using an 4-camera infrared system from Advanced Realtime Tracking.

The desktop system is a Tablet PC (Toshiba Portégé M200, 1.8 GHz CPU, GeForce Go 5200 graphics card, 12-inch TFT touchscreen at 1400x1050 pixels). The Tablet PC is placed on a desk approximately 2 meters in front of the screen, tilted at approximately 60 degree for convenient readability. The user is seated at the desk so that both stereo wall and Tablet PC are within the field of view as shown in Figure 1.

3.2 Hybrid Interaction

When referring to hybrid interaction, it is important to differentiate between two approaches: serial and parallel integration. Using serial integration, 2D and 3D methods are used in a sequential order, one after each other. In parallel integration, 2D methods are quasi embedded and used directly to control and adapt the data in the immersive environment. In the virtual liver planning system, the interaction makes use of serial integration, in which the 2D and and the VR system are two separate systems that can be synchronized.

The combination of 2D and 3D interactions can have considerable advantages. 2D actions can be performed with relatively high precision, whereas 3D actions are executed at high speeds in specific task situations. As such, a clear speed-accuracy trade-off can be noticed, depending on the task at hand. In that respect, the virtual liver planning application contains actions that are inherently 2D (like contour editing or point-based segmentation refinement) or 3D (including visual inspection of mixed data, or approximation of surfaces).

The main factor is the flow of action at macro and micro level, for which the mapping and mode changes of functionality, the synchronization of desktop and spatial environments and the focus of attention play a key role.

At macro level, the performance of the actions is influenced by the work process preferences of the end-users: the radiologist prefers the desktop, whereas the surgeon can better work within a spatial setting. In order to access the functionality, an effective system control method is needed, that allows consistent interaction at both desktop and spatial environment. Redundant mapping of functionality is deliberately chosen, in order to support the work preferences of the end-users: all actions can be performed in the desktop and the spatial environment.

At micro-level, one of the issues that affect the flow of action in an application is the effectiveness of performing mode changes. The high amount of functions cannot be accessed effectively by any of the currently available 3D system control techniques. Therefore, the only possible way for mode changes is to access most of the functions on a standard GUI-style menu on the desktop screen. However, to support frequent interaction loops in the immersive environment, some functions are mapped to the input device. Specifically, manipulation actions are mixed with visual inspection actions (e.g. navigation) at high regularity. Therefore both CT data movement and general camera movement actions are directly mapped to two buttons on the input device.

Due to the different locations of the desktop display and the stereo wall in relation to the user, switching between desktop and spatial interaction (for example during mode change) necessarily results in a change of visual focus. Best practice demands that head rotation and focal plane difference are as limited as possible, without the desktop display occluding the stereo wall.

In the current hardware setup, the Tablet PC is placed at a table, and put in a tilted angle towards the users. The user can conveniently use the touch screen for selecting menu items or manipulating objects. The user's arm may be placed on the table to reduce fatigue. The table is placed at a specific distance from the stereo wall, so that stereo objects are viewed in a depth place that seems to be above or just behind the visuals viewed at the desktop screen. Consequently, both the angular movements of the head are limited, as well as the change of focus between depth planes. There are still field of view differences. Combining a smaller stereo wall with a larger touch screen in an L-shape like configuration may improve this issue.

Performing the different actions in desktop or spatial mode necessarily leads to different kinds of input. In the spatial environment, most actions are coarse, mixed with some more fine-grained actions, whereas at the desktop, all actions are fine-grained. The different kinds of performances, and the necessity to make use of a pen-like device to control the touch screen lead to different kinds of dynamical coupling between hand and device. This is mostly caused by the different kinds of grips to the device that match the precision needed to perform the task, as will be illuminated in the next section.

3.3 Hybrid Input Device

For the design of the new device, a close analysis of the tasks and the associated hand-device couplings and movements was made. In initial tests, a specific device for two-handed interaction was not found necessary because all tasks could be performed well with one hand. However, we observed that the device would need to allow for both power and precision grasps. Rotational movements for putting clipping plane or CT data plane are generally performed in high-speed and lower accuracy (sweeping task), whereas other tools like deformation demand lower speed and higher accuracy, and are better performed in precision grip.

To get an idea of a basic form of the device, we tried to match the movement and rotation patterns to existing devices. It was found that the hand activity matches partially a flying mouse, partially a pen-like device. The pen characteristics were considered important, since the device needed to function as pen-input device for the Tablet PC.

This resulted in an attempt to merge flying mouse and pen shapes into one single design, which allows for an unobtrusive switching between power and precision grasps. From clay models, we arrived at the shape shown in Figure 3. Due to the visual shape of the device, it was nicknamed *Eye of Ra*. The form needed to be large enough to enclose the electronics, which were taken from an EZ5 Optical Pen Mouse, which has a very small circuit board. The wireless connection was tested and found suitable when combined with a longer antenna in the device. The final device was made from carbon and fiberglass mats layered with epoxy glue, which results in a lightweight yet sturdy surface. The original button casing from the EZ5 was directly included in the design, in order to make a stable connection between device casing and electronics. Finally, the tip of the electromagnetic pen for the Tablet PC was embedded at the front of the device, and retro-reflective markers required for tracking were rigidly mounted on the body of the device.

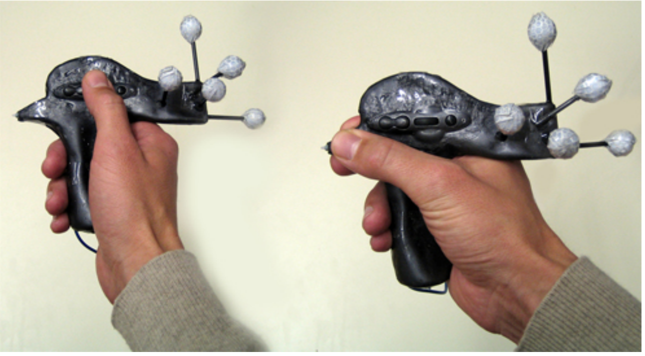

Figure 3: Eye of Ra - Input device for the hybrid user interface: The tip contains a conventional Tablet PC stylus tip for 2D interaction. Two buttons and a scroll wheel are used to trigger 3D interaction tasks. It is equipped with retro-reflective targets for optical tracking. The device is connected to the Tablet PC via a cable. Note the two different ways of grasping the device, the power grasp on the left and the precision grasp on the right.

The shape of the hybrid interaction device allows for easy switching between flying mouse and pen mode. By pronating the forearm, and slightly changing the position of the fingers (mostly moving the thumb), the user can easily change between the different modes. This allows for dynamic coupling between device and hand without the user actively noticing it.

3.4 Software

Following the overall hybrid approach, the software of the system consists of two collaborating applications, a desktop application and a VR application. Both parts of the system are closely coupled and share a large portion of the code. The dataset is visualized on both systems simultaneously. Interaction with the data can take place in either application, while system control tasks like loading the datasets or setting parameters are limited to the 2D menu system on the Tablet PC. All user interaction is performed using the *Eye of Ra* input device described in Section 3.3, which acts like a normal Tablet PC stylus in the desktop application, while 6-DOF tracking, a scroll wheel and buttons on the device are used to trigger input in the VR application.

Both applications involve 3D rendering based on Coin[1], a scene graph library compatible to the Open Inventor standard. The desk-

[1] `http://www.coin3d.org`

top application uses the Qt [2] framework for graphical user interface programming. The VR application is based on the *Studierstube* [24] VR/AR library, which builds on top of Coin and provides handling of VR devices such as stereo wall and tracking as well as convenient programming of 3D interaction with the scene graph.

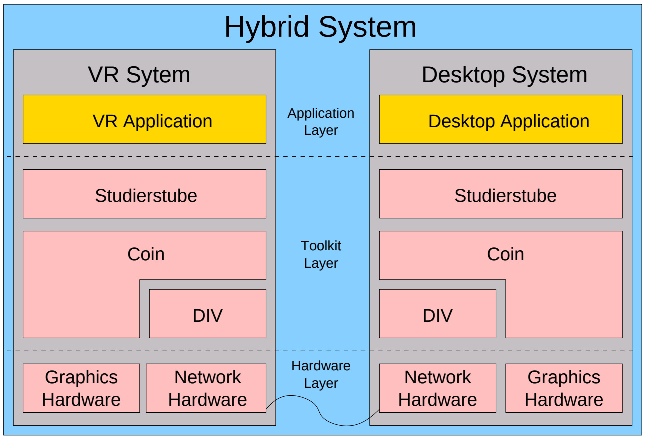

Figure 4: Software Architecture: The two separate applications share data via a network connection based on a distributed scene graph.

Synchronization between the desktop and VR application is based on a distributed shared scene graph extension called DIV [17]. The two applications share synchronized copies of the scene graph, which stores all geometric and application relevant data. Modifications to one copy of the scene graph will be propagated to the other copy, and vice versa. The synchronization happens automatically within the scene graph library, and need not be managed by the application programmer explicitly. Figure 4 gives an overview of the system architecture of the hybrid system.

4 Interaction Tools

There are three main tasks necessary to improve a segmented surface model:

- **Model Inspection** - The user tries to locate errors in the surface model by comparing raw CT data to the boundary of the surface.
- **Error Marking** - Regions of the surface model that were found erroneous in the inspection step are marked for further processing. This allows to restrict the following correction step to the erroneous regions, and avoids accidentally modifying correct regions.
- **Error Correction** - Marked regions are corrected using special correction tools based on mesh deformation.

Usually these tasks are not performed in a strict order. For example, model inspection is repeatedly required throughout the correction task. Individual corrected parts should be marked as final after successful correction.

The following sections will give an overview of the functionality of the system and the tools provided, mostly from the user's point of view, while technical details of the implementation of the segmentation refinement tools will be described in Section 5.2.

4.1 Model Inspection

The first step, model inspection, can be performed on the Tablet PC screen using the *Eye of Ra's* tip for interaction with the rotation, movement and scaling controls in the 2D user interface. In VR the model can be moved and rotated by pressing the scroll-wheel button on the input device, which fixes the model to the input device, while moving the device. The model navigation feature is bound to the scroll-wheel and is permanently available.

CT data is visualized on a 2D cutting plane that can be arbitrarily placed inside the CT scan volume. On the Tablet PC the plane can be manipulated by dragging 3D control widgets provided by the scene graph library. In the VR system the cutting plane, visualized as a rectangle attached to the input device, can be set by dragging in 3D with a specific button pressed. Like the model transformation, the cutting plane manipulation feature is permanently available.

The user may also configure most visualization parameters. For example, they can choose to show the surface model in any combination of wireframe, Gouraud shading and textured with the CT data. An optional plane clipping the 3D model slightly above the cutting plane allows to inspect the surface model near the clipping plane more efficiently.

4.2 Error Marking

For efficient organization of the correction procedure, the user may mark regions according to the type of observed error by painting a ”traffic light” color code - green, yellow or red. Green indicates that a portion of the surface is correct and will be immutable by subsequent correction operations. Yellow indicates that the surface is mostly correct but may be moderately altered from its current state as needed for example to smooth out differences at region boundaries. Finally, red indicates the surface is incorrect and may be drastically altered by the error correction tools. The marking is done by painting on the surface either on the Tablet PC or in the VR environment with a brush of adjustable size.

4.3 Error Correction

The presented system allows for correction of segmentation errors using a number of different tools for interactively deforming the surface representation of the object.

The *sphere deformation tool* consists of a sphere of user-defined radius which can be interactively placed in the datasets. In the VR system this is moved in place by moving the input device, while the tool position in the desktop setup is calculated as the position on the cutting plane corresponding to the 2D position of the cursor. Triggering sphere deformation causes object surface parts located within the sphere shape to be successively moved out of the sphere on the shortest possible path. Therefore, placing the sphere tool so that most parts of it are outside the object, causes its surface to move inwards, while outward movement is achieved by placing the sphere mostly inside the object. Moving the input device, while the deformation tool is active, causes the tool to respond, just like if one was deforming a piece of clay using a real world modeling tool.

The *plane deformation tool* is much like the sphere deformation tools, except that its behavior is similar to modeling using a scraper. It can be used to flatten the object's surface. In the VR system the position and orientation of the tools is directly determined by the input device, while cutting plane and the pen stroke direction define the tool's behavior in the desktop setup.

Fine grained deformation can be achieved using the *point dragging tool*, which can be used to pick individual surface vertices and move them directly to the desired location, while the surface deforms like a rubber sheet in the vicinity. Figure 5 shows a screenshot of the tools described above.

[2] http://www.trolltech.no/products/qt

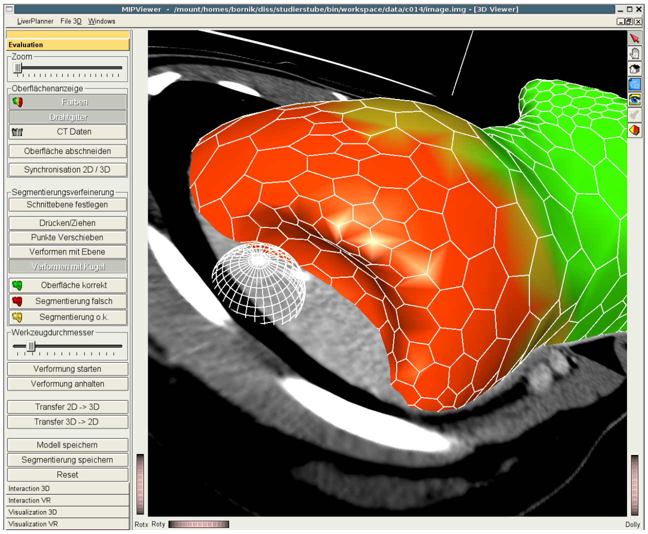

Figure 5: Interaction Screenshot: Deformation of the model using the sphere deformation tool on the Tablet PC. The correctness of the model can be verified using the cutting plane chowing CT data. Correct and erroneous regions have been marked before.

5 Methods

5.1 Cutting Planes

The 2D cutting planes are based on OpenGL 3D textures derived from the CT data by downsampling to fit the texture volume into the graphics card memory (256^3 in the VR setup and 128^3 on the Tablet PC). The 3D texture is displayed while the plane is interactively moved. Once the user has fixed the plane's position, a 2D texture for the plane is sampled at the full resolution of the dataset, delivering maximum quality. Plane sampling is decoupled from rendering, since it would impact the frame rate.

5.2 Surface Model

The presented tools are all based on deformable simplex meshes [7]. A simplex mesh is a special surface mesh with the property, that each vertex has exactly three neighbors, which makes it easy to calculate surfaces properties such as curvature and consequently to set up a deformable model based on a Newtonian law of motion, involving regulating forces, and other forces to deform the mesh. In order to avoid mesh degeneration, polygon splitting and merging operations are performed based on mesh quality criteria.

In our system we formulate forces towards the boundary of the binary segmentation calculated using an automated segmentation approach. This leads to a mesh accurately representing the segmentation.

In a VR setup the frame rates must be high, in oder not to discomfort the user, while the model should still be accurate. On average this leads to simplex meshes of around 250.000 polygons. We employ vertex buffer objects available on newer graphics hardware. They allow for mapping graphics memory into main memory which makes selective updates on the graphics card possible.

For efficiency, we keep a set of active vertices. When a vertex does not move significantly over several iterations, it is removed from this set and only added to it again, when neighboring vertices move significantly or new forces are applied. The refinement tools described in Section 4 alter the mesh locally, so the set of active vertices is always much smaller than the total number of vertices. Note, that only vertices altered in an iteration need to be updated in graphics memory in each frame.

Painting the mesh surface with the error marking tools introduced in Section 4.2 does not only alter the surface color. Painting the mesh in red affects the deformable model in a way, that forces towards the erroneous segmentation boundary are not calculated anymore. The mesh does not deform in these regions until until refinement tool induced forces apply. If the mesh is painted green, indicating that the surface is correct, no forces are calculated for the affected vertices. The standard color is yellow. In this case the forces towards the segmentation apply. Regularizing forces may still cause limited response to nearby tool based deformation.

Applying the sphere deformation tool presented Section 4.3 results in force calculation for all red colored surface vertices located within the tool's influence range. The force vector direction is spanned by the sphere center and the affected vertex. The length of the vector is the distance from the affected vertex to the sphere border. The plane deformation tool works similarly.

The point dragging affects the vertex closest to the input device location when it is triggered. The vertex can be placed freely in space. It is immediately removed from the set of active vertices, but its three neighbors are added to the active set, in case they are either red or yellow. When the vertex is released it keeps its current position, while the red of yellow adjacent parts of the mesh deforms and/or restructures until all quality criterions have been met.

6 Evaluation

6.1 Methodology and Procedure

The overall goal of the evaluation is to investigate the validity of our hybrid interface for liver surgery planning by comparing spatial (3D) and constrained (2D) tasks. Therefore, the evaluation is planned in two steps:

1. The first step examined the general spatial manipulation tools described in Section 4.

2. In the second step specific constrained tasks like segmentation refinement based on local contour drawing will be evaluated.

In this paper, we only report on the first evaluation step. The different modes of the system, namely desktop, spatial interaction, or hybrid mode were tested in a comparative study. The evaluation included mostly empirical testing, with some analytical methods, using a variety of data collection methods. The evaluation addressed several issues relevant in complex interaction tasks, in particular learning curve effects and mode switching. The evaluation included both qualitative and quantitative components, in which the user attitude and psycho-physiological abilities were collected and analyzed. All results were cross-compared to see if there were any notable differences between the user attitude towards the system, and the data collected through observation and recording.

The qualitative component of the evaluation was dominated by the subjective measurements obtained from the questionnaires, and the quasi thinking aloud protocols. The thinking aloud protocol was more or less a notification of the thoughts that were expressed by the subjects. The subjects were asked to speak, but not forced. As such, results from the thinking aloud protocols differed between users, since expression levels differed between persons. The questionnaire was mostly focused at the user satisfaction, by validating 17 questions. The main factors included were user learning curve, attitude towards tools, easiness and effectives of tools, user comfort including fatigue and device ergonomics, and attention. Hence, the questionnaires focused at the main issues specified in the hybrid interaction methodology applied, as described in Section 4.

The quantitative data were collected from the external observer's notes, the quality of the final liver model delivered by the subjects, and the logging files that tracked duration and changes of interaction modes. The observer noted all question asked by the user as well as user behavior (grasps, observable dexterity in fingers and wrist, arm-hand steadiness, attention to desktop and projection screen). Furthermore, the work-flow was observed and later compared with logs. Finally, a comparison was performed between the data produced by the subjects, and the best-practice model provided by an expert user.

The different steps of the evaluation were as follows: Subjects were first introduced to the system by the instructor, taking a 10 minute tour through the software. After the introduction, users could make use of the system for 12 minutes and ask questions. Next, the tests were performed. Users were instructed to only ask questions when absolutely necessary. All questions were recorded and analyzed to get a impression of the learning behavior.

6.2 Experimental Design

The evaluation consisted of a basic set of functions focusing at the visual inspection and segmentation of a liver dataset. The overall goal was to perform segmentation refinement on pre-segmented liver models that had artificially induced segmentation errors. The errors were designed to be obvious even to novices after introduction to the problem domain and system function. Three steps had to be performed: visual inspection, failure marking and segmentation refinement at the identified erroneous spot. Hence, the task consisted of navigation (zooming and camera movement actions), next to mesh marking, and segmentation modification (mesh contour clipping, mesh freezing and unfreezing, plane and sphere deformation, surface mesh deformation). Overall, the task can be categorized as highly complex.

The tools to perform the task were either used in desktop or spatial mode, or in a mixed way. Tool selection always took place at the menu on the desktop. Zooming and camera movement, and the placement of the cutting plane in spatial mode were possible by pressing a button on the input device.

6.3 Results

Eighteen subjects (11 female, 7 male) aged between 21 and 46 participated in the evaluation. All users had a medical background. Most of them were students of human medicine, whereas some of the users already had more extensive medical skills. Their experience with computer systems differed widely, from only incidental usage to expert users. Most users had only used a mouse before, about half of the users had also used a touch screen. None had any real experience with immersive environments. A complete evaluation session took about one hour per subject.

The questionnaires had a 7 point Likert scale. In order to get an impression of the users' attitudes, both averages and standard deviations were calculated. The results were compared with the protocol of the observer that included thinking-aloud statements, and the analysis of dataset results.

Tool sets For the task set given to them, the subjects generally prefered the 3D tool set above the 2D tool set. Overall, 37% were very satisfied with the ease of use of the 2D tool set, whereas 63% were satisfied with the 3D tool set. The mean values showed that most tools were rated in the range of very much acceptable up to very good (means between 5 and 6), only a limited amount of tools performed bad in either of the modalities. However, this user preference data is highly biased by the tasks evaluated, which are inherently 3D tasks. Tools for less complex actions (visual inspection tools) were rated considerably higher than for more complex ones (segmentation refinement tools). Clearly the deformation with the plane was not very well rated in both modalities, whereas deformation with the sphere was very well rated in 3D. The error marking task could be performed very well in both 2D and 3D, which was confirmed by external observations. Table 1 gives an overview the precentage of user satisfied with individual tools and the average tool rating concerning easyness and effectiveness.

	Easiness				**Effectiveness**			
Tool	**2D**		**3D**		**2D**		**3D**	
zooming	64%	5.7	89%	6.6	53%	5.2	78%	6.2
camera move	62%	5.1	88%	6.5	35%	4.6	71%	5.9
move c-plane	27%	4.5	69%	6.4	27%	4.5	71%	5.8
mark surface	64%	5.7	78%	6.0	57%	5.3	71%	6.0
plane def.	9%	3.9	37%	4.8	18%	3.4	33%	4.6
sphere def.	53%	5.2	94%	6.2	36%	4.5	61%	5.6
free def.	25%	3.7	40%	5.2	13 %	4.1	30%	4.9
point dragging	14%	3.4	33%	4.6	13%	3.8	27%	3.9

Table 1: Evaluation Results: percentage of fully satisfied users and average marks.

In a direct comparison of separate tool preference, 70% voted for 3D, 16.4% for desktop, and only 13.6% for mixed tools. The overall preference for complete tool sets was almost exactly in between mixed and 3D tool sets, being in line with the expectation that the 3D tool set would be preferred above 2D tool set. When interpreting the results, it should be taken into consideration that the tasks used in the evaluation have strong 3D characteristics. Therefore they illustrate only certain aspects of the complex overall liver surgery planning system.

Assessment of individual tools Looking at the actual performance of tools, users noted only little problems on basic actions like rotation or translation. The means for object rotation (5.33 in 2D and 5.94) and translation (5.00 in 2D, 5.78 in 3D) where rather high. Comparing truly satisfied users revealed that performance of atomic actions was far more appreciated in 3D (around 70% truly satisfied) than in 2D (only around 31% truly satisfied). With respect to precision of performance, 3D was marked much higher than 2D interaction. A highly diverse mix of user feedback could be noted: 44% were truly satisfied with the precision in 3D, against only 6% in 2D. Average marks were mediocre: a rating of 3.22 for the 2D environment, and 4.67 for the spatial setup. The mediocre marks for the 2D precision probably were caused by the size of the desktop display, and it can be expected to increase considerably by using a larger touch screen display. Looking at the level of visual details, 53% were very much satisfied with the desktop display, whereas the stereo wall got 83% full satisfaction. Furthermore, the better marks for the precision of 3D tools could have been biased by the ease of interaction. Through user experience and by using a larger 2D display, marks would be expected to become both higher and leveled in comparison to each other.

User observations showed a rather diverse image of performance: some users could correct the error extremely fast in 3D. Well performing users obviously made use of strategy taught to them in the introduction. Some users had too many problems with the complexity of the tasks and tools that they could not even apply the strategy. Regarding flow of action during action performance, some users had problems with switching between navigation and manipulation modes in the desktop interface. This disturbance was caused since users forgot to click on a specific button that was placed inconsistently in the user interface – a problem that can easily be solved.

Learning curve effects Due to the complexity of the evaluated task, it was expected that understanding the task and its tools would greatly influence the performance of using the tools. Learning of tools turned out to be not so easy, but not too hard: both user observations and user satisfaction showed learning curve issues. 35% were completely satisfied with the speed of learning the 2D interface, in the 3D interface this was even 56%. Average marks reflected the complexity though. 2D scored an average of 4.47, whereas 3D scored 5.17. Subjects often clearly noted that they could have used the more complex tools rather easily, if they had been given more practicing time, which was confirmed by user observations. Though the users did not always learn fast how to perform an action, they seemed well enough informed on what they were doing, stating only little mode errors or problems with feedback. 63% were completely satisfied with feedback in 2D, against 78% in 3D (averages 5.59 in 2D, 5.89 in 3D). Experienced computer users did not always learn to use the tools faster, and did not necessarily perform better. As such the experience-performance tradeoff was not always fully true in this experiment, possibly affected by the complexity of the task, and necessary understanding of medical data and performance strategies.

Focus The majority of users did not have a problem of switching between focusing at the large stereo wall or at the desktop. 50% were completely satisfied, and 56% noted they noticed no problems at all with changing between desktop and projection screen for interaction purposes. The averages for both issues were at 5.1. This mark shows that there is still space for improvement.

Eye of Ra The new input device was well accepted. Users rated the weight of the device as being excellent (almost 100% satisfaction rate). This rating was especially good when compared to the actual duration of usage of the device (also in free-air) of about 40 minutes. About 56% were highly satisfied with the device ergonomics (average of 5.56). The mark might be biased by the fact that most of the users did only use a mouse before. Most users need to get used to the new form of the device, since it is slightly unusual. Observations showed that users seemed to be performing well with the device, when focusing at the grips they used, and the amount of re-grasping, being a possible sign of problems. Most users seemed to handle the device very naturally, using dynamic coupling. Switching between flying mouse and pen-mode did not seem to cause any problems, and users often took an intermediate grip between two grip-modes, supported by the form of the device. The intermediate grip did not seem to be uncomfortable for the users.

Fatique Users noted some hand discomfort after 40 minutes of usage. As could be expected, for desktop interaction, more users were satisfied (59%) than for spatial operation (28%). The marks were highly variable in the rating of spatial interaction. Some tools in 3D (notably the plane deformation tool) were difficult to use, which definitely explained some of the worse results. However, the significance of the marks is unclear, since no comparison with any other 3D input device was made. Also, close observations of the users' hands by the external observer showed that most (but not all) users had very steady hands during operation. Hence, since device ergonomics were rated well we tend to say that users did not have significant problems with fatigue. Also, all users did not use 3D input devices before, so usage was probably very strange to them, making a comparative rating difficult.

6.4 Discussion and conclusions

Visual inspection and segmentation refinement can be successfully performed using the developed system. The users' attitude towards the spatial tool set was better than to the desktop tool set, as expected. Less complex tools were rated higher on ease of use and effectiveness than more complex tools. Nonetheless, several of the tools did perform below acceptance level (below a mark of 4.5), notably the plane and point deformation tools. The plane deformation tool seems to produce some ergonomic hand rotation problems that would require a redesign of the tool to make it more useful.

It was interesting to see how users seemed to interpret precision in relation to the size of the display, and not in terms of the resolution of input. Even though the input at the touch screen was steadier, users felt better in control when working with the large model.

The macro-level flow of action worked out rather well. The majority of users did not have a problem mixing the modalities, even though it should be stated that most users predominantly worked in one of the modalities. Hence, effects were lower than if they had changed between the modalities continuously.

At micro-level, some users had problems with switching between modes, both at interaction and at focal level. All of these users also expressed learning problems, which may indicate that they were not used to working in a 3D environment. In particular, we observed problems with using the 2D system controls correctly. Most of these problems were related to learning deficiencies, since the problem was not the actual selection of the menu item, but rather which item needed to be selected.

In general, learning effects affected the outcome of the test to a large extent. To all users, working in a Virtual Environment was new, and many users did not have much computer experience either. Hence, learning how to use the interfaces affected the usability of the system considerably. Users who quickly grasped the concept of solving the task could effectively solve the task within time, even sometimes much faster than expected. Several users solved the problem within 7 minutes, which was extremely fast. We believe that during longer evaluation sessions, learning effects would affect the marks in a positive way.

It can be concluded that the evaluation provided useful insights on the performance of the selected tool set, and several hybrid interaction issues. Results are encouraging in relation to the complexity of the task at hand. However, the validity of the hybrid system as a whole cannot be currently claimed, since the second part of evaluation – examining the 2D constrained contour tools, which were not completed at the time of evaluation – needs to be performed. Some results of the current evaluation may indicate a clear advantage of the 3D tool set against using 2D techniques, but with the complete system in mind, the usage of a hybrid system still seems valid. An expert radiologist testing the system also felt comfortable using it and noted its great potential for a variety of medical applications.

7 Acknowledgements

We would like to thank Thomas Pock for helping with the manufacturing of the *Eye of Ra*. Furthermore we would like to thank Prof. Erich Sorantin and all medicine students of the Medical University of Graz who participated in the evaluation of the system. This work was funded by the Austrian science fund FWF under grants P17066 and Y193.

References

[1] W. A. Barret and E. N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, 1997.

[2] R. Beichel, S. Mitchell, E. Sorantin, F. Leberl, A. Goshtasby, and M. Sonka. Shape- and appearance-based segmentation of volumetric medical images. In *Image Processing, 2001. Proceedings. 2001 International Conference on Image Processing*, volume 2, pages 589–592, 2001.

[3] A. Bornik, R. Beichel, B. Reitinger, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Computer aided liver surgery planning: An augmented reality approach. In Robert L. Galloway, editor, *Medical Imaging 2003: Visualization and Display, Proceedings of SPIE*, volume 5029. SPIE, SPIE Press, May 2003.

[4] A. Bornik, B. Reitinger, R. Beichel, E. Sorantin, and G. Werkgartner. Augmented-reality-based segmentation refinement. In *Proceedings of the SPIE, Volume 5367, pp. 77-87 (2004).*, pages 77–87, May 2004.

[5] B. Buxton. Input device sources and resources, 2005.

[6] R. P. Darken and R. Durost. Mixed-dimension interaction in virtual environments. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 38–45, New York, NY, USA, 2005. ACM Press.

[7] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, September 1999.

[8] H. Delingette and N. Ayache. Hepatic surgery simulation. *Communications of the ACM*, 48(2):31–36, February 2005.

[9] T. Emerson, J. Prothero, and S. Weghorst. Medicine and virtual reality: A guide to the the literature (MedVR). Technical Report B-94-1, Human Interface Technology Laboratory, University of Washington, Seattle, USA, 2001.

[10] A. X. Falçao and J. K. Udupa. A 3D generalization of user-steered live-wire segmentation. *Medical Image Analysis*, 4:389–402, 2000.

[11] S. Feiner and A. Shamash. Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. In *ACM Symposium on User Interface Software and Technology (UIST'91)*, 1991.

[12] G. Fitzmaurice, S. Zhai, and M. Chignell. Virtual reality for palmtop computers. *ACM Transaction on Information Systems*, 11(3), 1993.

[13] B. Fröhlich and J. Plate. The cubic mouse: A new device for three-dimensional input. In *Proceedings of the 2000 ACM Conference on Human Factors in Computing Systems (CHI 2000)*, pages 526–531. ACM Press, 2000.

[14] M. Hachet, P. Guitton, and P. Reuter. The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In *Proceedings of the 2003 ACM Symposium on Virtual Reality Software and Technology (VRST 2003)*, pages 205–212. ACM Press, 2003.

[15] H. K. Hahn, B. Preim, D. Selle, and H. O. Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 395–402, Washington, DC, USA, 2001. IEEE Computer Society.

[16] T. Heiman, M. Kunert, and H-P. Meinzer. New methods for leak detection and contour correction in seeded region growing segmentation. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXV, 2004.

[17] G. Hesina, D. Schmalstieg, A. Fuhrmann, and W. Purgathofer. Distributed open inventor: a practical approach to distributed 3d graphics. In *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 74–81, New York, NY, USA, 1999. ACM Press.

[18] K. Hinckley, R. Pausch, J. Goble, and N. Kassell. Passive real-world interfaces props for neurosurgical visualization. In *Proceedings of the 1994 ACM Conference on Human Factors in Computing Systems (CHI'94)*, pages 452–458. ACM Press, 1994.

[19] M. Jackowski, A. Goshtasby, and M. Satter. Interactive tools for image segmentation. In Kenneth M. Hanson, editor, *Medical Imaging 1999: Image Processing, Proceeding of SPIE*, volume 3661. SPIE, SPIE Press, May 1999.

[20] H-P. Meinzer, P. Schemmer, M. Schöbinger, and et al. Computer-based surgery planning for living liver donation. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXV, pages 291–295, 2004.

[21] K. Montgomery, M. Stephanides, S. Schendel, and M. Ross. User interface paradigsm for vr-based surgical planning: Lessons learned over a decade of research. Technical report, National Biocomputation Center, Stanford University, Stanford CA 94305, 2001.

[22] J. Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *ACM Symposium on User Interface Software and Technology*, pages 31–39, 1997.

[23] C. Sandor, A. Olwal, B. Bell, and S. Feiner. Immersive mixed-reality configuration of hybrid user interfaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2005)*, 2005.

[24] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavàri, L. M. En carna çao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. *Presence: Teleoper. Virtual Environ.*, 11(1):33–54, 2002.

[25] Z. Szalavári and M. Gervautz. The personal interaction panel - a two-handed interface for augmented reality. *Computer Graphics Forum*, 16(3):335–346, 1997.

[26] B. Ullmer and H. Ishii. The metadesk: Models and prototypes for tangible user interfaces. In *Proceedings of the 1997 ACM Symposium on User Interface Software and Technology (UIST'97)*, pages 223–232. ACM Press, 1997.

[27] M. Wloka. Interacting with virtual reality. In J. Rix, , S. Haas, and J. Teixeira, editors, *Virtual Prototyping - Virtual Environments and the Product Development Process*. Chapman & Hall, 1995.