Intuitive and Lightweight User Interaction for Medical Augmented Reality

Jan Fischer, Dirk Bartz, Wolfgang Straßer

Visual Computing for Medicine at WSI/GRIS, University of Tübingen
Sand 14, 72076 Tübingen, Germany
Email: `fischer@gris.uni-tuebingen.de`

Abstract

The graphical overlay of additional medical information over the patient during a surgical procedure has long been considered one of the most promising applications of augmented reality. While many experimental systems for augmented reality in medicine have reached an advanced state and can deliver high-quality augmented video streams, they usually depend heavily on specialized dedicated hardware. We have described a novel medical augmented reality application, which is based almost exclusively on existing, commercially available, and certified medical equipment [5].

In this paper, we show how the capabilities of a so-called image guided surgery system for tracking surgical instruments can be utilized for user interaction. Our method enables the user to define points and freely drawn shapes in 3D and provides selectable menu items, which can be located in immediate proximity to the patient. This eliminates the need for conventional touchscreen- or mouse-based user interaction without requiring additional dedicated hardware like dedicated tracking systems or specialized 3D input devices. We demonstrate our new input method with an application for creating operation plan sketches directly on the patient in an augmented view.

1 Introduction

In augmented reality (AR), virtual graphical objects are added to the real environment of the user [1]. One central aspect of useful reality augmentation is the correct spatial alignment of virtual objects with respect to the user's surroundings. In order to achieve this augmentation, the user's head or the digital video camera have to be tracked in real-time.

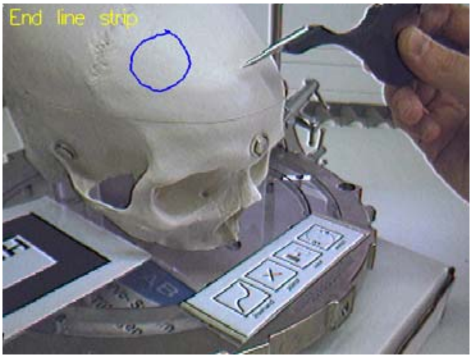

Figure 1: Example drawing created directly on a plastic skull. The tracked tool used for user interaction can be seen in the top right part of the image. On the bottom, the row of four menu "icons" is visible.

Ways of using augmented reality for the support of medical therapy have been in the focus of active research. During the last years, a number of experimental augmented reality systems for medicine have been presented. The design of most early medical AR systems favored the use of immersive display devices like head mounted displays (HMDs). Many of the more recent approaches use projector-based display systems or semi-transparent LCD screens, which are better accepted for medical applications. The major disadvantage of most existing augmented reality systems for medicine is their reliance on specific hardware components. Such devices, like dedicated magnetic, ultrasonic or infrared trackers and specialized displays, can have a negative impact on the usability of the system. These components, most of which are intended for use in VR research or engineering applications, tend to be ill-suited for medical settings. They can require tedious setup procedures, can consume plenty of space, are generally not certified for medical applications, and are often very expensive.

VMV 2005 Erlangen, Germany, November 16–18, 2005

Our previously introduced medical augmented reality system *ARGUS* does not require additional dedicated hardware [5]. All information necessary for a useful reality augmentation is obtained from medical standard hardware. A so-called image guided surgery (IGS) device, which is equipped with built-in infrared cameras, is accessed using a specialized network interface. In order to provide a more realistic and more useful display of virtual objects, we have also developed a method for handling occlusion in our AR system [4].

One important task for any kind of augmented reality system is to provide useful facilities for user interaction. These should usually include methods for triggering application-specific actions, for the selection or manipulation of virtual objects, and for the definition of points or more complex shapes in space. Many different techniques for user interaction in AR have been proposed. Most of these are also based on specialized hardware, e.g., magnetic trackers or specialized 3D input devices. As stated above, such dedicated system components can be problematic for use in medical applications.

In this paper, we present a novel method for user interaction in medical AR. Our approach is based exclusively on the information delivered by the image guided surgery system. The pose data of tracked surgical tools is processed in order to recognize a set of basic "gestures", i.e. simple user interaction elements. Our application uses these gestures to implement a flexible, configurable menu system. This system makes it possible to trigger actions by performing a "click" at a certain position. These clicks can be executed in immediate proximity to the patient. Moreover, the user can define positions and freely drawn shapes in three-dimensional space. All of these interactions, which are performed with wireless passive input tools, are supported without requiring any kind of additional hardware. Figure 1 shows an example of a simple operation plan element created using our user interaction method.

The menu system presented in this paper might even be used as a replacement for existing user interfaces provided by medical devices. These are usually based on touchscreens or conventional screens with mouse input. Unlike such traditional user input systems, our immersive approach works in immediate proximity to the patient and does not require the user to shift the attention focus on a screen built into a medical device.

2 Related Work

An augmented reality application for ultrasound-guided needle biopsies has been described by State et al. [9]. The system offers accurate tracking of both the user and the ultrasound probe as well as stereoscopic rendering, but relies on a number of proprietary technologies. These include a hybrid optical-magnetic tracker for the user's head and a mechanical arm for probe registration.

In recent years, AR systems were created for the support of various medical application scenarios. The Varioscope AR is an optical see-through head mounted display for computer aided surgery [3]. It consists of a newly designed display component, which is tracked by a dedicated optical camera system. Sauer et al. have described an AR system for the visualization of ultrasound images [7]. Their system is composed of a video see-through HMD and a pair of cameras for acquiring a stereo video stream of the user's surroundings. The Medarpa system presented by Schwald et al. is based on a semi-transparent LCD screen, which is mounted over the patient on a swivel arm [8]. They use a combination of infrared and magnetic tracking for the display, the patient and surgical instruments. Bornik et al. have presented a medical augmented reality system for liver surgery planning [2].

Methods for user interaction in virtual and augmented reality have been an area of active research. Most approaches require the use of specific interaction devices. Wormell and Foxlin have given an overview over some recent devices for user input in VR/AR [10].

3 Medical Augmented Reality based on Image Guided Surgery

Our medical augmented reality system *ARGUS* utilizes the capabilities of an image guided surgery device (*ARGUS* is an acronym for Augmented Reality based on Image GUided Surgery). Image guided surgery (IGS) is a widespread technology for the visualization of preoperatively scanned image data and information about tracked surgical tools during an intervention. *ARGUS* is based on a VectorVision® IGS system produced by the BrainLAB company. The VectorVision device is a mobile unit containing a touchscreen, a computer running the IGS software, and a highly accurate infrared tracking camera. The infrared camera is capable of tracking surgical instruments with attached infrared marker clamps.

A special network interface, VectorVision Link (VVL), is used for exchanging various types of data between *ARGUS* and the IGS system. These include tracked tools information, the current patient volume dataset, patient registration information, and operation plan elements like planned access or entry points and trajectories. Note that patient registration, i.e. the computation of a transformation from the world coordinate system to the patient volume dataset, is performed using the IGS system in a standardized and reliable way. Our augmented reality application does not need to generate the patient registration, but simply utilizes the information provided by the VectorVision system.

ARGUS is a video see-through AR system. The major innovation required for implementing IGS-based augmented reality is to find a way for tracking the camera using the existing medical equipment. An infrared marker clamp, which was originally intended for tracking surgical instruments, is attached to the webcam used for acquiring the AR background image. This setup is shown in Figure 2.

Figure 2: Webcam with attached marker clamp.

The position and orientation of the marker clamp are retrieved using the VectorVision Link interface in real-time. We have devised a one-time calibration step, which computes the transformation from the marker clamp to the camera coordinate system. This calibration step has been described in detail in [5].

After the one-time calibration has been performed, only the tracking information delivered by the IGS system is required for achieving a correct reality augmentation. This way the large trackable volume and high accuracy of the image guided surgery device are utilized for augmented reality without requiring any additional dedicated hardware.

4 The New User Interaction Method

We have developed a new technique for providing comprehensive user interaction based on the information delivered by the image guided surgery system. The IGS device is capable of tracking multiple infrared marker clamps simultaneously. One of the instrument clamps is attached to a pointer-like tool, as illustrated in Figure 3. Alternatively, a standard pointer tool can be used, which is supplied with the IGS system. The tool is used for triggering actions by indicating menu markers located at previously defined positions and for the definition of points or more complex shapes in 3D.

Figure 3: Example of a tracked tool used for interaction.

The *ARGUS* system continually acquires the position and orientation of tracked tools from the image guided surgery device. It then looks up the user-defined interaction tool, which is identified by a unique name string. Consecutive tool positions and orientations are compared in order to detect basic gestures and the triggering of menu actions.

4.1 Basic Gestures

Our user interaction system supports two basic gestures. One is the so-called “still click”, which is detected when the movement of the tool is below a given threshold for a certain amount of time. The still click is easy to execute, but it is also often triggered inadvertently, e.g. when the tool is put down.

The “angle click” also requires the position of the tool to remain practically constant for a certain duration. However, additionally the direction vector of the tool has to change continually during that period. Note that the position reported by the IGS system always is that of the tip of the tracked instrument. An angle click is executed if the user holds the tip of the instrument at a certain point while rotating the instrument. The angle click is significantly more complex to perform and thus is rarely triggered unintentionally. Figure 4 illustrates the tool motion necessary for triggering an angle click.

Figure 4: Illustration of the “angle click” user interaction gesture.

We have devised the following algorithm for detecting tool gestures. In every time step, the position pos_t and the direction vector dir_t of the tracked interaction tool are received from the IGS system. The software permits a maximum tip movement of $threshold_{pos}$ per time step. Since the coordinate system of the infrared camera is calibrated to have a unit length of one millimeter, $threshold_{pos}$ is also defined in that unit. A typical value for $threshold_{pos}$ is less than one millimeter. A second threshold parameter, $threshold_{dir}$, determines the minimum change of the direction vector required for a valid angle click step. The number of consecutive time steps during which the respective gesture conditions have to be fulfilled is given as parameter $clickDuration$. The threshold and click duration values can be configured at run-time.

Figure 5 shows the program flow chart of the click detection algorithm for a single time step. After the tool information has been downloaded, the movement of the tip is checked. If it is above $threshold_{pos}$, the current gesture state is reset, and the algorithm stops. This means that any ongoing click period is interrupted by the tool motion, and any tool gesture can only begin in a later time step. If the movement is small enough, the counter $numSteps_{still}$ is incremented. $numSteps_{still}$ counts the number of time steps without significant tool motion and is used for triggering still clicks. Subsequently, the current direction vector is compared to the tool direction measured in the last time step. If the direction change is above $threshold_{dir}$, the counter variable $numSteps_{angle}$ is incremented. The value of $numSteps_{angle}$ is used for detecting angle clicks.

After the current changes in tool position and direction have been considered, the conditions for triggering a click event are checked. If $clickDuration$ time steps without significant motion have passed, a still click has been executed. The algorithm then checks whether the current tool position is the same as the position recorded during the last still click. This position is stored in the vector $lastStillClick$. If the locations of the last and the current click are very close, the user interaction event is suppressed. Otherwise, a still click event is generated for further processing by the software, and the current tool position is saved in $lastStillClick$. If the application provides support for menu interaction, the click event is processed by the menu system (see Section 4.2).

Finally, the algorithm checks whether an angle click has been executed. If the conditions of an angle click were continuously met for $clickDuration$ time steps, the location of the current and the last angle click are compared. In case the current tool position is too close to the last angle click, which is stored in the vector variable $lastAngleClick$, no click event is generated. Otherwise, a user interaction event is sent to the application and optionally the menu system.

Please note that using this method, a still click event is always generated simultaneously or right before an angle click. We assume the application logic to account for this fact by filtering click events according to the semantics of the current top-level user interaction sequence. Moreover, the algorithm could be easily modified to operate exclusively in one of two separate still click or angle click modes.

4.2 Menu System

In addition to the basic gestures recognition, the second main innovation presented in this paper is a method for implementing a configurable menu using the basic gestures. Markers, which have icons for the menu items printed on them, are placed in-

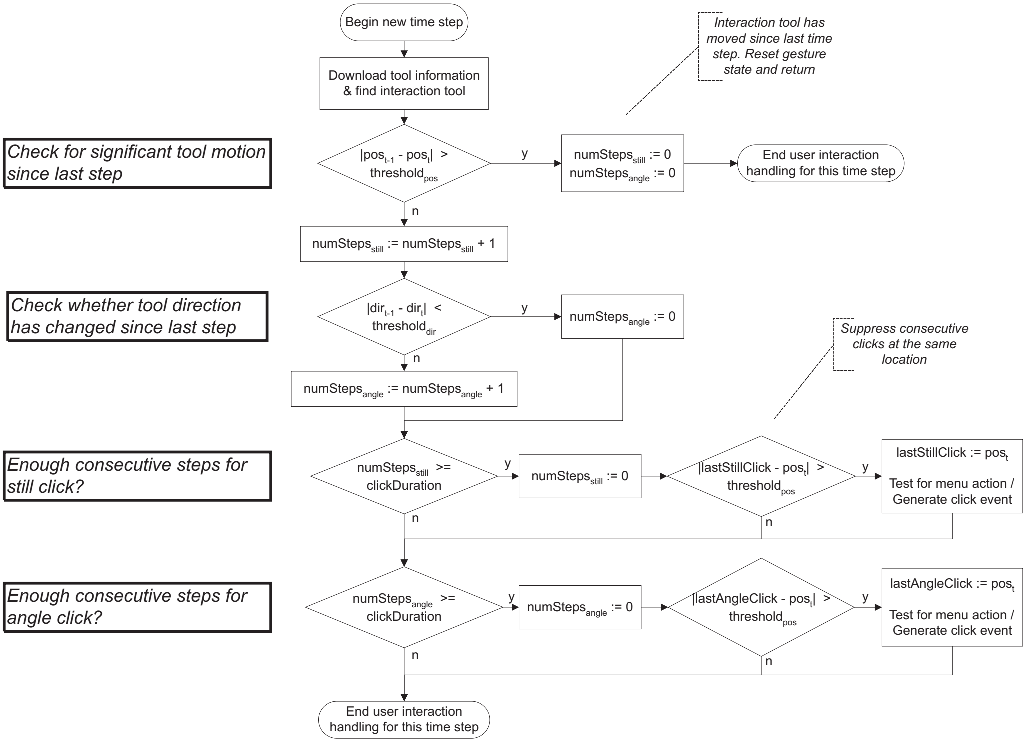

Figure 5: Program flow for updating the gesture state and generating click events for a single time step.

side the trackable volume of the infrared camera. The user can then activate a menu action by performing a click with the interaction tool on one of the markers. Figure 6 shows an example of a simple menu, which is used by our demonstration application for patient drawings (see Section 5). The same menu markers can be seen in Figure 1 at the base of the plastic skull.

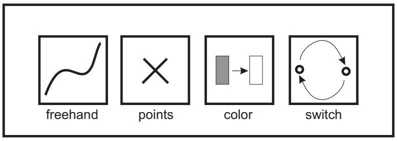

Figure 6: Simple menu used by our patient drawing example application.

Our software contains an editor for the interactive definition and modification of menu items. The data structure for each item consists of the position of the center of the marker, a radius, and name and description strings. In the editor, the marker position is defined by a click gesture with the interaction tool. The other data are entered with a graphical user interface. Table 1 lists the attributes for a menu item and their data types.

Table 1: Attributes of a menu item

Attribute	Data Type
Center position	3 float values (x,y,z)
Radius	float
Name	string
Description	string

An application using the menu system can be configured to react to still clicks or angle clicks for menu actions. When a click event of the respective type is generated by the basic gestures algorithm (see Figure 5), it is analyzed by the menu system. The click location is compared with the position of each menu item. If the distance between the click location and the center of a menu item is smaller than its radius, a menu action event is generated. The menu event is parameterized with the name of the menu item and can thus be easily interpreted by the application. Note that due to the comparison of the Euclidean distance with a radius, the real marker is approximated by a sphere. However, this distance criterion could be easily modified to be sensitive to a shape which more closely matches the physical appearance of the marker. Moreover, we have found the sphere criterion to work reliably and intuitively in the practice.

5 Example Application: Operation Plan Drawings

A demonstration application has been implemented for proving the usefulness of our interaction method. Using the software, points and free formed line strips can be drawn directly on the patient. This way, operation plan elements or related visual aids can be created interactively. The application uses the menu items shown in Figure 6.

The software continually waits for menu action events. Once the menu event “freehand” is received, a special drawing mode is initiated. From that moment, the software waits for the next still click. After the still click event has been triggered, the line strip is recorded. Every new position of the tip of the interaction tool is stored and added to the geometry of the line strip. This continues until another still click event is received. The click ends the interactive drawing of the line strip. Figure 7 illustrated this user interaction sequence. The sequence can be easily and intuitively perfomed by the user. After clicking on the menu item, the interaction tool has to be moved to the intended starting point of the drawing. Then the user has to wait for a short period. Since the basic gestures algorithm suppresses consecutive clicks at the same location (see Figure 5), the line strip never inadvertently begins at the menu item. The drawing is created in a continuous motion. When the drawing is complete, the user again has to wait for a short while in order to leave the drawing mode. This way the entire functionality is controlled without any additional hardware, only using the tracking information of the interaction tool.

The application also contains a mode for setting individual points. When the respective menu action event has been triggered, the software waits for the next still click. A point is then generated at the location of the click. In addition to the line strip and point drawing modes, two utility functions are supported. The menu action “color” changes the current color by traversing a preset color palette. The next point or free formed line strip to be generated is then displayed in the new color. By selecting the menu action “switch”, the user can cycle through several drawing slots. Only one of several drawings is displayed and can be edited at a time. The switch command advances to the next stored drawing or displays nothing, if the slot has not been used yet. Altogether the demonstration software shows that our lightweight interaction method can provide all the functionality required for a useful drawing application in an easy-to-use and intuitive way.

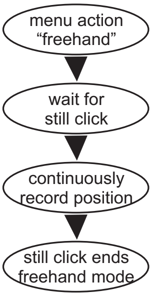

Figure 7: Interaction sequence for drawing a free formed line strip in the demonstration application.

6 Results

In the actual experimental setup, the four menu items are laid out in a row as shown in Figure 8. However, it would be possible to place the items freely within the trackable volume of the infrared camera due to the configurability of the menu. A standard XML file contains the description of the menu items and their placement.

Figure 9 illustrates the procedure for creating a single line strip drawing. The drawing mode is actived in Fig. 9(a) by a still click on the respective menu item. The following click at a different location determines the starting point of the line strip (see Fig. 9(b)). The freely drawn line strip is extended until the interaction tool again remains still

(a) Activation of drawing mode

(b) After initial still click

(c) End of interaction

Figure 9: Interaction mode for free formed line strips. The mode is activated by a still click on the "freehand" menu item. A second still click begins the drawing, and a third click ends it. In the top left corner of the image, the name of the last detected user interaction is displayed.

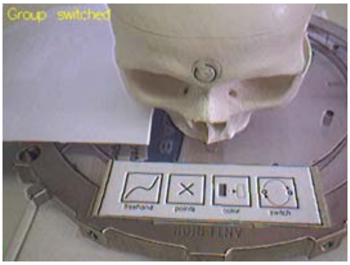

Figure 8: Real menu used by our example application.

for a period of time.

In Figure 10, the effect of the color change action can be seen. Figure 10(a) shows the augmented reality display, after a white line strip has been drawn. The user then activates the color change item using a still click. This causes the next line strip to be rendered in a different color (see Fig. 10(b)). Due to the fact that the tracking information delivered by the image guided surgery system contains full spatial position data, the drawings created with our user interaction method are three-dimensional. This is illustrated in Figure 10(c), which shows the scene with the virtual drawings from a different point of view.

Our augmented reality system is capable of generating augmented images at a frame rate of approximately 15 fps on the average. The presented user interaction method including the menu system does not negatively affect the performance of the application. Since the tracked tool data are continually requested from the IGS system for tracking the webcam anyway, the amount of data transferred over the network does not increase.

7 Conclusion

We have presented a novel concept for wireless user interaction in medical augmented reality. Unlike previously described interaction methods, our approach is based exclusively on existing, commercially available medical equipment. Whereas specialized 3D tracking or interaction devices are usually designed for applications in VR or engineering and ill-suited for medicine, image guided surgery systems are fully certified for medical settings.

The interaction modes supported by our method can be used for a wide range of applications. The menu system could provide support for changing the parameters of an advanced information display like volume or multi-modal rendering. Even conventional functions like loading a patient dataset or initiating the patient registration procedure could be triggered using our novel menu system.

The user interaction system presented in this paper has also been used as the basis for a new method for semiautomatic volume classification. In this application, intuitive augmented reality interaction delivers the input data for a machine learning algorithm, which computes the volume rendering transfer function [6]. This demonstrates that our new user interaction method can support complex application cases.

(a) Change of color after white line has been drawn

(b) Different color used for circular line strip

(c) Top view of scene

Figure 10: The example application provides a menu item for changing the current drawing color. Figure 10(c) illustrates that the created drawings are three-dimensional.

Acknowledgements

This work has been supported by project VIRTUE in the focus program on ”Medical Navigation and Robotics” (SPP1124) of the German Research Foundation (DFG).

References

[1] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[2] A. Bornik, R. Beichel, B. Reitinger, E. Sorantin, G. Werkgartner, F. Leberl, , and M. Sonka. EG2003 Medical Prize Competition: Augmented Reality based Liver Surgery Planning. *Computer Graphics Forum*, 22(4):795–796, December 2003.

[3] M. Figl, W. Birkfellner, C. Ede, J. Hummel, R. Hanel, F. Watzinger, F. Wanschitz, R. Ewers, and H. Bergmann. The Control Unit for a Head Mounted Operating Microscope Used for Augmented Reality Visualization in Computer Aided Sugery. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 69–75, 2002.

[4] J. Fischer, D. Bartz, and W. Straßer. Occlusion Handling for Medical Augmented Reality using a Volumetric Phantom Model. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2004.

[5] J. Fischer, M. Neff, D. Freudenstein, and D. Bartz. Medical Augmented Reality based on Commercial Image Guided Surgery. In *Proceedings of Eurographics Symposium on Virtual Environments (EGVE)*, pages 83–86, June 2004.

[6] M. Köbele. Augmented-Reality Based Classification for Volume Rendering in Medical Visualisation. Diplomarbeit, Universität Tübingen, April 2005.

[7] F. Sauer, A. Khamene, B. Bascle, L. Schimmang, F. Wenzel, and S. Vogt. Augmented Reality Visualization of Ultrasound Images: System Description, Calibration, and Features. In *Proceedings of IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 30–44, 2001.

[8] B. Schwald, Helmut Seibert, and T. Weller. A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 261–262, September 2002.

[9] A. State, M.A. Livingston, G. Hirota, W.F. Garrett, M.C. Whitton, H. Fuchs, and E.D. Pisano. Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. In *Proceedings of ACM SIGGRAPH 96*, pages 439–446, August 1996.

[10] D. Wormell and E. Foxlin. Advancements in 3D Interactive Devices for Virtual Environments. In *Proceedings of Eurographics Workshop on Virtual Environments (EGVE)*, pages 47–55, May 2003.

(a) Activation of drawing mode

(b) After initial still click

(c) End of interaction

Figure 9: Interaction mode for free formed line strips. The mode is activated by a still click on the "freehand" menu item. A second still click begins the drawing, and a third click ends it. In the top left corner of the image, the name of the last detected user interaction is displayed.

(a) Change of color after white line has been drawn

(b) Different color used for circular line strip

(c) Top view of scene

Figure 10: The example application provides a menu item for changing the current drawing color. Figure 10(c) illustrates that the created drawings are three-dimensional.