Eurographics Italian Chapter Conference (2010)
E. Puppo, A. Brogni, and L. De Floriani (Editors)

# Mixed Reality and Gesture Based Interaction for Medical Imaging Applications

A.F. Abate, M. Nappi, S. Ricciardi, G. Tortora

Dipartimento di Matematica ed Informatica, Università di Salerno, Italy

**Abstract**

*This paper presents a framework providing a collection of techniques to enhance reliability, accuracy and overall effectiveness of gesture-based interaction applied to the manipulation of virtual objects within a Mixed Reality context. We propose an approach characterized by a floating interface, operated by two-hand gestures, for an enhanced manipulation of 3D objects. Our interaction paradigm, exploits one-hand, two-hand and time-dependent gesture patterns to allow the user to perform inherently 3D tasks, like arbitrary object rotation, or measurements of relevant features, in a more intuitive yet accurate way. A real-time adaptation to the user's needs is performed by monitoring hands and fingers motions, in order to allow both direct manipulation of virtual objects and conventional keyboard-like operations. The interface layout, whose details depend on the particular application at hand, is visualized via a stereoscopic see-through Head Mounted Display (HMD). It projects virtual interface elements, as well as application related virtual objects, in the central region of the user's field of view, floating in a close-at-hand volume. The application presented here is targeted to interactive 3D visualization of human anatomy resulting from diagnostic imaging or from virtual models aimed at training activities. The testing conducted so far shows a measurable and user-wise perceptible improvement in performing 3D interactive tasks, like the selection of a particular spot on a complex 3D surface or the distance measurement between two 3D landmarks. This study includes both qualitative and quantitative reports on the system usability.*

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities

## 1. Introduction

Thanks to continuous advances in technology and research [ABF*01], AR/VR mixed interfaces are becoming increasingly effective in providing users with a richer level of interactivity. They merge the physical space in which the user lives and works with the virtual space in which the user interacts with digital information [PTB*02]. Actually, since its first introduction, the joint use of virtual and augmented reality has addressed a wide variety of applications. Typical examples are military simulations which were the first candidates to take advantage of the new technologies, apart from entertainment applications. A further privileged field to employ AR/VR is training/learning [BRI91]. Along a conceptually similar line, related frameworks can be used for the simulation of production sequences [DFG*05], or even for the rapid evaluation of prototypes before production [BKFT00]. Within this technological context, gesture based interfaces have been often regarded by human-computer-interaction literature as a very natural way to navigate through and interact with virtual reality environments. However, even though virtual reality technology was already available in the early '90, costs as well as performance bottlenecks in the required equipments delayed its diffusion [BRO99]. It is only thanks to more recent progress and an overall cost reduction of related hardware, that a broader and growing spectrum of applicative fields can nowadays be addressed [KHLE97]. In particular, medical applications exploiting virtual reality began to emerge in the early '90s. [KR93] These include VR surgical simulators, tele-presence surgery, complex medical database visualization, and more. Such applications represent a paradigm shift in the field of medicine, but, despite its potential advantages, the usage of such kind of interface is actually rather limited due to practical issues. In the field of medical imaging, the research on gesture based interfaces is mainly focused on replacing conventional input devices, e.g. keyboard and mouse, with a contact-less interface that is better suited for sterile operating room interventions. Indeed, computers and their peripherals, as well as other electronic equipment, are difficult to sterilize. Therefore, when interactivity is required, the surgeon must be supported by an assistant who operates the computer. An alternative approach is

© The Eurographics Association 2010.

DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2010/033-040

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

represented by voice recognition, in association with dictation services. However, dictation services in critical situations require accurate and reliable speech recognition systems, that in turn often need to be trained by the final user. This may limit their use when some user actions require an articulated set of utterances. As a result, a gesture-based interaction [KS98] would be preferred for such kind of applications. This must be supported by a robust visual approach to gesture recognition which however may limit the available gesture patterns due to occlusion issues. Graetzel et al. [GFGB04] exploit stereo cameras to capture both color and depth information to achieve reliable, high-speed hand detection and tracking within a user-specified workspace; in such workspace, hand gestures are interpreted as mouse commands (pointer movements and button presses). The Gestix system by Wachs et al. [WSEG*07] is also designed for HCI in sterile settings; the authors propose a vision based gesture capture system, which interprets in real-time the user's gestures performed to navigate through and manipulate an image and data visualization environment. Dynamic navigation gestures are translated to commands according to their relative positions on the screen. Gesture recognition relies on tracking of the user's hand, based on color-motion cues; a finite state machine switches from navigation gestures to others, such as zoom and rotate. On the other hand, medical applications for which sterile operation is not required (diagnostic imaging, training, pre-operation planning) may instead take advantage from instrumented gloves and non-image-based tracking techniques. To this regard, Tani et al. [TMW07] discuss the use of a glove-driven interface in radiological workstations, and present a prototype that aims at integrating common functions, such as virtual manipulation and navigation control, with a basic gesture interface. The user can control the mouse by simply pointing to the screen and moving the hand, or perform gestures conventionally associated with specific commands in the given context. However, common functions like image rotation with respect to a given point, which are easily performed in a two-dimensional space, become more complex in three dimensions, as a greater degree of freedom is required to make the system effective. Therefore, approaches focusing on recognition accuracy, which map mouse and keyboard operation in a 2D space onto gesture patterns in a 3D space, might not exploit the full potential of this interaction technique.

In this study, we propose a framework aimed to enhance reliability, accuracy and overall effectiveness of gesture-based interaction applied to the interactive manipulation of virtual objects within a Mixed Reality context. The main goal is to effectively combine a small set of simple (user-wise) one hand and two-hand gestures with a context adaptive virtual interface. The interface's layout is visualized close-at-hand within the user's field of view, via a stereoscopic see-through Head Mounted Display (HMD). The interaction paradigm concurrently exploits both hands to perform precise manipulation of anatomic models reducing the interaction effort, allowing both complex actions on 3D objects or even more conventional operations. For instance, a keyboard can be visualized within the virtual space while fingers capturing enables virtual typing. We also describe how the aforementioned objectives have been pursued in a prototype application.

The rest of the paper is organized as follows. Section 2 presents the overall architecture of the proposed system; Section 3 describes the experiments conducted to assess system's usefulness and usability, and Section 4 draws some conclusions.

## 2. System description

The proposed framework exploits head/wrists tracking, gesture recognition and mixed-reality in order to provide effective human-computer interaction. The proposed architecture is shown in Fig. 1. We can summarize the overall system's operation as follows.

A brief calibration session is required to initialize glove sensors and check the trackers; afterwards, the system enters in operating mode. Two separate data channels, for the left and right hand respectively, are preprocessed and then both feed the Gesture Recognition Engine (GRE). This module analyzes posture tokens obtained from data-gloves, combined with motion data captured by the tracking system. As a result of this process a (one-hand or two-hand) gesture is recognized and passed to the Interaction Engine (IE), which eventually selects a function of the virtual keyboard (in the visual interfaces) or translates the gesture to a transformation (rotation/translation) of the 3D model sent to the Visualization Engine (VE).

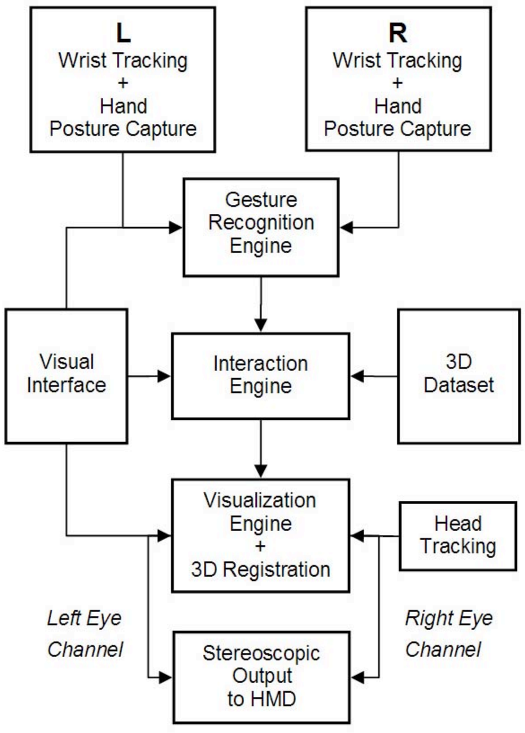
L
Wrist Tracking + Hand Posture Capture
R
Wrist Tracking + Hand Posture Capture
Gesture Recognition Engine
Visual Interface
Interaction Engine
3D Dataset
Visualization Engine + 3D Registration
Head Tracking
Left Eye Channel
Right Eye Channel
Stereoscopic Output to HMD

**Figure 1.** *The overall system's architecture.*

© The Eurographics Association 2010.

The latter computes two renderings (left and right) of the whole scene (interface plus scene model), which are coherently displayed by the HMD. The main hardware and software components of system architecture are described in detail through the following subsections 2.1 to 2.4.

## 2.1. Hands and head motion capture

The user input module is responsible for the user's hands tracking within 3D space, and hand posture acquisition. When the application is not aimed at sterile operations, for which an image-based hand tracking would be a design constraint, can be exploited a more accurate and reliable capture technique based on wireless instrumented gloves and ultrasonic tracking devices. Such choice simplifies the posture/gesture recognition stage, since for example, in this case inter-hands and inter-fingers occlusions are not an issue: each single finger has individual sensors for flexion and abduction, which are unaffected by any other finger. In our case, four fingers are considered excluding the thumb. Left and right hand posture acquisition is performed via a couple of wireless 5DT Data-glove 14 ultra (Fig. 2), featuring fourteen channels for finger flexion and abduction measurement each with 12 bit (0 to 4095) of sampling resolution. Additionally a binary (open/closed finger) value, resulting from the comparison of normalized joint flexion values to a threshold, leads to $2^4$ different combinations or postures of the four tracked fingers. It would be possible to access an even much more accurate finger status representation via each sensor's raw values, however, the adopted classification is more appropriate for the posture recognition in the addressed setting, as it simplifies both user training (partially flexed fingers do not compromise posture recognition) and the design of the gesture recognition engine. The four hand postures used in this study (fist, index finger point, not index finger point and flat hand) are shown in Fig. 3a. They have been selected as they are the simplest to perform for most users, and among the most used in natural interaction. As data-gloves do not provide any spatial information, the system relies on Intersense IS-900/VET six degrees-of-freedom ultrasonic motion tracking hardware (Fig. 2), with, to detect head and wrists position in 3D space, as well as their rotation on three axes (yaw, pitch and roll). The use of such data will be presented in the following, when we will discuss the Gesture Recognition Engine (GRE). The scalable wide capture volume is among the advantages of such setup, which frees the user from the need to be positioned in a precise place within the camera field of view, typical of video based solutions. Moreover, the setup is robust to electrical and magnetic fields, eventually present for example in a radiology facility (as long as they do not interfere with wireless transmitter frequencies). Furthermore, it provides an optimal accuracy in the range of one millimeter for distances and of 0,25 degrees for angles, as well as a high sampling rate (up to 180 measures per second) allowing an accurate capture of fast movements. A preprocessing is applied to each one of the six channels (for each hand), which filters capture noise by means of a high frequency cut and a temporal average of sampled values. Afterwards, both left and right data streams are obtained, each including the captured basic hand postures as well as the positional/rotational information.

## 2.2. Gesture Recognition Engine (GRE)

GRE checks for particular predefined posture tokens, which trigger associated interaction activities. The performed analysis is based on timed automata [AD94], able to detect one-hand and two-hand timed posture patterns, which are associated to manipulation functions. Timed automata are labeled transition systems used to model the behavior over time of single components in real-time systems. Classical state-transition graphs are further annotated with timing constraints. Accordingly, a timed automaton performs time-passage actions, in addition to ordinary input, output and internal actions. In more detail, a timed automaton is a standard finite-state automaton extended with a finite collection of real-valued clocks. The transitions of a timed automaton are labelled with a guard (a condition on clocks), an action, and a clock reset (a subset of clocks to be reset). Intuitively, a timed automaton starts execution with all clocks set to zero. Clocks increase uniformly with time while the automaton is within a node. A transition can be executed if the clocks fulfil the guard. By carrying out the transition, all clocks in the clock reset will be set to zero, while the remaining keep their values. Thus transitions occur instantaneously. Semantically, a state of an automaton is a pair composed by a control node and a clock assignment, i.e. the current setting of the clocks. Transitions are either labelled with an action (if it is an instantaneous switch from the current node to another) or a positive real number, i.e. a time delay (if the automaton stays within a node letting time pass). A timed automaton accepts timed words, i.e. infinite sequences in which a real-valued time of occurrence is associated with each symbol.

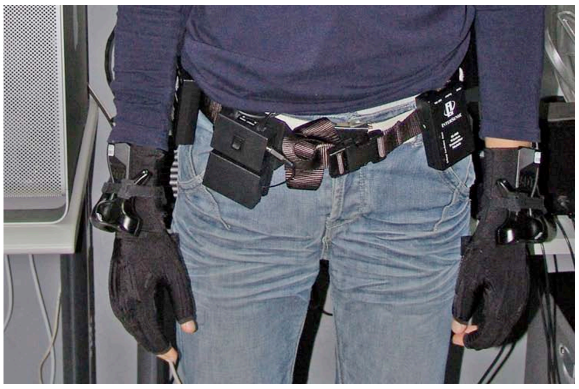

**Figure 2.** *A user wearing left and right handed wireless instrumented gloves (Data Glove 14 ultra from 5DT) plus the IS-900/VET Inertial 6DOF Tracking system from Intersense used for wrists/head wireless motion tracking.*

In this way, timed automata provide a feature that classical finite automata do not address in any way, namely timing. Embedding time allows to change the status of involved entities according to time-based events; this enhances the quality of user-system interaction when a feedback is required in a reasonable time, or when the time elapsed between elementary actions can influence the interpretation of their composition. The aim here is to augment the basic one-hand postures through timed patterns, or via a combination of left and right hands for a simple yet more powerful interaction. The use of timed automata offers a further key benefit for the proposed architecture, as it enables the framework designer to formally verify the interaction model by means of well established model checking procedures. In our framework, eight gestures are used as shown in Fig. 3. Four of them are defined as basic postures (Fig. 3a), two are defined through a two hand combination of basic postures (Fig. 3b), while the last two (Fig. 3c) are defined by a timed sequence of basic postures (for instance, fist-flat_hand-fist, or double pointing). Recognized gestures are represented by a vector including gesture index, first hand x-y-z spatial coordinates, first hand yaw-pitch-roll angles, second hand x-y-z spatial coordinates, second hand yaw-pitch-roll angles.

### 2.3. Interaction Engine (IE)

Each time a valid gesture is fully recognized by the GRE, the corresponding vector is provided to the Interaction Engine (IE), which deploys a similar architecture. It is also based on timed automata and is responsible for any visual interaction allowed by the system, by translating gestures into actions. Gestures are evaluated according to the current interaction status, so that the same gesture may trigger different actions in different operative contexts (rotation, measurements, landmark assignment, etc).

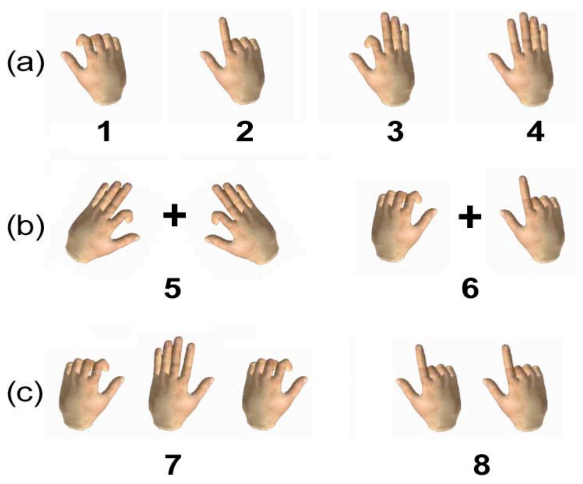

**Figure 3.** *One-hand (a), two-hands (b) and (c) time-based gestures.*

Operational modes and manipulation function are selected via a virtual interface. Such virtual interface is displayed as a frame surrounding the 3D application-related content (see Fig. 6), and may include textual information related to the ongoing operations (numerical values for coordinates and angles, distances, etc.). The layout of the interface is visually perceived as a floating object, thanks to the stereoscopic rendering. It is displayed in a close-at-hand position along the depth of the visual field, in agreement with the results of a calibration procedure. During such procedure, the user touches with the finger a sequence of small targets positioned at various depths, thus allowing an adaptation of the parameters that regulate the stereo effect. One of the main aims of the interface design is to minimize the number of gestures required to operate it. In some situations the need arises to perform classical keyboard-and-mouse supported operations, such as selecting an object from a menu of models or entering parameters for a complex operation. In this case, switching to a “real” device would break off the interaction flow in a disturbing way, as the user might need to move or change position to reach a different place or equipment, so interrupting the task operation flow. For this reason, a virtual keyboard is projected onto the actual environment, so that the user can comfortably switch from manipulation operations to system interaction operations without physically shifting his/her locus of attention/operation. Moreover, the familiar point-and-click pattern is re-proposed through fingers movements, to extend the range of available functions while keeping the number of basic gestures to a minimum. For this reason a gesture adaptation of the classical point-and-click interaction paradigm is adopted: selection is triggered hitting an active area by index fingertip (see Fig. 3, gesture #2), an action or a confirmation is triggered by double hitting (see Fig. 3, gesture #8), a cancel/escape command is triggered by a fist-flat_hand-fist sequence (see Fig. 3, gesture #7). Though only one-hand gestures are used to operate the interface, the design would allow an experienced user to exploit both hands to operate in a faster and more comfortable way (for instance, typing characters in a text field by both hands results in a much faster operation than via a mouse-like character-by-character selection, and would not require a physical keyboard). Visual and acoustical feedbacks are provided to confirm the “pressure” of a key or the acknowledgment of a particular command, thus reducing wrong operations. If required, interface layout can be hidden at any time via a gesture toggle. At present, only a basic set of 3D actions has been implemented, allowing to rotate/move a virtual object, to place landmarks over its surface and to take distance measurements between landmarks. Object pan is conventionally achieved with any of the two hands. Object rotations in 3D space are the actions that fully highlight the advantage of two-hand gestures. They are addressed by associating object rotation over three axes (though a user may lock one or two of them) with the rotation of a vector connecting the two points of contact between index finger and thumb on each hand (see gesture #5 in Fig. 3, and Fig.4).

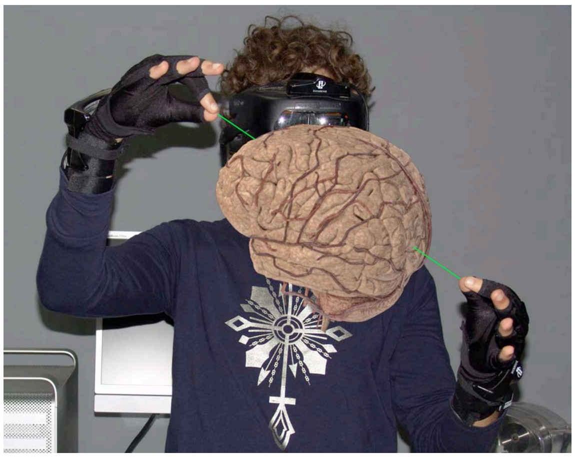

**Figure 4** - *User performing object rotation by means of two virtual handles corresponding to the extremities of the green vector. Due to the optical see-through design of the HMD this image is simulated as the scene was observed from a third person point of view. The brain model is the result of actual medical imaging processed to generate a 3D geometry and further optimized (about 625,000 triangles). The color texture shown is fictional.*

We found this approach to be much more precise than the typical one-hand based interaction often seen, for instance, in VR applications. As a matter of fact, it allows a greater control of rotation (two hands better visualize the overall rotation), a more comfortable operation (it does not matter how each wrist rotates during interaction, as only the vector connecting the two hands is relevant), and a less jerky interaction, yet without losing responsiveness (a weighted average of both hands spatial information improves tracking). Once the rotation operation has been selected, the user can set the rotation handle for each hand (the anchor points used to interact with the model) by gesture #3 in Fig. 3 by simply moving the fingers along the object's surface. A valid handle location, i.e. one that lies within a valid region, is highlighted by a colored spot. If the gesture #5 (Fig. 3) is recognized, then the vector connecting the two handles is visualized and the interactive rotation is performed until this condition is true. Landmark positioning over an object surface may be accomplished according to two different operation patterns. In the first case, a rotation of the object is performed as explained before, to expose the location of interest, and then a landmark is placed by double hitting (gesture #8 in Fig. 3). A more intuitive, though less precise, operation pattern requires to perform the rotation of the object by a single hand, grasping the object (see gesture #1 in Fig. 3) in a position which acts as the pivot point, and double pointing the location on which the landmark is to be assigned. For any task involving positioning in 3D space, a precise calculation of the actual finger positions is performed through a combination of a forward-kinematics approach applied to a 3D parametric hand model, which is adapted to the real user's hand measures during a calibration session. In this case the raw flexion values are exploited for each finger. This setup is performed only once, and may be saved and retrieved during system start up.

## 2.4. Visualization Engine (VE) and software environment

3D objects, as well as the virtual interface, are processed by the Visualization Engine (VE), which is responsible for real time transformation and stereo rendering of 3D scenes. It manages all the graphic tasks typically required to convert 3D data in a sequence of rendered frames. Similarly to the previously described engines in subsections 3.2 and 3.3, the VE is built on the Quest3D graphics programming environment and on the underlying DirectX API. Quest3D is a commercial authoring environment for real-time 3D applications. Its most peculiar features are the graphical nature of the software environment and the "edit-while-executing" philosophy. Dynamic simulation is accomplished by means of the Open Dynamics Engine (OpenDE, a.k.a. ODE) open-source library or even via the Newton Dynamics API.
Some examples of such channels implemented for hand gesture recognition are provided at different levels of programming detail in Fig. 4a and 4b. Almost any graphic format for 3D objects can be imported in a Quest3D scene. To provide the AR environment, the Visualization Engine exploits the user's head position and orientation. These data are processed to transform the virtual content as seen from the user's point of view, and coherently with a 3D model of the surrounding environment.

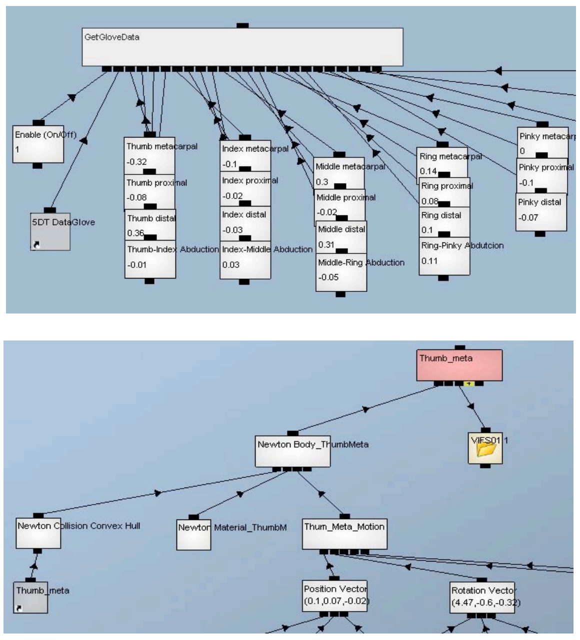

**Figure 4a-4b.** *Sample fragments of the GRE implemented via the Quest3D graphical programming environment. 4a) Dataglove and wrist tracker handling . 4b) Metacarpal thumb control.*

Such crucial task is referred to as 3D registration. Actually, any AR architecture requires a precise registration of real and virtual objects. In other words, the objects in the real and virtual world must be properly aligned with respect to each other, or the illusion that the two worlds coexist will be compromised. To achieve such non trivial goal two main requirements have to be satisfied: a) the position and orientation of the user's head have to be precisely tracked at a high sampling rate; b) the physical world, or at least the relevant objects for the application, have to be precisely measured in the same 3D space where the user operates. At runtime, two rendering cameras (one for each eye) are built, matching the exact position/orientation of user's eyes, and accordingly transforming each vertex of each virtual object to be displayed onto the real scene. Two renderings (left and right) are then computed and coherently displayed through an optical see-through HMD (a Cybermind Visette SXGA), a helmet which works by placing optical combiners in front of the user's eyes. These combiners are partially transmissive, so that the user can look directly through them to see the real world. The combiners are also partially reflective, so that the user sees virtual images bounced off the combiners from head-mounted LCD monitors. It is worth to note that, due to the aforementioned nature of the viewing device, the overall (virtual + real) images as seen by the viewer during the experiments cannot be recorded, so that figures 5a and 5b provided hereafter have been simulated by overlaying the virtual content outputted by one channel of the visualization engine onto images of the working environment. The rendering engine has been tailored to the optical see-through HMD, but it can be adapted to video see-through displays. Eventually, a selective culling of a virtual object may be performed whenever it is partially or totally behind a real object. The suitability of such process depends on the application at hand, also considering the overhead required to model the real environment in a more accurate and complete way. However, according to our experience with the implemented prototype for medical imaging, an optical see-through HMD is preferable over a video see-through solution (featuring comparable display resolution) if the quality of its optical combiners is high enough to provide a reasonably wide field of view. Unfortunately this feature usually makes this equipment very expensive.

## 3. The case study: medical imaging

The application we present is aimed to support visualization/manipulation of 3D medical data. In this context, it is very common to observe radiologists or even surgeons preferring to work on 2D sections of CT or NMR (for instance to place a landmark or to delimitate a region) as they often do not feel confident with the 3D tools provided by commercial diagnostic systems, which are considered visually appealing but not reliable. As the three dimensional data generated by these systems are the direct product of the (reliable) 2D sections, the reason behind this belief possibly resides in the way these data are made accessible and on how complex becomes to interact with them through a bi-dimensional display and interface. Though stereoscopic displays have been already recommended for this kind of applications, the main issue is conditioned by the interface side. Indeed, to operate on a 3D surface or volume requires a more powerful way to specify actions or locations in 3D space which is not well addressed by a usual 2D based interaction paradigm, involving mouse or trackball The software module developed to address this problem is only a part of a wider project which aims to improve the usage of three dimensional data in medical imaging practice. Consequently, since from an early stage of this study, the main concern has been to assess if the proposed approach to 3D manipulation could match the requirements for a more accurate (therefore more useful) and intuitive way to deal with complex data. At the same time many conventional functions have to be accessible and easy to use in a way similar to the one with common devices (mouse and keyboard). This compatibility is achieved by means of the floating interface and virtual keyboard visualized according to the aforementioned AR setting (see Fig. 5a – 5b).

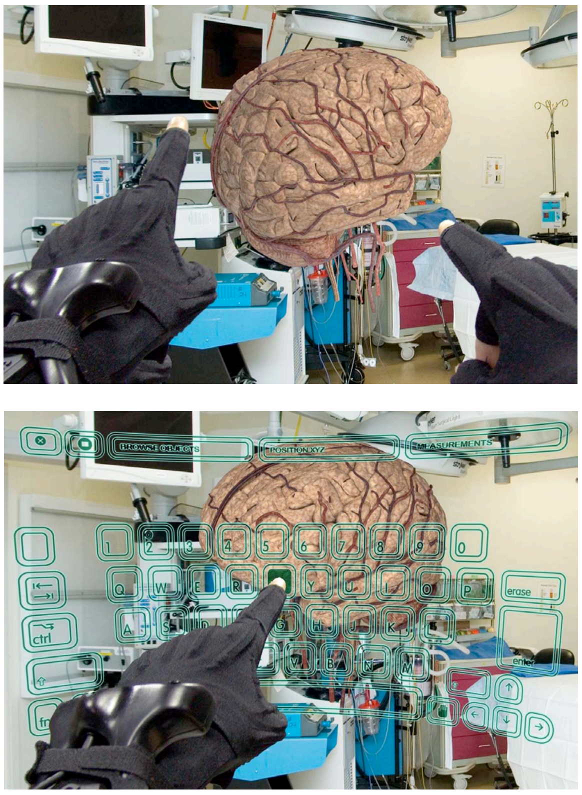

**Figure 5a-5b:** *A simulated example of the user's field of view during gesture interaction within an AR based environment. The floating interface layout (which can be hidden (top) or shown (bottom) via a gesture toggle) is projected onto the central region of the field of view to enable the selection of the required functionalities as well as the interaction with virtual objects.*

This design allows eliminating distractions and breakdowns in the gesture patterns, as the user has not to move from his place or change his position to type commands to the computer. Due to the way the 3D models are computed (from CT or NMR), the exploration provided by the implemented system can proceed by layers, allowing to explore the desired anatomical district at different depths. Such setting is intended to mainly address training or learning activities, but it can also be exploited in operative contexts. The functionalities already implemented include: object browsing, two-hand operated object rotation/traslation with respect to any axis, object transparency setting, landmark positioning and landmark-based measurements.For testing purposes we exploited a library of anatomical 3D models, since the visualization engine is currently suited to operate on polygonal-based objects rather than on voxel-based datasets, usually resulting from the processing of diagnostic images (see Fig. 7). As an alternative, we could generate a polygonal mesh from the iso-surface resulting after a segmentation process applied to a voxel-based model. Whatever the technique adopted to create the virtual anatomy, the inherent geometrical complexity of human organs often leads to a high polygon count for the 3D scene, with a value ranging from many tens of thousands to even million of triangles, depending on the required accuracy. The real-time rendering hardware is based on a workstation including two quad-core Xeon Processors coupled with a Nvidia Quadro FX-5600 graphics board featuring 128 parallel cores and 1,5 GB of VRAM. During experiments, this hardware setup has been capable to render in stereo scenes featuring more than 5 million of polygons (lighted and textured), at an output resolution of 2x1280x1024 pixels, with a frame rate always above 30 fps. Expert evaluation and some preliminary and informal tests were performed with a small group of users, through a specific walkthrough method. A suitable evaluation grid was prepared to obtain comperable outputs from different experts. The grid was organized in four sections: graphic presentation, architecture (structure and navigation), functionality (suitability and correctness of functions), and support to the user. Each section includes specific items to account for, and, for each of these, encountered problems, positive aspects and suggestions to be listed by the experts. User testing is crucial to confirm the above observations, and to highlight possible further problems. To this aim, we prepared a user questionnaire to assess the perceived quality of the interaction after performing a number of tasks. The first evaluation sessions involved seven users, part doctors and part radiology technicians, and the obtained results are encouraging. All subjects reported an advantage in either achieving the right landmark placement, or in the object control, with an overall good confidence feeling during interaction. In the following we list some examples of tasks performed by the users:

• Remove buttons from visualization.
• Progressively remove anatomical layers of tissue (available in the model) from the visualized zone, till to possibly reach the inner one.
• Progressively add anatomical tissues (available in the model) to the visualized zone, till to possibly reach the outer layer.
• Modify the transparency of a layer.
• Load a new object within the visualization space.
• Select an object within the visualization space.
• Move an object across the visualization space.

In the final questionnaire, the questions were presented using a five-point Likert scale, were respondents specify their level of agreement to a statement. In order to avoid any bias, some statements were in positive form and others in negative one. This was taken into account in the final assessment of results. We list the proposed statements.

1. Tools available to interact with the visualization zone (glasses, glove, etc.) allow natural actions in a comfortable way
2. The number of gestures provided by the interface is not sufficient
3. The meaning of the interface gestures is made clear for the user
4. There is a suitable number of helps to understand how to perform actions
5. Navigation within the application is very difficult
6. Acting upon visualization objects is very easy
7. The type and number of provided actions to interact with objects is not sufficient
8. It is easy to select objects within the visualization space

Results are summarized in the following table

| Question | I strongly agree | I agree | I do not know | I disagree | I strongly disagree |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 2 | 1 | 0 |
| 2 | 0 | 2 | 1 | 4 | 0 |
| 3 | 5 | 2 | 0 | 0 | 0 |
| 4 | 0 | 7 | 0 | 0 | 0 |
| 5 | 0 | 2 | 2 | 3 | 0 |
| 6 | 0 | 6 | 1 | 0 | 0 |
| 7 | 0 | 0 | 2 | 5 | 0 |
| 8 | 5 | 2 | 0 | 0 | 0 |

We also performed a subsequent interview with users, to better analyze the questionnaire results. As for answers to question 1, the reason for having only 4 out of 7 agreements was due to the lack of familiarity with virtual environments and related interaction tools, as it was already highlighted by the expert evaluation. Answers to question 5 show a consistent result, since some users reported some difficulties due to lack of familiarity with the environment. The number of recognized gestures was considered quite satisfying, as resulting from 4 disagreements to the negative statement in question 2. However, some users would have preferred to have more freedom in their gesturing. It is interesting to notice that a slight different result had been obtained in some previous testing sessions, which were performed in a very early design stage. During different sessions, we found that a higher number of gestures, while resulting more attractive to users, can be at the same time a factor of insecurity. This is due to the higher uncertainty of having performed the right sequence of elementary movements. Responses to questions 6, 7 and 8 consistently underline that users were quite satisfied with the possible actions on visualization objects. Responses to questions 3 and 4 highlight the users' overall satisfaction with both the interaction model and the available help.

## 4. Conclusions and future enhancements

In this paper, we presented a framework for gesture-based interaction aimed at 3D data manipulation in a mixed reality environment. The proposed architecture exploits easy to perform one-hand, two-hand and timed interaction patterns, combined with a virtual floating interface, enabling a wider set of commands and functionalities. The main aim is to provide the user with a more natural and effective interaction level, improving at the same time the accuracy of the user-system interaction during the usage. In particular, we explored how to address medical imaging scenarios. To this aim, we performed both expert evaluation based on a walkthrough method, and user testing by means of a specific questionnaire and subsequent interviews. The obtained results, although on a small group of users (seven), are encouraging. All subjects reported an advantage in either achieving the right landmark placement, or in the object control, with an overall good confidence during interaction. With regards to the visual engine and the typologies of 3D content supported, we are currently working to enable the visualization of voxel-based and polygon-based objects at the same time. Moreover, as the timed-automata based gesture recognition approach is very suited to support multi-user interaction, it would be interesting to work on collaborative interaction, enabling cooperation between multiple users, eventually by means of combined gesture patterns or coordinated actions within a common AR environment.

## References

[ABF*01] Azuma, R. Baillot, Y. Behringer, R. Feiner, S. Julier, S. MacIntyre, B. (2001) Recent advances in augmented reality, IEEE Computer Graphics and Applications, Volume: 21, Issue: 6, 2001, pp.34-47

[AD94] Alur, R and Dill, D. L. (1994). A theory of timed automata. Journal of Theoretical Computer Science, 126(2):183–235.

[BKFT00] S. Balcisoy, M. Kallmann, P. Fua, D. Thalmann (2000). A framework for rapid evaluation of prototypes with augmented reality, Proceedings of the ACM symposium on Virtual reality software and technology, Seoul, Korea, pp. 61 – 66, 2000

[BRI91] M. Bricken (1991). Virtual reality learning environments: potentials and challenger, ACM Computer Graphics, Volume 25 , Issue 3, pp. 178 – 184, 1991

[BRO99] Brooks, F.P., Jr. (1999) What's real about virtual reality? IEEE Computer Graphics and Applications, Volume 19, Issue 6, pp. 16- 27

[DFG*05] W. Dangelmaier, M. Fischer, J. Gausemeier, M. Grafe, C. Matysczok and B. Mueck (2005). Virtual and augmented reality support for discrete manufacturing system simulation. Computers in Industry, Volume 56, Issue 4, May 2005, Pages 371-383

[GFGB04] C. Graetzel, T. Fong, S. Grange, and C. Baur. (2004). A Non-Contact Mouse for Surgeon-Computer Interaction. Technology and Health Care, IOS Press, vol. 12, no. 3, pp. 245-257.

[KHLE97] C. Krapichler, M. Haubner, A. Lösch, and K. Englmeier, (1997) "Human-Machine Interface for Medical Image Analysis and Visualization in Virtual Environments", IEEE conference on Acoustics, Speech and Signal Processing, ICASSP-97. Vol 4, pp. 21-24.

[KR93] Kaltenborn K.-F., Rienhoff O. (1993) Virtual Reality in Medicine. Methods of information in medicine. Vol. 32, N 5, 1993, pp.407-417

[KS98] Kohler, M. and Schroter, S. (1998). A Survey of Video-based Gesture Recognition - Stereo and Mono Systems. Technical Report 693, Informatik VII, University of Dortmund.

[PTB*02] Poupyrev, I., Tan, D.S., Billinghurst, M., Kato, H., Regenbrecht, H., Tetsutani, N. (2002) Developing a generic augmented-reality interface, Computer, Volume: 35, Issue: 3, 2002, pp. 44-50

[TMW07] Brian S. Tani, Rafael S. Maia, Aldo von Wangenheim. (2007). A Gesture Interface for Radiological Workstations. Proceedings of the Twentieth IEEE International Symposium on Computer-Based Medical Systems

[WSEG*07] Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Craig Feied, Mark Smith, and Jon Handler. (2007) Gestix: A Doctor-Computer Sterile Gesture Interface for Dynamic Environments. A. Saad et al. (Eds.): Soft Computing in Industrial Applications, Springer, ASC 39, pp. 30–39.