Evaluation of Gesture Based Interfaces for Medical Volume Visualization Tasks

Can Kirmizibayrak[1,§], Nadezhda Radeva[1,ψ], Mike Wakid[1], John Philbeck[2], John Sibert[1], James Hahn[1]

[1] Department of Computer Science
The George Washington University
Washington, DC, USA

[2] Department of Psychology
The George Washington University
Washington, DC, USA

Abstract

Physicians are accustomed to using volumetric datasets for medical assessment, diagnosis and treatment. These modalities can be displayed with 3D computer visualizations for physicians to study the overall shape and internal anatomical structures. Gesture-based interfaces can be beneficial to interact with these kinds of visualizations in a variety of medical settings. We conducted two user studies that explore different gesture-based interfaces for interaction with volume visualizations. The first experiment focused on rotation tasks, where the performance of the gesture-based interface (using Microsoft Kinect) was compared to using the mouse. The second experiment studied localization of internal structures, comparing slice-based visualizations via gestures and the mouse, in addition to a 3D Magic Lens visualization. The results of the user studies showed that the gesture-based interface outperformed the traditional mouse both in time and accuracy in the orientation matching task. The traditional mouse was the better interface for the second experiment in terms of accuracy. However, the gesture-based Magic Lens was found to have the fastest target localization time. We discuss these findings and their further implications in the use of gesture-based interfaces in medical volume visualization.

CCS Categories: I.3.6 [Computer Graphics]: Interaction Techniques, J.3 [Life And Medical Sciences]

Keywords: volume visualization, gesture-based interfaces, human-computer interaction, user study, medical visualization

Links: DL PDF

1. Introduction

Medical assessment, diagnosis and treatment rely heavily on volumetric imaging modalities, such as CT and MRI scans. The common practice of visualizing 3D modalities as a series of 2D slices requires physicians to mentally reconstruct the volume by cognitively merging the 2D images, thereby building a 3D mental object [Dev 1999]. The task of reconstruction and understanding the spatial relationships between external and internal parts of the objects is challenging. Therefore, 3D medical visualizations that allow displaying and interaction of CT and MRI data are widely used in training and diagnosis [Silén et al. 2008]. However, perceiving the depth relationship among three-dimensional objects on a 2D screen presents a major challenge in the field of medical visualizations. Some of the issues of such visualizations are volume occlusion and ambiguity (or absence) of depth cues.

To help alleviate this problem, various manipulation tools have been created and tested. Examples include changing the viewpoint to use parallax effects or changing rendering parameters such as transparency. Some of the most widely used interaction interfaces are the mouse, electromagnetic and optical trackers. Even though these have been proven accurate and helpful in volume visualizations, they present challenges when introduced in surgical systems used in the operating room (OR).

The mouse is a device most physicians are comfortable using in their work, but it has disadvantages when mapping the 2 degree of freedom (DOF) motion of the mouse to a 6-DOF motion in the 3D real world. Electromagnetic trackers are accurate in detecting motion. However, electromagnetic interference with other medical equipment might present problems. Furthermore, they are costly and may be difficult to set-up, which is a problem also applicable to optical trackers. The mouse and trackers can introduce contamination and require special sterilization procedures when used in the operating room.

In this paper we study the prospect of using hand gestures acquired using depth cameras (Microsoft Kinect) in manipulating 3D medical visualizations. We compared accuracy and interaction speed of Kinect with respect to the same parameters measured when using the mouse. We conducted two user studies, which aimed to study the performance of interaction methods in rotation tasks and understanding the internal structures in a volumetric dataset.

2. Related Work

In earlier work, Hauptmann [1989] suggested that users prefer to use gestures and speech when interacting with computer images. The most used interactions were interaction with several fingers at the same time, moving both hands in all three degrees of freedom for performing various tasks, and ultimately using a combination of fingers, hands and several words to execute any desired task. Two-handed interactions for 3D volume manipulations have been observed to be more natural and better performed when implemented with the user and the task in mind [Zeleznik et al. 1997; Hinckley et al. 1994]. Zeleznik et al. developed various two-handed cursor techniques for 3D object transformation and to manipulate the position of the camera in the 3D world.

§ email: kirmizi@gwmail.gwu.edu
ψ email: nradeva@gwmail.gwu.edu

VRCAI 2011, Hong Kong, China, December 11 – 12, 2011.
© 2011 ACM 978-1-4503-1060-4/11/0012 $10.00

Until recently most gesture systems would rely on trackers or some other technology that would require wires running to the computer ports. However, Breuer et al. [2007] and Ahn et al. [2009] point out two of the most important principles in Human Computer Interaction systems – the action of the user should not be invasive and the interaction device should not disturb the user. This is also the motivation for gesture-based systems such as Gestix [Wachs et al. 2007]. This makes many of the methods used in motion tracking inadequate because of the nature of the technology and the wires running from the device to the computer. However, these efforts were more geared towards implementation aspects of such systems, and no user studies have been done to validate the importance and the adequacy of the system and its usability.

In the past couple of years more attention has been paid to the integration of game controllers like the Wii remote [Gallo et al. 2008] and the XBox game controller in the manipulation of 3D objects and tracking of the user. Ardito et al. [2009] conducted a user study to test the performance of users when manipulating 3D virtual objects using three different methods: the mouse, Wiimote and the XBox controller, concluding the mouse can outperform the game controllers but user performance is heavily dependent on familiarity of the game interfaces.

3. Gesture Based Interfaces for Medical Volume Visualization

The current standard interaction interfaces for medical applications can be classified in two groups: image-space and object-space. Image-space interactions are performed by interacting with the produced visualization in the virtual world, and are most of the time in 2D. Object space interactions enable the users to interact with the visualization in the real world, and can perform 3D interactions. Currently, the most widely used image-space interfaces use mouse input, while object space interactions are performed by using trackers. Both of these groups have drawbacks, some of which are explained in detail by Johnson et al. [2011] from an interventional radiology context. Sterilization is an important concern in this domain. Also, physicians can lose attentional focus at crucial time-points if there is a need to move to a different location to interact with the system. The use of gestures can provide an alternative that can overcome the aforementioned limitations of tactile interaction interfaces. The introduction of Microsoft Kinect has provided researchers with a cheap and robust alternative to track user movements and translate them into interactions, with applications in many diverse areas [Tanz 2011]. In this paper, we provide human factors analysis of a Microsoft Kinect based system with the ultimate goal being development of a device useful for use in medical visualization settings, especially in the operating room.

In this section, we will describe some of the interaction methods using Kinect for medical visualization applications. All of these interaction methods used the tracked location of one or both user hands, which were extracted using the OpenNI framework [OpenNI Organization 2011]. Our aim in providing these methods here is to describe how they were implemented in our experiments, while the alternatives are presented for comparison purposes.

- **Rotation**

The rotation with two hands can be implemented in a variety of ways. The method we chose for our experiments was using the relative locations of two hand positions in the real world. In this method, the vector between the hand positions (right to left in our implementation) was used to calculate the X- and Y- plane rotations of the object. This is shown with examples in Figure 1. One way to understand this rotation is to imagine that the user is holding the object from its sides (as indicated by L and R in Figure 1) and performing the rotation accordingly (this was the way the interface was described to the subjects in our experiments). Other techniques such as ratcheting the rotation incrementally were found to be neither intuitive nor robust in our pilot studies. This was mainly caused by the difficulty of establishing an engagement/disengagement trigger to start/stop rotation (thresholding the hand distance was used in our pilot studies for ratcheting).

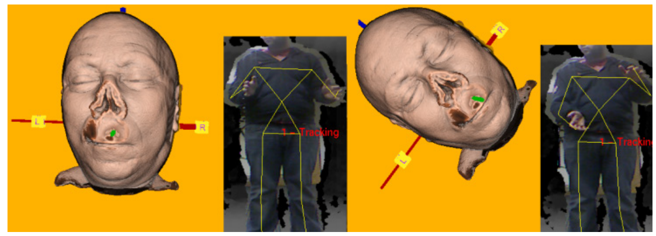

Figure 1. Rotation by hand locations. *The three main axes of rotation are used as rotation references. The yellow cubes denoted by L and R in the visualization respectively correspond to left and right hand locations of the user, the correspondence to user's hand locations can be seen in the right side of the figure.*

- **Gestures as a Mouse**

The projected hand position can be used to control the cursor location. For this, we need a trigger for the mouse click: two possibilities are using the dominant or non-dominant hand distance to the camera as triggers. For this, the depth value of hand locations are compared with the depth of another joint (e.g. right hand with right shoulder), and if the distance is more than a threshold, a mouse button down action is registered.

- **Slice-based Visualization**

2D slices are the most commonly used visualization method for 3D medical datasets because of their familiarity to the radiologists and surgeons and their success in presenting size information since no perspective projection is used. These visualizations can be augmented with 3D renderings where the location of the slice can be shown as a placeholder to help physicians understand the relative location of the current slice to the rest of the dataset. The location of these slices in X, Y and Z planes can be changed by gesture based interactions. In our implementation we used the relative height (y value) of the right hand with respect to the torso location of the user extracted by Kinect to select the slice location on the XZ plane, with the shoulder width used to normalize the height to [0,1] range of the dataset.

- **3D Magic Lens**

The Magic Lens interface was proposed by Bier et al. [1993] and later was applied to volume visualization by several researchers [Viega et al. 1996; Kirmizibayrak et al. 2010]. The location of a volumetric Magic Lens can be controlled by a gesture-based interface. An example can be seen in Figure 2(a). In this example, the Magic Lens is used to apply different transfer functions to inside and outside the currently selected region. In our recent work, we have proposed to use lenses as volumetric brushes to perform volume editing tasks. Implementation details for such rendering approaches are beyond the scope of this paper. Interested readers can refer to our previous work [Kirmizibayrak et al. 2011]. An example volume editing result is shown in Figure 2(b).

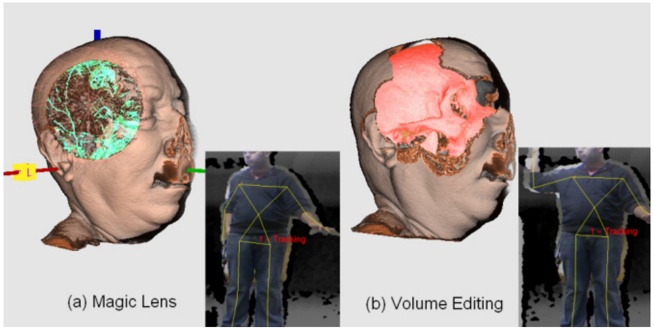

Figure 2. Examples of Magic Lens visualization and volume editing. *The volumetric lens location in (a) and brush location used to create (b) are controlled by the user's right hand, while the editing mode is activated by raising left hand.*

The methods presented in this section can be useful for a variety of medical visualization tasks. In order to evaluate the effectiveness of gesture-based interfaces, we have conducted user studies of two of the most widely used tasks for interactive volume visualizations; rotation and finding structures inside a volume. These two volume interactions are a necessary part in the exploration of medical volumes, allowing the physician to study both the overall shape of the object and its inner structures.

4. Experiment I: Rotation

The first experiment asked the users to rotate a volume rendered visualization to match a target visualization, which was rotated randomly by one of several pre-defined possible orientations. The computer screen was split into two parts. The right part displayed the target orientation of the volume. The left side showed the same volume but in a different orientation. The user had control over the left-side volume and rotated the object so that it matched the orientation shown on the right hand side. In order to limit the amount of time spent by each user, we have limited the possible random rotations to 0, -45 and 45 degrees in the X and Y planes, resulting in 9 possible orientations for each study. These orientations were presented to users in a randomized order. The users performed all 9 matching tasks in one interaction interface before switching to next. The order of the interfaces was also selected randomly. After all possible interfaces were tested, the process was started again for a second test, with an independent random order. Our aim with this randomization of interface orders was to minimize any systematic effects of familiarity and learning by distributing these effects randomly across users, and the analysis of results showed no order effects were present in either of the experiments.

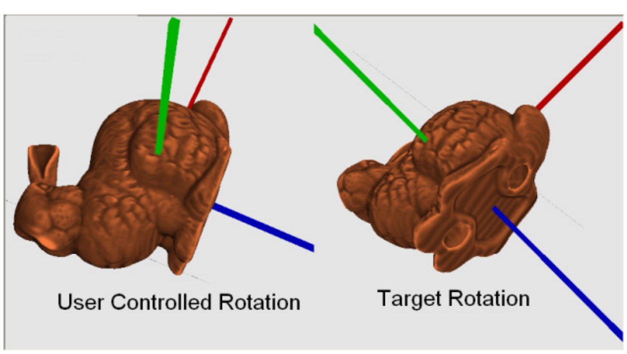

Figure 3. A sample screen for Experiment I. *The users try to match the rotation on the left side of the screen to the target orientation seen on the right.*

The subjects were instructed to get close to the target as fast as possible and to indicate when they think they are reasonably close to the target shape. The test administrator then advanced to the next target. We chose not to use an automatic or more direct method for advancing because of the difficulty for the users to indicate a match when both hands are used for interaction, and since automatic advancement methods were found to introduce undesirable errors in our pilot studies.

Training

All users provided their informed consent to participate. The experimental protocol was reviewed and approved by the GW ethics committee. A short description of the system was given next. The goals of the experiment were not explained in detail to the users to avoid any bias towards one of the interfaces, except explaining that the performances of different interfaces for some common volume visualization tasks were being compared. Different datasets were used for training and the experiments (Piggy Bank and the Stanford Bunny [The Volume Library] respectively).

Pilot Study

Before performing our experiment, we conducted a pilot study with five members of our research group (not including the authors). We were especially concerned about the use of gestures to simulate a mouse, since the clicking gesture using a depth threshold felt awkward to use in our pre-experiment tests. Similarly, using the other hand to perform the clicking was thought to be distracting and not very intuitive.

In the pilot study, three interfaces to perform rotation tasks were tested: Kinect two-handed rotation (K2HR), Kinect mouse rotation (KMR) and traditional mouse rotation (TMR). For K2HR, the rotation was performed as explained in Section 3, by using the vector from user's right hand to left hand. For TMR, the rotation was performed around the center of gravity of the object (as indicated by the intersection of the displayed axes) using the GLUT-Manipulator class rotate function [Everitt 2000]. We chose a standard and widely used rotation function to perform mouse rotation to ensure an accurate comparison. The function rotates the object around the center of rotation of the object when the user left-clicks and drags the mouse. The axis of rotation is perpendicular to the direction of the current mouse movement. For KMR, the threshold for clicking was selected to be the plane 40 centimeters closer to the camera than the right shoulder plane (both planes perpendicular to the image plane), while the projected right hand location was used to control the mouse location as described above. The results of the pilot study reinforced our assumptions that KMR in this implementation would not be comparable to the other two interfaces. The users commented the interface was frustrating to use and the overall time spent on these experiments was significantly higher than on the other two interfaces – average time for KMR was 29.3s versus K2HR time of 14.8s and TMR time of 17.0s . Therefore, we decided to omit KMR from our experiment and decided to proceed with using only TMR and K2HR.

Controlled User Study

The study group consisted of 15 people between the ages of 22 and 38, with an average age of 29.4. Out of the fifteen users, 12 were male and 3 were female. Our subjects were all college-educated adults. None of the subjects indicated they have used a gesture-based interaction interface before, or played with the Microsoft Kinect gaming platform. 7 of the subjects indicated they occasionally use software that produces 3D renderings, while 8 said they never use such software. On average it took about 6 minutes to perform the training and 12 minutes to perform and collect all data. The subjects stood about 6 feet from the monitor (a 42-inch LCD screen) for the Kinect interactions, and a mouse was provided around the same distance to be used standing up to ensure consistency between the interfaces. The monitor orientation was constant across trials, and was set to face the approximate centerline between mouse and Kinect locations.

We hypothesized that users would find it easier or of equal difficulty to rotate and match the target orientation when using K2HR than when using TMR, because gestures can be considered as a more intuitive interface. However, the TMR interface could prove to be more accurate than K2HR since it is possible that precise small movements could be performed more easily with the mouse.

Data Analysis and Results

The data collected were analyzed by a 2-tailed, paired-sample t-test with 14 degrees of freedom (each subject's mean value across all trials was used for analysis). The dependent variables were accuracy and time (analyzed in separate t-tests), while the independent variable for each test was the interface type (K2HR or TMR). For the accuracy measure, we used the quaternion representation and used quaternion norm of the difference between target and the user selected rotations to measure rotation error. The box plots of the results are displayed in Figure 4 for rotation error, and in Figure 5 for time.

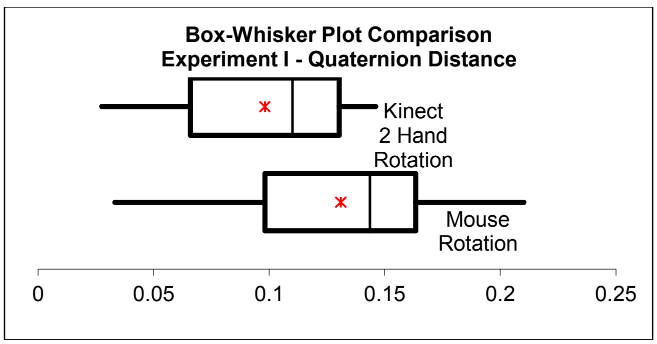

Figure 4. Box plot of results for Experiment I (rotation error). Left to right: TMR, K2HR.

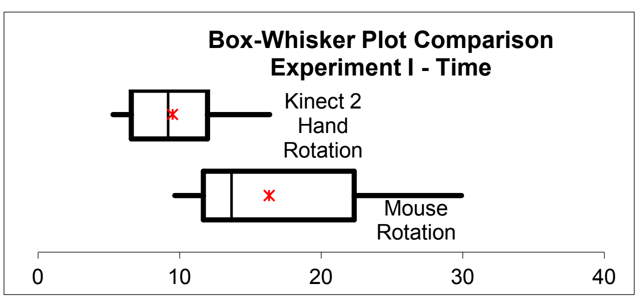

Figure 5. Box plot of results for Experiment I (time). Left to right: TMR, K2HR.

The results showed that the average time it took users to rotate the object to its target orientation was 16.3s with TMR and 9.4s with K2HR. Furthermore, the variance in time for K2HR was much lower than the variance for TMR. This shows that users quickly adjusted to the K2HR interface. Moreover, the rotation error analysis showed that the accuracy in orientation matching with K2HR (average error of 0.09 quaternion units) is higher than the accuracy using TMR (average error of 0.13 units). The t-test produced the following results: for error, the p-value was 0.0066. For time, the p was < .0001. Therefore, we can conclude that for both testing conditions the gesture-based interface performed statistically significantly better than the mouse as a rotation interface. The results are significant in showing that the K2HR is an accurate and fast interface for rotation matching tasks.

5. Experiment II: Target Localization

For this experiment, the task was localizing artificially created targets inside a volume. This task aimed to replicate the way physicians display volumetric datasets in a variety of medical diagnosis and treatment applications – i.e., by studying consecutive 2D slices of the volume to detect abnormalities or changes in the internal structures. Three interfaces were tested: Mouse slice (MS), Kinect slice (KS) and 3D Magic Lens (ML). For MS and KS, the users controlled the location of a cross-section, shown in the right half of the window. The slice location with respect to the volume was also displayed on the left half of the window, along with an opaque volume rendering of the dataset, which did not show any of the target structures within. The ML interface was provided as an alternative interface to explore the inner structures in 3D instead of using 2D slices. The Magic Lens filter was defined as showing the target structures but not the dataset boundary surface, in effect acting as a transparent volumetric sub-region that reveals the inner structures. A cylinder was selected as the lens shape. The boundaries of the lens were indicated by drawing lines showing the shape of the cylinder. The center of the lens (which the subjects are instructed to match to the target) was displayed as a small square on the top of the cylinder.

The targets were created by copying a smaller volume texture to specific locations of the volume dataset, which were defined manually beforehand and were common across all subjects. 10 possible target locations were defined. In each trial, 9 of these were used as distractors, while the remaining one was the target for that trial. The difference between targets and distractors was the size: targets had a radius of 15 voxels while distractors had a radius of 10 voxels. The users were instructed to search the volume by either changing the slice location (using the mouse or Kinect) or moving the Magic Lens around. When the target was located, users were instructed to try to center it and instruct the test administrator to advance to the next trial. Two example results are shown in Figure 6 for the slice-based and Magic Lens interfaces. We used the same kind of advancement method to maintain continuity with our previous experiment.

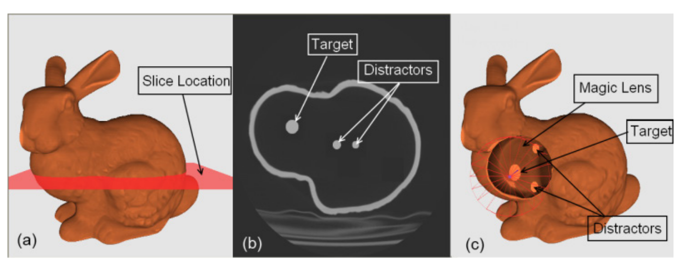

Figure 6. (a) and (b) A sample screen for Experiment II (for slice-based visualizations), (c) a sample screen for Experiment II (for Magic Lens visualization).

Pilot Study

We again performed a pilot study beforehand with the same group of pilot users. Since the performance of the three possible interfaces was found to be comparable, we decided to include all three interaction methods in our experiment.

Controlled User Study

This experiment was done with the same group of people who participated in Experiment I, and same datasets were used for training and experiments. Experiment I and Experiment II were performed sequentially (as were the pilot studies, as we believed the tasks to be reasonably different from each other and learning effects should not affect the performance). The users were allowed to do a training session similar to the one in Experiment I. On average, it took about 3 minutes to perform the training and 6 minutes to perform and collect all data. Each interface was tested with 5 target locations, and the possible target locations in the training and experiment were different from each other (one half of the 10 used in training, while the remaining half used for the experiment). After all three interfaces were tested, the experiment was repeated again in an independently random order, in the same manner of Experiment I.

Our hypothesis for the experiment was that users would locate the target object slightly faster when using the MS rather than using KS because of the mouse's familiarity. Moreover, the lack of an apparent frame of reference could be confusing to the users. Furthermore, precise movements could be more difficult with the KS and ML because of the accuracy of the skeleton extraction from depth images. We also expected that the time it takes for the user to locate abnormal structures in the inside of the volume when looking at 2D slices would be close to the time it takes to use the Magic Lens interface, since the user is presented with the 3D shape using ML. We aimed to determine if the ML helps the user better understand the internal structures of the volume and the spatial relationship between objects.

Data Analysis and Results

We performed two separate 1x3 repeated measures analyses of variance (ANOVA) on the accuracy and time data, with interface type as the repeated measures factor. For accuracy, either the slice distance from the target center (for MS and KS), or the projected pixel distance in Y-axis (for consistency) of the Magic Lens center was used for analysis (for ML). However, comparing an essentially 2D visualization of slices with the Magic Lens (3D visualization) might produce some errors in this analysis, which will be discussed in more detail in Section 6. The results of Experiment II are presented in box plots in Figure 7 and Figure 8.

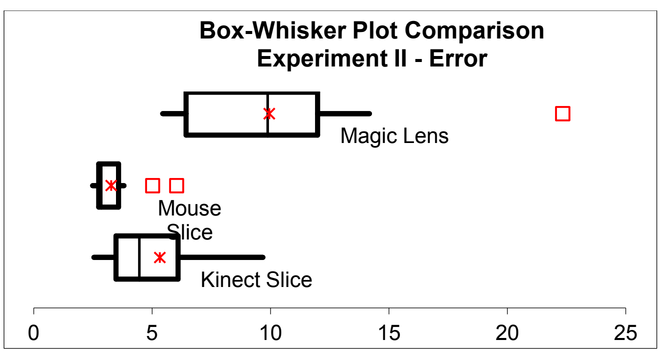

Figure 7. Box plot of results for Experiment II (error). Left to right: KS, MS, ML.

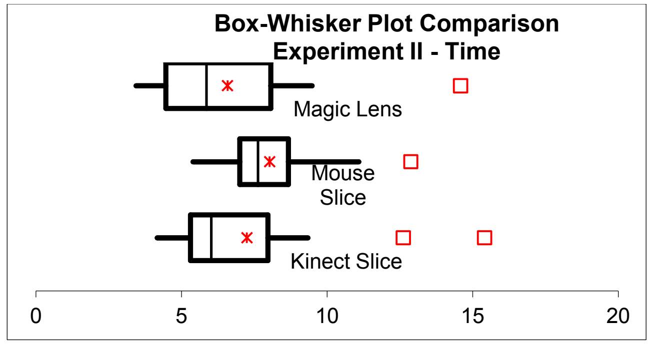

Figure 8. Box plot of results for Experiment II (time). Left to right: KS, MS, ML.

The mean performance for KS (7.2s) was on average faster than MS (8.0s). However, the between-subjects variance for MS was lower than KS. Furthermore, both the error in identifying the target slice and the response variance was lowest for MS. Thus, in general, performance in using MS to select the correct slice was highest along several criteria. ML is a slightly different interface for studying the internal structure of objects than MS and KS. ML had a better mean time (6.6s) than KS (7.2s) and MS (8.0s), but had a higher variance. Furthermore, the accuracy of ML (9.9 units) was lower than the accuracy of MS and KS. This is not a surprising result as we expected the mouse could perform precise movements better, and the high variance in error shows that most users did not perform as consistently when interacting with this interface as compared to using the mouse. The p-value of when all three interfaces were considered was 0.056. Compared to each other, only ML and MS demonstrated statistically significant difference in terms of time ($p=0.017$). In terms of error, KS and MS significantly outperformed ML ($p<0.01$) and the p-value between MS and KS was 0.068. These results lead us to believe that even though ML can be used as an interface for quick exploration of datasets, the mouse can perform precise targeting tasks better. We discuss possible reasons and further implications of these results in Section 6.

6. Discussion

Our experiments have yielded several interesting results. In Experiment I, the performance using the gesture-based interface exceeded our expectations. Users performed significantly better using Kinect as compared to the mouse in the rotation task. We believe the success of the interface comes from its similarity to an action that users can relate to (holding and rotating an object), as opposed to the mouse rotation, which is a more abstract mapping. Furthermore, these results were achieved after a short training time using an unfamiliar interface, which points to the intuitiveness of using gestures for rotation tasks.

In the second experiment, the mouse outperformed both gesture-based interfaces in terms of accuracy, which was an expected result given the suitability of the mouse in making precise movements. For the Magic Lens interface, some other factors might have contributed to the high error rate. In slice-based visualizations, an accurate match requires the slice to be in an exact position since only a cross-section of the data is displayed. However, even though Magic Lens is not perfectly centered at the target location, the target might be inside the lens volume and completely visible. Furthermore, perspective projection might make it difficult to center the lens exactly on the target. These factors, combined with the fact that gesture-based interfaces outperformed the mouse in terms of time makes us believe that gesture based interfaces might be suitable in exploration of volumetric datasets. Moreover, the Magic Lens interface was received favorably by users, and the fact that it can present the inner structures of the dataset in 3D means that it can contribute to understanding of medical datasets and shapes of internal structures. Furthermore, the users could locate targets more quickly with Magic Lens, therefore in situations where the user has to compare information between several spatial locations (e.g. if the experiment had more than one target with varying sizes larger than the distractors), the Magic Lens can prove to be effective for quick spatial exploration.

7. Conclusion

In this paper, we presented and evaluated effectiveness of gesture-based interfaces to be used in volume dataset rotation and exploration tasks. The gesture-based interface outperformed the mouse for rotation tasks. For locating targets, the mouse outperformed gesture-based interfaces in terms of accuracy but not in terms of time. We hope these results will provide insight for further exploration of these kinds of interfaces, as their use in the operating room might solve several problems currently encountered. Especially, methods to increase the precision of gesture-based interfaces and smooth ways to perform engagement/disengagement actions would be valuable. When designing these kinds of gestures, the limited space of the operating room, accurate differentiations between gestures to interact with the system and expressive or explanatory gestures, and maintaining the surgical workflow need to be considered. We are also aiming to extend the studies using medical datasets and medically trained users to test if our findings successfully transfer to medical applications.

Acknowledgement

The authors would like to thank all the volunteers and members of our research group for their time and invaluable comments that contributed greatly to this study.
We would also like Stefan Roettger, Stanford 3D Scanning Repository and the Osirix Foundation for making the volume datasets used in the experiments and the figures available, and the OpenNI Organization for the Kinect pose tracking framework.

References

AHN, Y.K., PARK, Y.C., CHOI, K.S., PARK, W.C., SEO, H.M., JUNG, K.M., 2009. 3D Spatial Touch System Based On Time-Of-Flight Camera. *WSEAS Transactions On Information Science And Applications 6*, Pp1433-1442

ARDITO, C., BUONO, P., COSTABILE, M. F., LANZILOTTI, R., SIMEONE, A. L., 2009. Comparing Low Cost Input Devices For Interacting With 3D Virtual Environments. *Proceedings Of The 2nd Conference On Human System Interactions 2009* (Pp. 289-294). IEEE Press, Piscataway, NJ, USA

BIER, E. A., STONE, M., PIER, K., BUXTON, W., DEROSE, T., 1993, Toolglass And Magic Lenses: The See-Through Interface. *Proceedings Of SIGGRAPH 93*, 73-80.

BREUER, P., ECKES, C., MÜLLER, S., 2007. Hand Gesture Recognition With A Novel IR Time-Of-Flight Range Camera-A Pilot Study. *In Proc. MIRAGE, 2007*, Pp.247-260.

DEV, P., 1999, Tutorial: Imaging And Visualization In Medical Education. *IEEE Computer Graphics And Applications, May/June*, Pp. 21-31

EVERITT, C., 2000. GlutManipulator Class. From: http://twars.googlecode.com/svn-history/r2/trunk/twars/common/nvglutmanipulators.hh

GALLO, L., PIETRO, G. D., MARRA, .I, 2008, 3D Interaction With Volumetric Medical Data: Experiencing The Wiimote. *In Ambisys 08 Proceedings Of The 1st International Conference On Ambient Media And Systems (ICST)*, Pp. 1-6.

HAUPTMANN, A.G., 1989, Speech And Gestures For Graphic Image Manipulation. *SIGCHI Bull. 20, SI* (March), 241-245

HINCKLEY, K., PAUSCH, R., GOBLE, J., KASSELL, N., 1994, Passive Real-World Interface Props For Neurosurgical Visualization. *CHI'94*, 452-458.

JOHNSON, R., O'HARA, K., SELLEN, A., COUSINS, C., CRIMINISI, A., 2011, Exploring The Potential For Touchless Interaction In Image-Guided Interventional Radiology. *Proceedings Of Chi 2011*, Vancouver, BC, Canada: ACM, P 3323-3332.

KIRMIZIBAYRAK, C., WAKID, M., HAHN, J. K., BIELAMOWICZ, S., 2010, Interactive Visualization For Image Guided Medialization Laryngoplasty. *Proc. Of Computer Graphics International*

KIRMIZIBAYRAK, C., WAKID, M., YIM, Y., HAHN, J. K., 2011, Interactive Focus+Context Medical Data Exploration And Editing. *Computer Animation And Virtual Worlds. (In Submission)*

OPENNI ORGANIZATION, Openni User Guide. Retrieved 05/01, 2011, From http://www.openni.org/documentation.

SILÉN, C., WIRELL, S., KVIST, J., NYLANDER, E., SMEDBY, O., 2008. Advanced 3D Visualization In Student-Centred Medical Education. *Med Teach. 2008* Jun;30(5):E115-24.

TANZ, J., 2011. Kinect Hackers Are Changing The Future Of Robotics. *Wired Magazine.*

THE VOLUME LIBRARY, FROM: http://www9.informatik.uni-erlangen.de/External/vollib (2010)

VIEGA, J., CONWAY, M. J., WILLIAMS, G., PAUSCH, R., 1996. 3D Magic Lenses. *Presented At The Proceedings Of The 9th Annual ACM Symposium On User Interface Software And Technology*, Seattle, Washington, United States

WACHS, J., STERN, H., EDAN, Y., GILLAM, M., FEIED, C., SMITH, M., HANDLER, J., 2007. Gestix: A Doctor-Computer Sterile Gesture Interface For Dynamic Environments. *In: Soft Computing In Industrial Applications. Volume 39, Advances In Soft Computing:* Springer Berlin / Heidelberg; Pp 30-39.

ZELEZNIK, R., FORSBERG, A., STRAUSS, P., 1997. Two Pointer Input For 3D Interaction. *In SI3D '97: Proceedings Of The 1997 Symposium On Interactive 3D Graphics*