Slicing Techniques for Handheld Augmented Reality

Paul Issartel* Florimond Guéniat† Mehdi Ammi‡

LIMSI-CNRS
Université Paris-Sud

Abstract

Manipulating slice planes is an important task for exploring volumetric datasets. Since this task is inherently 3D, it is difficult to accomplish with standard 2D input devices. Alternative interaction techniques have been proposed for direct and natural 3D manipulation of slice planes. However, they also require bulky and dedicated hardware, making them inconvenient for everyday work. To address this issue, we adapted two of these techniques for use in a portable and self-contained handheld AR environment. The first is based on a tangible slicing tool, and the other is based on a spatially aware display. In this paper, we describe our design choices and the technical challenges encountered in this implementation. We then present the results, both objective and subjective, from an evaluation of the two slicing techniques. Our study provides new insight into the usability of these techniques in a handheld AR setting.

Keywords: Augmented reality, 3D interaction, tangible user interface, scientific visualization

Index Terms: H.5.1 [Information Interfaces And Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 Introduction

Exploration of volumetric datasets is important in many scientific fields, such as physics, medical science or architecture. During exploration, scientists often use *slice planes* to visualize the internal structures of complex volumes. The orientation and positioning of a slice plane within a volume is inherently a 3D interaction task. Currently, this manipulation is generally done with standard desktop computers. However, interaction techniques are limited by the mouse, which does not provide sufficient degrees of freedom for 3D interaction. This leads to suboptimal and unnatural mappings. To overcome this issue, alternative approaches have been proposed. They support direct 3D manipulation of slice planes through real-world objects, called "tangible objects". The object is tracked in the 3D space, and its movements are directly mapped to the slice plane position and orientation.

These approaches can be divided into two groups. In the first group, the tracked object is used as an tangible slicing tool, and the result of slicing is shown on an external display. In the PassProps [3] interface, the slicing tool is a rectangular plate. De Guzman et al. [2] proposed a fork-shaped tool, which could be either attached to an adjustable arm or freely moved. Qi et al. [5] and Mulder et al. [4] used a pen, where the slice plane was orthogonal to its axis. In each case, however, the system assumes a fixed display (computer screen or workbench) and fixed tracking equipment (magnetic or optical). None of these interfaces are thus portable.

In the second group, the tracked object is a spatially aware display, thus the slice can be shown on the slicing tool itself. There is no need for any other input device. Examples of such interfaces include the mechanical device proposed by Bertelsen et al. [1], the PaperLens [7] and the WYSIWYF approach proposed by Song et al. [6]. This configuration seems particularly suited to make the system fully portable. Yet, the aforementioned works are all constrained in this regard. The display in [1] is attached to a mechanical arm for tracking purposes. The display in the PaperLens interface is a passive sheet of paper, and thus requires a top projector to visualize the resulting slice. The WYSIWYF interface uses a handheld device with embedded sensors, however this is only one part of the whole system, which still depends on a fixed wall display.

We believe that a portable and self-contained system would be more convenient for end users. A portable system does not force users to go to a dedicated place. It can be brought inside the workplace, or even at home. Unlike fixed installations, it is also conceivable to deploy a portable system to all target users. Indeed, handheld devices have become a popular platform for designing portable and self-contained interactive systems, due to their ubiquity. When combined with augmented reality, they become a good platform for 3D interaction.

In this paper, we describe the adaptation of the two tangible slicing approaches identified above ("tangible slicing tool" and "spatially aware display") for a handheld AR interface. We discuss the design choices and technical challenges encountered during the adaptation. Finally, we present an evaluation of the two techniques in this new environment.

2 Portable augmented reality platform

One of our main requirements was to use common, off-the-shelf hardware. The interface was implemented on an Android tablet: a Toshiba AT270. It has a 7.7" screen and weighs 332 g. This form factor was chosen to be easy to hold in one hand, while still having a relatively large screen. The tablet has a rear camera, which is required for handheld augmented reality.

Since the system was to be portable and self-contained[1], we could only rely on the tablet's own sensors. However, in order to support 3D interaction in augmented reality, we also had to make the volume dataset appear to be outside the screen (i.e. move independently). Therefore, we needed a frame of reference outside the tablet. In our interface, we added an additional tangible object for this purpose: the *reference object* (Figure 2). It provides an anchor point for the volume dataset. The dataset is shown at this location, and thus appears to be outside the tablet, in a 3D space. The reference object is tracked – relative to the tablet – using ARToolKit[2] markers. Unlike markerless or environment tracking, which depend on a sufficient amount of visual features, an object with fiducial markers can be tracked reliably in any environment (such as the user's home or workplace). An alternative could be non-optical tracking, such as magnetic or inertial sensors, but this would require adding new sensors to the tablet and/or the reference object. Another benefit of a tangible reference object is to allow direct ma-

*e-mail: paul.issartel@limsi.fr
†e-mail: florimond.gueniat@limsi.fr
‡e-mail: mehdi.ammi@limsi.fr

[1] The associated tangible objects are considered part of the system.

nipulation of the dataset: the user can translate and rotate the volume by manipulating the object.

The volume dataset is loaded and processed using the VTK library[3]. It is then rendered with OpenGL ES 2.0, and overlaid on the reference object. In this configuration, the visual and interaction spaces become co-located. As shown by Ware et al. [9], this can significantly facilitate 3D manipulation.

3 Slicing Techniques

We implemented the two main slicing approaches described in the introduction. For the "tangible slicing tool" method, we used a tangible stylus. For the "spatially aware display" method, we used the tablet itself as a slicing tool.

3.1 Stylus-based slicing

We designed a pen-shaped tangible object, called the *stylus*, to be used as a slicing tool. In this mode, a virtual plane is attached to the stylus tip (which is tracked with markers). This plane defines a slice through the data volume. The result is shown in the augmented reality view, on the tablet screen (Figure 1). When the stylus is moved, the slice is updated in real-time, giving the impression of effectively slicing the virtual data with a physical tool.

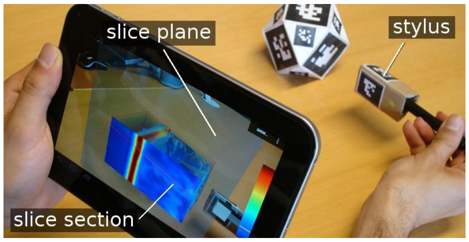

Figure 1: A virtual slice plane is attached to the stylus, which is manipulated with the fingers to slice through the dataset.

The elongated shape of the stylus "suggests" to use it like a pen: it has a handle, and the tip is an effector. This means the stylus benefits from the dexterity previously acquired from the use of pens. When the stylus handle is held with the fingers, a few controlled movements are enough to change the position and the orientation of the slice plane in the data volume. This type of grasp is known as a "precision grasp" [8]. As a precision tool, the stylus is expected to be more suited for accurate adjustment of the slice plane. It might however be less efficient for large movements within the dataset.

We chose to align the virtual plane with the stylus axis, instead of making it orthogonal as in previous works (e.g. [5]). This choice is justified by the constraints of handheld AR on a self-contained device. Since the camera used for tracking has to be located on the back of the tablet, users need to bring the stylus behind the tablet for it to be detected. The usable workspace is further limited by the user's arm length (because both the tablet and the stylus are held at the same time) and the camera's field of view. If the slice plane was orthogonal to the stylus axis, the users's wrist or arm would be blocked by the tablet. Therefore, we chose to align it with the stylus. Whether this change affects performance remains to be tested experimentally.

Since most handheld devices do not yet support 3D textures, slicing cannot be implemented directly in OpenGL. Instead, we used the VTK library to render the slice section into a texture, based on the geometrical transformation between the stylus tip and the reference object (i.e. the data volume). This texture is then applied to a quad – a rectangular polygon – aligned with the stylus, in the augmented reality view.

3.2 Tablet-based slicing

In the second slicing technique, the tablet itself is used for slicing. It has a planar shape, and its position and orientation relative to the data volume (the reference object) are known from tracking. Thus the tablet is actually a spatially aware tangible plane, directly controlled by the user in the 3D space. This tangible plane defines a slice through the data volume. Any slice plane can be obtained by positioning the tablet and the reference object relative to each other. The slice section is displayed on the tablet screen (Figure 2), and updated in real-time during manipulation.

However, the tablet and the reference object are both physical, tangible objects, so they cannot interpenetrate. A strict interpretation of this metaphor would make some slice planes inaccessible. We worked around this issue by shifting the virtual slice plane a few dozen centimeters behind the tablet. This offset is called the *slicing distance*. Unfortunately, this makes it more difficult to rotate the slice plane within the data volume, since the center of rotation remains located on the tablet. To overcome this limitation, users can rotate the data volume instead, by manipulating the reference object.

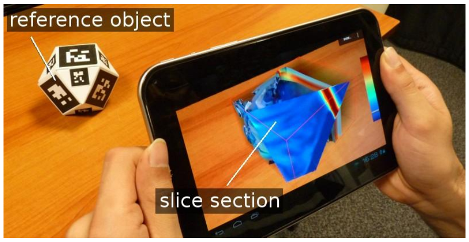

Figure 2: The slice plane is located at a fixed distance behind the tablet surface. The user moves the tablet to manipulate the slice.

Compared to the stylus, the tablet is larger and relatively heavier. It will be held firmly by the fingers, possibly with both hands. This type of grasp is known as a "power grasp". Manipulating an object held in such a grasp involves large muscle groups, such as the elbow and shoulder, since fingers cannot be used. These muscles allow larger movements, but the manipulation is less accurate. Because of that, and because of the rotation issue described above, this interaction mode is expected to be more suited for quickly skimming through the dataset (i.e. large translational movements) than for precise manipulation of the slice plane.

The slice section is computed according to the relative transformation between the tablet and the reference object, and rendered into a texture using VTK. Since the slice plane is always aligned with the tablet, this texture is then applied to a fullscreen quad. To avoid occlusion artifacts with other virtual objects, the fullscreen quad is artificially given a depth value the depth buffer (equal to the slicing distance).

4 Evaluation

In order to evaluate the two techniques (tablet-based and stylus-based), we compared them with each other in an exploration task. The goal was to highlight differences in performance and usability in this handheld AR configuration.

To assess their usefulness relative to more common techniques, we also compared them with a baseline. Currently, most scientists use either command-line or WIMP-style visualization tools to manipulate slice planes. Thus, a 2D mouse technique was deemed to be a good baseline. This technique was carefully reimplemented on a tablet, so as to be functionally equivalent to the slice plane widget found in ParaView[4], a popular visualization software on desktop computers. When the right mouse

[3] http://www.vtk.org/

[4] http://www.paraview.org/

button was held down, mouse movements resulted in a rotation ("arcball" rotation) of the slice plane within the 3D volume. When the left button was held down, mouse movements resulted in a translation of the slice plane along its normal vector. It should be noted that this technique cannot support simultaneous translations and rotations, because of the limited degrees of freedom provided by a 2D mouse (only 2 DOF). Since this is an inherent limitation of WIMP interaction, we believe the mouse technique described here is representative of current software.

The entire experiment was conducted on a single tablet. For the baseline mouse technique, we connected a wireless ambidextrous mouse to the tablet.

4.1 Participants

Nine unpaid participants (1 female, 8 male) from 23 to 29 years old (mean=25.0, SD=1.7) were recruited from the laboratory. One reported to be left-handed.

4.2 Experimental design

A CT-scan of a human head was loaded into the software and attached to the reference object (Figure 3a). Inside the head, 3 spherical targets (∅ 1 cm) were added. Their color was strongly contrasted with the rest of the data, to make them stand out among other cranial features (Figure 3b). Targets could not be seen from outside the volume, and were only visible on slice planes. Slice planes were controlled by the technique to be evaluated.

Participants were instructed to *find the three targets* within the data volume, and to make *a slice plane intersecting all three spherical targets*. This state had to be maintained for 0.8 seconds before the task was automatically validated. There was a 5 minute time limit on each trial.

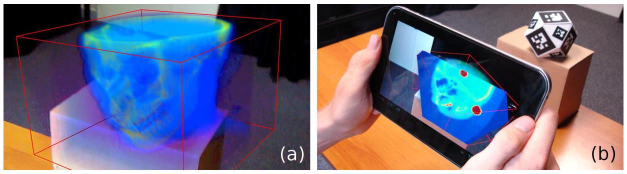

Figure 3: Experimental task. (a) The head dataset. (b) Using tablet-based slicing to locate the targets (two targets, the red circles, are visible in the slice section).

The three techniques were evaluated one after the other. To compensate for learning effects, the technique order was permuted between participants. We generated 18 predefined tasks (triplets of target positions). They were screened to exclude degenerate cases, such as overlapping targets, positions being too close to each other or to the volume bounds. All participants were presented the same 18 tasks, in random order. Each block of 6 consecutive tasks was accomplished with a different technique. Each technique block was preceded by 2 training tasks. Participants were allowed to take a short break between each block.

During the experiment, participants were seated in front of a table, on which the reference object was (initially) placed on a small cardboard box. This box allowed to raise the object slightly above the table surface, increasing the usable 3D workspace. Participants were permitted – and encouraged – to manipulate the reference object freely, since it is an integral part of the interface. Although being seated seems to forgo the advantages of portability, we argue that this setting is representative of a realistic use case. Even though the system itself is portable, in many cases the user does not have the physical space, or willingness, to move around the table. We still emphasize the importance of portability for the system itself, but in this experiment the user remained seated during the interaction. Therefore, manipulation of the reference object was often *required* to reach certain slice planes.

The main quantitative measure was the task completion time, but the entire slice plane trajectory was also logged. With 9 participants, 3 techniques and 6 tasks per technique, 162 trials were logged in this experiment. At the end of the experiment, participants completed a questionnaire. They were asked to grade each technique, on 6-point Likert scales, according to the perceived fatigue and their overall preference. They were also asked to summarize, in plain text, the positive and negative points of each technique.

4.3 Results and discussion

In this section, we present the preliminary results of this experiment. We intend to recruit more subjects to increase the reliability of these results.

Objective measures

The completion times were not normally distributed (Shapiro-Wilk W=0.79, p<0.01), but the data conformed to a lognormal distribution (Kolmogorov-Smirnov D=0.05, p=0.76). We thus applied a log transformation to the time data. Statistical analysis was performed on transformed data, but time values are reported untransformed. Variances were homogeneous in the transformed values (Levene F=0.55, df=2, p=0.57), meeting the prerequisites for parametric tests.

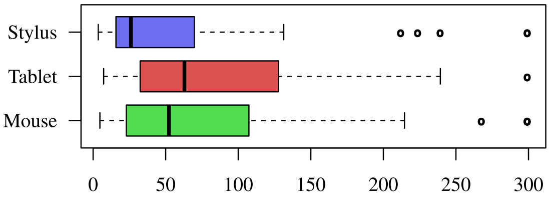

Figure 4: Task completion time (s)

The completion times are shown in Figure 4. The mean time was 56.5 seconds with the stylus, 84.6 seconds with the tablet, and 78.7 seconds with the mouse. A few trials had to be interrupted at the 5 minute time limit: 2 with the stylus, 1 with the tablet, and 3 with the mouse. An ANOVA revealed a significant difference among the mean completion times (F(2,16)=4.52, p<0.05). Post-hoc t-tests, with Holm correction, revealed that the mean completion time with the stylus was significantly shorter than with the tablet (p<0.01) and the mouse (p<0.05). There was however no significant difference between tablet and mouse (p>0.05).

From these preliminary results, the stylus appears to be significantly more efficient than the baseline technique for this task. This confirms our assumption that a direct 3D positioning technique can be more efficient than a mouse-based technique for slice plane manipulation. The stylus completion times were also strongly skewed toward the left (γ_1=2.16). This means that the stylus was much more efficient under some conditions than in others. In our interpretation, supported by observations, this difference was due to clutching. Whenever the target slice plane could be reached with small movements of the fingers, wrist or arm, the task was completed fairly quickly. When the target slice plane could not be reached because of anatomical constraints, participants had to interrupt the manipulation to change position (clutching), which caused longer completion times.

The tablet mode did not outperform the mouse for this task. This could indicate that tablet-based slicing was not suited for the experimental task: locating and reaching a given slice plane. To understand why, we analyzed the trajectories of the slice planes. Figure 5 shows the mean translation speed of the slice cross section. The speed was measured at the cross section centroid. Since the slice plane itself is manipulated very differently between the three techniques, we chose to analyze the trajectory of the *useful* part of the slice plane: the cross section. The mean speed was 1.2 cm/s with the stylus, 2.6 cm/s with the tablet, and 0.65 cm/s with the mouse. An ANOVA revealed a significant difference among the mean translation speeds (F(2,16)=38.77, p<0.01). Post-hoc t-tests, with Holm correction, confirmed that the three techniques were significantly different (p<0.01) in terms of translation speed. As shown in Figure 5, the cross section of the slice plane thus moved significantly faster in tablet mode.

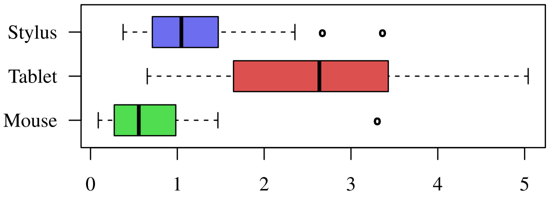

Figure 5: Mean translation speed of the slice cross section (cm/s)

We then analyzed the expansion of this trajectory. Expansion was computed as the mean distance between all points of the trajectory and their isobarycenter – that is, the standard deviation of the trajectory points. It was 7.5 cm (SD=3.1 cm) with the stylus, 12.5 cm (SD=2.7 cm) with the tablet, and 6.9 cm (SD=3.7 cm) with the mouse. Statistical analysis revealed that the trajectory expansion with the tablet was significantly larger than with the other techniques (p<0.01 for both comparisons). Therefore, the cross section covered a significantly larger space in tablet mode.

Even though participants were performing the same task with all techniques, the cross section of the slice plane moved faster within the dataset in tablet mode, and covered a larger space. This suggests that tablet-based slicing is more suited for exploration than the other techniques. However, participants did not complete the task faster with the tablet. Since the above results suggest this technique was faster for the initial exploration phase (i.e. locating the targets), we can assume it was much less efficient for the final alignment phase. This is consistent with our expectations. As explained before, the tablet is held with a firm grasp, or "power grasp", which involves large muscle groups. These muscles are suited for large and fast movements, rather than small and precise ones – for which the stylus is better. Furthermore, the slicing distance causes the slice plane to be shifted away from the tablet. While it does not affect translations, it causes the center of rotation to be located on the tablet itself, rather than on the slice plane. This restricts the range of possible orientations relative to the data, assuming users are seated. To reach further angles, users were forced to interrupt their manipulation to rotate the data volume instead, which decreased task performance. In contrast, the stylus mode was much less affected by this issue, since the distance between the stylus handle and tip is very small.

Subjective measures

All three techniques were almost equally rated in terms of fatigue. The mean ratings were: stylus=4.0, tablet=3.9, mouse=3.6 (on a 6-point Likert scale, 1=not tiring, 6=tiring). Differences were not statistically significant. It seems that the technique itself did not have a major effect on user fatigue. Indeed, some participants commented that the tablet had to be held in the hand all the time, and that this factor alone was more tiring than any of the three techniques.

The overall preference ratings were more contrasted: stylus=5.0, tablet=2.7, mouse=4.0 (1=dislike, 6=like). The lower rating for the tablet mode was likely due to its inadequacy for the experimental task, as suggested by the objective measures. The difference between stylus and mouse was not statistically significant, which means that participants had diverging opinions. However, the preference rating for the mouse may have been biased by the familiarity with the device, and the stylus may have been rated lower due to occasional tracking issues. We expect that a larger number of participants will make the difference significant.

Participants were also asked to list in plain text the positive and negative points of each technique. Interestingly, many participants used similar words or expressions to describe the same point, while a few others had very specific opinions. The baseline mouse technique was found to be "accurate" by three participants, "stable" by three others. On the other hand, five participants said it was "unintuitive for controlling the slice plane, especially rotations". Five participants found the stylus "intuitive", and three others said it was "convenient for rotating the slice plane". On the negative side, three of them said it was "difficult to keep in the field of view" (behind the tablet), and three others complained about "tracking issues". One participant found the stylus to be "too sensitive", which may be related either to tracking issues, or to the novelty of tangible interaction. Regarding the tablet, three participants appreciated the fact that it can be "held with two hands". However, three participants found it "difficult and/or inaccurate for positioning the slice plane". Another one complained that the tablet had to be held "too close to the eyes". We indeed noticed that it was an issue for some participants, presumably because of the default slicing distance. Therefore, we recommend that implementations of this technique let users customize this distance.

5 Conclusion and Future Work

This paper presents the adaptation of two slicing techniques for handheld AR: using a tangible slicing tool, and using a spatially aware display. Our implementation and user study provided new insight into the usability of these techniques in handheld AR. The results suggest that the tangible slicing tool is efficient for manipulating the slice plane, and the spatially aware display is better for exploration. On the other hand, the tangible slicing tool suffers from ergonomic issues, such as alignment with the handheld device and wrist constraints, while the spatially aware display is less convenient when the user remains seated. Future work should focus on improving the tangible slicing tool as well as the camera's field of view. The evaluation could also be repeated in a different configuration, where the user is moving and the data volume remains fixed. Finally, the two techniques could be combined to get the best out of each metaphor.

References

[1] A. Bertelsen, P. Irarrazaval, and R. F. Cadiz. Volume visualization using a spatially aware mobile display device. *Computerized Medical Imaging and Graphics*, 36(1):66–71, Jan. 2012.

[2] E. De Guzman, F. W.-l. Ho-Ching, T. Matthews, T. Rattenbury, M. Back, and S. Harrison. EEWWW!!: tangible interfaces for navigating into the human body. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, page 806–807. ACM, 2003.

[3] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 452–458. ACM, 1994.

[4] J. D. Mulder and R. Van Liere. The personal space station: Bringing interaction within reach. In *Proceedings of the virtual reality international conference, VRIC*, volume 2002, page 73–81, 2002.

[5] W. Qi and J.-B. Martens. Tangible user interfaces for 3D clipping plane interaction with volumetric data: a case study. In *Proceedings of the 7th international conference on Multimodal interfaces*, ICMI '05, page 252–258. ACM, 2005.

[6] P. Song, W. B. Goh, C.-W. Fu, Q. Meng, and P.-A. Heng. WYSIWYF: exploring and annotating volume data with a tangible handheld device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 1333–1342. ACM, 2011.

[7] M. Spindler and R. Dachselt. PaperLens: advanced magic lens interaction above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, page 7:1–7:1. ACM, 2009.

[8] S. Subramanian. *Tangible interfaces for volume navigation*. PhD thesis, Technische Universiteit Eindhoven, 2004.

[9] C. Ware and J. Rose. Rotating virtual objects with real handles. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 6(2):162–180, 1999.