Automatic Zooming Interface for Tangible Augmented Reality Applications

Gun A. Lee* Huidong Bai† Mark Billinghurst‡

The Human Interface Technology Laboratory New Zealand, University of Canterbury

Abstract

Tangible Augmented Reality (AR) interfaces use physical objects as a medium for interacting with virtual objects. In many cases, they track physical objects using computer vision techniques to attach corresponding virtual objects on them. However, when a user tries to have a closer look at the virtual content, the tracking can fail as the viewpoint gets too close to the physical object. To prevent this, we propose an automatic zooming method that helps users to achieve a closer view to the scene without losing tracking. By updating the zoom factor based on the distance between the viewpoint and the target object, a natural and intuitive zooming interaction is achieved. In a user study evaluating the technique, we found that the proposed method is not only effective but also easy and natural to use.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities;

Keywords: tangible augmented reality, interaction method, zooming interface

1 Introduction

In recent years augmented reality (AR) interfaces [Azuma 1997] have emerged as an important medium for its various applications in education, entertainment, and industry. By overlaying computer graphics image on top of live video stream and showing the virtual scene registered to the real world, AR provides natural views of 3D virtual objects such as CAD (Computer Aided Design) models.

While AR interfaces work mainly as visual displays, there is a need for investigating interaction methods for AR applications. One of the widely adopted AR interaction metaphors is Tangible AR [Kato et al. 2001]. Tangible AR is an approach that combines tangible user interface [Ishii and Ullmer 1997] input methods with AR display as output. In this way the virtual content in the AR interfaces can be manipulated using physical objects, making these interfaces extremely intuitive to use.

In many cases, Tangible AR systems use computer vision technique to track physical objects with fiducial markers [Kato et al. 2001] or natural visual features on the object [Clark 2010]. When users manipulate the physical object (whether the tracking target or the camera) to investigate the virtual object, moving the viewpoint too close to the physical object can cause a tracking failure due to there being not enough visual features available for the tracking algorithm.

*e-mail:gun.lee@hitlabnz.org
†e-mail:huidong.bai@pg.canterbury.ac.nz
‡e-mail:mark.billinghurst@hitlabnz.org

In this paper, we propose a novel interaction method based on a zooming metaphor that can help users overcome this limitation and have a much closer look at the AR scene. Compared to other approaches, such as using additional sensors [You and Neumann 2001], the proposed method is purely based on a software technique which could be widely applied without requiring additional hardware.

2 Related Work

Zooming is a well-studied interaction metaphor in 2D graphical user interfaces (GUI) [Perlin and Fox 1993; Bederson 1996; Igarashi and Hinckley 2000], and many software tools, such as image editors or maps on the web, provide interfaces for controlling the zoom level of the view. However, there are not many works in AR interfaces that investigated zooming metaphor.

Mulloni et al. [2010] used a zooming metaphor in an outdoor AR system to provide a smooth transition between different views, such as an egocentric AR view and an exocentric panorama. In their system, the user manually switched the zoom level to view the information in different contexts.

In comparison, our work mainly focuses on investigating a zooming interface for Tangible AR applications, and we also introduce a novel automatic control of the zoom level.

3 Automatic Zooming Interface

Zooming is useful for showing a closer view of a distant scene without physically moving the viewpoint (or camera) closer. In a Tangible AR application, it can be especially useful when the viewpoint cannot get closer to the AR scene in order to keep the tracking pattern within view. However, manually controlling the zoom level can be quite difficult in a Tangible AR environment where the users hands are mostly occupied by manipulating physical objects.

Igarashi and Hinckley [2000] demonstrated a zooming interface for browsing large documents, which automatically controls zooming based on scrolling speed. The view automatically zooms out when the user scrolls rapidly so that the perceptual scrolling speed in screen space remains constant.

In this paper, we apply a similar concept to controlling zoom automatically in a Tangible AR application. We propose automatically controlling the zoom level based on the distance between the camera and the AR tracking target, and applying a higher zoom factor as the camera gets closer to the target.

The zoom (or scaling) factor z can be calculated as,

$$z = \begin{cases} 1 & \text{if } d \geq t2 \\ 1 + (z_{max} - 1)(d - t_1)/(t_2 - t_1) & \text{if } t_1 < d < t_2. \\ z_{max} & \text{if } d \leq t_1 \end{cases}$$

where d is the distance between the camera and the tracking target, t_1 is the near distance threshold, t_2 is the far distance threshold ($t_1 < t_2$), and z_{max} is the maximum zoom factor ($z_{max} > 1$). While the equation above uses simple linear interpolation, for smoother transition at near and far thresholds, ease-in/out functions can be applied when interpolating the zoom factor. For instance,

in our prototype implementation, we applied sine function to the interpolation to achieve smooth transition of the zoom factor.

While zoomed view is useful for observing the AR scene, it becomes harder to decide whether the tracking marker is within the actual view. To overcome this problem, zooming is instantly turned off when the tracking is lost, in order to help the user manipulating the viewpoint to make sure the tracking target stays in view.

4 Implementation

We built a prototype system that implements the proposed automatic zooming technique. We used the osgART[1] library with AR-Toolkit[2] and OpenSceneGraph[3] libraries for AR tracking and visualization. The prototype software ran on a laptop computer (Dell Studio XPS 1645, Microsoft Windows 7) with a camera (Logitech QuickCam S7500) that provides video stream with 640×480 pixel resolution at 30 frames per second.

Figure 1 shows the prototype software visualizing an AR scene with and without automatic zooming while the camera is placed at the same physical distance from the tracking target.
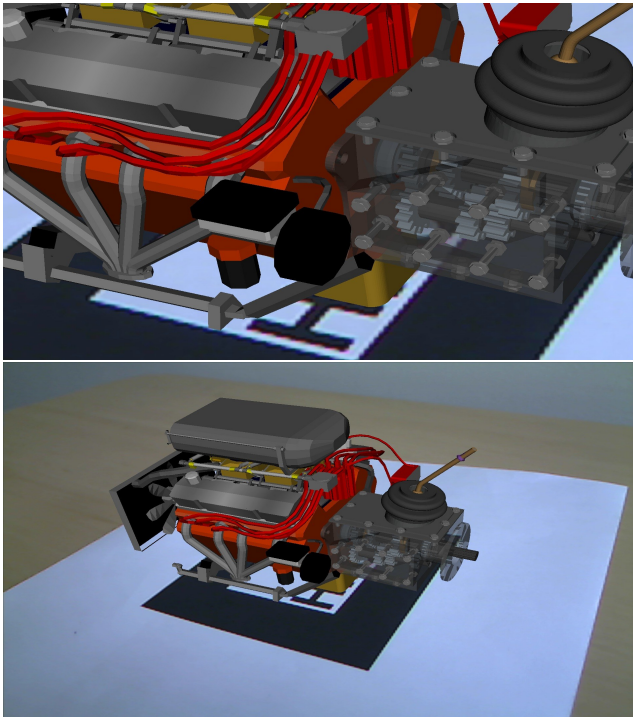

Figure 1: *Prototype AR system showing a virtual object with (upper) and without (lower) automatic zooming from the same viewing distance.*

5 User Study

In order to investigate the usability of automatic zooming method for Tangible AR, we conducted a pilot user study that compares the proposed method with a conventional Tangible AR approach which does not support zooming.

---

[1] http://www.osgart.org
[2] http://www.hitl.washington.edu/artoolkit
[3] http://www.openscenegraph.org

5.1 Experimental Design

The experiment was designed as a within subject experiment with one control variable: the type of interaction method (normal and automatic zooming).

In the experimental setup, participants sat in front of a desk with the prototype system described in the previous section (see Figure 2). They were asked to hold the camera and manipulate the marker to investigate the AR scene.

Figure 2: *The experimental setup.*

The experimental task was to observe a tangible AR scene with a virtual box, and to find, read and write down the letters on one of the labels on the box (see Figure 3). The letters on the label were randomly generated consisting of five alphanumeric characters, and the label was placed on a random side of the box. The participants had to do the task with five boxes for each condition. The order of conditions was counter balanced between participants.
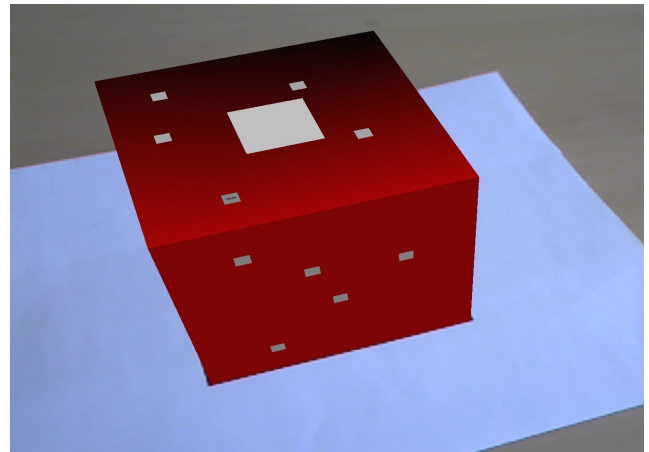

Figure 3: *AR scene of the experimental task.*

Participants answered a usability questionnaire after completing the task with each interaction method. The whole procedure took about 15 minutes. We also measured the user performance in terms of task completion time and error, and to investigate the system performance, we measured the number of times the tracking failed.

Table 1: *Usability questions*

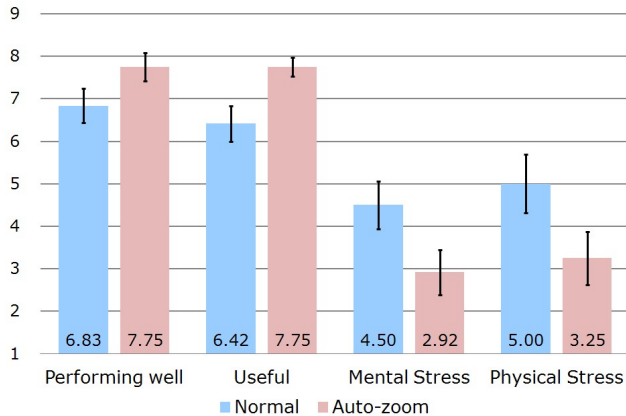| Usability questions |
| --- |
| Q1. I was performing well. |
| The given interface was: |
| Q2. Useful to complete the task |
| Q3. Mentally stressful |
| Q4. Physically stressful |
| Q5. Intuitive |
| Q6. Natural |
| Q7. Easy to use |
| Q8. Easy to learn |

5.2 Experimental Result

Twelve students were recruited aged 21 to 41 years old ($M = 27, S.D. = 5.7$). Two of them were female, and all of them had previous experience of using 3D and AR interfaces.

We analyzed the results from the questionnaire and measurements, using a within-subject one-factor ANOVA test ($\alpha = 0.05$).

We asked eight usability questions for each trial (see Table 1), and the participants answered on a 9-point Likert scale (1: strongly disagree $\sim$ 9: strongly agree).

In the results of the first four questions related to user performance, we found a significant difference between the two interfaces in all of the questions (Q1: $p = 0.042$, Q2: $p = 0.015$, Q3: $p = 0.032$, Q4: $p = 0.009$). Participants answered that they felt they were performing better and the interface was more useful when using the automatic zooming. They also answered that they felt less stress using the automatic zooming (see Figure 4).

**Figure 4:** *Results – Q1∼Q4 (Error bar: +/- S.E.).*

Participants rated the automatic zooming interface slightly better in the later four questions related to the usability (see Figure 5). However, the ANOVA test results found a significant difference between the two interfaces only in Q7 that asked if the interface was easy to use (Q5: $p = 0.054$, Q6: $p = 0.067$, Q7: $p = 0.046$, Q8: $p = 0.096$).

In terms of user performance, participants took significantly less time ($p = 0.023$) to complete the task when using the automatic zooming interface ($M = 124sec., S.E. = 9.98$), compared to the normal Tangible AR interface ($M = 161sec., S.E. = 13.70$). However, there was no significant difference in terms of error as measured by number of correct answers to reading the labels ($p = 0.151$).
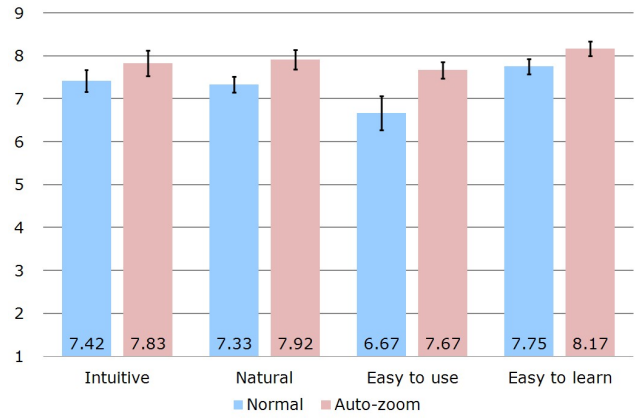
Figure 5: *Results – Q5∼Q8 (Error bar: +/- S.E.).*

The automatic zooming interface also significantly reduced the number of times the tracking failed during the experimental trials ($p < 0.001$). The tracking failure count with automatic zooming was to one third ($M = 8.7, S.E. = 1.62$) of that of the normal condition ($M = 25, S.E. = 3.16$). To the question asking which interface they prefer to use, eight participants chose automatic zooming, while other three answered both, and one chose the normal condition.

Figure 6 summarizes the results of user and system performance.
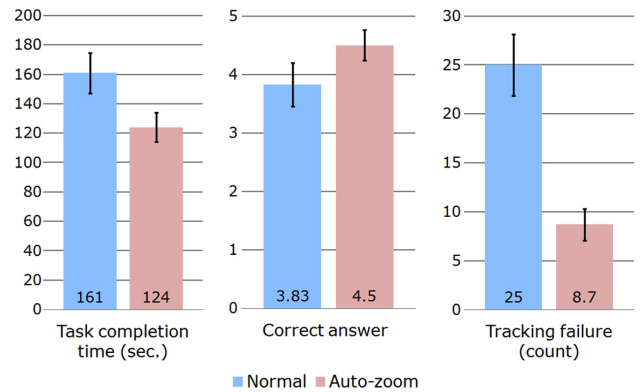
Figure 6: *Results – User and system performance (Error bar: +/- S.E.).*

6 Discussion

According to the experimental results, we can see that the automatic zooming helped users to perform faster with better usability, without compromising the naturalness or intuitiveness of the conventional Tangible AR interface. The proposed method especially reduced tracking failure significantly which contributed to better usability overall.

In debriefing, we asked participants what difference they recognized between the two interfaces. It is interesting that three of the participants did not even recognize the difference between the two interfaces. Only two participants mentioned of an unnatural mapping between the distance and image size in the automatic zooming condition.

When asked about problems, four participants mentioned that shak-

ing of the viewpoint got worse as the camera moved closer to the target.

7 Conclusion and Future Work

We proposed to use automatic zooming for enabling users to have a closer look at the Tangible AR scene, without suffering from frequent tracking failure. Through a user study, we showed the proposed method is not only effective but also easy and natural to use.

For future work, we plan to improve the stability of the view by applying filtering techniques. We also plan to further investigate the zooming technique by combining it with other AR interaction techniques.

References

AZUMA, R. 1997. A survey of augmented reality. *Presence-Teleoperators and Virtual Environments 6*, 4, 355–385.

BEDERSON, B. 1996. Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics. *Journal of Visual Languages & Computing 7*, 1 (Mar.), 3–32.

CLARK, A. 2010. *OPIRA: The Optical-Flow Perspective Invariant Registration Augmentation and Other Improvements for Natural Feature Registration*. PhD thesis, University of Canterbury, Christchurch, New Zealand.

IGARASHI, T., AND HINCKLEY, K. 2000. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '00, 139–148.

ISHII, H., AND ULLMER, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ACM, New York, NY, USA, CHI '97, 234–241.

KATO, H., BILLINGHURST, M., POUPYREV, I., TETSUTANI, N., AND TACHIBANA, K. 2001. Tangible augmented reality for human computer interaction. In *Proceedings of Nicograph 2001*.

MULLONI, A., DÜNSER, A., AND SCHMALSTIEG, D. 2010. Zooming interfaces for augmented reality browsers. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, ACM, New York, NY, USA, MobileHCI '10, 161–170.

PERLIN, K., AND FOX, D. 1993. Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '93, 57–64.

YOU, S., AND NEUMANN, U. 2001. Fusion of vision and gyro tracking for robust augmented reality applications. In *Proceedings of of IEEE VR2001*, IEEE, 71–78.