AiRType: An Air-tapping Keyboard for Augmented Reality Environments

Necip Fazıl Yıldıran*
University of Central Florida

Ülkü Meteriz-Yıldıran†
University of Central Florida

David Mohaisen‡
University of Central Florida

(a) **Default Setup.** Increasing the FoV. (b) **Flexible Setup.** Maximizing the FoV. (c) **Overlapping Setup.** More immersive.

Figure 1: Illustrations of possible positioning options of AiRType in the augmented reality environment. FoV: Field-of-View.

Abstract

We present AiRType for AR/VR HMDs that enables text entry through bare hands for more natural perception. The hand models in the virtual environment mirror hand movements of the user and user targets and selects the keys via hand models. AiRType fully leverages the additional dimension without restraining the interaction space by users' arm lengths. It can be attached to anywhere and can be scaled freely. We evaluated and compared AiRType with the baseline—the built-in keyboard of Magic Leap 1. AiRType shows 27% decrease in the error rate, 3.3% increase in character-per-second, and 9.4% increase in user satisfaction.

Index Terms: Human-centered computing—Text input; Human-centered computing—Empirical studies in interaction design Human-centered computing—Ubiquitous and mobile devices

1 Introduction

AR technologies came closer to the consumer market for ubiquitous usage. After extensive efforts by the research community, development and deployment speeds of sophisticated AR technologies, such as head-mounted displays (HMDs), have considerably increased, opening avenues for the integration of AR HMDs in our daily lives.

Text entry is one of the fundamental interaction mechanisms with AR environments. The new dimension made possible with the AR environments reinvigorated search for new interaction solutions. However, the mobility and efficiency requirements of AR bring along new challenges to the design space. Firstly, the virtual objects should not occlude the real-world. Therefore, designing a text entry method that balances the trade-off between usability and the real-world visibility creates a challenge. Secondly, new text entry methods should not impede the mobility with additional input devices, such as remote controllers. Lastly, to maintain convincing perception, virtual objects should not stand out among the real-world objects.

Recognizing those challenges, we introduce AiRType, an air-tapping keyboard for AR environments. AiRType enables users to type on the virtual keyboard using the hand models mirroring the hand's of the user. The freedom of changing the position and scale of the keyboard exhaust the possibilities of the new interaction dimension introduced by the augmented reality technologies without limiting the range of interaction with the user's arm length. Such freedom also enables users to adaptively optimize the virtual and real-world visibility trade-off solving *the first design challenge*. AiRType utilizes hand tracking sensors of AR HMDs to mirror the hand movements, thus, it does not require any additional device which solves *the second design challenge*. The keyboard design of AiRType follows the conventional QWERTY keyboard design. It supports seamless transition from real-world to AR, promotes convincing perception that overcomes *the third design challenge*.

To measure the usability of AiRType, we perform a small comparative usability study.We compare AiRType with the built-in keyboard of Magic Leap 1 in which targeting and selection of keys are done with a remote controller. Our user study revealed that AiRType reduces the error rate by 27%, increases the typing speed by 3.3%, and increases the user experience by 9.4%.

2 AiRType Design

AiRType is an in-air tapping keyboard for AR/VR HMDs that supports ten-finger typing experience. AiRType renders the keyboard plane and two hand models as virtual objects. The targeting and selection of the keys in AiRType are done through the hand models that mirror the user's hands via hand tracking feature of AR HMDs.

In designing AiRType, we utilized the conventional US keyboard as our default layout. Figure 1(a) illustrates the default ratio of the keyboard size with the field-of-view of the AR HMD. In the default setup, the size of the keyboard is 40 cm $\times$ 16.8 cm, and the size of the keys alphanumerical keys is 2.5 cm $\times$ 2.5 cm.

AR allows users to freely change the size of the virtual objects. As such, we designed our keyboard to be mobile. Even though the actual size of the keyboard does not change in this configuration, users can push the keyboard and hand models further to make them smaller. The user can increase the keyboard depth to make it look smaller and pin it to a position where it does not block any significant object in the real-world, e.g., the empty space on the table in Figure 1(b).

Another configuration might be to directly overlap the hand models with the hands of the user for more immersive experience. In this setting, the fingertips of the user directly touch the keys in the augmented environment. However, as illustrated in Figure 1(c), this requires us to design a large keyboard that occludes most of the real-world view. Irrespective of the decrease in the field-of-view, our design does not restrict the user's freedom of scaling, as users are free to scale the AR objects in the virtual environment controlled by T_s, and making the keyboard too small or too large has a detrimental effect on the usability. As shown in Figure 2, a smaller keyboard ($T_s = 0.5$) makes it difficult to pick the correct key, while a larger keyboard ($T_s = 2.0$) increases the distance the hands have to travel

*e-mail: yildiran@knights.ucf.edu
†e-mail: meteriz@knights.ucf.edu
‡e-mail: mohaisen@ucf.edu

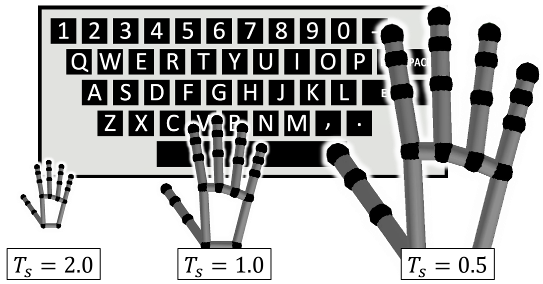

Figure 2: The effect of scaling factor, T_s, on the keyboard. For more concise illustration, the keyboard size is fixed and hand models are inversely scaled.

to touch the keys. Moreover, with a too large keyboard, the keyboard may not fit into the AR HMD frame. Figure 2 shows the scale of the hands with respect to the keyboard with borderline scaling factors.

To deploy AiRType for Magic Leap 1, we developed a Unity3D application where the users can easily move and scale the keyboard and the hand models with associated sliders in the Unity interface.

Keyboard. The keyboard consists of 42 cubes, each with an embedded collision interface defining the actions upon fingertips collision with the keys. Absent a tactile feedback, AiRType provides audio-visual feedback. When a user presses a key, the cube travels in, becomes green, play a "click" sound, put the associated character into the text box, and returns back to its original position.

Hand Models. Hand models are driven with the coordinates of the fingertips, palm center and finger joints, which is read from the hand tracking sensor. In the first version of the AiRType for Magic Leap 1, we utilized the hand tracking information coming from the built-in sensor. However, unlike Hololens 2, the current hand tracking of Magic Leap 1 does not provide a steady and sensitive data. Therefore, to facilitate a fair comparison, we used the data obtained from the Leap Motion Controller (LMC). We calibrate the coordinate systems of the devices using Iterative Closest Point (ICP) algorithm, thus the users are free to attach LMC to HMD or put LMC on a table. We note that it is expected to have an upgrade on the hand tracking feature of Magic Leap 1, which will enable us to remove the LMC from the setup altogether.

3 User Study

In this section, we review the test setup and the results of our comparative usability analysis. For the comparative evaluation, we use the system keyboard of Magic Leap 1 as the baseline keyboard. In the baseline keyboard, users move the cursor through the ray coming out of the remote to target letters, then pull the trigger button to select the keys. The background becomes blurry when the keyboard opens, which occludes the real-world environment. The experimental setup for AiRType is quite simple. The size of the keyboard is kept to the default size (Figure 1(a)) and LMC is placed on the table.

We measured the usability based on the standard ISO 9241-11 model of usability. To comply with ethical guidelines, the user study has been approved by our Institutional Review Board.

3.1 Usability Test Setup

We carried out the experiments with two groups, each of five novice participants with no experience with any AR HMD, and no training before the experiments. We formed the groups randomly and kept the groups exclusive to avoid any bias through sample leakage. The users from both groups typed the same target word sequences (an e-mail of length 250, 15 random strings of length 8, and 15 passwords). The ISO usability model defines three metrics: effectiveness, efficiency, and satisfaction. *Effectiveness* measures the proportion of erroneous key taps. *Efficiency* measures the task completion time by the user. *Satisfaction* depicts the System Usability Scale (SUS) score

Table 1: The average ISO usability test results.

	Effectiveness	**Efficiency**	**Satisfaction**
AiRType	5.833%	0.624 cps	81.5
Baseline	8.024%	0.604 cps	74.5

of the design. We measure the satisfaction by a standardized SUS questionnaire score [1]. The score range of 61-70 corresponds to an average score, whereas a score of >80 is considered satisfactory.

3.2 Results and Discussion

Table 1 shows the usability measurements of both keyboard designs. For the *effectiveness*, we observe that AiRType is more effective than the baseline, since it decreased the error rate by 27%. On the baseline keyboard, users tend to hit the trigger button by mistake while moving the remote to target the actual key which increased the error rate. The difference in the *efficiency* of the keyboards is negligible. Since the participants were inexperienced with the AR environment, they were moving their hands or the remote controller slowly to type correctly. Therefore, we anticipate the cps to improve significantly when the participants gain experience. Finally, we observe 9.4% increase in the SUS score with AiRType keyboard. The score of the baseline is slightly above the average satisfaction (> 70), while the score of the AiRType is satisfactory (> 80). Considering the metrics in combination, we conclude that AiRType offers a more usable option than the baseline keyboard.

4 Future Work

AiRType shows significant and promising initial results. To this end, we acknowledge various directions to address in our future work.

First, we will extend our user study to cover different AiRType configurations. Second, we will explore the effect of experience by conducting a training session for improving users familiarity before measuring the efficiency. Third, we are going to add appearance preferences, such as transparency and color theme, to AiRType and measure the user response under different appearances. Fourth, we are going to perform comparative usability study considering different interaction methods, such as dwelling, gesture, gaze-then-gesture, etc. Finally, we are going to deploy AiRType in VR environments and compare its usability to its counterparts.

5 Conclusion

In this study, we present a new text entry method for AR and VR environments—AiRType, which overcomes three concrete challenges introduced along with the new design space: (1) optimizing the visibility trade-off between real-world and AR environment, (2) maintaining the mobility feature of AR HMDs, and (3) supporting seamless switching between environments with familiar designs. AiRType focuses on increasing the user's freedom in the AR environment by mirroring the interaction medium, i.e., hands of the user, to the AR environment. Although counter-intuitive, our user study showed that mirroring a real-world object to AR environment is helpful to increase the usability. Moreover, AiRType takes the most out of the additional interaction dimension without restraining the interaction radius with the arm length of the users. We measured and compared the usability of AiRType with the system keyboard of Magic Leap 1 and showed that AiRType is a better option than the baseline in terms of efficiency, effectiveness, and satisfaction. Our work opens the door for several design and usability directions.

Acknowledgement. This work was supported in part by NRF grant 2016K1A1A2912757 (GRL) and CyberFlorida Seed Award (2020).

References

[1] J. Brooke. *"SUS-A quick and dirty usability scale." Usability evaluation in industry*. CRC Press, June 1996. ISBN: 9780748404605.