Context-based Information Visualization for Smart AR Glasses

Choonsung Shin, Youngmin Kim, Jisoo Hong, Sung-Hee Hong, Hoonjong Kang
VR/AR Research Center
Korea Electronics Technologies Institute
Seoul, Republic of Korea
email:{cshin, rainmaker, jshong, hshong, hoonjongkang} @keti.re.kr

***Abstract*—This paper presents context-based information visualization architecture for smart augmented reality (AR) glasses. In order to support context-based information visualization, the proposed system consists of four parts: object recognition and tracking, context management, interaction management, and AR visualization. With these components, the proposed architecture understands nearby objects and the user's situation, then composes and visualizes virtual information over a real environment. We implemented the proposed architecture on Hololens and tested it with an example AR scenario.**

Keywords-augmented reality; context-aware; smart glasses.

I. INTRODUCTION

With recent technical advances, augmented reality has received a great deal of attention not only from researchers but also from consumers. These advances have resulted from both the software and hardware used to realize augmented reality. Global companies, such as Google and Apple have released their own AR software development kits (SDKs) for rapid AR app development that support natural feature tracking and recognition. Additionally, Microsoft released Hololens, which supports spatial tracking and mapping while Meta released a similar AR glass that supports a wide field of view and hand interaction [1][2]. With these advanced in AR, users have the ability to engage in a more immersive AR experience. More interestingly, these wearable AR glasses are promising for industry and daily life since these devices are able to more practically and intuitively assist workers and users with free hand interaction.

Several studies have examined supporting context-based information visualization in a mobile and wearable AR environment. An early work, the touring machine supported 3D visualization based on localization and interaction with a heavy desktop system combined with external sensors [4]-[6]. Later research focused on more lightweight systems, including a system that used an AR head mounted display (HMD) connected to a smartphone and a wearable interaction device, although this combination was still limited to a lab study [7]. Google Glass later showed the possibility of mobile and wearable information visualization for consumers [3]. Recently, Hololens has shown more immersive and impressive 3D visualization by enabling spatial mapping and head tracking technologies [1]. While previous work has solved crucial problems related to practical AR systems for end-users, these systems have still been limited to using contextual information about users and environment.

In this paper, we introduce context-based information visualization architecture for smart AR glasses. The proposed architecture combines context-awareness with wearable AR. It thus understands user's situation and detects nearby objects; then, it proactively visualizes with virtual object and information related to the objects detected. We implemented the proposed system on Hololens and tested it with a representative use case scenario.

II. CONTEXT-BASED INFORMATION VISUALIZATION

Our aim is to investigate context-based guidance using wearable AR systems that provide contextual information by understanding nearby objects and environments. These types of applications require the recognition of physical objects and location while also providing related information with end-user services. The touring machine used an external global positioning system (GPS) and magnetic sensors to determine the position and orientation of the user [4]. However, recent AR devices, such as Google Glass and Hololens have been equipped with a number of sensors such as a GPS, orientation sensor, touch pad, and accelerometer. In our work, we focus on how to architecturally combine the information from the sensors with augmented reality.

In order to achieve this aim, we introduce context-based information visualization architecture to provide architectural support for intelligent visualization on smart AR glasses. The proposed system consists of an object recognition and tracking component, context management, interaction management, and AR visualization. Figure 1 shows the architecture of the proposed information visualization system.

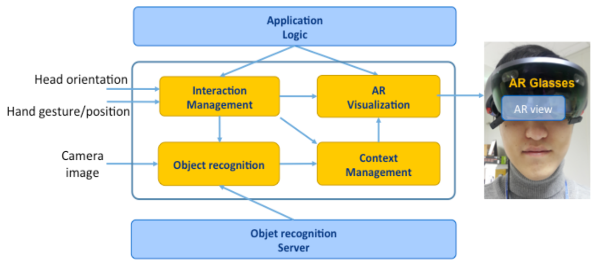

Figure 1. Context-based information visualization for smart AR glasses

The object recognition and tracking component detects, identifies, and tracks physical objects from the camera image stream. These objects include visual markers, 2D patterns, and 3D objects. This component thus includes a marker recognition/tracking library. For 2D recognition, feature

 ISBN: 978-1-61208-658-3

detection and matching are applied to a given camera image while optical flow is applied to support frame-by-frame tracking. To deal with 3D objects, it is connected to object recognition server that classifies the object based on a deep learning model. The context management component stores and offers a wide range of contextual information about a user and his/her device and environment from sensory information and the object recognition component. The interaction management component detects and deals with user input and events with respect to the AR scene being rendered. The AR visualization component has a role in presenting 3D information based on the AR scene and contextual information. This component thus aligns the geometric relationship between the display screen and virtual scene. It then visualizes the 3D information over a real space.

III. Implementation

We implemented the proposed context-based information visualization system on Hololens. Hololens is a stand-alone AR glass platform, which contains the components required for AR such as head tracking, spatial mapping, gaze tracking, hand gesture recognition, and inertial measurement unit (IMU) sensors. In order to support image processing, we included the OpenCVForUnity library, which is a C# wrapper library connected to the raw OpenCV library [8]. We also added a camera image-grabbing library since the Hololens application programing interface (API) itself does not provide a method to retrieve the camera stream [8]. We then implemented maker recognition and tracking and 2D image recognition and tracking based on the OpenCV library in a Unity 3D environment. Figure 2(left) shows an example of marker-based visualization.

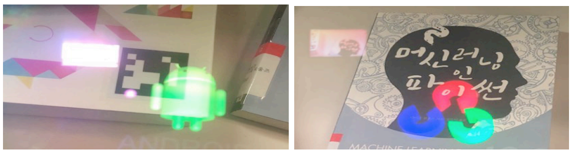

Figure 2. Example of maker and image recognition and tracking.

As seen in Figure 2(left), the proposed system detected and tracked a marker and the visualized a 3D object. In this example, the proposed system detected visual marker pattern from a camera image and then extracted the identification information. It then calculated 3d camera pose and applied it to Hololens camera matrix for visualizing the 3D object. We then also tested the 2D image-based recognition and tracking. As seen in Figure 2(right), the proposed system recognized and tracked a 2D image pattern. It first captured 2D images and stored them in the local folder in the Hololens. The proposed system then loaded the images and matched them to the camera stream from Hololens. The proposed system computed geometrical transformation related to the image it matched when the image from the camera stream was found in the images loaded. The system then estimated the 3D camera pose and applied it to the Hololens camera matrix in order to visualize the 3D contents over the object detected.

Last, we evaluated the performance of the proposed system. As marker-based tracking is simple and produces real-time performance, we focused more on 2D recognition and tracking. We thus evaluated the performance of the 2D image-based recognition and tracking. For this purpose, we measured the time to recognize and track 2D image patterns and compared the Hololens performance with an Android smartphone and a desktop PC. Hololens has a 1 GHz CPU while the Android smartphone has a 2.7 GHz octal core and the Desktop PC has a 3.3 GHz quad core.

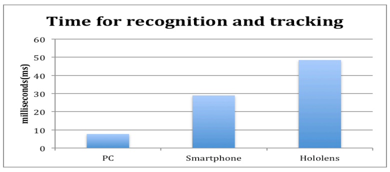

Figure 3. Time required for recognition and tracking.

As seen in Figure 3, Hololens took more time to recognize and track 2D patterns than the smartphone and the desktop PC. While the desktop PC's time was 16 ms (60 fps) and the smartphone's time was 25 ms (40 fps), Hololens took 46 ms. When we consider the camera frame rate (30 fps) of the Hololens, the overall time for tracking and rendering was 77 ms (15 fps), which needs to be improved for practical use.

IV. Conclusion

This work is the first step toward supporting information visualization for smart AR glasses. There are still technical problems that need to be improved. We first would like to add deep learning to improve 3D object tracking and recognition. We would also like to improve tracking and recognition performance on smart AR glasses.

Acknowledgement

This work was supported by the Ministry of Science and, ICT (MSIT) (Cross-Ministry Giga Korea Project).

References

[1] Hololens, https://www.microsoft.com (access date: 2018 Mar. 24)

[2] Meta Glass, http://www.metavision.com (accse date: 2018 Mar.24)

[3] Google Glass, https://www.x.company (accse date: 2018 Mar.24)

[4] S.Feiner, B.MacIntyre, T.Höllerer, and A.Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," PUC, Vol. 1, No 4, pp. 208–217, 1997.

[5] T. Starner, et al., "Augmented Reality Through Wearable Computing", In Presence, Vol. 6, No. 4, pp. 386-398, 1997

[6] R. Tenmoku, M. Kanbara, and N. Yokoya, "A Wearable Augmented Reality System for Navigation Using Positioning Infrastructures and a Pedometer," Proceedings of the ISMAR '03, pp. 344.

[7] J. K. Steven, et al., "Wearable mobile augmented reality: evaluating outdoor user experience," In Proceedings of VRCAI '11, pp. 209-216.

[8] https://github.com/EnoxSoftware/HoloLensWithOpenCVForUnityExample (access date: 2018 Mar. 24)