A Scalable Mixed Reality Platform for Remote Collaborative LEGO Design

Xinyi Yao Jiangong Chen Ting He Jing Yang Bin Li
School of EECS, Pennsylvania State University, University Park, Pennsylvania, USA
{xinyi, jiangong, tinghe, yangjing, binli}@psu.edu

***Abstract*—Mixed reality (MR) is a new paradigm that merges both real and virtual worlds to create new environments and visualizations. This together with the rapid growth of wireless virtual/augmented reality devices (such as smartphones and Microsoft HoloLens) spurs collaborative MR applications that provide an interactive and immersive experience for a group of people. In this demo, we develop a scalable MR-based platform for remote collaborative LEGO design. To provide the best immersive experience, the system should provide 1) high-speed and high-resolution image rendering: the rendering should achieve screen resolution of the mobile device and at least 60 frames-per-second; 2) extremely low delay guarantees: the motion-to-display latency of each user should be below 20ms; 3) synchronization: the synchronization latency should be small enough to enable the smooth collaboration; 4) scalability: the number of users should not have a significant impact on the system performance. To achieve all these goals, we introduce a central server to facilitate user synchronization via exchanging small messages. Each user reports to the server with its LEGO design progress, which is then distributed to all other users by the server; all other users render the corresponding virtual LEGO models in its own design space. We demonstrate via real-world implementations and evaluations that: 1) our system performance (e.g., synchronization delay, frame rate) does not degrade with the increase of the number of users; 2) our developed system not only yields a motion-to-display delay of 11 ms (i.e., 90 frames per second) but also achieves a screen resolution of each user's mobile device (e.g.,** 2400×1080 **pixels for Google Pixel 6).**

I. Introduction

Mixed reality (MR) creates new environments that are hybrids of both real and virtual worlds. Such technology together with the rapid deployment of cloud/edge computing enables the collaborative design among geo-distributed users. In particular, it provides an immersive environment for collaborative tasks that traditionally require workers to work side by side, which is particularly meaningful during the current pandemic. Such technology is expected to greatly improve productivity by stimulating people's imagination and bringing together expertise from all over the world.

In this demo, we demonstrate the collaborative mixed-reality-based platform development based on LEGO design, which extends the single-user mixed-reality-based LEGO construction in [1]. Each user with actual LEGO units can design her part with the help of mixed reality technology. For example, a virtual LEGO unit will appear on the user's display to be recommended for the next step, while the units assembled by other users will also be displayed as virtual objects. A user without actual LEGO units can still see the current design as virtual models. Despite being a toy example, this application demonstrates capabilities of great value to many real collaborative applications: it provides an interactive and immersive 3D environment that can be simultaneously accessed by geographically-distributed users, which is not currently available.

This research has been supported in part by NSF grants: CNS 2152658, CNS 2152610.

However, it is quite challenging to provide the best immersive and collaborative experience, since the system design should provide (see [2]) 1) high-speed and high-resolution image rendering: achieving screen resolution of the mobile device and having at least 60 frames-per-second; 2) extremely low delay guarantees: the motion-to-display latency of each user should be below 20ms for immersive user experience; 3) synchronization: the synchronization latency should be small enough to enable the smooth collaboration; 4) scalability: the number of users should not have a significant impact on the system performance. To achieve all these goals, our system performs all local rendering and coordinates users' design progress via the central server.

II. System Architecture

In this section, we develop a scalable MR-based platform for remote collaborative LEGO design. In such a system, each user renders virtual bricks to guide her own design in the physical space as well as to display all other users' virtual models to enable remote collaborations. As such, our system consists of a server and multiple users, which communicate with each other via Wi-Fi/Internet, as shown in Fig. 1. Next, we introduce the detailed implementations of the users and the server.

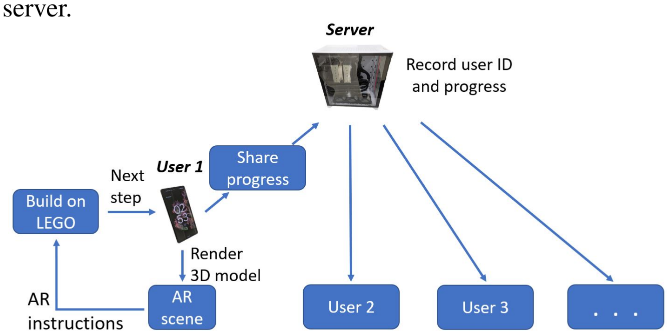


Fig. 1: System Architecture

Users: To support the scalable implementation, all virtual models are pre-loaded to the users. As such, each user's device can render virtual bricks to guide her LEGO design, where it localizes the LEGO sets based on AprilTag [3], [4]. After each step, each user reports its ID and design step to the server. Note that the pose (i.e., position and rotation) of the virtual model are traced locally, and thus each user does not necessarily know the pose of other users' virtual objects. In the meantime, each user receives the design progress of all other users from the server and displays corresponding virtual models in its own design space for smooth collaboration.

Server: The main purpose of the server is to enable the collaborative LEGO design among remote users. Recall that users report their progress to the server. Upon receiving each user's design progress, the server records its latest design profile and sends it to all other users via established TCP connections.

III. Demonstration

In our system, we deploy the server on a Lambda workstation with Intel Core i9-10980XE CPU @ 3.00GHz ×36, NVIDIA GeForce RTX 3070 Graphic Card ×4, 128 GB memory, and Ubuntu 20.04 LTS. Most users are deployed on commercial Android phones, including 10× Google Pixel 6, 2× Google Pixel 5, and 3× Google Pixel 4. Each Google Pixel displays the virtual model in the user's design space. Fig. 2 shows the user interface of one mobile device for the system consisting of 2 users, where user two (Bob) can see the next virtual brick to guide his design as well as the virtual model of user one (Alice). The demo video is available at [5]. To examine the scalability of our system, we also connect 10× Raspberry Pi 4B and emulate the rendering process, which makes the total number of users as 25. The server communicates with those users via a commercial router TP-Link Archer AX50.

To emulate the geo-distributed remote communications, we add extra communication delay between each user and the server using Linux TC [6]. In particular, we separate 25 devices into five groups. Each group contains 3× Google Pixel and 2× Raspberry Pi 4B. For each group, we add 10 ms, 20 ms, 30 ms, 40 ms, 50 ms extra communication delay for each device, respectively. We start the experiment with only one group of devices, i.e., five users, and collect around 1000 samples (including the synchronization delay and the rendering delay) for each user. Here, rendering delay refers to the time interval between two continuous frames displayed by each user. In contrast, the synchronization delay is the duration from one user reporting her progress until the other user updating the corresponding virtual model, including the communication delay and the rendering delay. Then, we add one group of users to our system and conduct a new trial. Such progress is repeated until all five groups of users, i.e., 25 users, are connected. We repeat ten times of our experiments to get the average results.

The evaluation results of our system are shown in Fig. 3. From Fig. 3a, we can observe that both the communication delay and the rendering delay are independent of the number of users. Specifically, the average motion-to-display latency

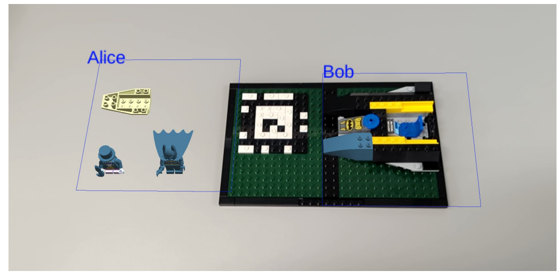
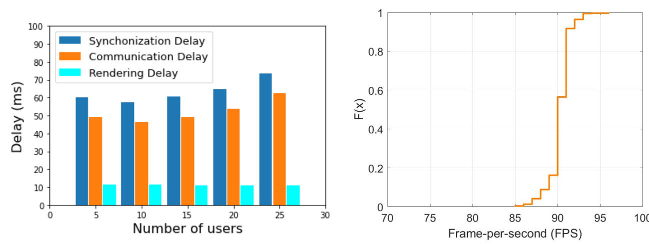


Fig. 2: Mixed-reality-based collaborative LEGO design for two users: the screen of mobile user 2.

(rendering delay) is about 11 ms for each setup. Meanwhile, the average communication delay is around 52 ms. The empirical cumulative distribution function of the frame rate for mobile users is shown in Fig. 3b, where each mobile user achieves 90 FPS on most occasions.



(a) delay comparison

(b) CDF of frame rate

Fig. 3: Performance evaluation

IV. Conclusion

In this paper, we developed a scalable mixed reality based platform for remote collaborative LEGO design. In such a system, each mobile device provides mixed-reality-based instructions to guide its user design by displaying the virtual brick on the user design space. Moreover, each mobile device displays all other users' virtual models coordinated by the server. To ensure the scalability of the our system, all virtual models are pre-loaded to the users. Our system evaluation demonstrates that the performance of our system design does not degrade as the number of users increases. Moreover, it achieves desired immersive and collaborative performance, i.e., having the motion-to-display latency of 11 ms and a screen resolution of each user's mobile device (e.g., 2400 × 1080 pixels for Google Pixel 6).

References

[1] W. Yan, "Augmented reality instructions for construction toys enabled by accurate model registration and realistic object/hand occlusions," *Virtual Reality*, pp. 1–14, 2021.

[2] "Keeping the virtual world stable in vr." [Online]. Available: https://www.qualcomm.com/news/onq/2016/06/29/keeping-virtual-world-stable-vr

[3] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407.

[4] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[5] "Demo Video." [Online]. Available: https://www.personal.psu.edu/jjc7191/demo/INFOCOM2022ARdemo.mp4

[6] "Linux TC." [Online]. Available: http://man7.org/linux/man-pages/man8/tc.8.html