ReplayAR: A Tool for Visual Evaluation of Mixed Reality

Zijian Huang*, Cary Shu*
{zijianh,caryshu}@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Hang Qiu
hangq@ucr.edu
University of California, Riverside
Riverside, California, USA

Jiasi Chen
jiasi@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Abstract

In world-locked mixed reality (MR), virtual content is locked in place with respect to the real world. Pose estimation is a key component to create world-locked MR experiences by estimating the device's position and orientation in order to render the virtual content accordingly. Current methods of evaluating world-locked MR include user studies, which are time consuming, and absolute trajectory error (ATE), which does not directly represent what is shown on the user's display. In this work, we propose ReplayAR, a tool that can replay user movement traces and output the corresponding visualizations (renderings) of the MR display. ReplayAR can be used to compare renderings from different MR pose estimation methods side by side, using our proposed Visual Difference metric. We implemented ReplayAR on a Hololens 2 MR headset and used it to evaluate open and closed-source pose estimation methods on standard datasets and our own collected traces. The results suggest that Visual Difference better reflects what is shown on the MR display compared to ATE. We hope that ReplayAR can encourage reproducible evaluation of world-locked MR, and towards this, we release the open-source code.

CCS Concepts

• **General and reference → Evaluation**.

Keywords

Augemented Reality, Evaluation Tools, Visualization

*Both authors contributed equally to this research.

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3696213

ACM Reference Format:
Zijian Huang*, Cary Shu*, Hang Qiu, and Jiasi Chen. 2024. ReplayAR: A Tool for Visual Evaluation of Mixed Reality. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3636534.3696213

1 Introduction

World-locked mixed reality (MR) is an important class of mixed reality experiences. For example, in an MR furniture placement app [11], users expect the virtual furniture to remain locked in place as the user moves around the real world, viewing the virtual furniture from different angles and locations in the room. To support world-locked MR experiences, an MR device typically senses the environment using its cameras and inertial measurement unit, runs a pose estimation method to determine its pose (*i.e.,* position and rotation), and renders the virtual content accordingly on its display. For example, if the pose estimation method determines that a user is far away from a virtual sofa placed in the corner of the room, the virtual sofa will appear smaller on the display. Common pose estimation algorithms include marker-based (*e.g.,* ArUco markers [6]) or marker-free (*e.g.,* ORB-SLAM3 [2]) approaches. In this work, we focus on marker-free pose estimation methods because they are common on commercial MR devices such as smartphones, Microsoft Hololens 2, and the Apple Vision Pro.

Evaluating world-locked MR is challenging and, and there are currently two main methods, as illustrated in Fig. 1. Evaluation methods from the HCI community include user questionnaires [9, 12]. While user studies are the gold standard, they are time-consuming to run and can be difficult to reproduce, hindering rapid prototyping and iterative improvements of world-locked MR. Evaluation metrics from the robotics community (where marker-free pose estimation is common), include Absolute Trajectory Error (ATE) [2]. ATE measures the distance between two camera pose trajectories (*e.g.,* the ground truth and estimated pose from a candidate SLAM method); however, it misses a key aspect of MR – what is rendered in the field of view – because it does not directly measure camera orientation, only position. In other words,

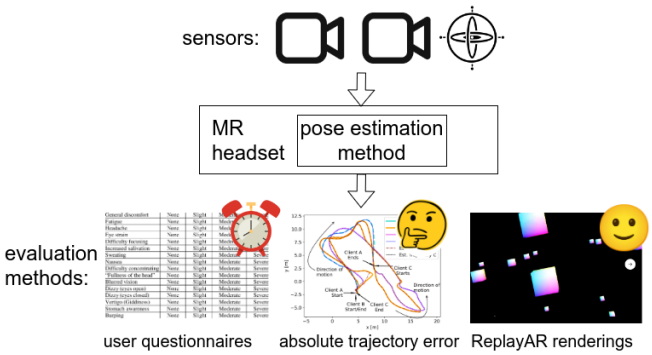


Figure 1: To evaluate MR, user questionnaires are slow and absolute trajectory error is hard to interpret and may not reflect user experience. ReplayAR's renderings show what the user actually sees, and can be used to derive metrics like our proposed Visual Difference.

ATE could be very good (close to 0), but the MR display could be very different from the ground truth if the device thinks it faces a different direction. Furthermore, it is hard to interpret ATE, which is commonly reported in meters, and understand how this maps to what the users see. Overall, neither of these methods (user questionnaires or ATE) allows visualization of what the MR user sees.

To overcome this, we propose a tool, ReplayAR, to visualize renderings of an MR experience based on previously collected user traces. ReplayAR enables researchers and developers to compare the outputs of different pose estimation methods or user movement patterns. The comparison can be done quantitatively using the Visual Difference metric that we propose, which scores the difference between two renderings from the same MR experience (*e.g.,* to compare two pose estimation methods from the same MR user trace). We implement ReplayAR on a Hololens 2 and use it to evaluate several pose estimation methods; namely, the open-source ORB-SLAM3 and the closed-source pose estimation method of Hololens 2. We evaluate these methods using our own traces collected by a Hololens 2 as well as on a standard dataset. To encourage reproducible evaluation of world-locked MR by others, we release our tool, data, and data collection methodology to the community[1].

In summary, the contributions of this paper are as follows.

- We build the ReplayAR tool, which takes an MR user trace as input and outputs the resulting rendered holograms on a Hololens 2.
- We propose an evaluation metric, Visual Difference, to quantify the differences between two MR renderings.
- We use ReplayAR to compare two popular pose estimation methods (ORB-SLAM3 and the default method on Hololens 2 with stereo cameras), and find there are significant differences.
- We collect our own user movement traces on a Hololens 2, in addition to evaluating on a standard public dataset.
- Our results suggest that both rotation error and translation error contribute to the final visual quality in terms of Visual Difference, which is not captured by ATE alone.

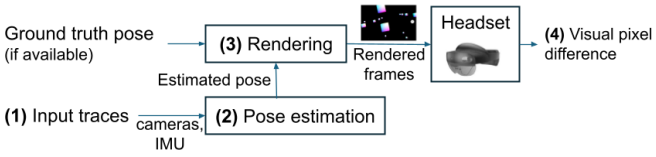


Figure 2: System overview. ReplayAR enables the MR headset (a Hololens 2 in our implementation) to replay input traces collected from an MR headset and visualize the output renderings.

2 ReplayAR Design

2.1 Data Processing Pipeline

The goal of ReplayAR is to visualize what would be displayed on the MR headset from a trace of the user's movements, and record the visualized rendering for further analysis via the Visual Difference metric. We break this down into the processing pipeline shown in Figure 2:

(1) **Input traces:** The headset records a data trace consisting of sensor readings from cameras and an IMU. Optionally, the headset's estimated pose can also be saved for later comparison.
(2) **Pose estimation:** A pose estimation method (*e.g.,* ORB-SLAM3) receives the input trace and estimates the poses of the headset over time.
(3) **Rendering:** The rendering module receives the estimated poses, renders the virtual objects in the MR scene for every frame, and sends the rendered frames to the headset for display. The example outputs are shown in Figure 3.
(4) **Visual Difference**: The system saves the visualized frames for later comparison with another trace using our proposed metric, Visual Difference.

In our implementation, we choose the Hololens 2 as because it is widely commercially available, has a multitude of sensors for accurate pose estimation, and reasonably open research APIs enabling collection of raw sensor data. We specifically utilize the Holographic Remoting functionality [18], which consists of a Player app running on the headset and a Remote app running on a desktop computer or a server. This provides an ideal interface to visualize the results because the majority of the replay functionality, which is computationally heavier (pose estimation and rendering in Steps 2 and 3) can be done on a computer, and the final results

[1]https://github.com/mavens-lab/replayAR

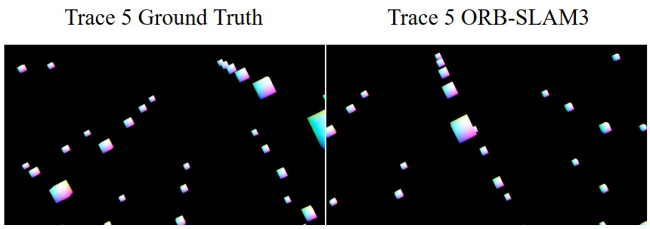


Figure 3: Example outputs from ReplayAR (corresponding to trace 5, use case 2 in §3). The differences between the ground truth (left) and ORB-SLAM3 pose estimation method (right) are shown. The colorful cubes are the virtual objects, and a black background is rendered instead of the real world environment for clarity.

streamed to the headset for display. The example results from the headset are shown in Figure 3. Note that we placed many virtual objects in the environment so that a subset of them will always be visible regardless of the pose of the user.

Challenge: Replaying camera movements. In order to recreate the MR renderings in step (3) above, the movements of the virtual camera in the scene must be replayed following the user movement trace, so that the appropriate holograms are rendered with the right orientation and position. However, a key challenge we faced is that we could not modify the device (camera) pose of the Hololens during real-time operation. This is because the camera pose values are written directly by the MR operating system, based on sensor inputs, and thus cannot be modified by developer code. We found this to be true for the Hololens 2 and other headsets such as the Quest Pro. Therefore, to recreate the rendering, we came up with a "mirroring" solution: we simulated the user's movement by moving the *virtual content* instead, resulting in the same visualization. For example, if the user trace showed the camera moved to the left by 2 meters, we instead moved the virtual content to right by 2 meters.

The detailed mathematical operations behind this idea are as follows. Let R be a 3×3 rotation matrix, t be a 3×1 translation vector, and K is the total number of frames in a trace. We are given trajectory α with poses $\tau^\alpha = (R_0^\alpha, t_0^\alpha), \ldots, (R_K^\alpha, t_K^\alpha)$, along with a comparison trajectory τ^β. The initial pose of the headset is $(R_0^{\text{HMD}}, t_0^{\text{HMD}})$ and the pose of a virtual object $(R_0^{\text{obj}}, t_0^{\text{obj}})$. To move a virtual object so that the rendering result is the same as if the user had moved according to the trajectory τ^α, for frame k, we compute the virtual object's pose as:

$$t_k^\alpha = (R_k^\alpha)^{-1} R_0^\alpha (t_0^{\text{HMD}} + R_0^{\text{HMD}} t_0^{\text{obj}} - t_k^\alpha + t_0^\alpha) \quad (1)$$

$$R_k^\alpha = (R_k^\alpha)^{-1} R_0^\alpha R_0^{\text{HMD}} R_0^{\text{obj}} \quad (2)$$

Essentially, these equations find the inverse transformation of the user's movements in order to apply them to a virtual object. The equations account for the effect of different headset pose initializations each time after the device restarts, the virtual object's initial pose in the system and in the trajectory, and counteracts the translation and rotation of the user's movements through subtraction and the matrix inverse operations. These calculations are implemented in the rendering module in step (3) of Figure 2. More details are provided in Appendix A.

2.2 Quantifying Visual Differences

Next, we discuss how to quantify the differences between two renderings, in Step (4) of Figure 2. We propose two variants of our Visual Difference metric below.

Visual Difference (pixels). We define Visual Difference (pixels) as:

$$\text{Visual Difference (pixels)} = \frac{1}{KNM} \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbb{1}_{P_{ijk}^\alpha \neq P_{ijk}^\beta} \quad (3)$$

where P_{ijk}^α is value of pixel (i, j) in frame k from trace α, after the images are converted to grayscale. This equation counts the number of pixels that are not the same in two images, summed across all frames, and normalized by the total number of frames and pixels. A lower value means that the two sequences of MR renderings are similar to each other.

Visual Difference (IoU). One issue with Visual Difference (pixels) is that it is sensitive to the user's distance to the virtual objects. For example, a larger distance and hence a small virtual object will cause a smaller values of the Visual Difference (pixels), because it is normalized to the total number of pixels in the frame. Therefore, we propose an alternative measure of Visual Difference based on the IoU of the virtual objects. Specifically, we define:

$$\text{Visual Difference (IoU)} = \frac{1}{K} \sum_{k=1}^{K} \frac{A_k^\alpha \cap A_k^\beta}{A_k^\alpha \cup A_k^\beta} \quad (4)$$

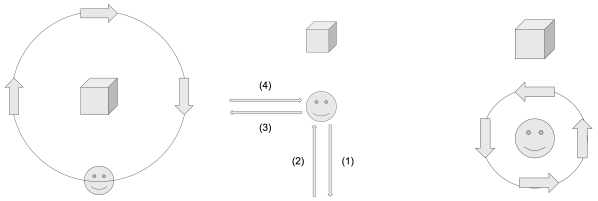
where A_k^α is the area of virtual objects in frame k of trajectory α. A higher value means the two sequences of MR renderings are similar to each other.

3 Experiments and Results

We consider three evaluation scenarios for ReplayAR, summarized in Table 1. In Scenario 1, the goal is to show the basic performance of ReplayARand to show that it can accurately replay MR traces. In scenario 2, the goal is to visualize the MR renderings on a standard dataset, comparing the renderings from an imperfect pose estimation method (ORB-SLAM3) to the ground truth. In sceanrio 3, the goal is to compare two pose estimation methods, that of ORB-SLAM3 and the default pose estimation method running on a Hololens 2 (which

Scenario	Input data	Pose estimation method	Visualization	Goal
1	Custom Hololens	Hololens 2	***Live vs. ReplayAR***	Accurately replay AR traces at a later time.
2	EuRoC	***ORB-SLAM3 vs. ground truth***	ReplayAR	Visualize MR results from a SLAM pose estimation method.
3	Custom Hololens	***ORB-SLAM3 vs. Hololens 2***	ReplayAR	Compare MR visualizations from different pose estimation methods.

Table 1: Summary of evaluation scenarios and their goals.



(a) Trace 1: User moves in a circle **(b) Trace 2: User translates** **(c) Trace 3: User spins in place**

Figure 4: User movement patterns in our own Hololens dataset. The initial distance to the virtual cube is approximately 2 meters.

is closed source). We evaluated the scenarios qualitatively by watching their replay videos, as well as quantitatively by computing their Visual Difference.

3.1 Experimental Setup

Datasets. To show the effectiveness of our method, we evaluate our method on the following two datasets.

- **EuRoC** [1]: The EuRoC dataset is one of the most popular benchmark datasets for SLAM-based pose estimation. It is collected in 2 different scenarios (Machine Hall and Vicon Room) with varying difficulty levels by a drone equipped with stereo cameras and an IMU sensor. A Leica MS50 laser tracker or a Vicon 6D motion capture system provide the ground truth pose.
- **Custom Hololens dataset:** Although there are many excellent datasets to evaluate SLAM methods (*e.g.,* [1, 7, 14, 21]), they are insufficient for MR evaluation because they do not reflect the real environment captured by the headsets. Therefore we collected our own dataset using the HoloLens 2 to record sensor data [5] and also log the poses estimated by the headset at the same time. The dataset consists of 5 trajectories of controlled movements, with Traces 1-3 illustrated in Figure 4. Traces 4 and 5 consist of more complicated movements recorded in an office environment, moving between tables and a corridor with a longer walking distance.

Evaluation Metrics. To evaluate the performance of open-source pose estimation algorithms like ORB-SLAM3 against commercial SLAM algorithms such as the one deployed in HoloLens 2 and show that ATE is not the ideal metric in general MR scenarios, we compute the ATE and Visual Difference between them. Additionally, we also compute the rotation error as a helper metric to further understand the results.

	Trace 1	Trace 2	Trace 3
Visual Difference (pixels)	0.030	0.036	0.069
Visual Difference (IoU)	0.27	0.68	0.89

Table 2: Scenario 1: Comparison between live recording and ReplayAR on traces from our custom Hololens dataset. The Visual Difference generally suggests a good degree of agreement between the live and offline replay.

- **Absolute Trajectory Error (ATE):** ATE is a popular way to evaluate a pose estimation method. It first requires two aligned pose trajectories τ^α and τ^β, for example aligned using the Horn method [10]. These two trajectories could be the ground truth trajectory and the trajectory from another pose estimation method. The ATE is the summation of the l_2 distances between each pair of positions in the two trajectories.
- **Rotation Error**: Compared to ATE, rotation error receives less attention, but it is actually very important given that rendering issue and MR sickness is highly affected by rotation error [15]. For rotation matrix R_1 and rotation matrix R_2, we compute the rotation error as the angle between these two rotation matrices, which is $\arccos(\text{trace}(R_2 R_1^{-1}))$.
- **Visual Difference (pixels):** Our proposed metric, defined earlier in (3).
- **Visual Difference (IoU)**: Our supplementary metric, defined earlier in (4).

3.2 Scenario 1: Accurately replay MR traces

The goal in Scenario 1 is to evaluate whether the offline replay from ReplayAR renders the same images as in the live experiment. To test this, we recorded a live trace on the Hololens 2, then replayed the same trace using ReplayAR. Then, using a video recorded from a remote view of the Hololens display during both the live run and replay,

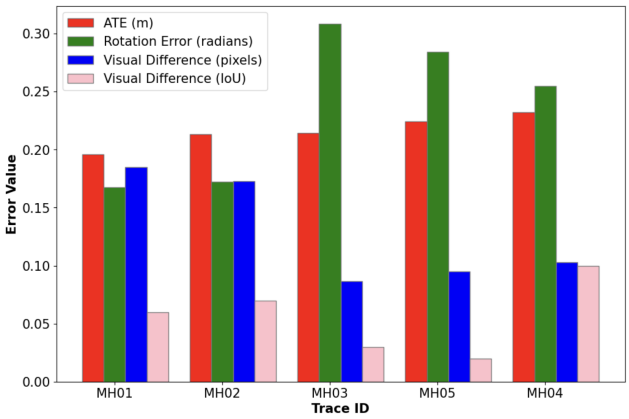


Figure 5: Scenario 2: Compared to the ATE and Rotation Error, Visual Difference (IoU) can more accurately capture the rendering error caused by SLAM pose estimation, as it accounts for rotation error. Traces sorted by increasing ATE.

the Visual Difference between the live run and replay was calculated.

The results of these trials are shown in Table 2. The Visual Difference (pixels) is low, close to 0, indicating good agreement between the live trial and the replay. Similarly, the Visual Difference (IoU) is fairly high. When we watched the videos, Trace 3 had the highest agreement between live operation and offline replay, suggesting that the IoU variant of Visual Difference most accurately captures user perception. However, a formal user study is needed in the future to fully evaluate this.

During watching the videos, we noticed that the replays are not always perfectly identical to the live run. This is borne out in Table 2, as the IoU is not exactly 1 or the pixel difference exactly 0. We hypothesize that this is due to the Hololens OS tweaking the hologram poses slightly during the live run. Specifically, lower Visual Difference (pixels) is observed when there is greater change in depth from the initial position during the trace; the change in depth may change the focus point of the Hololens and cause misplaced virtual content, as an artifact of the Microsoft Mixed Reality Capture [19]. However, these tweaks by the OS are out of our control, and we believe them to be negligible upon examining the videos. ***Overall, the results suggest that ReplayAR replays traces accurately, with slight discrepancies due to run-time tweaks by the Hololens OS.***

3.3 Scenario 2: Visualize MR results from a SLAM pose estimation method

In Scenario 2, we seek to evaluate our hypothesis that ATE is not the ideal metric in MR environments. To do so, we calculate the ATE, rotation errors, and Visual Difference between ReplayAR's renderings from the ground truth trajectory vs the ORB-SLAM3 estimated trajectory. The results are shown in Figure 5. From Figure 5, we can see that even though the difficulty level of the EuRoC MH01 to MH05 traces is generally from the easiest to the hardest, and is sorted that way on the horizontal axis of the plot, the Rotation Error and Visual Difference do not correlate well with ATE. Instead, Visual Difference is actually worse in the MH03 and MH05 traces, although ATE is increased compared to MH01 and MH02. We believe this is due to the high rotation error in MH03 and MH05. In other words, the translation error (ATE) and the rotation error *both* contribute to the differences between the final renderings (represented by Visual Difference), not just ATE alone.

Considering the relative ATE and Rotation Error between the traces in case 2, we can see that the rotation error correlates inversely with IoU when the ATE is close in general (e.g. MH02 vs MH03), while the ATE also correlates inversely with IoU when the rotation error is close (e.g. MH03 vs MH05). We hypothesis that Visual Difference (IoU) is better than Visual Difference (pixels) due to the bias caused by the depth issue in Visual Difference (pixels).

The MH04 trace is an exception, as it has the highest ATE and moderate Visual Difference values, bucking the trend of negative correlation between ATE and Visual Difference. Upon examining the replay videos, we observed that the device was mostly stationary at the beginning of the trace, contributing to the high IoU value and skewing the results of this specific trace. ***Generally, the results suggest that high rotation errors contribute to worse Visual Difference, which ATE alone cannot capture.***

3.4 Scenario 3: Compare AR visualizations from different SLAM methods

In Scenario 3, the goal is to test whether different pose estimation methods produce substantially different results. These experiments were carried out with our custom Hololens dataset as input, fed into open source SLAM (ORB-SLAM3) vs. commercial SLAM (on a HoloLens 2), and the resulting renderings from ReplayAR compared. An example trace is shown in Figure 6. The Visual Difference (pixels) changes over time, indicating that the two pose estimation methods produce different visualizations despite receiving the same raw sensor data. In particular, the lower values correspond to when the user is farther from the virtual object and the object is smaller in the field of view. The intermediate values during the middle of the trace are because there are few features in the environment at that time, causing noisier pose estimation and hence less accurate final renderings.

The evaluation metrics for the five traces are plotted in Figure 7. From Figure 7, we can see that the Visual Difference (both pixel and IoU) is significant even though the

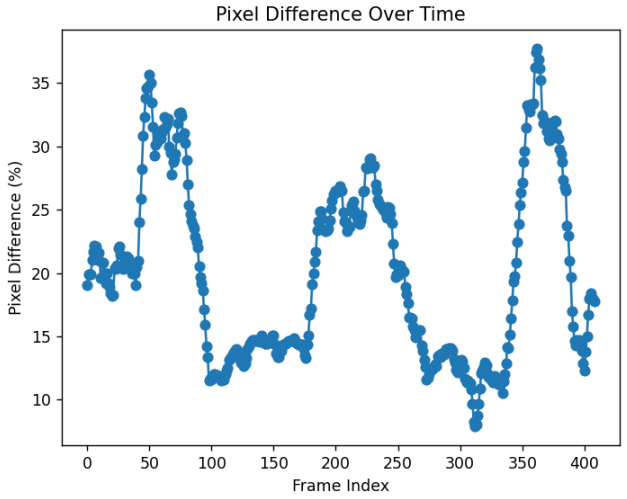


Figure 6: Time series of Visual Difference (pixels) for trace 2 in case 3. The values change over time, indicating that the two pose estimation methods produce different visualizations despite receiving the same sensor inputs.

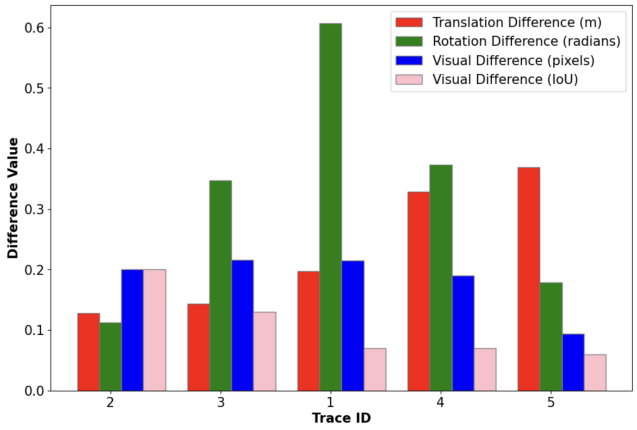


Figure 7: Scenario 3: There is significant Visual Difference between traces, and ATE can only be a good measure when there is less rotation error, such as Trace 2. Traces sorted by increasing ATE.

translation difference (*i.e.,* ATE) between the pose trajectories estimated by the two SLAM pose estimation methods is small, for example in traces 2 and 3. The significant difference in Trace 1 is because this trace is mainly moving around a center which is mostly caused by rotation difference; in the contrary, the trace 4 and 5 is relatively a larger map and mostly movements are translation, which means that most of the visual difference is caused by translation difference. ***The results suggest that there is still a significant difference between commercial and open-source pose estimation methods. Overall, these results demonstrate that the visual effect in MR systems is complicated and it is not sufficient to compare the translation difference / error only.***

4 Related Work

MR Evaluation. A large proportion of the MR literature focuses on tracking [17], which is based on SLAM pose estimation methods that are evaluated with the ATE metric [4, 13]. However, with the rapid development of MR technology, some MR evaluation tools have been built by researchers. [20] proposes to use ArUco markers as a reference to compare with the rendered virtual objects, in order to evaluate the spatial inconsistency and drift estimation performance of an MR system. [22] extends this to evaluate the spatial drift of a virtual object over time by tracking the position of virtual objects. Other methods to evaluate MR experiences include user studies and questionnaires [9, 12] as discussed previously. Other recent work [23] comparing open-source SLAM (ORB-SLAM3) with commercial headsets (Meta Quest 3, rather than Hololens 2 in this work) and reached a similar conclusion that commercial SLAM generally has better performance.

MR Evaluation Datasets. HoloSet [3] includes traces collected by a HoloLens in a variety of environments (indoor, outdoor) and scene setups (trails, suburbs, downtown) under multiple user action scenarios (walk, jog). However, the ground truth pose is missing, making it hard to use for evaluation. Standard visual-inertial odometry datasets such as EuRoC [1] or KITTI [8] are recorded by drones and cars. The SenseTime dataset [16] is recorded by handheld mobile phones, which is closer to MR but still differs from head-mounted devices.

5 Conclusions and Future Work

We propose a new tool called ReplayAR to evaluate pose estimation methods in MR in terms of their visual renderings. We show that ReplayAR can accurately replay traces and evaluate the visual output of an MR device better than traditional metrics such as ATE. Our analysis reveals that rotation error has a significant impact on the visual result, and Visual Difference (IoU) can reveal user experience quality better according to our results. In the future, we plan to render real world backgrounds in ReplayAR's output, and also transfer the current offline operation into an online tool. Furthermore, we plan to conduct a user study to evaluate the correlation between ATE, rotation error, IoU, Visual Difference, and user MOS scores, in order to develop a better understanding of what metrics best reflect the MR user experience.

Acknowledgments

Thanks to Garv Shah for his help in the initial stages of the project. This project is supported in part by NSF 2312762.

References

[1] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research* (2016). https://doi.org/10.1177/0278364915620033

[2] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* (2021).

[3] Yasra Chandio, Noman Bashir, and Fatima M Anwar. [n. d.]. Dataset: HoloSet-A Dataset for Visual-Inertial Pose Estimation in Extended Reality. ([n. d.]).

[4] Aditya Dhakal, Xukan Ran, Yunshu Wang, Jiasi Chen, and K.K. Ramakrishnan. 2022. SLAM-Share: Visual Simultaneous Localization and Mapping for Real-time Multi-user Augmented Reality. In *Proc. ACM CoNEXT*.

[5] Juan C. Dibene and Enrique Dunn. 2022. HoloLens 2 Sensor Streaming. arXiv:2211.02648 [cs.MM] https://arxiv.org/abs/2211.02648

[6] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3354–3361.

[9] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[10] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *Josa a* 5, 7 (1988), 1127–1135.

[11] IKEA. 2017. IKEA Place app launched to help people virtually place furniture at home. urlhttps://www.ikea.com/global/en/newsroom/innovation/ikea-launches-ikea-place-a-new-app-that-allows-people-to-virtually-place-furniture-in-their-home-170912/.

[12] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. 1993. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology* 3, 3 (1993), 203–220.

[13] Georg Klein and David Murray. 2007. Parallel tracking and mapping for small AR workspaces. In *Proc. IEEE ISMAR*.

[14] S Klenk, J Chui, N Demmel, and D Cremers. 2021. TUM-VIE: The TUM Stereo Visual-Inertial Event Dataset. In *International Conference on Intelligent Robots and Systems (IROS)*. arXiv:2108.07329 [cs.CV]

[15] Steven LaValle. 2023. *Virtual Reality*. Cambridge University Press.

[16] Jinyu Li, Bangbang Yang, Danpeng Chen, Nan Wang, Guofeng Zhang*, and Hujun Bao*. 2019. Survey and Evaluation of Monocular Visual-Inertial SLAM Algorithms for Augmented Reality. *Journal of Virtual Reality and Intelligent Hardware* (2019). https://doi.org/10.3724/SP.J.2096-5796.2018.0011

[17] Leonel Merino, Magdalena Schwarzl, Matthias Kraus, Michael Sedlmair, Dieter Schmalstieg, and Daniel Weiskopf. 2020. Evaluating mixed and augmented reality: A systematic literature review (2009-2019). In *IEEE ISMAR*. IEEE, 438–451.

[18] Microsoft. 2022. Holographic Remoting Overview. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/holographic-remoting-overview.

[19] Microsoft. 2022. Mixed reality capture overview. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/mixed-reality-capture-overview.

[20] Xukan Ran, Carter Slocum, Yi-Zhen Tsai, Kittipat Apicharttrisorn, Maria Gorlatova, and Jiasi Chen. 2020. Multi-user augmented reality with communication efficient and spatially consistent virtual objects. In *Proc. ACM CoNEXT*.

[21] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers. 2018. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In *International Conference on Intelligent Robots and Systems (IROS)*.

[22] Carter Slocum, Xukan Ran, and Jiasi Chen. 2021. RealityCheck: A tool to evaluate spatial inconsistency in augmented reality. In *Proc. IEEE International Symposium on Multimedia (ISM)*.

[23] Fan Yang, Kaijian Huang, Tianyi Hu, Ying Chen, and Maria Gorlatova. 2024. Tracking Performance Analysis of Commercial XR Headsets and Open-Source Platforms (internal poster). (2024).

Appendix

A Implementation details

In practice, we do not need to account for the movement of the Hololens during the replay, as this remains stationary and the output can be viewed from a remote window as a result of the Holographic Remoting functionality. Moreover, since the Hololens initializes its own position at the origin, we do not need to account for this when creating the initial transformation for the replay. This allows us to simplify the implementation from equations (1) and (2) to the following using 4x4 transformation matrices, letting P^{obj} be the 4x4 matrix of the pose of the virtual object in the space:

$$t_k = R_k^\alpha R_0^\alpha (t_k^\alpha - t_0^\alpha) R_0^{\text{HMD}} P^{\text{obj}}[3, 0:3] \tag{5}$$

$$R_k = (R_k^\alpha R_0^\alpha (t_k^\alpha - t_0^\alpha) R_0^{\text{HMD}} P^{\text{obj}})^{-1}[:3, :3] \tag{6}$$

Where [3,0:3] denotes the translation component of the matrix and [:3,:3] denotes the rotation component, where the indices of the matrix are 0-indexed.