User-in-the-Loop View Sampling with Error Peaking Visualization

Ayaka Yasunaga[1], Hideo Saito[1], and Shohei Mori[2,1]
[1]Keio University, [2]University of Stuttgart

Paper ID: 1930

Project Page

LinkedIn

Introduction

Our error peaking Focus peaking

Conventional AR visualizations [Mildenhall et al., 2019]

Background

- **Goal**: Explore view sampling strategy for novel view synthesis with fewer multi-view images
- **Key idea :** Augmented reality (AR) can provide ways to visualize missing view samples.
- **Existing work:** Determine view intervals according to the plenoptic sampling theory
  → Requiring uniform view sampling , thereby ignoring the structure of the target scene

Remaining challenges in AR visual guidance

1. Hide the subject to be photographed and hinder the entertaining nature of photography
2. The alignment task known to be mentally demanding.
3. The globally applied minimum view intervals that may not make sense at the moment.
4. Fixed visualization limiting users' ability to explore beyond predetermined areas

Contribution

- AR-based error-peaking visualization of the area of imperfect local light field reconstruction inspired by the focus peaking visualization for tuning the in-focus of camera lens

Proposed Method

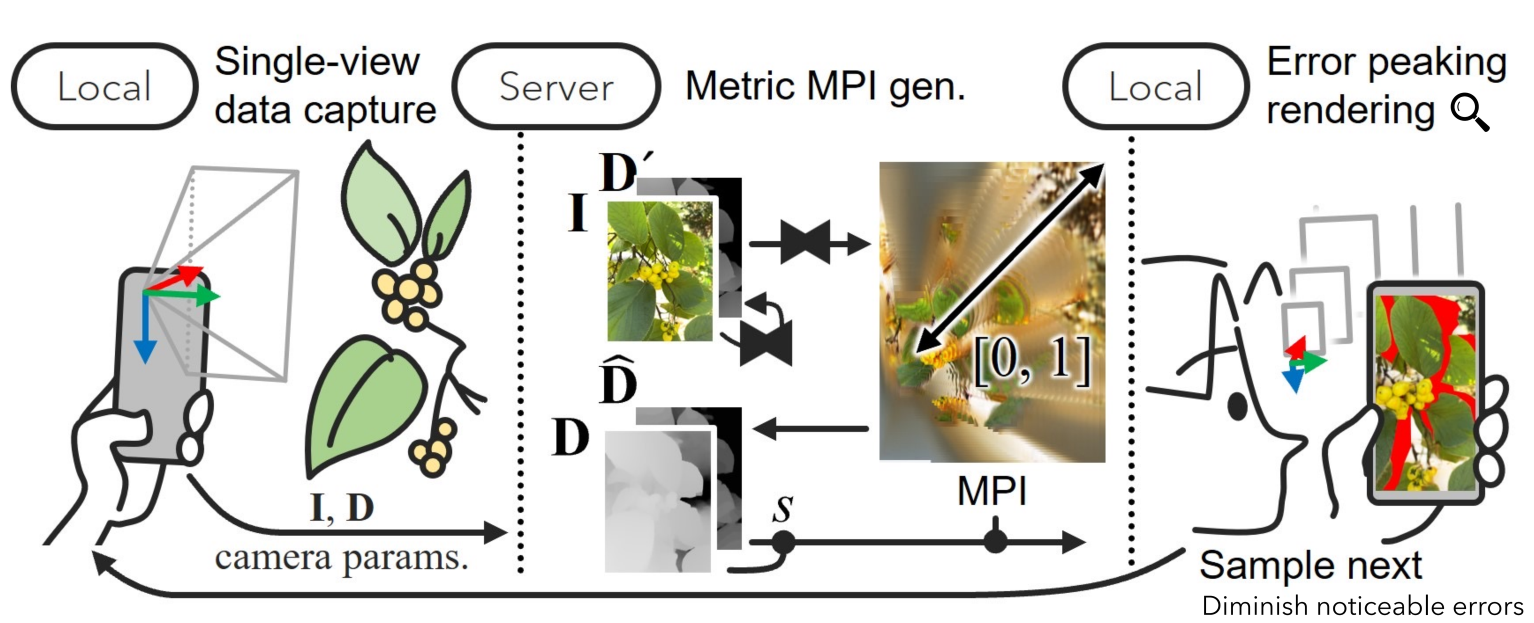

Fig. User-in-the-loop view sampling system overview

Metric MPI Generation

Single-view data
Our system allows users to capture single-view data (RGB image $\mathbf{I}$, and metric depth map $\mathbf{D}$) on AR-supported mobile phone.

Multi-Plane Images (MPI) from single-view data
Given $\mathbf{I}$ and $\mathbf{D}$, we generate an MPI using AdaMPI [Han et al., 2022].

Blending multiple MPI
To avoid frequent switching artifacts between the nearest MPI to another,
→ Blending k nearest MPI volumes ($k = 3$ by default).

Error Peaking Visualization for Next View Samples

1. Calculate the difference between the video frame and MPI rendering
2. Highlight pixels with significant errors in red
3. Diminish these errors by inserting more view samples.

View synthesis quality

Dataset:
We recorded a real image dataset with varying depths and camera motions.
We evaluated the rendering quality of MPI and 3D Gaussian splatting (3DGS).

Result:
Our approach produces fewer artifacts than Uniform (corresponds to LLFF) and Random.

Table. View synthesis quality of different view sampling strategies

Method (MPI)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Random	0.89 (0.73)	0.92 (0.56)	1.09 (0.67)
Uniform	0.97 (0.87)	0.96 (0.78)	1.03 (0.86)
Ours	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
Method (3DGS)	**PSNR (↑)**	**SSIM (↑)**	**LPIPS (↓)**
Random	0.92 (1.74)	0.986 (1.98)	1.30 (1.96)
Uniform	0.96 (1.27)	0.997 (1.12)	1.09 (1.28)
Ours	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)

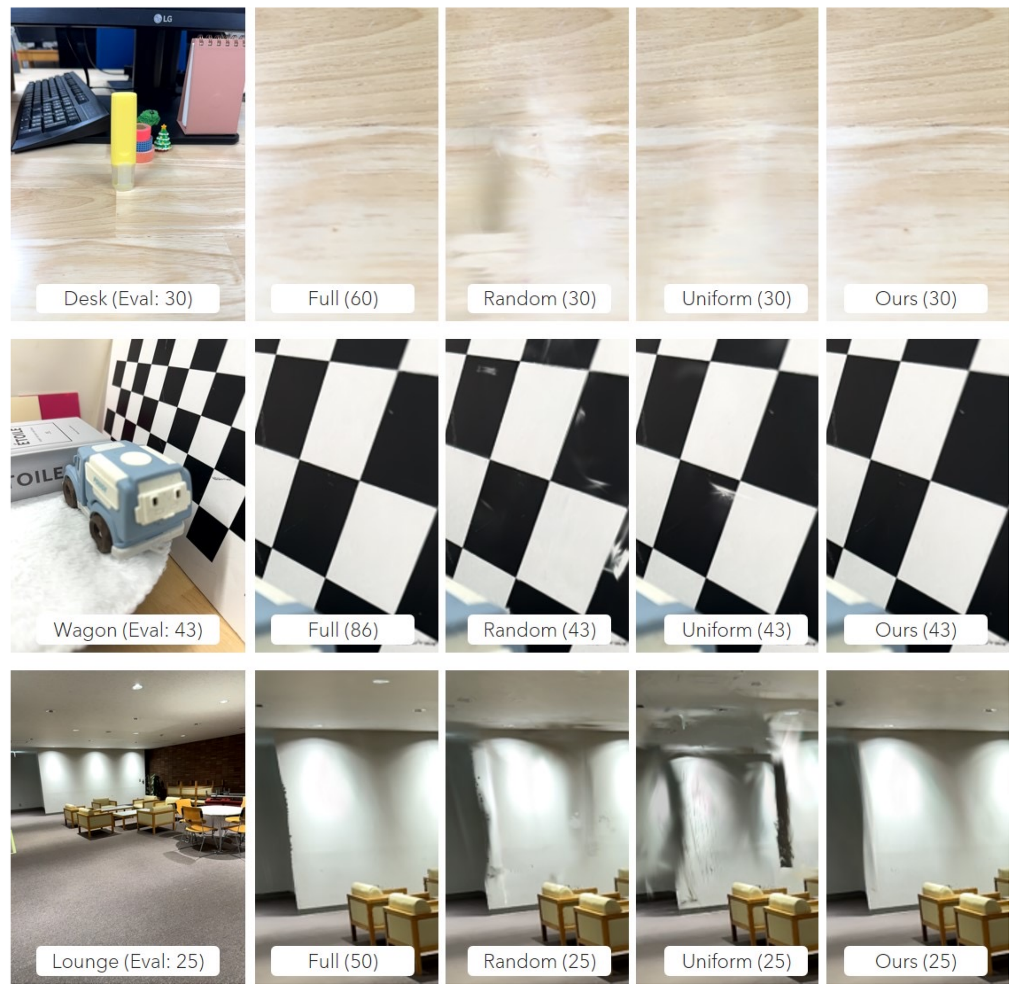

Fig. 3DGS rendering results

Contact

Ayaka Yasunaga: ayaka.yasunaga@keio.jp, Hideo Saito: hs@keio.jp, Shohei Mori: s.mori.jp@ieee.org

User Study

Goal: We validate the capability of users to spontaneously sample views under our error-peaking visualization in a user study.

Baseline implementation: The LLFF visual guidance based on plenoptic sampling theory

Metrics
- Usability: SUS, NASA-TLX
- Completion time: TCT, NCT
- Self-Confidence: *"Q. The captured results will meet my expectations."*
- Satisfaction: *"Q. The captured results met my expectations."*
- Scene Focus: *"Q. I focused on the scene of interest while capturing. ."*

Participants:
We collected 20 participants (five female and 15 male, $\bar{X}$ =22.4 (SD=1.4) years old, all right-handed and corrected vision).

Scene: Designed an office scene with varying depths during the scene capture.

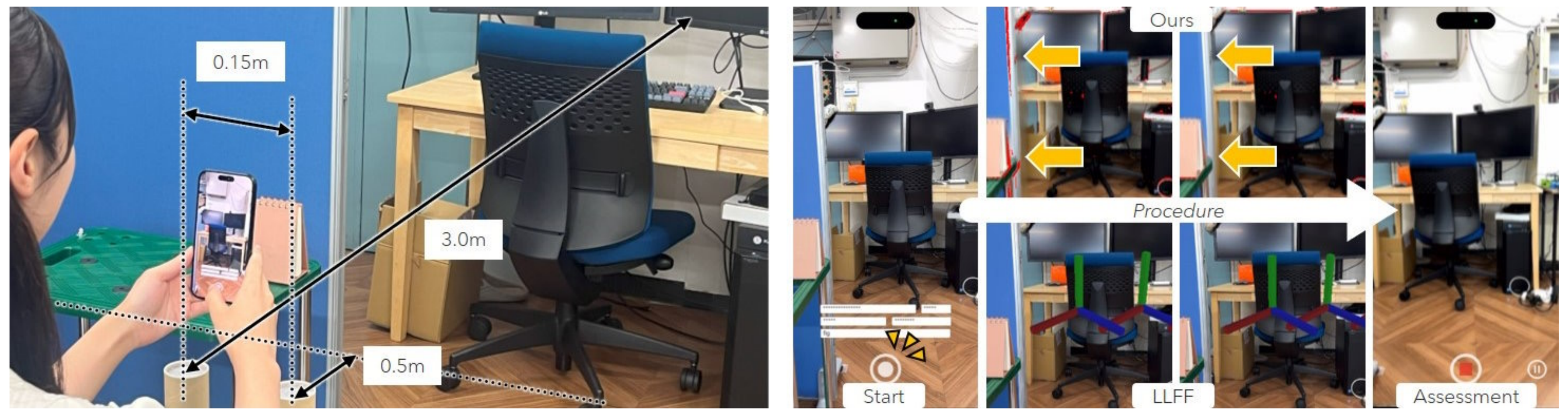

Fig. Experimental setup and procedure

Result:
Ours gives higher self-confidence during view sampling with better scene focus and lower disappointment in the final rendering quality with fewer views.

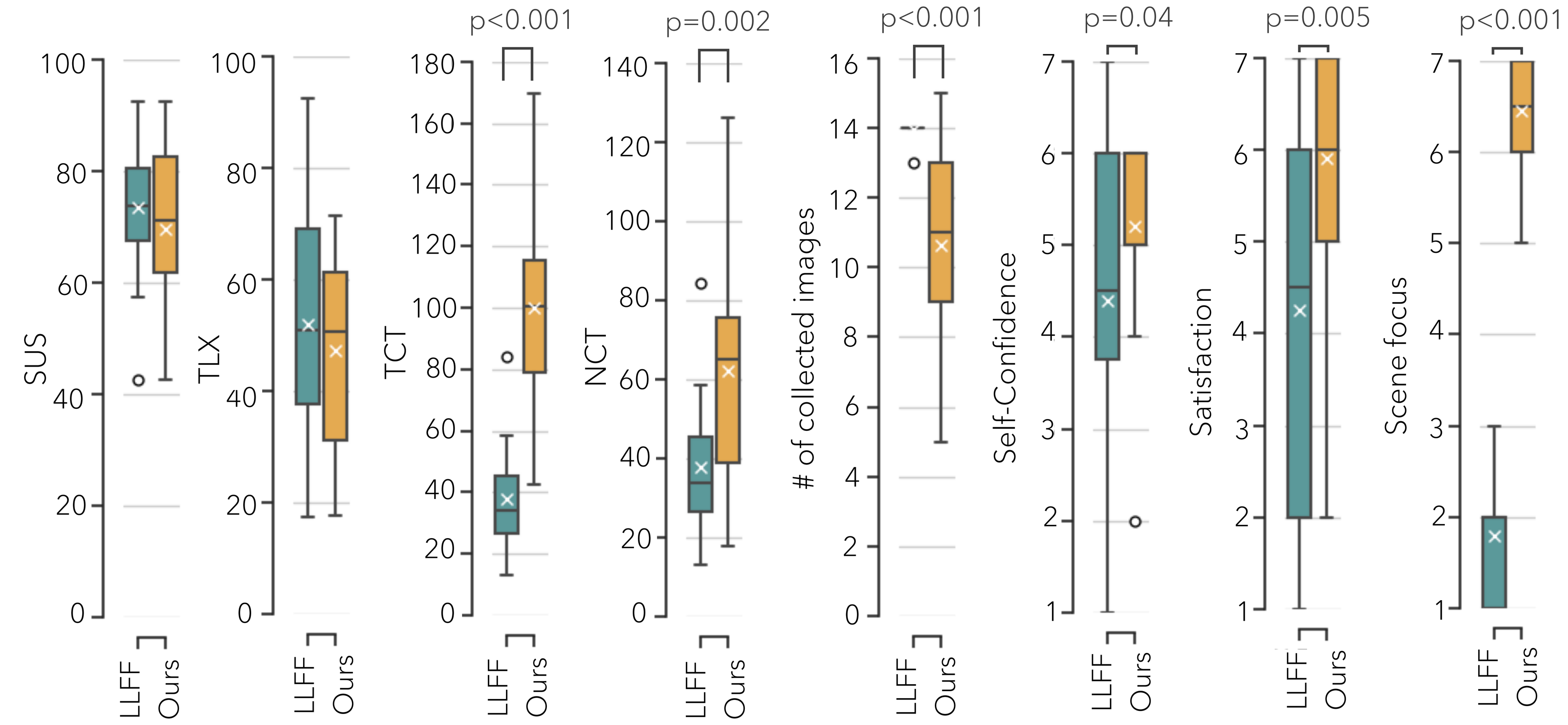

Fig. User study results

Limitations

Latency in communication and network inference
- Our system still experiences delays
  - Due to network inference and communication time between the server and the client.

Further user analysis
- Missing factors: the screen resolution, the distance from the AR display to the users' eyes, and the users' motion while observing the results