This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101092889, Topic HORIZON-CL4-2022-HUMAN-01-14

SHARESPACE

Embodied Social Experiences in Hybrid Shared Spaces

Project Reference No	101092889
Deliverable	D4.9. Photo-realistic scene neural rendering v1
Workpackage	WP4: Interaction-Aware Avatar Animation and Rendering
Nature	D (Deliverable)
Dissemination Level	PU - Public
Date	05/06/2024
Status	v1.0
Editor(s)	Marcel, Rogge (DFKI)
Involved Institutions	DFKI
Document Description	This deliverable presents the first results on neural rendering with spherical images for large spaces.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Table 1: List of Abbreviations

Term / Abbreviation	Definition
AI	Artificial Intelligence
MSI	Multi-Sphere Image
VR	Virtual Reality
SfM	Structure from motion
HMD	Head-mounted Display

1 INTRODUCTION

Neural rendering approaches have advanced drastically in the last few years. They can produce photo-realistic reconstructions of objects and scenes. However, rendering speed has been one of the main challenges. Real-time rendering speeds are difficult to obtain but necessary for many applications. Especially in VR, high frame rates are necessary to provide good immersion and prevent side effects such as motion sickness.

We present the first results of our approach for photo-realistic scene neural rendering. We combine the best available neural rendering methods with a scene representation that enables real-time rendering. We extended the MSI scene representation to be able to represent very large scenes, which would be too large to fit a single MSI. Additionally, we provide a plugin for Unreal Engine that enables the integration of our novel multi-MSI representation into projects within Unreal Engine.

2 BACKGROUND

Neural rendering deals with learning a scene's representation and rendering any view of it using a neural network. Neural rendering makes it possible to easily capture models of real-life scenes, which otherwise would be difficult and expensive to achieve, simply using images as input data.

One of the requirements for SHARESPACE is to have virtual scenes, in which people can come together. For example, the Sport scenario requires a virtual scene in which people in VR can ride on bicycles. This will require the scenes to be exceptionally large with multiple kilometers of road. Additionally, the scenes should be photo realistic to provide the highest possible immersion. Creating such scenes with classical methods would be prohibitively expensive.

Prior neural rendering methods dealt only with object [3] or small-scale scene [4] rendering. Later methods expanded on this to work for large-scale scenes as well [5]. However, rendering frame rates are low, which would be a problem for the user experience.

Our method builds on our work called "SOMSI" Tewodros et al. [1], which enables very high frame rate rendering using standard hardware. Even though only small-scale scenes were shown, there is a reasonable expectation to be able to extend this to large-scale scenes.

3 APPROACH

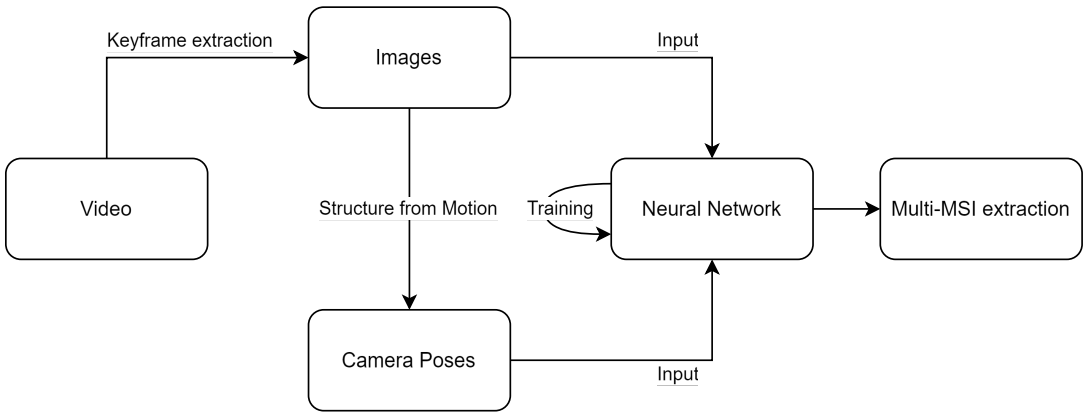

Figure 1 – Overview of our pipeline.

We developed and investigated the following approaches:

- Construction of a capturing device
- Studied different SfM and Bundle Adjustment frameworks because the quality of the MSI generation depends heavily on the accurate calculation of the position and orientation of the camera in the scene.
- MSI and multi-MSI for large scenes using SOMSI
- Extracting MSI from Nerf

Our approach starts with recording a video of any scene. As we target to capture as much as possible from an outdoor, large environment, we use spherical and fisheye cameras provided by the SHARESPACE partner Ricoh. Figure 2 shows a capturing rig that we constructed in collaboration with Ricoh to use up to six of their cameras simultaneously. We then extract individual frames from the video. The frames are used to perform SfM, which returns the camera poses belonging to each frame. Given the frames with the position information, we train a neural network to represent the real scene accurately. After the training is concluded, we can extract a multi-MSI representation that can be rendered in real time. An overview of this pipeline can be seen in Figure 1. A more detailed explanation is given in the following.

Figure 2 – Bicycle capturing rig with six Theta Z1 cameras from Ricoh.

There are various ways of recording a scene. To simplify the data capture, we record videos with one or multiple cameras instead of taking individual images. Each camera is pre-calibrated using Matlab's camera calibration tool. We extract keyframes from the videos in a way that ensures sufficient motion between each frame, e.g. every 10^{th} frame. The extracted video frames are then used to perform SfM with Colmap. Colmap then returns the camera's position and orientation within the scene at the time that each frame was captured.

We use both, the extracted video frames and their corresponding camera poses, as input for the neural network. The neural network encodes an initially empty scene. A classical volume rendering procedure is performed on the encoded scene for each of the camera poses. The neural network then updates the internal scene representation by comparing the rendered images with the actual video frames. By

involving the volume rendering procedure, the network is forced to learn a meaningful result. This enables it to learn the real scene geometry.

After the network learned a high-quality representation of the scene, it is possible to render even previously unknown images of the scene. We then extract our multi-MSI representation from the network. Each MSI is a simplified scene representation of multiple spheres centered at one point. Each sphere contains only scene contents that it intersects with. Their advantage lies in the simple real-time rendering. By arranging multiple of these MSI close together, it is possible to visualize scenes that are too large for a single MSI. An example of the MSI representation is shown in Figure 5.

We implemented an Unreal Engine plugin that allows the real-time rendering of multi-MSI. Based on the position of the user's HMD, we load all MSI that are within a predefined radius of the user. If the user moves, any MSI that comes within the radius will be loaded and any MSI that goes out of the radius will be unloaded. At any given point, only the single MSI that is closest to the user will be rendered. Dynamic loading is essential to reduce the hardware requirements for the system.

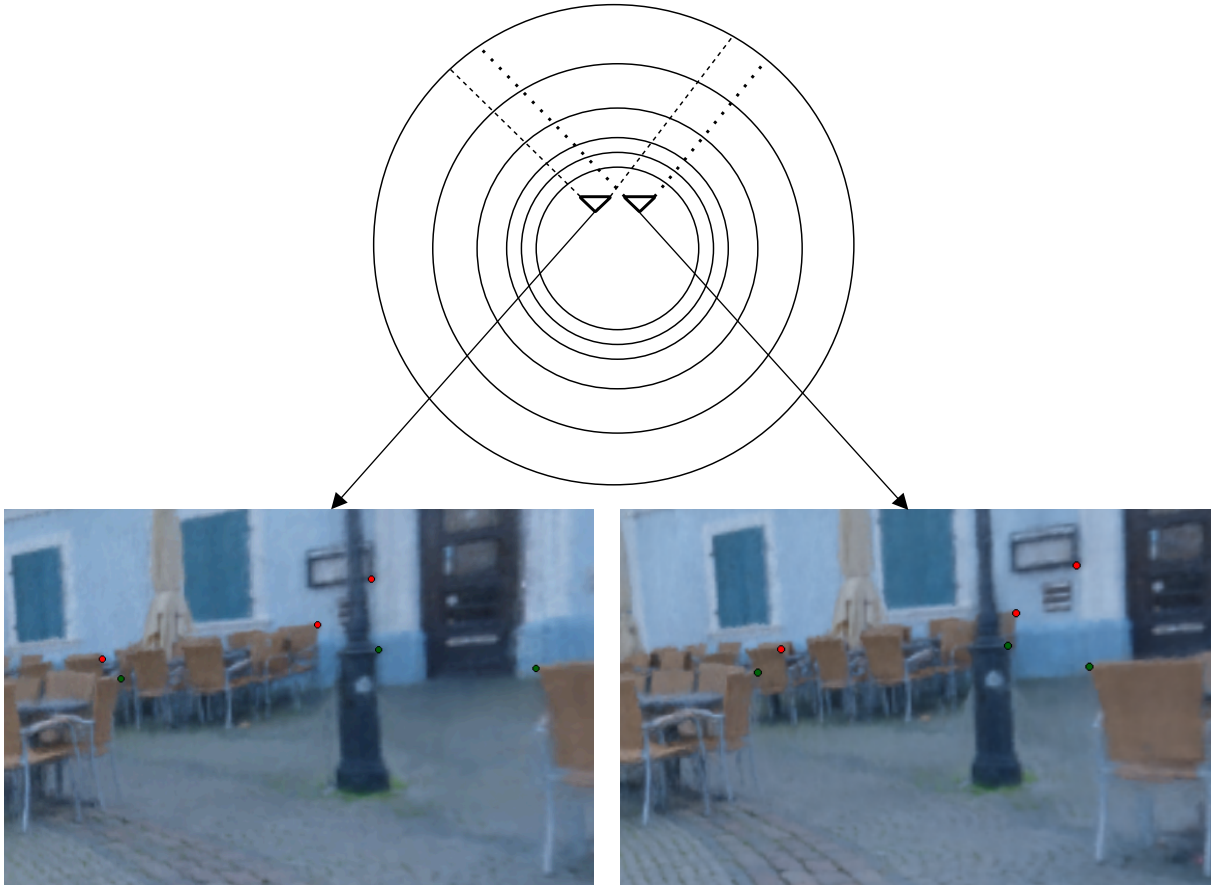

Figure 3 – Example of an MSI rendering for two views from different positions. A few points in the background and foreground have been highlighted to emphasize the motion.

4 RESULTS

We implemented a multi-MSI renderer as an Unreal Engine plugin. It can render multi-MSI scenes in real time. Multi-MSI scenes are created through our neural rendering approach described in section 3. Figure 3 shows a screenshot from Unreal Engine while running our plugin. A video of the live rendering is also available to be viewed on YouTube: www.youtube.com/watch?v=ChyHFYZYhSw

Figure 4 – Screenshot of the Unreal Engine plugin while rendering a multi-MSI scene. Optionally, a mini-map in the top right shows the locations of the VR user and the MSI.

We examined the memory requirements of differently sized multi-MSI scenes. This was done by tracking the overall system memory usage while operating only the Unreal Engine plugin. We observed that the importing of a scene requires a substantially larger amount of memory than the later usage of that scene. The import of a scene is a necessary step that needs to be performed once when loading a new scene for the first time. After a scene has been imported once, it can be loaded without an import in the future. We observe that an average of 0.3 GB per MSI is necessary to load and play a multi-MSI scene. The initial import of a multi-MSI scene, however, requires approximately 1.75 GB per MSI. Even though the import requires significantly more memory, this step should be performed before the whole system is taken into operation. This means that other SHARESPACE systems do not have to be running, which will leave more system resources available. An overview of the memory requirements from our tests is shown in Table 2.

Table 2: Memory requirements for loading multi-MSI.

	1 MSI	2 MSI	3 MSI	4 MSI	9 MSI
Load & Play	0.4 GB	0.5 GB	0.9 GB	1.3 GB	2.5 GB
Import	2.1 GB	3.5 GB	5.5 GB	7.0 GB	14.8 GB

5 LIMITATIONS AND NEXT STEPS

We show that we can scale the MSI representation to larger scenes. However, there are some concerns about how well it scales, especially considering the extremely large scene requirements. One scaling factor is time. Training a high-quality reconstruction can take more than 24 hours on medium-sized scenes. Scenes of the required multiple kilometers would take multiple times that. Additionally, loading large multi-MSI scenes require a lot of memory. We already addressed this by having a dynamic loading of the necessary MSIs. However, it is not yet clear how many MSI need to be loaded at any given time and how much memory the overall system would need to have.

The results of the multi-MSI rendering show that there are some flickering effects within the scene. This happens because each MSI discretizes the scene in different areas. This can cause small details or lighting effects to be represented in different ways. These differences between the MSI will be recognized as a flickering effect whenever the MSIs are quickly switched.

More recent advances in neural rendering show a novel approach which supports real-time rendering [2]. Previously, it was not possible to integrate a real-time rendering pipeline based on neural rendering in e.g. Unreal Engine. This was the advantage of the MSI representation because it enables real-time rendering that can easily be integrated in other software. The downside of the MSI is that they are not a complete representation of the scene. The new gaussian splatting approach would not suffer from the flickering effects we saw with multi-MSI. Additionally, the memory and training time constraints are lower than those of the multi-MSI. This is why we are now experimenting with using gaussian splatting for large-scale scenes and target at developing a solution that combines both approaches into a global hybrid rendering framework.

6 REFERENCES

[1] Habtegebrial, T., Gava, C., Rogge, M., Stricker, D., & Jampani, V. (2022). Somsi: Spherical novel view synthesis with soft occlusion multi-sphere images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15725-15734).

[2] Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 42(4), 1-14.

[3] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), 99-106.

[4] Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., & Kanazawa, A. (2022). Plenoxels: Radiance fields without neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5501-5510).

[5] Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P. P., ... & Kretzschmar, H. (2022). Block-nerf: Scalable large scene neural view synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8248-8258).