Efficient 3D Reconstruction, Streaming and Visualization of Static and Dynamic Scene Parts for Multi-client Live-telepresence in Large-scale Environments

Leif Van Holland[1] Patrick Stotko[1] Stefan Krumpen[1] Reinhard Klein[1] Michael Weinmann[1,2]

[1]University of Bonn [2]Delft University of Technology

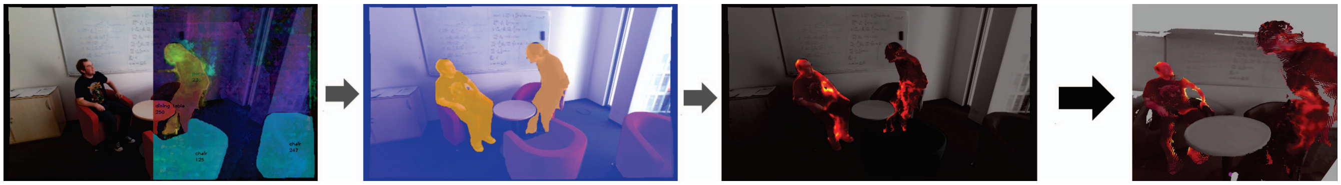

Figure 1: Visualization of the key components of our proposed pipeline. The color image is blended with class and instance information, and shown along with the optical flow with respect to the previous frame (first image). This information is integrated to produce a mask that segments the frame into static and dynamic regions (second image). Together with an accumulated 3D motion estimate (third image), the scene is streamed to one or multiple remote clients for immersive exploration in VR (fourth image). In this example, the user chose to view the accumulated 3D motion.

Abstract

Despite the impressive progress of telepresence systems for room-scale scenes with static and dynamic scene entities, expanding their capabilities to scenarios with larger dynamic environments beyond a fixed size of a few square-meters remains challenging.

In this paper, we aim at sharing 3D live-telepresence experiences in large-scale environments beyond room scale with both static and dynamic scene entities at practical bandwidth requirements only based on light-weight scene capture with a single moving consumer-grade RGB-D camera. To this end, we present a system which is built upon a novel hybrid volumetric scene representation in terms of the combination of a voxel-based scene representation for the static contents, that not only stores the reconstructed surface geometry but also contains information about the object semantics as well as their accumulated dynamic movement over time, and a point-cloud-based representation for dynamic scene parts, where the respective separation from static parts is achieved based on semantic and instance information extracted for the input frames. With an independent yet simultaneous streaming of both static and dynamic content, where we seamlessly integrate potentially moving but currently static scene entities in the static model until they are becoming dynamic again, as well as the fusion of static and dynamic data at the remote client, our system is able to achieve VR-based live-telepresence at close to real-time rates. Our evaluation demonstrates the potential of our novel approach in terms of visual quality, performance, and ablation studies regarding involved design choices.

1. Introduction

Sharing immersive, full 3D experiences with remote users, while allowing them to explore the respectively shared places or environments individually and independently of the sensor configuration, represents a core element of *metaverse* technology. Beyond pure 2D images or 2D videos, 3D telepresence is defined as the impression of individually *being there* in an environment that may differ from the user's actual physical environment [98, 38, 49, 170, 28]. This offers new opportunities for diverse applications including remote collaboration, entertainment, advertisement, teaching, hazard site exploration, rehabilitation as well as for joining virtual sports events, work meetings, remote inspection, monitoring and maintenance, consulting applications or simply enjoying social gatherings. In turn, the possibilities for virtually bringing people or experts together from all over the world in a digital twin of a location as well as the live-virtualization of such environments and events may reduce the effort regarding on-site traveling for many people, which not only helps to reduce our CO_2 footprint and increase the efficiency of various processes due to time savings, but also facilitates economically less well-situated or handicapped people to access such environments or events.

The creation of an immersive telepresence experience relies on various factors. Respective core features are visually convincing depictions of a scenario as well as the subjective experience, vividness, and interactivity in terms of operating in the scene [138, 133]. Therefore, the involved aspects include display parameters (e.g., resolution, frame rate, contrast, etc.), the presentation of the underlying data, its consistency, low-latency control to avoid motion sickness, the degree of awareness and the suitability of controller devices [98, 38, 49, 170, 28, 138, 133]. Furthermore, experiencing 3D depth cues like stereopsis, motion parallax, and natural scale also contribute to the perceived level of immersion and copresence [45, 103].

However, such immersive 3D scene exploration experience becomes particularly challenging for telepresence in live-captured environments due to the additional requirement of accurately reconstructing the digital twin of the underlying scene on the fly as well as its efficient streaming and visualization to remote users under the constraints imposed by available network bandwidth and client-side compute hardware. Among many approaches, impressive immersive AR/VR-based live-3D-telepresence experiences have only been achieved based on advanced RGB-D acquisition for dynamic scenes on the scale of rooms, i.e. areas of only a few square-meters, using special expensive static capture setups [47, 162, 92, 91, 21, 127, 34, 194, 113, 59, 69, 145, 29, 114, 23, 76] and display technology [76], as well as for static scenes beyond that scale based on low-cost and light-weight incremental scene capture with a moving depth camera [101, 139, 142, 141, 140]. For the latter category, bandwidth requirements have been reduced from hundreds of MBit/s for a single user [101] to around 15MBit/s for group-scale sharing of telepresence in live-captured environments while also handling network interruptions [139, 142, 140], thereby even allowing live-teleoperation of robots [141]. However, expanding the capabilities and, thereby, overcoming the aforementioned limitations in large dynamic environments for many users with low-cost setups still remains an open challenge.

In this paper, we aim at sharing 3D live-telepresence experiences in large-scale environments beyond room scale with *both static and dynamic scene entities* at practical bandwidth requirements and based on light-weight scene capture with a single moving consumer-grade RGB-D camera. For this purpose, we propose a respective system that relies on efficient 3D reconstruction, streaming and immersive visualization for dynamic large-scale scenes.

In particular, the key contributions of our work are:

- For the sake of efficiency, our system leverages a hybrid volumetric scene representation, where we use optical flow and instance information extracted from the input frames to detect static and dynamic scene entities, thereby allowing the combination of a classic implicit surface geometry representation enriched with the object semantics as well as their accumulated dynamic motion over time, with a point-cloud-based representation of dynamic parts.
- We achieve efficient data streaming to remote users by the separate yet simultaneous streaming of both static and dynamics scene information, where we seamlessly integrate potentially moving but currently static scene entities in the static model until they are becoming dynamic again. Additionally, the fusion of static and dynamic data at the remote client allows VR-based visualization of the scene at close to real-time rates.
- We demonstrate the potential of our approach in the scope of several experiments and provide an ablation study for respective design choices.

Furthermore, while not being among the main contributions of our work, our approach also inherits the robustness of previous techniques to network interruptions for the reconstruction of the static scene parts as well as the scalability to group-scale telepresence [139, 142, 141]. An overview of our proposed system is depicted in Figure 1.

2. Related Work

Telepresence Systems Despite almost two decades of progress, the development of systems that allow immersive telepresence experiences remains challenging due to the prerequisite of simultaneously achieving high-fidelity real-time 3D scene reconstruction, the efficient streaming and management of the reconstructed models and the high-quality visualization based on AR and VR equipment. Early approaches were limited by the capabilities of the available hardware [41, 64, 104, 157, 152, 71] or inaccurate silhouette-based reconstruction techniques [121, 88]. Depth-based 3D scanning led to improved reconstruction quality and allowed telepresence at the scale of rooms [55, 91, 93, 99, 62, 42], however, remaining artifacts induced by the high sensor noise and temporal inconsistency in the reconstruction process still impacted the visual experience. More recently, advances in 3D scene capture, streaming, and visualization technology led to impressive immersive AR/VR-based live 3D telepresence experiences. Live-telepresence for small-scale scenarios of a few square-meters has been achieved based on light-weight capture setups for teleconferencing [109, 61, 32, 117, 4, 19] and other collaborative scenarios [185, 137, 89, 46, 153, 33] as well as based on expensive multi-camera static and pre-calibrated capture setups [47, 162, 92, 91, 21, 127, 34, 194, 113, 59, 69, 145, 29, 114, 23, 76]. Furthermore, live-telepresence for scenarios beyond a few square-meters has been achieved based on low-cost and light-weight incremental scene capture with a moving depth camera [7, 101, 139, 141, 142, 140, 182], allowing remote users to immersively explore a live-captured environment independent of the sensor configurations. Regarding the latter approaches, impractical bandwidth requirements of up to 175 MBit/s for immersive scene exploration by a single user [101] have been overcome by more recent approaches that allow group-scale sharing of telepresence experiences in live-captured environments while also handling network interruptions [139, 142, 141, 140] as well as live-teleoperation of robots [141]. Furthermore, mechanisms for annotation, distance measurement [141] and efficient collaborative VR-based 3D labeling were added [193]. However, practical sharing of live-captured 3D experiences in dynamic large-scale environments for many users with low-cost setup still remains an open challenge. The same applies for immersive robot teleoperation, where approaches focused on small-scale scenarios with dynamics [120, 70, 85, 156, 168, 129, 106] and large-scale, static scenarios [141].

In contrast to the aforementioned approaches, we propose a live-telepresence system for large-scale environments while also taking scene dynamics into account.

3D Reconstruction and SLAM Techniques Current state-of-the-art telepresence systems rely on depth-based simultaneous localization and mapping (SLAM) techniques. Examples are the use of depth-sensor-based 3D scene capture based on surfels [51] or extensions of Kinect-Fusion [108, 55] in terms of voxel block hashing techniques [110, 75, 73, 74, 122] for incremental scene capture for large-scale telepresence applications [101, 139, 142, 141, 140]. To avoid the need for depth sensors, more recent SLAM approaches for incremental scene capture – that might be applicable in respective telepresence applications – leveraged principles of deep learning [72, 178, 66, 68, 177, 25, 169]. Further approaches investigated 3D reconstruction from multiple synchronized cameras [100, 2, 31, 1, 54].

Recently, neural scene representation and rendering techniques [154, 155] have led to significant improvements in reconstruction quality for small-scale objects or scenes. The underlying idea originates from novel view synthesis and consists of training a neural network to represent a scene with its weights, so that respectively synthesized views match the input photographs. In particular, this includes implicit scene representations based on Neural Radiance Fields (NeRFs) [97] and respective extensions towards speeding up model training [125, 39, 16, 26, 10, 164, 147, 105, 37, 9, 11, 188, 179, 102] with training times of seconds, the adaptation to unconstrained image collections [94, 13, 63], deformable scenes [115, 123, 43, 158, 124, 111, 160, 118, 116, 12, 87, 58, 83, 37] and video inputs [82, 174, 30, 119, 44, 81, 151, 79], the refinement or complete estimation of camera pose parameters for the input images [181, 165, 146, 20, 192, 191, 130, 187, 95, 84, 57, 173, 90, 6, 17, 15, 14, 52, 148, 86], combining NeRFs with semantics regarding objects in the scene [163, 189, 40], incorporating depth cues [166, 26, 128, 126, 3] to guide the training and allow handling textureless regions, handling large-scale scenarios [150, 161, 96], and streamable representations [18, 149]. However, despite promising results, current solutions [146, 192, 191, 130, 187, 90] do not yet reach real-time performance but only reach 12 FPS on a high-end GPU [130] or less within completely static environments. Further improvements regarding efficiency and the handling of dynamic scenes are required to achieve real-time performance for the joint camera pose estimation and neural scene reconstruction in a SLAM setting in dynamic environments.

Particularly addressing dynamic environments, various approaches focused on filtering dynamic objects and only reconstructing the static background [67, 134, 35, 5, 183, 186, 176, 175, 24, 77, 171, 36] or additionally reconstructing the dynamics based on rigid object tracking and reconstruction [144, 172, 80, 132, 131, 50] and non-rigid object tracking and reconstruction [78, 65, 180, 107, 48, 167, 53, 167, 27, 184, 135, 56, 136, 143, 159, 8]. Taking inspiration of the non-rigid scene reconstruction approaches in terms of separating static and dynamic scene parts, the 3D reconstruction approach involved in our live-telepresence system is particularly designed for capturing large-scale environments (i.e., beyond scenarios limited to a small area of a few square-meters) with both static and dynamic entities based on a single moved RGB-D camera. Our hybrid volumetric scene representation leverages semantic and instance information to detect dynamic scene entities and combines a voxel-based scene representation for the static parts, where we also accumulate information on whether and how significant objects have been moved, with a point-cloud-based representation of dynamic parts. A major contribution of our work is the separate but simultaneous streaming of both static and dynamics scene information and its VR-based visualization at close to real-time rates.

3. Methodology

As shown in Figure 2, our live-telepresence system for large-scale environments with scene dynamics at practical bandwidth requirements takes a continuous stream of RGB-D images $(I_1, D_1), (I_2, D_2), ...$ from a moving depth camera as input, where $I_k(u) \in \mathbb{R}^3$ represents the red, green, and blue color values of frame k, and $D_k(u) \in \mathbb{R}$ the corresponding raw depth measurement at pixel $u \in \mathcal{U} \subset \mathbb{N}^2$, with $\mathcal{U}$ being the image domain. The main challenge consists in an efficient processing of these measurements, their efficient integration into a consistent model and the efficient streaming of the latter over the network at practical band-

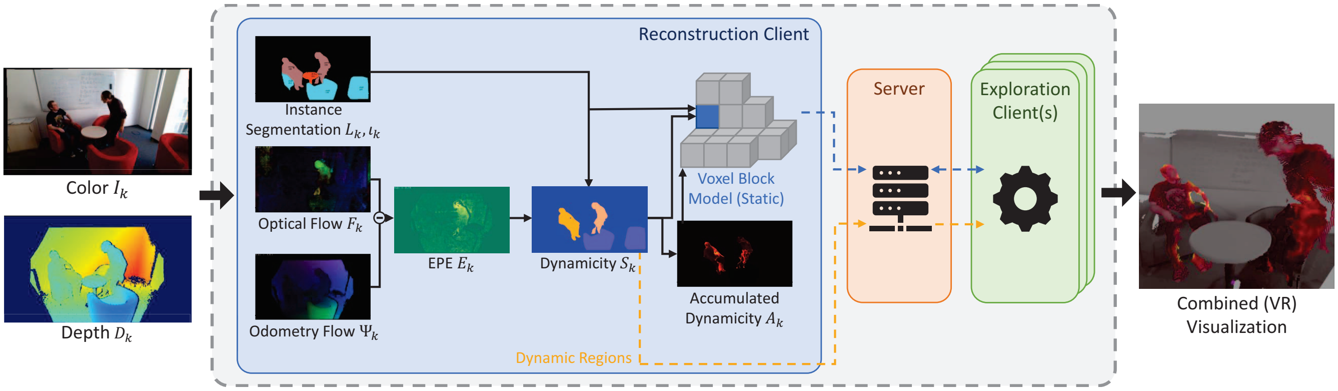

Figure 2: Visualization of different processing stages for the k-th RGB-D frame in the pipeline. Starting with color I_k and depth D_k, instance segmentation L_k (class labels) and ι_k (instance IDs), optical flow F_k and odometry flow Ψ_k (i.e., the flow generated from the estimated camera motion) are computed. Next, the end-point-errors (EPE) between the flows are computed, normalized and propagated using the instance segmentation to generate the dynamicity scores S_k. The scores are accumulated in A_k and L_k, ι_k, S_k and A_k are used to integrate information about static regions in the voxel block model. New static voxels and current dynamic regions are sent to the server, which forwards this information to the exploration clients appropriately.

width requirements to remote clients, where it has to be visualized at adequate visual quality and at tolerable overall latency. For this purpose, we use a hybrid scene representation that separately handles static and dynamic scene parts, thereby allowing the combination of efficient large-scale 3D scene mapping techniques, that face problems with dynamic regions, with efficient point-based reconstruction for the dynamic parts. In more detail, we segment the frames of the input stream into static and dynamic regions by determining score maps S_k, where $S_k(u) \in \mathbb{R}$ describes the amount of dynamicity in frame k at pixel u. This separation allows us to efficiently reconstruct, stream and immersively visualize static regions using existing state-of-the-art large-scale telepresence techniques [139, 142] while simultaneously reconstructing, streaming and visualizing dynamic scene parts based on a point-based representation in terms of a partial RGB-D image and its corresponding estimated camera pose, thereby limiting the amount of data to be transferred and reducing the processing time. After streaming the hybrid scene representation to remote users, its static and dynamic parts are joined in a combined 3D visualization.

In the following subsections, we explain the different steps of our pipeline. Please refer to the supplemental material for more details.

3.1. Segmentation into Static and Dynamic Regions

For the sake of efficiency, we segment the RGB-D frames of the input stream into static and dynamic regions, which will later allow the efficient treatment of the different types of scene parts. For this purpose, we compute the aforementioned score maps S_k. In the following, we will ensure that these scores are normalized in the sense that a pixel is deemed static if $S_k(u) \leq 1$, and dynamic if $S_k(u) > \tau$, where $\tau \geq 1$ is a threshold that allows for a region of uncertainty between the static and dynamic labels.

Instance Segmentation To compute the dynamicity score S_k of frame k, we first detect objects in I_k using instance segmentation [60], which yields both a class label and an instance ID for each pixel in the image, i.e. $(L_k, \iota_k) = f_{\text{seg}}(I_k)$ of I_k, where $L_k(u) \in \mathbb{N}$ is the predicted class label and $\iota_k(u) \in \mathbb{N}$ is the instance ID at pixel u. The raw output of the segmentation network may consist of multiple, potentially overlapping region proposals, which we integrate into the instance and label maps using non-maximum suppression. The resulting indices are then associated with the IDs from the previous frame to get the final map ι_k of instance IDs. The label map L_k is set to the class labels corresponding to the instances. In our experiments, we used YOLOv8 [60] as the segmentation network.

Optical Flow Estimation Next, we estimate the backward optical flow $F_k = f_{\text{flow}}(I_k, I_{k-1})$, where $F_k(u) \in \mathbb{R}^2$ is the corresponding flow vector at pixel u, such that u in I_k corresponds to $u + F_k(u)$ in I_{k-1}. For f_{flow}, we use the NVIDIA Optical Flow Accelerator (NVOFA) [112]. We additionally generate a map of confidence weights $W_k(u) \in [0, 1]$ based on the agreement with the inverse flow and per-pixel costs given by the NVOFA to reduce the influence of bad correspondences.

Odometry Subsequently, we estimate the camera motion $\xi_k \in \mathfrak{se}(3)$ between the previous and current frame, yielding an absolute camera pose $T_k \in \mathbb{R}^{4\times4}$ when we assume T_1 to be centered at the world origin. Our implementation uses a standard point-to-plane RGB-D registration implementation [190].

End-point-error Based on F_k, T_k and W_k, we determine a per-pixel end-point-error E_k between the estimated flow and the flow Ψ_k we expect from a completely static scene where only the camera is moving by ξ_k, i.e.,

$$E_k(u) = W_k(u) \cdot \|F_k(u) - \Psi_k(u)\|_2. \tag{1}$$

Dynamicity Score To decide which of the resulting scores $E_k(u)$ indicate dynamic regions, we found that a simple thresholding is not sufficient, because the average error varies too strongly, especially for frame pairs where the camera tracking or optical flow network yield poor estimates. To reduce the influence of these fluctuations, we instead analyze the histogram of per-pixel errors for all pixels of each instance i. More specifically, we are selecting the rightmost mode $s_k(i)$ that is above a minimum size threshold by finding the corresponding bin index and choosing $s_k(i)$ as the center of that bin.

We normalize all scores by subtracting the smallest mode from them, assuming that at least one of the detections is of static nature. Together with an empirically chosen linear rescaling by a factor $\delta \in \mathbb{R}_{\geq 0}$, we get the normalized scores

$$E'_k(u) = \delta \cdot (E_k(u) - \min_i\{s_k(i)\}), \tag{2}$$

which fulfill the previously mentioned criterion that scores $S_k(u) \leq 1$ are indicating a static object, while higher scores indicate dynamic regions.

While $E'_k(u)$ can now be used for the segmentation into static and dynamic regions, we found the visualization of moving regions to be more coherent if the segmentation happens on the object level. This is particularly important for articulated or non-rigid objects like humans, where potentially only a small part of the object (e.g. an arm) is moving. To accomplish this, we use the normalized modes $s'_k(i)$, which result from applying the transformation from Equation (2) to $s_k(i)$. An instance i is deemed as dynamic if its normalized mode is above the dynamic threshold τ, i.e., $s'_k(i) \geq \tau$. To represent this in the resulting score map, we propagate $s'_k(i)$ in the final score map by setting $S_k(u) = s'_k(i)$ for all pixels u with $\iota_k(u) = i$, i.e., all pixels belonging to instance i. The score of each instance is temporally smoothed to be more robust against outliers.

Score Accumulation As the object tracking is only performed in 2D for efficiency reasons, we also accumulate the dynamicity scores of each instance over time in 2D by updating an accumulation map $A_k(u) \in \mathbb{R}_{\geq 0}$. To increase the interpretability of the scores, we compute a 3D end-point-error between the last and current frame by using F_k for the correspondences between the pixels and backprojecting the respective coordinates of into 3D using the corresponding depth maps and camera poses. The resulting 3D flow $\hat{F}_k(u) \in \mathbb{R}^3$ is then combined with the warped previous accumulated score A'_{k-1} to $A_k(u) = A'_{k-1}(u) + \|\hat{F}_k(u)\|_2$. We also experimented to use $\hat{F}_k$ as an input for the end-point-error calculation, but found that the signal was too noisy for our method to robustly distinguish between static and dynamic scene parts.

3.2. Update of the Static Model

With the score map S_k computed, we are able to integrate the static part of the frame into the static model. For this purpose, we use a modified version of real-time 3D reconstruction based on spatial voxel block hashing [110], with an added extension for concurrent retrieval, insertion, and removal of data [139]. However, in order to further increase the efficiency of the approach, we seamlessly shift potentially dynamic but currently static scene parts into the static scene representation until they become dynamic again. This requires us to additionally consider the following situations:

1. Dynamic regions should not be integrated into the static model. In case this happens erroneously, they should be removed as quickly as possible.
2. Regions that change their state from dynamic to static (e.g. a box was placed on a table) should be integrated into the static model seamlessly.
3. Regions changing their state from static to dynamic (e.g. a box is picked up) should be removed from the static model immediately.
4. Static regions that changed while not in the camera frustum should be updated as soon as new information is available.

Following the suggested modification of the weighting schema for dynamic object motion by Newcombe et al. [108], we truncate the updated weight, which effectively results in a moving average favoring newer measurements. We extended the schema to incorporate the previously computed dynamicity scores.This helps in situations 1 and 3, since dynamic regions are updated with new information more quickly, as well as in situation 4, as the weight is truncated even for static regions.

In addition, we aid the timely removal of dynamic regions from the static model (situations 1 and 3) by setting the SDF value to -1 for voxels where the associated dynamicity score $S_k(u)$ exceeds a threshold $\tau_{\text{SDF}} > 0$. In conjunction with the modified integration weight, this invalidates the existing surface estimate at that location.

Situation 2 is covered by the temporal smoothing of the scores. Details can be found in the supplementary material.

3.3. Visualization

After having streamed the hybrid scene representation to remote users' devices, the static and dynamic scene entities have to be combined within an immersive scene exploration component, where we focus on VR-based immersion of users into the live-captured scenarios. For this, we created a client component that receives updates of the static model as well as the dynamic regions of the current RGB-D frame.

The static model is visualized as a mesh, where the local mesh representation of the static scene is updated using received MC voxel block indices and rendered in real time, thereby following previous work [139]. In contrast, the dynamic parts are shown as a point cloud at the corresponding location relative to the static mesh. For this, we backproject the dynamic pixels of the current RGB-D frame using the known camera intrinsics and the current camera pose.

The user is then able to individually and independently of the sensor explore the captured scene by physically looking and walking around, or use a teleportation functionality for locomotion. The current position and orientation of the RGB-D sensor and other users is also shown.

3.4. Streaming

To be able to run the described method with low latency from the time of capturing to the visualization at remote locations, we use a server-client architecture. The server receives and distributes data packages over a network to the appropriate processing clients. The RGB-D capture, segmentation into static and dynamic regions as well as the integration into the static model are performed in the *reconstruction client*.

Updates of this representation are then broadcasted to one or multiple *exploration clients*, which in turn update a mesh representation of the static scene using the MC indices. At the same time, the server also sends updates of the dynamic regions as masked RGB-D images together with the current camera pose estimate, such that the RGB-D pixels can be projected into the scene as a point cloud.

For all network communication, we use a general-purpose lossless data compression scheme [22] to reduce the bandwidth requirements.

3.5. Implementation Details

To take advantage of modern multiprocessor architectures, the stages shown in Figure 2 are running concurrently, such that each stage can start with the next item once the processing of the current one has been completed. While this leads to overhead due to inter-process communication, the processing speed of the pipeline is no longer bound to the latency, but the processing duration of the slowest stage in the pipeline. We provide a more detailed, per-stage performance analysis in the supplemental material.

4. Experimental Results

To evaluate the performance of the proposed pipeline, we ran experiments on 10 self-recorded sequences captured with a Microsoft Azure Kinect RGB-D sensor in different office environments, and measured both speed and bandwidth metrics.

The scenes contain varying types of motion, and we categorized them into three groups. Fixed (F.) are scenes that have no camera motion once dynamic entities can be seen in the camera, whereas Moving (M.) describes scenes with an always-moving camera and simultaneous object motion. A third category Outside (O.) contains a scene where the camera is hand-held, but object motion only happens outside the camera view. A short description and some exemplary images of each scene are shown in the supplemental material. To validate design choices, we also conducted an ablation study regarding certain components of the pipeline and compared them to baseline methods. Following that, we will discuss the impact and limitations of the approach.

4.1. Experimental Setup

We set up three computers in a local network that each run one of the three processes shown in Figure 2. All devices use the same hardware except for the GPU, which is an Nvidia GeForce RTX 3090 for the reconstruction client and an Nvidia GeForce GTX 1080 for both server and exploration client, as they require less GPU performance.

We measured three different metrics in this setup: The end-to-end latency of an RGB-D frame from the camera to the exploration client, the frame-rate at which RGB-D frames are being processed by the components of the pipeline, and the network bandwidth between server and connected clients. The latency and frame-rate is measured using timestamped logs that are synchronized between all computers to ensure a minimal deviation. The frame-rate is given as the averaged arrival time difference between consecutive dynamic RGB-D images at the exploration client, and the latency is the average between the emission times of RGB-D frames into the pipeline and the corresponding arrival times at the exploration client.

The hyperparameters used for the performance evaluation and visualization were fixed for all scenes and are listed in the supplemental.

4.2. Evaluation of Performance and Visual Quality

Table 1 shows the results of the frame-rate and latency measurements. Here, the performance is largely independent of the type of scene and exhibits an average of around 0.4 seconds in end-to-end latency and a frame-rate of 18.8 frames per second (FPS). A closer analysis reveals that the frame-rate is upper-bound by the single image inference speed of the instance segmentation network. We refer to the supplemental for details.

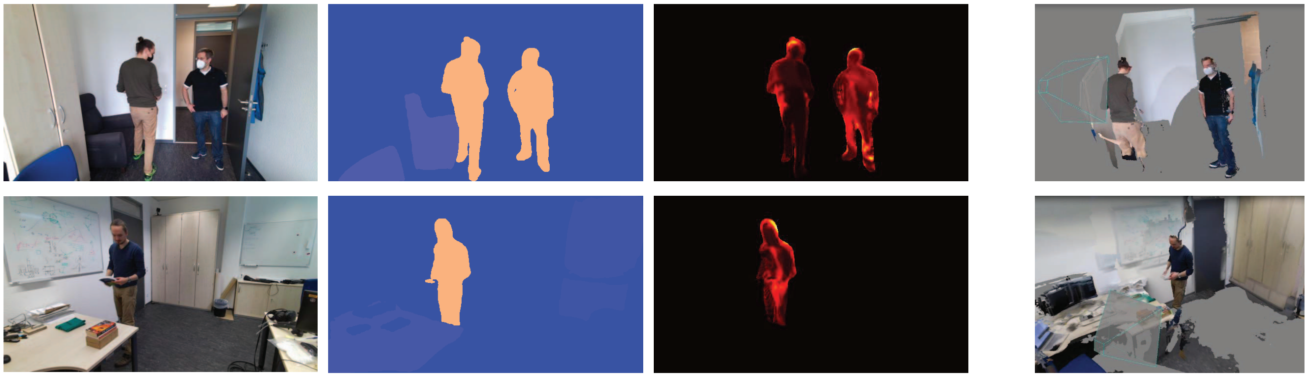

Figure 3: Results of our approach on different scenes. Left to right: Input color image; resulting segmentation into static (blue) and dynamic (yellow) regions; the accumulated 3D flow magnitude; a novel view of the scene as visualized in the exploration client.

Scene	F.	M.	O.	end-to-end [s]	FPS [1/s]
items_1	✓			0.40 (0.02)	18.95 (5.75)
items_2	✓			0.40 (0.03)	18.67 (5.95)
people_1	✓			0.39 (0.02)	19.33 (5.94)
people_2		✓		0.41 (0.03)	17.92 (5.12)
people_3		✓		0.43 (0.03)	17.73 (4.81)
people_4		✓		0.40 (0.03)	17.98 (5.34)
people_5		✓		0.40 (0.02)	19.01 (5.64)
ego_view		✓		0.40 (0.02)	18.93 (5.79)
oof_1			✓	0.40 (0.08)	19.81 (6.05)
oof_2			✓	0.40 (0.02)	19.56 (6.39)

Table 1: Performance results on the 10 self-recorded scenes. The F., M., O. columns indicate the type of motion that was captured (F: fixed camera when object motion is seen, M: camera always in motion, O: static scene manipulation outside of camera view). Latency and FPS columns show both the mean and standard deviation (in parentheses) of the respective metrics.

Type	F.	M.	O.
TSDF	44.80 (77.90)	69.72 (92.14)	65.70 (81.21)
MC	3.89 (6.13)	6.10 (7.31)	6.30 (5.83)
Dyn.	7.06 (7.37)	7.41 (8.43)	2.66 (4.42)

Table 2: Required mean bandwidth and respective standard deviation (in parentheses) in MBit/s of the different types of data packages over the types of recorded scenes (F: fixed camera when object motion is seen, M: camera always in motion, O: object motion only outside of camera view).

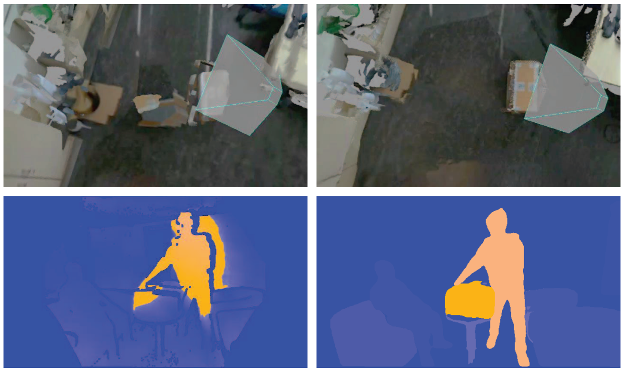

Figure 4: Comparison of design choices of the proposed pipeline. Top row: An example output from the exploration client using the standard voxel block weighting schema (left) vs. exponential weight decay via weight capping. The second approach yields a reconstruction of the box with fewer artifacts. Bottom row: Thresholding of the normalized EPE before (left) and after (right) propagation of the error modes into the static (blue) and dynamic (yellow) object masks. Again, the second approach produces a more plausible segmentation into static and dynamic regions.

The network bandwidth requirements are summarized in Table 2. Here, the measured package sizes are split up in the type of data. TSDF represents the values of the truncated signed-distance function generated by the voxel block hashing of the reconstruction client, MC labels the Marching Cubes indices the server generates from the TSDF representation and sends to the exploration client(s). The dynamic RGB-D that results from the segmentation of the reconstruction client and that is subsequently sent to the exploration client(s) is called Dyn. The results indicate that the majority of data is transferred between reconstruction client and server. The Marching Cubes indices and dynamic RGB-D data, which are selectively streamed to the exploration client(s), allow for multiple connections, even over the Internet, considering modern bandwidth availability. Furthermore, we provide qualitative results in Figure 3.

4.3. Ablation Study

To validate some design choices of our approach, we show the effects of removing certain elements of the pipeline on the results. Figure 4 illustrates the effect of the weighting function from Section 3.2, as well as the difference between error thresholding with and without propagation into the object mask (Section 3.1, Dynamicity Score).

In the weighting example, we show that the update of inconsistent measurements results in less artifacts while walking around the box when using an exponential decay. This motivates our choice to enable this weighting schema for regions with recent object motion. At the same time, the floor texture shows slightly more artifacts as the more recent measurements are favored, but collide visually with regions that were not recently seen by the camera. This effect is reduced in the original weighting schema, which motivates the extension of the schema mentioned in Section 3.2.

The bottom row of Figure 4 shows how the propagation of the error modes into the object masks aids to correctly identify potentially dynamic objects. Due to weak motion boundaries produced by f_{flow}, a large region of pixels behind the moving person is considered dynamic after normalization. This can be filtered out completely in this case using our approach.

We also conducted a performance comparison with different optical flow and instance segmentation approaches to validate our choice. The results can be found in the supplemental material.

4.4. Limitations

While our approach shows promising results and is designed with modularity and extensibility in mind, there are also some limitations to consider. Most importantly, the pipeline only runs at frame-rates close to real-time due to the performance limitations inherited by the involved neural network approaches. In our scenario, we require high single-image inference speed, which is not a functionality most modern deep learning approaches are particularly tuned for. Furthermore, our approach requires the segmentation network to detect objects to be able to identify dynamic regions, which limits its capabilities on out-of-distribution samples (Figure 5). This is also the case for the optical flow network, as it is also limited by the quality of the training data and the domain overlap with the scenes we recorded. However, due to the modular nature of our approach, future developments with improved accuracy of the predictions might address this current limitation of our approach. Furthermore, future developments on increasing the efficiency of the networks for the respectively involved subtasks will further improve the overall performance.

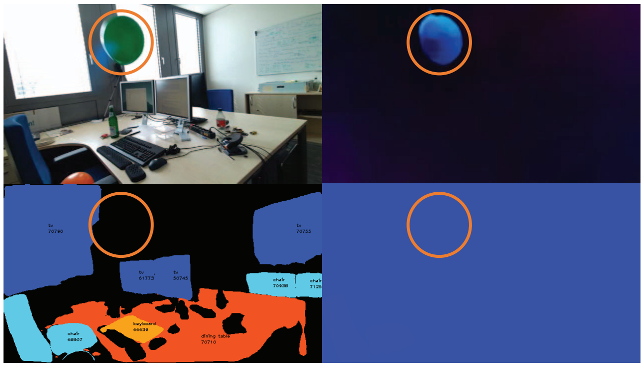

Figure 5: Failure case of our method. Shown are RGB (top left), optical flow (top right), instance segmentation (bottom left) and resulting segmentation into static and dynamic (bottom right). Even though a clear motion cue is available in the optical flow image, due to a missing object detection, our method fails to correctly identify the dynamic region (orange circle).

5. Conclusions

We presented a novel live-telepresence system that allows immersing remote users into live-captured environments with static and dynamic scene entities beyond an area of a few square-meters at practical bandwidth requirements. In order to allow the respectively required efficient 3D reconstruction, data streaming and VR-based visualization, we built our system upon a novel hybrid volumetric scene representation that combines a voxel-based representation of static scene geometry enriched by additional information regarding object semantics as well as their accumulated dynamic movement over time with a point-cloud-based representation for dynamic parts, where we perform the respective separation of static and dynamic parts based on optical flow and instance information extracted for the input frames. The separation, determined frame-by-frame on the 2D RGB-D data, remains unaffected by the length of the input sequence and scale of the scene and therefore does not impact the performance of the static reconstruction technique employed. As a result of independently yet simultaneously streaming static and dynamic scene characteristics while keeping potentially moving but currently static scene entities in the static model as long as they remain static, as well as their fusion in the visualization on remote client hardware, we achieved VR-based live-telepresence in large-scale scenarios at close to real-time rates.

With the rapid improvements in hardware technology, particularly regarding GPUs, we expect our system to soon reach full real-time capability. Also, the modularity of our system allows replacing individual components with newer approaches, which might be particularly relevant for the instance segmentation network as it represents the main bottleneck of our current system.

Acknowledgements

This work was supported by the DFG project KL 1142/11-2 (DFG Research Unit FOR 2535 Anticipating Human Behavior) and NFDI4Culture (DFG Project Number 441958017).

References

[1] Dimitrios Alexiadis, Dimitrios Zarpalas, and Petros Daras. Fast and smooth 3D reconstruction using multiple RGB-Depth sensors. In *IEEE Int. Conf. Visual Communications and Image Processing*. 2014. 3

[2] Dimitrios S Alexiadis, Dimitrios Zarpalas, and Petros Daras. Real-time, realistic full-body 3D reconstruction and texture mapping from multiple Kinects. In *IVMSP 2013*. 2013. 3

[3] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. TöRF: Time-of-flight radiance fields for dynamic scene view synthesis. *NeurIPS*, 2021. 3

[4] Tyler Bell and Song Zhang. Holo reality: Real-time low-bandwidth 3D range video communications on consumer mobile devices with application to augmented reality. *Electronic Imaging*, 2019. 2

[5] Berta Bescos, José M Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *RAL*, 2018. 3

[6] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. NoPe-NeRF: Optimising neural radiance field with no pose prior. *CoRR*, 2022. 3

[7] Gerd Bruder, Frank Steinicke, and Andreas Nüchter. Poster: Immersive point cloud virtual environments. In *3DUI*. 2014. 3

[8] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular RGB-D camera. In *NeurIPS*, 2022. 3

[9] Junli Cao, Huan Wang, Pavlo Chemerys, Vladislav Shakhrai, Ju Hu, Yun Fu, Denys Makoviichuk, Sergey Tulyakov, and Jian Ren. Real-time neural light field on mobile devices. *CoRR*, 2022. 3

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*. 2022. 3

[11] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*. 2022. 3

[12] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos. *arXiv preprint arXiv:2106.13629*, 2021. 3

[13] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*. 2022. 3

[14] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. *CoRR*, 2022. 3

[15] Yu Chen and Gim Hee Lee. DBARF: deep bundle-adjusting generalizable neural radiance fields. *CoRR*, 2023. 3

[16] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv preprint arXiv:2208.00277*, 2022. 3

[17] Zezhou Cheng, Carlos Esteves, Varun Jampani, Abhishek Kar, Subhransu Maji, and Ameesh Makadia. LU-NeRF: Scene and pose estimation by synchronizing local unposed nerfs. *CoRR*, 2023. 3

[18] Junwoo Cho, Seungtae Nam, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Streamable neural fields. In *ECCV*. 2022. 3

[19] SungIk Cho, Seung-wook Kim, JongMin Lee, JeongHyeon Ahn, and JungHyun Han. Effects of volumetric capture avatars on social presence in immersive virtual environments. In *IEEE VR*. 2020. 2

[20] Chi-Ming Chung, Yang-Che Tseng, Ya-Ching Hsu, Xiang Qian Shi, Yun-Hung Hua, Jia-Fong Yeh, Wen-Chin Chen, Yi-Ting Chen, and Winston H. Hsu. Orbeez-SLAM: A real-time monocular visual SLAM with ORB features and nerf-realized mapping. *CoRR*, 2022. 3

[21] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *TOG*, 2015. 2

[22] Y. Collet and C. Turner. Smaller and faster data compression with Zstandard. https://engineering.fb.com/2016/08/31/core-data/smaller-and-faster-data-compression-with-zstandard/, 2016. Accessed: 2023-07-19. 6

[23] Diana-Margarita Córdova-Esparza, Juan R Terven, Hugo Jiménez-Hernández, Ana Herrera-Navarro, Alberto Vázquez-Cervantes, and Juan-M García-Huerta. Low-bandwidth 3D visual telepresence system. *Multimedia Tools and Applications*, 2019. 2

[24] Linyan Cui and Chaowei Ma. SOF-SLAM: A semantic visual SLAM for dynamic environments. *IEEE Access*, 2019. 3

[25] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. DeepFactors: Real-time probabilistic dense monocular SLAM. *RAL*, 2020. 3

[26] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*. 2022. 3

[27] Mingsong Dou, Sameh Khamis, Yury Degtyarev, et al. Fusion4D: Real-time performance capture of challenging scenes. *TOG*, 2016. 3

[28] John V Draper, David B Kaber, and John M Usher. Telepresence. *Human Factors*, 1998. 1, 2

[29] Ruofei Du, Ming Chuang, Wayne Chang, Hugues Hoppe, and Amitabh Varshney. Montage4D: interactive seamless fusion of multiview video textures. In *I3D*. 2018. 2

[30] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4D view synthesis and video processing. In *ICCV*. 2021. 3

[31] Tobias Duckworth and David J Roberts. Camera image synchronisation in multiple camera real-time 3D reconstruction of moving humans. In *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*. 2011. 3

[32] Jörg Edelmann, Peter Gerjets, Philipp Mock, Andreas Schilling, and Wolfgang Strasser. Face2Face — a system

for multi-touch collaboration with telepresence. In *IEEE International Conference on Emerging Signal Processing Applications*. 2012. 2

[33] Fazliaty Edora Fadzli and Ajune Wanis Ismail. A robust real-time 3D reconstruction method for mixed reality telepresence. *International Journal of Innovative Computing*, 2020. 2

[34] A. J. Fairchild, S. P. Campion, A. S. García, R. Wolff, T. Fernando, and D. J. Roberts. A mixed reality telepresence system for collaborative space operation. *IEEE Trans. on Circuits and Systems for Video Technology*, 2016. 2

[35] Yingchun Fan, Hong Han, Yuliang Tang, and Tao Zhi. Dynamic objects elimination in SLAM based on image fusion. *Pattern Recognition Letters*, 2019. 3

[36] Yingchun Fan, Qichi Zhang, Yuliang Tang, Shaofeng Liu, and Hong Han. Blitz-slam: A semantic SLAM in dynamic environments. *Pattern Recognit.*, 2022. 3

[37] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia*. 2022. 3

[38] G. Fontaine. The experience of a sense of presence in intercultural and int. encounters. *Presence: Teleoper. Virtual Environ.*, 1992. 1, 2

[39] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*. 2022. 3

[40] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic NeRF: 3D-to-2D label transfer for panoptic urban scene segmentation. *arXiv preprint arXiv:2203.15224*, 2022. 3

[41] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and Takeo Kanade. Virtual space teleconferencing using a sea of cameras. In *Proc. of the Int. Conf. on Medical Robotics and Computer Assisted Surgery*, 1994. 2

[42] H. Fuchs, A. State, and J. Bazin. Immersive 3D telepresence. *Computer*, 2014. 2

[43] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *CVPR*. 2021. 3

[44] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*. 2021. 3

[45] Simon J Gibbs, Constantin Arapis, and Christian J Breiteneder. Teleport–towards immersive copresence. *Multimedia Systems*, 1999. 2

[46] Scott W Greenwald, Wiley Corning, Gavin McDowell, Pattie Maes, and John Belcher. ElectroVR: An electrostatic playground for collaborative, simulation-based exploratory learning in immersive virtual reality. In *International Conference on Computer Supported Collaborative Learning (CSCL)*. 2019. 2

[47] Markus Gross, Stephan Würmlin, Martin Naef, et al. blue-c: a spatially immersive display and 3D video portal for telepresence. *TOG*, 2003. 2

[48] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using L0 regularization. In *ICCV*. 2015. 3

[49] Richard Held. Telepresence. *The Journal of the Acoustical Society of America*, 1992. 1, 2

[50] Mina Henein, Jun Zhang, Robert Mahony, and Viorela Ila. Dynamic SLAM: The need for speed. In *ICRA*. 2020. 3

[51] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental robotics*. 2014. 3

[52] Hwan Heo, Taekyung Kim, Jiyoung Lee, Jaewon Lee, Soohyun Kim, Hyunwoo J. Kim, and Jin-Hwa Kim. Robust camera pose refinement for multi-resolution hash encoding. *CoRR*, 2023. 3

[53] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *ECCV*. 2016. 3

[54] ABM Islam, Christian Scheel, Ali Shariq Imran, and Oliver Staadt. Fast and accurate 3D reproduction of a remote collaboration environment. In *International Conference on Virtual, Augmented and Mixed Reality*. 2014. 3

[55] Shahram Izadi, David Kim, Otmar Hilliges, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *UIST*. 2011. 2, 3

[56] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez, and Daniel Cremers. Fast odometry and scene flow from RGB-D cameras based on geometric clustering. In *ICRA*. 2017. 3

[57] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*. 2021. 3

[58] Yifan Jiang, Peter Hedman, Ben Mildenhall, Dejia Xu, Jonathan T. Barron, Zhangyang Wang, and Tianfan Xue. AligNeRF: High-fidelity neural radiance fields via alignment-aware training. *CoRR*, 2022. 3

[59] Michal Joachimczak, Juan Liu, and Hiroshi Ando. Real-time mixed-reality telepresence via 3D reconstruction with HoloLens and commodity depth sensors. In *Proc. of the ACM International Conference on Multimodal Interaction*. 2017. 2

[60] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLOv8: The state-of-the-art YOLO model. `https://github.com/ultralytics/ultralytics`, 2023. Accessed: 2023-07-19. 4

[61] Andrew Jones, Magnus Lang, Graham Fyffe, Xueming Yu, Jay Busch, Ian McDowall, Mark Bolas, and Paul Debevec. Achieving eye contact in a one-to-many 3D video teleconferencing system. *TOG*, 2009. 2

[62] Brett Jones, Rajinder Sodhi, Michael Murdock, et al. RoomAlive: Magical experiences enabled by scalable, adaptive projector-camera units. In *UIST*. 2014. 2

[63] Kim Jun-Seong, Kim Yu-Ji, Moon Ye-Bin, and Tae-Hyun Oh. HDR-Plenoxels: Self-calibrating high dynamic range

radiance fields. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII*. 2022. 3

[64] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia*, 1997. 2

[65] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3D reconstruction in dynamic scenes using point-based fusion. In *3DV*. 2013. 3

[66] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *NeurIPS*, 2017. 3

[67] Deok-Hwa Kim and Jong-Hwan Kim. Effective background model-based RGB-D dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 2016. 3

[68] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning SfM from SfM. In *ECCV*. 2018. 3

[69] Ryohei Komiyama, Takashi Miyaki, and Jun Rekimoto. JackIn space: designing a seamless transition between first and third person view for effective telepresence collaborations. In *Proc. of the Augmented Human International Conference*. 2017. 2

[70] Dennis Krupke, Sebastian Starke, Lasse Einig, J Zhang, and F Steinicke. Prototyping of immersive HRI scenarios. In *CLAWAR*. 2018. 3

[71] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3D environment for remote collaboration and training of physical activities. In *IEEE VR*. 2008. 2

[72] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*. 2017. 3

[73] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. Hierarchical voxel block hashing for efficient integration of depth images. *RAL*, 2015. 3

[74] Olaf Kähler, Victor A Prisacariu, and David W Murray. Real-time large-scale dense 3D reconstruction with loop closure. In *ECCV*. 2016. 3

[75] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *TVCG*, 2015. 3

[76] Jason Lawrence, Dan B Goldman, Supreeth Achar, et al. Project Starline: A high-fidelity telepresence system. *TOG*, 2021. 2

[77] Ao Li, Jikai Wang, Meng Xu, and Zonghai Chen. DP-SLAM: A visual SLAM with moving probability towards dynamic environments. *Inf. Sci.*, 2021. 3

[78] Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, and Szymon Rusinkiewicz. Temporally coherent completion of dynamic shapes. *TOG*, 2012. 3

[79] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. In *NeurIPS*, 2022. 3

[80] Shile Li and Dongheui Lee. RGB-D SLAM in dynamic environments using static point weighting. *RAL*, 2017. 3

[81] Tianye Li, Mira Slavcheva, Michael Zollhoefer, et al. Neural 3D video synthesis from multi-view video. In *CVPR*. 2022. 3

[82] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*. 2021. 3

[83] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DynIBaR: neural dynamic image-based rendering. *CoRR*, 2022. 3

[84] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BaRF: Bundle-adjusting neural radiance fields. In *ICCV*. 2021. 3

[85] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *RAL*, 2017. 3

[86] Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. NeRF-Loc: Visual localization with conditional neural radiance field. *CoRR*, 2023. 3

[87] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *TOG*, 2021. 3

[88] C. Loop, C. Zhang, and Z. Zhang. Real-time high-resolution sparse voxelization with application to image-based modeling. In *Proc. of the High-Performance Graphics Conference*. 2013. 2

[89] Xinzhong Lu, Ju Shen, Saverio Perugini, and Jianjun Yang. An immersive telepresence system using RGB-D sensors and head mounted display. In *IEEE International Symposium on Multimedia (ISM)*. 2015. 2

[90] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-NeRF: Monte carlo localization using neural radiance fields. *ICRA*, 2023. 3

[91] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs. Enhanced personal autostereoscopic telepresence system using commodity depth cameras. *Computers & Graphics*, 2012. 2

[92] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras. In *ISMAR*. 2011. 2

[93] A. Maimone and H. Fuchs. Real-time volumetric 3D capture of room-sized scenes for telepresence. In *3DTV*. 2012. 2

[94] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*. 2021. 3

[95] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. GNeRF: GAN-based neural radiance field without posed camera. In *ICCV*. 2021. 3

[96] Zhenxing Mi and Dan Xu. Switch-NeRF: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *ICLR*. 2023. 3

[97] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF:

Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 3

[98] Marvin Minsky. Telepresence. *Omni*, 1980. 1, 2

[99] D. Molyneaux, S. Izadi, D. Kim, O. Hilliges, S. Hodges, X. Cao, A. Butler, and H. Gellersen. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Proc. of the Int. Conf. on Pervasive Computing*. 2012. 2

[100] Carl Moore, Toby Duckworth, Rob Aspin, and David Roberts. Synchronization of images from multiple cameras to reconstruct a moving human. In *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*. 2010. 3

[101] A. Mossel and M. Kröter. Streaming and exploration of dynamically changing dense 3D reconstructions in immersive virtual reality. In *ISMAR*. 2016. 2, 3

[102] Muhammad Husnain Mubarik, Ramakrishna Kanungo, Tobias Zirr, and Rakesh Kumar. Hardware acceleration of neural graphics. In *ISCA*. 2023. 3

[103] Lothar Muhlbach, Martin Bocker, and Angela Prussog. Telepresence in videocommunications: A study on stereoscopy and individual eye contact. *Human Factors*, 1995. 2

[104] J. Mulligan and K. Daniilidis. View-independent scene acquisition for tele-presence. In *Proc. IEEE and ACM Int. Symp. on Augmented Reality*. 2000. 2

[105] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 2022. 3

[106] Abdeldjallil Naceri, Dario Mazzanti, Joao Bimbo, Yonas T Tefera, Domenico Prattichizzo, Darwin G Caldwell, Leonardo S Mattos, and Nikhil Deshpande. The vicarios virtual reality interface for remote robotic teleoperation. *Journal of Intelligent & Robotic Systems*, 2021. 3

[107] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*. 2015. 3

[108] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, et al. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*. 2011. 3, 5

[109] Viet Anh Nguyen, Jiangbo Lu, Shengkui Zhao, Dung T Vu, Hongsheng Yang, Douglas L Jones, and Minh N Do. ITEM: Immersive telepresence for entertainment and meetings—a practical approach. *IEEE Journal of Selected Topics in Signal Processing*, 2014. 2

[110] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *TOG*, 2013. 3, 5

[111] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*. 2021. 3

[112] Nvidia Corporation. Nvidia optical flow sdk. `https://developer.nvidia.com/opticalflow-sdk`, 2019. Accessed: 2023-07-19. 4

[113] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, et al. Holoportation: Virtual 3D teleportation in real-time. In *UIST*. 2016. 2

[114] Viken Parikh and Mansi Khara. A mixed reality workspace using telepresence system. In *International Conference on ISMAC in Computational Vision and Bio-Engineering*. 2018. 2

[115] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*. 2021. 3

[116] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *TOG*, 2021. 3

[117] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. Room2Room: Enabling life-size telepresence in a projected augmented reality environment. In *Proc. of the ACM conference on computer-supported cooperative work & social computing*. 2016. 2

[118] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*. 2021. 3

[119] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*. 2021. 3

[120] Lorenzo Peppoloni, Filippo Brizzi, Carlo Alberto Avizzano, and Emanuele Ruffaldi. Immersive ROS-integrated framework for robot teleoperation. In *3DUI*. 2015. 3

[121] B. Petit, J.-D. Lesage, C. Menier, J. Allard, J.-S. Franco, B. Raffin, E. Boyer, and F. Faure. Multicamera real-time 3D modeling for telepresence and remote collaboration. *Int. Journal of Digital Multimedia Broadcasting*, 2010. 2

[122] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip HS Torr, and David W Murray. InfiniTAM v3: A framework for large-scale 3D reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017. 3

[123] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*. 2021. 3

[124] Amit Raj, Michael Zollhöfer, Tomas Simon, Jason M. Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *CVPR*. 2021. 3

[125] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In *ICCV*. 2021. 3

[126] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*. 2022. 3

[127] David J Roberts, Allen J Fairchild, Simon P Campion, John O'Hare, Carl M Moore, Rob Aspin, Tobias Duckworth, Paolo Gasparello, and Franco Tecchia. withyou — an experimental end-to-end telepresence system using video-

based reconstruction. *IEEE Journal of Selected Topics in Signal Processing*, 2015. 2

[128] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*. 2022. 3

[129] Eric Rosen, David Whitney, Michael Fishman, Daniel Ullman, and Stefanie Tellex. Mixed reality as a bidirectional communication interface for human-robot interaction. In *IROS*. 2020. 3

[130] Antoni Rosinol, John J Leonard, and Luca Carlone. NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 3

[131] Martin Runz, Maud Buffier, and Lourdes Agapito. Mask-Fusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *ISMAR*. 2018. 3

[132] Martin Rünz and Lourdes Agapito. Co-Fusion: Real-time segmentation, tracking and fusion of multiple objects. In *ICRA*. 2017. 3

[133] David W Schloerb. A quantitative measure of telepresence. *Presence: Teleoperators & Virtual Environments*, 1995. 2

[134] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments. In *ICRA*. 2018. 3

[135] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. KillingFusion: Non-rigid 3D reconstruction without correspondences. In *CVPR*. 2017. 3

[136] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. SobolevFusion: 3D reconstruction of scenes undergoing free non-rigid motion. In *CVPR*. 2018. 3

[137] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. BeThere: 3D mobile collaboration with spatial input. In *Proc. of CHI Conf. Hum. Fac. Comput. Syst.* 2013. 2

[138] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 1992. 2

[139] Patrick Stotko, Stefan Krumpen, Matthias B. Hullin, Michael Weinmann, and Reinhard Klein. SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence. *TVCG*, 2019. 2, 3, 4, 5, 6

[140] Patrick Stotko, Stefan Krumpen, Reinhard Klein, and Michael Weinmann. Towards scalable sharing of immersive live telepresence experiences beyond room-scale based on efficient real-time 3D reconstruction and streaming. In *CVPR Workshop*, 2019. 2, 3

[141] Patrick Stotko, Stefan Krumpen, Max Schwarz, Christian Lenz, Sven Behnke, Reinhard Klein, and Michael Weinmann. A VR system for immersive teleoperation and live exploration with a mobile robot. In *IROS*. 2019. 2, 3

[142] Patrick Stotko, Stefan Krumpen, Michael Weinmann, and Reinhard Klein. Efficient 3D reconstruction and streaming for group-scale multi-client live telepresence. In *ISMAR*. 2019. 2, 3, 4

[143] Michael Strecke and Jorg Stuckler. EM-Fusion: Dynamic object-level slam with probabilistic data association. In *ICCV*. 2019. 3

[144] Jörg Stückler and Sven Behnke. Efficient dense rigid-body motion segmentation and estimation in RGB-D video. *IJCV*, 2015. 3

[145] Po-Chang Su, Ju Shen, and Muhammad Usman Rafique. RGB-D camera network calibration and streaming for 3D telepresence in large environment. In *BigMM*. 2017. 2

[146] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*. 2021. 3

[147] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*. 2022. 3

[148] Jiankai Sun, Yan Xu, Mingyu Ding, Hongwei Yi, Chen Wang, Jingdong Wang, Liangjun Zhang, and Mac Schwager. NeRF-Loc: Transformer-based object localization within neural radiance fields. *RAL*, 2023. 3

[149] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *SIGGRAPH*. 2022. 3

[150] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *CVPR*. 2022. 3

[151] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben P. Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *CVPR*. 2022. 3

[152] T. Tanikawa, Y. Suzuki, K. Hirota, and M. Hirose. Real world video avatar: Real-time and real-size transmission and presentation of human figure. In *ICAT*. 2005. 2

[153] Theophilus Teo, Louise Lawrence, Gun A Lee, Mark Billinghurst, and Matt Adcock. Mixed reality remote collaboration combining 360 video and 3D reconstruction. In *Proc. of CHI Conf. Hum. Fac. Comput. Syst.* 2019. 2

[154] Ayush Tewari, Ohad Fried, Justus Thies, et al. State of the art on neural rendering. In *CGF*. 2020. 3

[155] Ayush Tewari, Justus Thies, Ben Mildenhall, et al. Advances in neural rendering. In *CGF*. 2022. 3

[156] Michail Theofanidis, Saif Iftekar Sayed, Alexandros Lioulemes, and Fillia Makedon. Varm: Using virtual reality to program robotic manipulators. In *PETRA*. 2017. 3

[157] H. Towles, W. Chen, R. Yang, S. Kum, H. Fuchs, N. Kelshikar, J. Mulligan, K. Daniilidis, C. C. Hill, L. Holden, B. Zeleznik, A. Sadagic, and J. Lanier. 3D telecollaboration over internet2. In *Proc. of the Int. Workshop on Immersive Telepresence*, 2002. 2

[158] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*. 2021. 3

[159] Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. DEMEA: deep

mesh autoencoders for non-rigidly deforming objects. In *ECCV*. 2020. 3

[160] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. CLA-NeRF: Category-level articulated neural radiance field. *arXiv preprint arXiv:2202.00181*, 2022. 3

[161] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable construction of large-scale NeRFs for virtual fly-throughs. In *CVPR*. 2022. 3

[162] R. Vasudevan, G. Kurillo, E. Lobaton, T. Bernardin, O. Kreylos, R. Bajcsy, and K. Nahrstedt. High-quality visualization for geographically distributed 3-D teleimmersive applications. *IEEE Trans. on Multimedia*, 2011. 2

[163] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. NeSF: Neural semantic fields for generalizable semantic segmentation of 3D scenes. *arXiv preprint arXiv:2111.13260*, 2021. 3

[164] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *European Conference on Computer Vision*. 2022. 3

[165] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

[166] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*. 2021. 3

[167] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *International Journal of Robotics Research*, 2016. 3

[168] David Whitney, Eric Rosen, Daniel Ullman, Elizabeth Phillips, and Stefanie Tellex. ROS reality: A virtual reality framework using consumer-grade hardware for ROS-enabled robots. In *IROS*. 2018. 3

[169] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *CVPR*. 2021. 3

[170] B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper. Virtual Environ.*, 1998. 1, 2

[171] Wenxin Wu, Liang Guo, Hongli Gao, Zhichao You, Yuekai Liu, and Zhiqiang Chen. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint. *Neural Comput. Appl.*, 2022. 3

[172] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *CVPR*. 2017. 3

[173] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. SiNeRF: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. In *BMVC*. 2022. 3

[174] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*. 2021. 3

[175] Linhui Xiao, Jinge Wang, Xiaosong Qiu, Zheng Rong, and Xudong Zou. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics Auton. Syst.*, 2019. 3

[176] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. MidFusion: Octree-based object-level multi-instance dynamic SLAM. In *ICRA*. 2019. 3

[177] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*. 2020. 3

[178] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *ECCV*. 2018. 3

[179] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. BakedSDF: Meshing neural sdfs for real-time view synthesis. In *SIGGRAPH*. 2023. 3

[180] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*. 2014. 3

[181] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IROS*. 2021. 3

[182] Jacob Young, Tobias Langlotz, Steven Mills, and Holger Regenbrecht. Mobileportation: Nomadic telepresence for mobile devices. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2020. 3

[183] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. DS-SLAM: A semantic visual SLAM towards dynamic environments. In *IROS*. 2018. 3

[184] Hao Zhang and Feng Xu. MixedFusion: Real-time reconstruction of an indoor scene with dynamic objects. *TVCG*, 2017. 3

[185] Shujun Zhang and Wan Ching Ho. Tele-immersive interaction with intelligent virtual agents based on real-time 3D modeling. *Journal of Multimedia*, 2012. 2

[186] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. FlowFusion: Dynamic dense RGB-D SLAM based on optical flow. In *ICRA*. 2020. 3

[187] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. NeRFusion: Fusing radiance fields for large-scale scene reconstruction. In *CVPR*, 2022. 3

[188] Xiaoshuai Zhang, Abhijit Kundu, Thomas A. Funkhouser, Leonidas J. Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3D scene representation from 2D supervision. *CoRR*, 2023. 3

[189] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*. 2021. 3

[190] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 5

[191] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. NICER-SLAM: Neural implicit scene encoding for RGB SLAM. *arXiv preprint arXiv:2302.03594*, 2023. 3

[192] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-SLAM: Neural implicit scalable encoding for SLAM. In *CVPR*. 2022. 3

[193] Domenic Zingsheim, Patrick Stotko, Stefan Krumpen, Michael Weinmann, and Reinhard Klein. Collaborative VR-based 3D labeling of live-captured scenes by remote users. *IEEE Computer Graphics and Applications*, 2021. 3

[194] Nikolaos Zioulis, Dimitrios Alexiadis, Alexandros Doumanoglou, Georgios Louizis, Konstantinos Apostolakis, Dimitrios Zarpalas, and Petros Daras. 3D tele-immersion platform for interactive immersive experiences between remote users. In *ICIP*. 2016. 2