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[2]

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

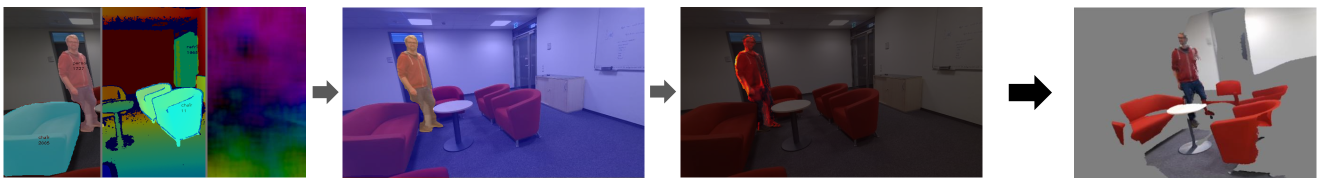


Figure 1. Visualization of the key components of our proposed pipeline. Color and depth images are blended with class and instance information, and shown along with the optical flow w.r.t. to the previous frame (first image). This information is integrated to produce a mask that segments the frame into 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).

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 squaremeters 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 interactive 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 from 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 [20, 28, 41, 84, 152]. 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 maintenance/consulting 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 save time but also facilitates economically less well-situated or handicapped people to access such 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 [117, 122]. 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 [20, 28, 41, 84, 117, 122, 152]. Furthermore, experiencing 3D depth cues like stereopsis, motion parallax, and natural scale also contribute to the perceived level of immersion and copresence [35, 88].

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 room scale using special expensive static capture setups [14, 16, 21, 26, 37, 52, 64, 68, 78, 79, 97, 98, 112, 129, 144, 170] and display technology [68], as well as for static scenes beyond room scale based on low-cost and light-weight incremental scene capture with a moving depth camera [87, 123–126]. For the latter category, bandwidth requirements have been reduced from hundreds of MBit/s for a single user [87] to around 15MBit/s for group-scale sharing of telepresence in live-captured environments while also handling network interruptions [123, 124, 126], thereby even allowing live-teleoperation of robots [125]. 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 as depicted in Figure 1.

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 interactive 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 [123, 125, 126].

2. Related Work

Telepresence Systems Despite almost two decades of developments, 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 [31, 58, 66, 90, 134, 140], inaccurate silhouette-based reconstruction techniques [76, 105]. Depth-based 3D scanning led to improved reconstruction quality and allowed telepresence at room [32, 47, 54, 78, 80, 85], 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 visualozation technology led to impressive immersive AR/VR-based live-3D-telepresence experiences. Live-telepresence for small-scale scenarios of a few squaremeters has been achieved based on light-weight capture setups for teleconferencing [4, 13, 24, 53, 94, 101] and other collaborative scenarios [25, 36, 77, 121, 136, 163] as well as based on expensive multi-camera static and pre-calibrated capture setups [14, 16, 21, 26, 37, 52, 64, 68, 78, 79, 97, 98, 112, 129, 144, 170]. Furthermore, live-telepresence for scenarios beyond room scale has been achieved based on low-cost and light-weight incremental scene capture with a moving depth camera [6, 87, 123–126, 160], allowing remote users to immersively explore a live-captured environment independent from the sensor configurations. Regarding the latter approaches, impractical bandwidth requirements of up to 175MBit/s for immersive scene exploration by a single user [87] have been overcome by more recent approaches that allow group-scale sharing of telepresence experiences in live-captured environments and handling network interruptions [123–126] as well as live-teleoperation of robots [125]. Furthermore, mechanisms for annotation, distance measurement [125] and efficient collaborative VR-based 3D labeling were added [169]. 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 [65,74,91,104,114,125,139,150] and large-scale, static scenarios [125].

In contrast to the aforementioned approaches, we propose a live-telepresence system for large-scale environments beyond room-scale and including scene dynamics.

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 [43] or extensions of KinectFusion [48, 93] in terms of voxel block hashing techniques [55–57, 95, 106] for incremental scene capture for large-scale telepresence applications [87, 123–126]. To avoid the need for depth sensors, more recent Simultaneous Localization and Mapping (SLAM) approaches for incremental scene capture - that might be applicable in respective telepresence applications - leveraged principles of deep learning [17,60,63,67,151,156,157]. Further approaches investigated 3d reconstruction from multiple synchronized cameras [1,2,23,46,86].

Recently, neural scene representation and rendering techniques [137,138] 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) [83] and respective extensions towards speeding up model training [7,11,18,29,89,110,132] with training times of seconds, the adaptation to unconstrained image collections [10, 81], deformable scenes [8, 33, 75, 96, 99, 100, 102, 107, 109, 141, 142] and video inputs [22, 34, 71, 72, 103, 154], the refinement or complete estimation of camera pose parameters for the input images [50,73,82,130,131,147,159,168], combining NeRFs with semantics regarding objects in the scene [30, 145, 166], incorporating depth cues [3, 18, 111, 113, 148] to guide the training and allow handling textureless regions, handling large-scale scenarios [133, 143], and streamable representations [12]. However, despite promising results, further improvements regarding efficiency are required for the joint camera pose estimation and neural scene reconstruction [131, 168] as required in a SLAM setting to achieve beyond the reported 5 fps on a current high-quality GPU (Nvidia RTX 3090) [168] while also reducing the jittering of the depicted scene during exploration.

Particularly addressing dynamic environments, various approaches focused on filtering dynamic objects and only reconstructing the static background [5, 27, 61, 118, 155, 161,164] or additionally reconstructing the dynamics based on rigid object tracking and reconstruction [42, 70, 115, 116,128,153] and non-rigid object tracking and reconstruction [19, 38, 45, 49, 59, 69, 92, 119, 120, 127, 149, 149, 158, 162]. 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 squaremeters) 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 interactive 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 efficiency when processing these measurements, integrating them into a consistent model and streaming the latter over the network at practical bandwidth requirements to remote clients, where it has to be visualized at adequate visual quality 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

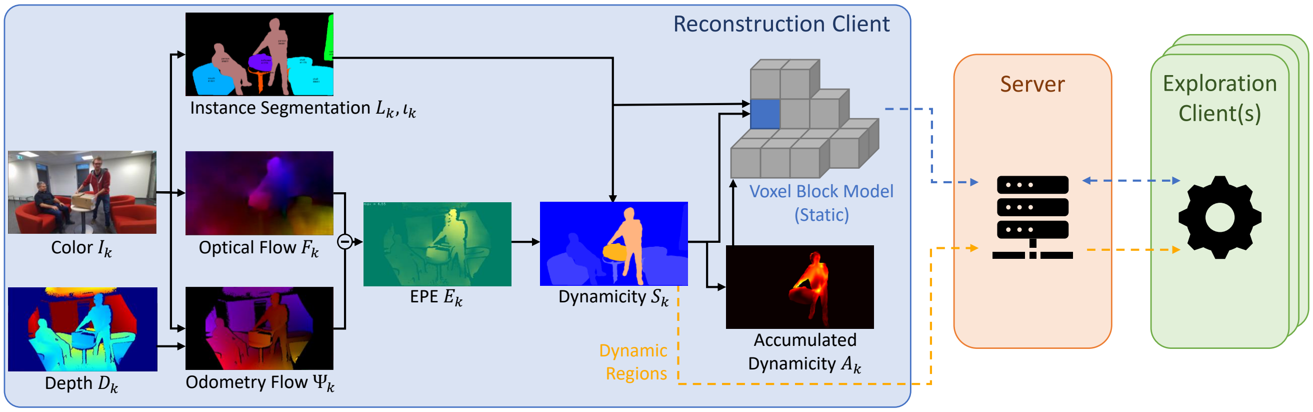


Figure 2. Visualization of different processing stages for the k-th RGB-D frame in the pipeline. Starting with color I_k and depth D_k, instance segmentation L_k (class labels) and ι_k (instance IDs), optical flow F_k and odometry flow Ψ_k are computed. Next the end-point-error (EPE) between the flows 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.

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 [123, 126] 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 provide more details on the different steps of our pipeline.

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 score maps S_k, where $S_k(u) \in \mathbb{R}$ describes the amount of dynamicity in frame k at pixel u. In the following, we will assume 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. To compute the dynamicity score S_k of frame k, we first detect objects in I_k using instance segmentation, 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 consists of multiple, potentially overlapping region proposals which we integrate into the instance and labels maps using non-maximum suppression. Pixels without any proposal, belong to a low confidence detection or to a region with pixel count below a certain threshold are ignored by setting $L_k(u) = \iota_k(u) = 0$. See the supplemental material for a detailed explanation of this procedure. In our experiments, we used SoloV2 [146] using pretrained weights from [9].

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 used a pretrained version [15] of LiteFlowNet2 [44].

Subsequently, we estimate the camera motion

$$\xi_k = f_{\text{pose}}(I_{k-1}, I_k, D_{k-1}, D_k) \in \mathfrak{se}(3) \tag{1}$$

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 of Open3D [167].

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

$$E_k(u) = \|F_k(u) - \Psi_{T_k}(u)\|_2. \tag{2}$$

Using D_k, we can compute $\Psi_{T_k}(u)$ as the offset between u and the corresponding point u' projected from frame $k-1$ into k. More specifically, let π^{-1} be the backprojection operation of some fixed pinhole camera, such that $v = \pi^{-1}(u, D_k(u)) \in \mathbb{R}^3$ is the 3D coordinate of pixel u

with depth measurement $D_k(u)$ in the local coordinate system the camera, and let π be the corresponding projection operation transforming v back to u. $\Psi_k(u)$ is then given as

$$\Psi_k(u) = [\pi \circ T_{k-1} \circ T_k^{-1} \circ \pi^{-1}(u, D_k(u))] - u. \quad (3)$$

Note that we imply proper conversion between euclidean and homogeneous coordinates by using the concatenation operator to keep the notation simple.

To decide which of the resulting scores $E_k(u)$ indicates dynamic regions, we analyze the histogram $H^i = (H_1^i, ..., H_n^i) \in \mathbb{N}^n$ of errors for each instance i. We chose the width of the n histogram bins empirically as $c = 0.25$ based on the error values produced by our approach. As an indicator of the highest motion of i, we look for the rightmost mode $s_k(i) \in \mathbb{R}_{\geq 0}$ of H^i that consists of at least $r \cdot \sum_{l=1}^{n} H_l^i$ values, where $r \in [0, 1]$ is a hyperparameter. This means, we look for the bin index $j^*(i)$ with

$$j^*(i) = \max \left\{ j \;\middle|\; H_{j-1}^i < H_j^i, H_{j+1}^i < H_j^i, \frac{H_j^i}{\sum_l H_l^i} \geq r \right\} \quad (4)$$

which, in turn, allows the mode m_i to be defined as the center of histogram bin $j^*(i)$, i.e. $s_k(i) = (j^*(i) + 0.5)c$.

We normalize all scores by subtracting the smallest mode from them, assuming that at least one of the detections is of static nature. This is done to remove shifts in the overall error that can be caused by inaccurate estimates produced in f_{flow} and f_{pose}. Together with an empirically chosen linear rescaling by a factor $\delta \in \mathbb{R}_{\geq 0}$, we get the normalized scores

$$E'_k(u) = \delta \cdot (E_k(u) - \min_i \{s_k(i)\}) \quad (5)$$

that fulfill the previously mentioned criterion that scores ≤ 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 that the visualization of moving regions is 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 (5)) to m_i. An instance i is deemed as dynamic if $s'_k(i) \geq \tau$. To represent this in the resulting score map, we propagate this value in the final score map by setting $S_k(u) = s'_k(i)$ for all pixels u with $\iota_k(u) = i$.

To make the dynamicity estimates more robust against noise in the error values when looking at multiple frames, we experimented with smoothing the values $s'_k(i)$ temporally using the maximum over the current and a decaying previous score, such that the smoothed score of instance i in frame k is given as

$$\hat{s}'_k(i) := \max\{\alpha \cdot \hat{s}'_{k-1}(i), s'_k(i)\}. \quad (6)$$

To make this work, we have to re-identify instance i from frame $k-1$ in frame k. A priori, instance IDs do not have any relation to each other, because f_{seg} is assumed to only be dependent on a single image. We use information about mask overlap between ι_k, L_k and ι'_{k-1}, L'_{k-1} , where latter maps result from warping ι_{k-1}, L_{k-1} according to flow F_k, aligning them with the maps of frame k. A confusion matrix C of the pairwise overlaps of the instance masks of the same class in ι_k and ι'_{k-1} is computed, such that

$$C_{ij} = \left| \{u \mid \iota_k(u) = i, \iota'_{k-1}(u) = j, L_k(u) = L'_{k-1}(u)\} \right|. \quad (7)$$

We identify instance i with instance j' from the previous frame, if $j' = \mathrm{argmax}_j\{C_{ij}\}$ and $C_{ij'}$ is larger than a minimum overlap count. In addition, we also keep track of the average dynamicity scores over time to be able to give a sensible initial score estimate when detecting a new instance. A detailed explanation of this initialization scheme can be found in the supplemental material.

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 last and current frame by using F_k for the correspondences between the pixels and unprojecting the respective coordinates of into 3D using π^{-1} with 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} as $A_k(u) = A'_{k-1}(u) + \|\hat{F}_k(u)\|_2$.

3.2. Updating 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 [95], where we added an extension for concurrent retrieval, insertion and removal of data [123]. However, in order to further increase the efficiency of our 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., we truncate the updated weight which effectively results in a moving average favoring newer measurements [93]. However, instead of having a global maximum weight $W_\eta > 0$, we store a separate value $W_{\eta,k}(v) > 0$ for each voxel v in our model and compute the new weight $W_k(p) \in \mathbb{R}_{\geq 0}$ as

$$W_k(v) = \min(W_{k-1}(v) + W'_k(v), W_{\eta,k}). \quad (8)$$

These maximum weights are computed directly from the dynamicity score. We found that a simple step function suffices, i.e.

$$W_{\eta,k}(v) = \begin{cases} \hat{W}_\eta, & S_k(u_v) \leq \tau_\eta \\ \check{W}_\eta, & S_k(u_v) > \tau_\eta \end{cases}, \quad (9)$$

given the corresponding raycasting source pixel $u_v \in \mathcal{U}$ of voxel v, a threshold $\tau_\eta > 0$ and weight caps $0 < \check{W}_\eta \leq \hat{W}_\eta$. 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 settings the SDF value to -1 for voxels where the associated dynamicity score $S_k(p)$ exceeds a threshold $\tau_{\text{SDF}} > 0$. This, together with the high integration weight from before, invalidates the existing surface estimate at that location.

Situation 2 is already covered by the temporal smoothing of the dynamicity scores in Equation (6), because the decay parameter α prevents to drop the scores too quickly, which leads to objects being considered dynamic for some time when they stop moving. Even though it takes a short time for the static reconstruction to integrate and stream the voxels of the state-changing object, we found this to be more intuitive when observing the live scene to have the object stop first than to suddenly disappear.

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 virtual reality (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 [123]. 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 known camera intrinsics and the current camera pose.

The user is then able to individually and independently from 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 capturing, 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.

3.5. Implementation Details

To take advantage of modern multi-processor architectures, the stages shown in Figure 2 are each running in separate processes, such that each stage can begin processing the next item once the current one has been processed. While this leads to overhead due to inter-process communication, the FPS of the pipeline is no longer bound to the latency, but the processing duration of the slowest stage in the pipeline. This can also be observed in our performance evaluation.

4. Experimental Results

To evaluate the performance of the proposed pipeline, we ran experiments on 8 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 of the camera view.

Scene	F.	M.	O.	Latency [s]	FPS [1/s]
items_1	✓			2.55 (0.07)	11.38 (5.51)
items_2	✓			2.41 (0.15)	10.44 (4.94)
people_1	✓			2.51 (0.06)	11.80 (5.79)
people_2		✓		2.46 (0.11)	10.14 (5.27)
people_3		✓		2.58 (0.19)	9.63 (4.81)
ego_view		✓		2.32 (0.11)	10.27 (5.53)
oof_1		✓		2.50 (0.09)	11.09 (4.93)
oof_2			✓	2.53 (0.14)	10.85 (6.15)

Table 1. Performance results on the 8 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: object motion only outside of camera view). Latency and FPS columns show both the mean and standard deviation (in parentheses) of the respective metrics.

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 merged the region proposals using a non-maximum suppression approach. Details and all hyperparameter choices can be found in the supplemental material.

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 pipeline, and the network bandwidth between server and connected clients. The latency and frame-rate is measured using timestamped logs that are synchronized via a "benchmark-start" broadcast from the server. This ensures that the timestamps do not deviate more than the local network latency. The frame-rate we report is given as the averaged arrival time difference between consecutive dynamic RGB-D images at the exploration client. 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.

4.2. Evaluation of Performance and Visual Quality

Table 1 shows the results of the frame-rate and latency measurements. It can be seen that the performance is largely independent of the type of scene and averages around 2.48 seconds in end-to-end latency and 10.7 frames per second.

Type	F.	M.	O.
TSDF	29.47 (63.42)	59.06 (95.06)	56.18 (84.14)
MC	3.20 (6.67)	5.91 (7.86)	5.51 (6.24)
Dyn.	4.89 (7.97)	2.62 (5.41)	1.27 (3.27)

Table 2. Required bandwidth in MBit/s of the different types of data packages, averaged 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). Given are both mean and standard deviation (in parentheses).

A closer analysis reveals that the frame-rate is upper-bound by the single image inference speed of the instance segmentation network. We provide respective details in the supplemental material.

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 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 show 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 of the 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 Equation (9) as well as the difference between error thresholding with and without propagation into the object masks. In the weighting example, we show that the update of false measurements is done with less artifacts while walking around the box when using a 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 dual approach shown in Equation (9).

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 be-

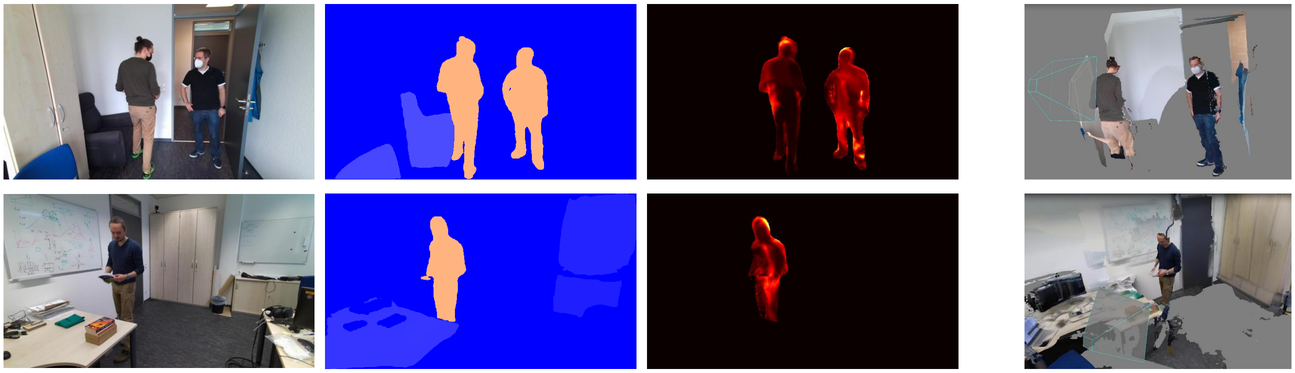


Figure 3. Results of our approach on different scenes. Left to right: Input color image; resulting segmentation into static (blue) and dynamic (yellow) regions; the accumulated 3D flow magnitude; a novel view of the scene as visualized in the exploration client.

Figure 4. Comparison of design choices of the proposed pipeline. Top row: An example output from the exploration client using the standard voxelblock weighting schema (left) vs. exponential weight decay via weight capping. The second approach yields a reconstruction of the box with less 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.

hind the moving person is considered dynamic after normalization. This can be filtered out completely in this case using our approach.

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 interactive frame-rates 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 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. 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. In the supplemental material, we show some failure cases where failed detections of both f_{seg} and $_{\text{flow}}$ cause artifacts in the static reconstruction.

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 room scale 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. 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 interactive 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, which 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).

References

[1] Dimitrios Alexiadis, Dimitrios Zarpalas, and Petros Daras. Fast and smooth 3d reconstruction using multiple rgb-depth sensors. In *2014 IEEE Visual Communications and Image Processing Conference*, pages 173–176. IEEE, 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*, pages 1–4. IEEE, 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. *Advances in neural information processing systems*, 34:26289–26301, 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(16):7–1, 2019. 2

[5] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 3

[6] Gerd Bruder, Frank Steinicke, and Andreas Nüchter. Poster: Immersive point cloud virtual environments. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 161–162, 2014. 2

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 3

[8] 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

[9] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 4

[10] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. 3

[11] 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

[12] Junwoo Cho, Seungtae Nam, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Streamable neural fields. In *European Conference on Computer Vision*, pages 595–612. Springer, 2022. 3

[13] 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 *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 26–34. IEEE, 2020. 2

[14] 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. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 2

[15] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. https://github.com/open-mmlab/mmflow, 2021. 4

[16] 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*, 78(15):21273–21290, 2019. 2

[17] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020. 3

[18] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3

[19] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 3

[20] John V Draper, David B Kaber, and John M Usher. Telepresence. *Human factors*, 40(3):354–375, 1998. 1, 2

[21] Ruofei Du, Ming Chuang, Wayne Chang, Hugues Hoppe, and Amitabh Varshney. Montage4d: interactive seamless fusion of multiview video textures. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, 2018. 2

[22] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021. 3

[23] Tobias Duckworth and David J Roberts. Camera image synchronisation in multiple camera real-time 3d reconstruction of moving humans. In *2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications*, pages 138–144. IEEE, 2011. 3

[24] Jörg Edelmann, Peter Gerjets, Philipp Mock, Andreas Schilling, and Wolfgang Strasser. Face2face—a system for multi-touch collaboration with telepresence. In *2012 IEEE*

International Conference on Emerging Signal Processing Applications, pages 159–162. IEEE, 2012. 2

[25] Fazliaty Edora Fadzli and Ajune Wanis Ismail. A robust real-time 3d reconstruction method for mixed reality telepresence. *International Journal of Innovative Computing*, 10(2), 2020. 2

[26] 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*, 27(4):814–827, 2016. 2

[27] Yingchun Fan, Hong Han, Yuliang Tang, and Tao Zhi. Dynamic objects elimination in slam based on image fusion. *Pattern Recognition Letters*, 127:191–201, 2019. 3

[28] G. Fontaine. The Experience of a Sense of Presence in Intercultural and Int. Encounters. *Presence: Teleoper. Virtual Environ.*, 1(4):482–490, 1992. 1, 2

[29] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3

[30] 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

[31] 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*, pages 161 – 167, 1994. 2

[32] H. Fuchs, A. State, and J. Bazin. Immersive 3D Telepresence. *Computer*, 47(7):46–52, 2014. 2

[33] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2021. 3

[34] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 3

[35] Simon J Gibbs, Constantin Arapis, and Christian J Breiteneder. Teleport–towards immersive copresence. *Multimedia Systems*, 7(3):214–221, 1999. 2

[36] 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. 2019. 2

[37] Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lamboray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, et al. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Transactions on Graphics (TOG)*, 22(3):819–827, 2003. 2

[38] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015. 3

[39] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 1831–1840, 2017. 16, 17

[40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 16, 17

[41] Richard Held. Telepresence. *The Journal of the Acoustical Society of America*, 92(4):2458–2458, 1992. 1, 2

[42] Mina Henein, Jun Zhang, Robert Mahony, and Viorela Ila. Dynamic slam: The need for speed. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2123–2129, 2020. 3

[43] 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*, pages 477–491. Springer, 2014. 3

[44] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn—revisiting data fidelity and regularization. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2555–2569, 2020. 4, 16, 17

[45] Matthias Innmann, Michael Zollh¨ofer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European conference on computer vision*, pages 362–379. Springer, 2016. 3

[46] 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*, pages 351–362. Springer, 2014. 3

[47] S. Izadi et al. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proc. of the ACM Symp. on User Interface Software and Technology*, pages 559–568, 2011. 2

[48] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 3

[49] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez, and Daniel Cremers. Fast odometry and scene flow from rgb-d cameras based on geometric clustering. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3999. IEEE, 2017. 3

[50] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 5846–5854, 2021. 3

[51] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021. 16, 17

[52] Michal Joachimczak, Juan Liu, and Hiroshi Ando. Real-time mixed-reality telepresence via 3d reconstruction with hololens and commodity depth sensors. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 514–515, 2017. 2

[53] 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. *ACM Transactions on Graphics (TOG)*, 28(3):1–8, 2009. 2

[54] B. Jones et al. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units. In *Proc. of the Annual Symp. on User Interface Software and Technology*, pages 637–644, 2014. 2

[55] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. Hierarchical voxel block hashing for efficient integration of depth images. *IEEE Robotics and Automation Letters*, 1(1):192–197, 2015. 3

[56] Olaf Kähler, Victor A Prisacariu, and David W Murray. Real-time large-scale dense 3d reconstruction with loop closure. In *European Conference on Computer Vision*, pages 500–516. Springer, 2016. 3

[57] 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. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015. 3

[58] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, 1997. 2

[59] 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 *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013. 3

[60] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 3

[61] Deok-Hwa Kim and Jong-Hwan Kim. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 32(6):1565–1573, 2016. 3

[62] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020. 16, 17

[63] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–713, 2018. 3

[64] Ryohei Komiyama, Takashi Miyaki, and Jun Rekimoto. Jackin space: designing a seamless transition between first and third person view for effective telepresence collaborations. In *Proceedings of the 8th Augmented Human International Conference*, pages 1–9, 2017. 2

[65] Dennis Krupke, Sebastian Starke, Lasse Einig, J Zhang, and F Steinicke. Prototyping of immersive hri scenarios. In *Human-Centric Robotics: Proceedings of CLAWAR 2017: 20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 537–544. World Scientific, 2018. 3

[66] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3D Environment for Remote Collaboration and Training of Physical Activities. In *IEEE Virtual Reality Conference*, pages 269–270, 2008. 2

[67] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017. 3

[68] Jason Lawrence, Dan B Goldman, Supreeth Achar, Gregory Major Blascovich, Joseph G. Desloge, Tommy Fortes, Eric M. Gomez, Sascha Häberling, Hugues Hoppe, Andy Huibers, Claude Knaus, Brian Kuschak, Ricardo Martin-Brualla, Harris Nover, Andrew Ian Russell, Steven M. Seitz, and Kevin Tong. Project starline: A high-fidelity telepresence system. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 40(6), 2021. 2

[69] Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, and Szymon Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics (TOG)*, 31(1):1–11, 2012. 3

[70] Shile Li and Dongheui Lee. Rgb-d slam in dynamic environments using static point weighting. *IEEE Robotics and Automation Letters*, 2(4):2263–2270, 2017. 3

[71] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 3

[72] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 3

[73] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 3

[74] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters*, 3(1):179–186, 2017. 3

[75] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose con-

trol. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021. 3

[76] 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*, pages 73–79, 2013. 2

[77] Xinzhong Lu, Ju Shen, Saverio Perugini, and Jianjun Yang. An immersive telepresence system using rgb-d sensors and head mounted display. In *2015 IEEE International Symposium on Multimedia (ISM)*, pages 453–458. IEEE, 2015. 2

[78] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs. Enhanced personal autostereoscopic telepresence system using commodity depth cameras. *Computers & Graphics*, 36(7):791 – 807, 2012. 2

[79] A. Maimone and H. Fuchs. Encumbrance-free Telepresence System with Real-time 3D Capture and Display Using Commodity Depth Cameras. In *Proc. of the IEEE Int. Symp. on Mixed and Augmented Reality*, pages 137–146, 2011. 2

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

[81] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 3

[82] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6361, 2021. 3

[83] 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*, 65(1):99–106, 2021. 3

[84] M MINSKY. Telepresence. *Omni*, 2(9):44–52, 1980. 1, 2

[85] 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*, pages 197–215, 2012. 2

[86] Carl Moore, Toby Duckworth, Rob Aspin, and David Roberts. Synchronization of images from multiple cameras to reconstruct a moving human. In *2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications*, pages 53–60. IEEE, 2010. 3

[87] A. Mossel and M. Kröter. Streaming and exploration of dynamically changing dense 3d reconstructions in immersive virtual reality. In *Proc. of IEEE Int. Symp. on Mixed and Augmented Reality*, pages 43–48, 2016. 2, 3

[88] Lothar Muhlbach, Martin Bocker, and Angela Prussog. Telepresence in videocommunications: A study on stereoscopy and individual eye contact. *Human Factors*, 37(2):290–305, 1995. 2

[89] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), jul 2022. 3

[90] J. Mulligan and K. Daniilidis. View-independent scene acquisition for tele-presence. In *Proc. IEEE and ACM Int. Symp. on Augmented Reality*, pages 105–108, 2000. 2

[91] 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*, 101(4):1–16, 2021. 3

[92] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 3

[93] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011*, 2011. 3, 6

[94] 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*, 9(3):546–561, 2014. 2

[95] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013. 3, 5

[96] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5772, 2021. 3

[97] S. Orts-Escolano et al. Holoportation: Virtual 3D Teleportation in Real-time. In *Proc. of the Annual Symp. on User Interface Software and Technology*, pages 741–754, 2016. 2

[98] Viken Parikh and Mansi Khara. A mixed reality workspace using telepresence system. In *International Conference on ISMAC in Computational Vision and Bio-Engineering*, pages 803–813. Springer, 2018. 2

[99] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 3

[100] 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. *arXiv preprint arXiv:2106.13228*, 2021. 3

[101] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. Room2room: Enabling life-size telep-

resence in a projected augmented reality environment. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*, pages 1716–1725, 2016. 2

[102] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 3

[103] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021. 3

[104] Lorenzo Peppoloni, Filippo Brizzi, Carlo Alberto Avizzano, and Emanuele Ruffaldi. Immersive ros-integrated framework for robot teleoperation. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 177–178. IEEE, 2015. 3

[105] 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

[106] 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

[107] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 3

[108] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10213–10224, 2021. 16, 17

[109] Amit Raj, Michael Zollhöfer, Tomas Simon, Jason M. Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11733–11742, 2021. 3

[110] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 3

[111] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 3

[112] 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*, 9(3):562–574, 2015. 2

[113] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 3

[114] Eric Rosen, David Whitney, Michael Fishman, Daniel Ullman, and Stefanie Tellex. Mixed reality as a bidirectional communication interface for human-robot interaction. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11431–11438. IEEE, 2020. 3

[115] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018. 3

[116] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017. 3

[117] David W Schloerb. A quantitative measure of telepresence. *Presence: Teleoperators & Virtual Environments*, 4(1):64–80, 1995. 2

[118] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3849–3856. IEEE, 2018. 3

[119] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017. 3

[120] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2646–2655, 2018. 3

[121] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. Bethere: 3d mobile collaboration with spatial input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 179–188, New York, NY, USA, 2013. Association for Computing Machinery. 2

[122] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992. 2

[123] 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. *IEEE Transactions on Visualization and Computer Graphics*, 25(5), 2019. 2, 3, 4, 5, 6

[124] 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 on Computer Vision for Augmented and Virtual Reality*, 2019. 2, 3

[125] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3630–3637. IEEE, 2019. 2, 3

[126] Patrick Stotko, Stefan Krumpen, Michael Weinmann, and Reinhard Klein. Efficient 3d reconstruction and streaming for group-scale multi-client live telepresence. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 19–25. IEEE, 2019. 2, 3, 4

[127] Michael Strecke and Jorg Stuckler. Em-fusion: Dynamic object-level slam with probabilistic data association. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2019. 3

[128] Jörg Stückler and Sven Behnke. Efficient dense rigid-body motion segmentation and estimation in rgb-d video. *International Journal of Computer Vision*, 113(3):233–245, 2015. 3

[129] Po-Chang Su, Ju Shen, and Muhammad Usman Rafique. Rgb-d camera network calibration and streaming for 3d telepresence in large environment. In *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, pages 362–369. IEEE, 2017. 2

[130] Shih-Yang Su, Frank Yu, Michael Zollhoefer, and Helge Rhodin. A-nerf: Surface-free human 3d pose refinement via neural rendering. *arXiv preprint arXiv:2102.06199*, 2021. 3

[131] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 3

[132] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3

[133] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 3

[134] 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 *Proc. of the Int. Conf. on Augmented Tele-existence*, pages 112–118, 2005. 2

[135] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 16, 17

[136] Theophilus Teo, Louise Lawrence, Gun A Lee, Mark Billinghurst, and Matt Adcock. Mixed reality remote collaboration combining 360 video and 3d reconstruction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–14, 2019. 2

[137] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 3

[138] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, W Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022. 3

[139] Michail Theofanidis, Saif Iftekar Sayed, Alexandros Lioulemes, and Fillia Makedon. Varm: Using virtual reality to program robotic manipulators. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 215–221, 2017. 3

[140] 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 Tele-Collaboration Over Internet2. In *Proc. of the Int. Workshop on Immersive Telepresence*, 2002. 2

[141] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 3

[142] 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

[143] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022. 3

[144] 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*, 13(3):573–584, 2011. 2

[145] 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

[146] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4, 16, 17

[147] 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

[148] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021. 3

[149] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 3

[150] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018. 3

[151] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021. 3

[152] B. G. Witmer and M. J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoper. Virtual Environ.*, 7(3):225–240, 1998. 1, 2

[153] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4671–4680, 2017. 3

[154] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 3

[155] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019. 3

[156] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020. 3

[157] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 817–833, 2018. 3

[158] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352, 2014. 3

[159] 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 *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021. 3

[160] Jacob Young, Tobias Langlotz, Steven Mills, and Holger Regenbrecht. Mobileportation: Nomadic telepresence for mobile devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–16, 2020. 2

[161] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174. IEEE, 2018. 3

[162] Hao Zhang and Feng Xu. Mixedfusion: Real-time reconstruction of an indoor scene with dynamic objects. *IEEE transactions on visualization and computer graphics*, 24(12):3137–3146, 2017. 3

[163] Shujun Zhang and Wan Ching Ho. Tele-immersive interaction with intelligent virtual agents based on real-time 3d modeling. *Journal of Multimedia*, 7(1):57, 2012. 2

[164] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328. IEEE, 2020. 3

[165] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6278–6287, 2020. 16, 17

[166] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 3

[167] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 4

[168] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 3

[169] 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*, 41(4):90–98, 2021. 3

[170] 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 *2016 IEEE International conference on image processing (ICIP)*, pages 365–369. IEEE, 2016. 2

A. Detailed Explanation of Components

Filtering of Instance Proposals. When given a color image I_k, the instance segmentation network f_{seg} used in our work outputs region proposals $M_1, ..., M_n$ in terms of Boolean masks that indicate the membership of each pixel $u \in \mathcal{U}$, i.e. $M_j(u) = 1$, if pixel u belongs to proposal j, and $M_j(u) = 0$ otherwise. In addition, each mask is associated with a class label l_j and a confidence score $c_j \in [0, 1]$. To produce the per-pixel class label L_k and instance ID maps ι_k, we have to integrate potentially overlapping region proposals, taking the confidence scores into account. We accomplish this by first removing proposals with a confidence smaller than a threshold τ_{conf}. For each pixel u, we then find the instance ID of the proposal with maximum confidence, i.e. for the filtered indices $j'_1, ..., i'_{n'}$ we compute $i^*(u) = \text{argmax}_i\{c_i \mid M_i(u) = 1\}$. The resulting assignment is again filtered by removing IDs that do not exceed a minimum pixel count. In other words, we set

$$\hat{i}(u) = \begin{cases} i^*(u), & \text{if } |\{i^*(u) = i\}| \geq \tau_{\text{count}} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

As described in the paper, the resulting indices are then associated with the IDs from the previous frame to get the final map ι_k of instance IDs and the label map is set to the corresponding class labels $L_k(u) = l_{\iota_k(u)}$.

Dynamicity Initialization. For instances that cannot be associated with an instance in the previous frame, we keep a mean for the dynamicity scores of each class that is smoothed according to the scheme shown in Equation (6) of the main paper. More specifically, let $\tilde{s}_k : \mathcal{L} \to \mathbb{R}_{\geq 0}$ be defined as a mapping from the set of class labels $\mathcal{L}$ to the mean dynamicity scores, where

$$\tilde{s}_k(l) := \max \left\{ \alpha_{\text{class}} \cdot \tilde{s}_{k-1}(l), \ \frac{1}{|\mathcal{I}_l|} \sum_{i \in \mathcal{I}_l} \hat{s}'_k(i) \right\}. \tag{11}$$

Figure 5 depicts the steps taken to compute the temporally smoothed score $s'_k(i)$ of some instance i with label l_i from the initial score $s_k(i)$. Importantly, we also handle the case that a class is observed for the first time in the current frame. In that case, we set the class mean to the only available observation $s_k(i)$.

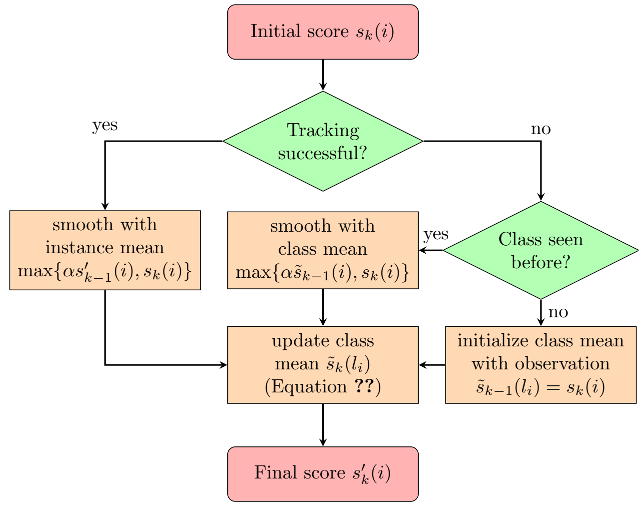


Figure 5. Flowchart visualizing the procedure for the temporal smoothing of the dynamicity score of instance i with class label l_i in frame k. If the tracking matches instance i with an instance from the previous frame, the resulting score is smoothed as shown in Equation (6). Otherwise, we first check whether the class was observed in a previous frame. In this case, we smooth the score similarly to Equation (6) but using the smoothed class mean from the last frame instead. In case the class is observed for the first time, we keep the initial score and use it as the new class mean for l_i. Lastly, the class mean for l_i is updated.

B. Hyperparameter Choices

The hyperparameters used for the performance evaluation and visualization were fixed for all scenes and are listed in Table 3.

C. Scene Descriptions

Table 5 contains a short description as well as some exemplary RGB images of each of the 8 self-recorded scenes used for our experiments. The scenes were all captured with a Microsoft Azure Kinect RGB-D sensor at a frame-rate of 30 Hz using the narrow FOV configuration and without depth binning.

D. Detailed Performance Comparison

In Table 6, we list the raw computation speed, measured in terms of frames per second (FPS), of individual components of our pipeline during evaluation. The results show that, on average, the inference of the instance segmentation network is the limiting factor for the overall speed of the pipeline. Note that the components run in parallel, such that actual processing speed of each component is limited by the output speed of the previous one. The measured values therefore only represent an upper bound for the FPS that each component can reach in our implementation. An example of this parallel execution is also visualized in Figure 7.

E. Comparison with Further Segmentation and Optical Flow Networks

To evaluate the choice of the instance segmentation [146] and optical flow [44] networks used in our approach, we compared the performance differences with some other recent segmentation [39,40,62,108] and optical flow [51,135,

Symbol	Description	Value
τ_{conf}	Minimum segmentation confidence	0.1
τ_{count}	Minimum pixel count for class acceptance	2300
τ	Dynamic threshold	1.1
τ_η	Voxel weight dynamicity threshold	1.1
τ_{SDF}	SDF invalidation dynamicity threshold	1.1
$\hat{W}_\eta$	Static max. voxel weight	255
$\check{W}_\eta$	Dynamic max. voxel weight	3
c	Histogram bin width	0.25
N/A	Histogram relative min. bin count	0.01
N/A	Instance tracking min. overlap ratio	0.2
α	Instance dynamicity decay factor	0.9
α_{class}	Class dynamicity decay factor	0.9
δ	Dynamicity normalization scale factor	0.4

Table 3. Choices for the hyperparameters of the pipeline used during the evaluation. Symbol "*N/A*" indicates that no symbol was given to this parameter in the main publication or the supplementary material.

Segmentation	Optical Flow	FPS [1/s]
Mask R-CNN [40]	LiteFlowNet2 [44]	6.91 (1.49)
PointRend [62]		5.34 (1.29)
SCNet [39]		4.17 (0.44)
DetectoRS [108]		2.39 (0.14)
SoloV2 [146]	MaskFlowNet [165]	9.57 (5.08)
	RAFT [135]	2.88 (0.09)
	GMA [51]	2.55 (0.08)
SoloV2 [146]	LiteFlowNet2 [44]	11.38 (5.51)

Table 4. Performance comparison of our pipeline using different networks for instance segmentation and optical flow. For this purpose, we provide the resulting frame-rate (in FPS) our approach reaches using the given combinations of networks.

165] techniques. Table 4 shows our pipeline's performance in terms of frames per second.

F. Failure Cases

Our approach relies on accurate predictions from both the segmentation and optical flow networks. Objects not detected by the instance segmentation network affect their assignment to the static or dynamic scene parts, which is shown in Figure 6. Here, the balloon is not detected as an object by the SoloV2 network and is therefore erroneously integrated into the static model. While the integrated voxels conflict with future measurements and are eventually removed, they cause visually unpleasant artifacts during reconstruction. 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.

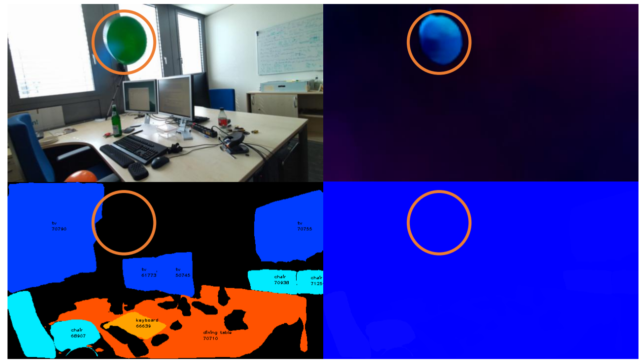


Figure 6. Failure case of our method. Shown are RGB (top left), optical flow (top right), instance segmentation (bottom left) and resulting segmentation into static and dynamic (bottom right). Even though a clear motion cue is available in the optical flow image, due to a missing object detection, our method fails to correctly identify the dynamic region (orange circle).

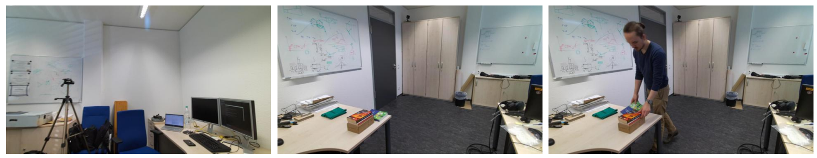
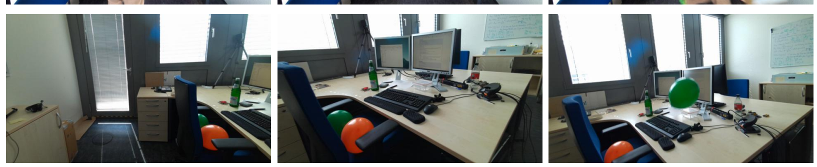
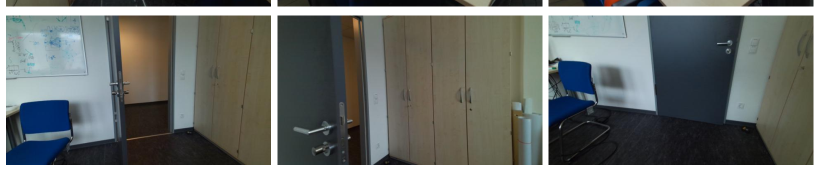
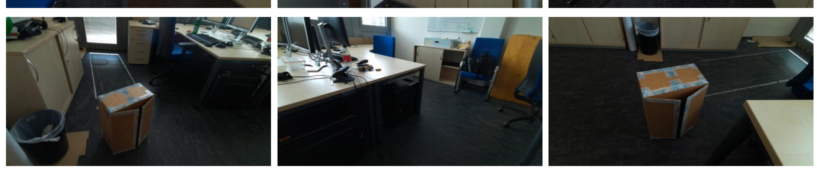
Scene	Description	Exemplary Images from Scene
items_1	A person moves around items (books and boxes) on an office table.	
items_2	A person picks up and drops off items on a table in a medium-sized office.	
people_1	Two persons meet at a coffee table and exchange a box.	
people_2	Chairs and a boxes are moved around in an office seating area.	
people_3	Two persons exchange a small box.	
ego_view	Balloons are kicked around in a medium-sized office.	
oof_1	An office door is opened and closed while not seen by the camera.	
oof_2	A box is moved multiple times while the camera is not observing it.	

Table 5. Short description and exemplary images for each of the scenes used for the evaluation.

Scene	System Components								
	Segmentation		Optical Flow	Odometry	Dynamicity				
	Inference	Tracking			EPE	Norm.	Smooth.	Prop.	Acc.
items_1	11.53	62.89	12.20	11.29	22.45	28.12	55.79	159.78	59.65
items_2	9.86	58.40	11.81	13.25	24.08	43.22	59.36	148.02	57.75
people_1	10.53	56.91	11.68	13.02	24.82	34.22	62.42	155.88	58.84
people_2	9.36	54.17	11.71	14.66	25.83	44.61	63.44	156.12	58.38
people_3	12.94	69.20	11.29	11.46	25.52	26.86	71.12	159.16	59.62
ego_view	9.73	51.88	11.92	16.29	25.76	49.17	56.43	156.67	59.23
oof_1	13.24	67.29	11.01	11.19	24.10	28.95	58.75	155.81	57.10
oof_2	10.63	56.02	11.87	14.99	25.45	41.48	59.37	154.40	58.75
Mean	10.98	59.59	11.56	13.27	24.65	37.08	60.84	155.73	58.67

Table 6. Raw computation speeds in FPS [1/s] for the major components of our pipeline, evaluated separately for each of the 8 scenes used for the evaluation. The last rows shows the mean over all scenes. The dynamicity computation is split into end-point-error computation (EPE), normalization (Norm.), temporal smoothing (Smooth.), object propagation (Prop.) and accumulation (Acc.).

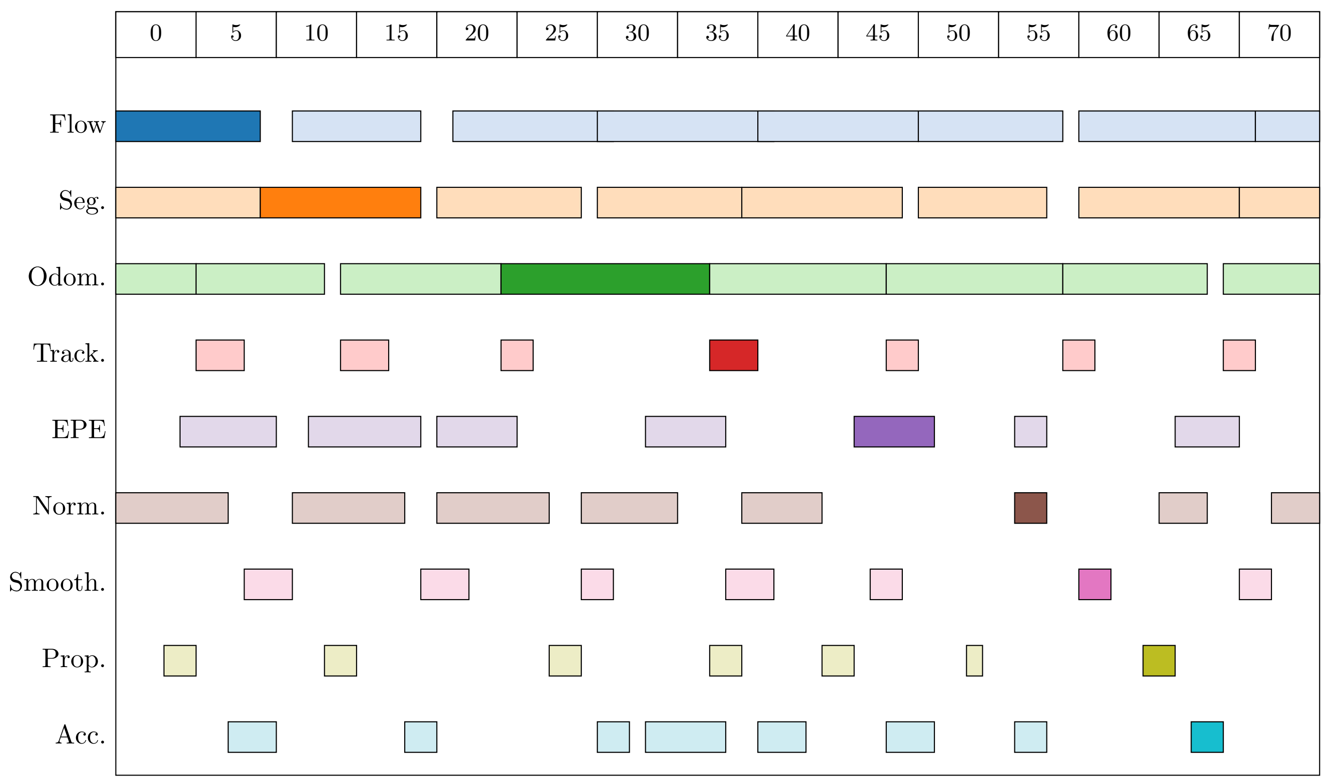


Figure 7. Gantt chart showing the compute durations of the major components of our pipeline in a 700ms long section from the evaluation of scene *items_1*. The highlighted bars correspond to the same input frame. System components are optical flow estimation (Flow), instance segmentation (Seg.), odometry (Odom.), instance tracking (Track.), end-point-error computation (EPE), dynamicity normalization (Norm.), temporal smoothing (Smooth.), propagation (Prop.) and accumulation (Acc.). It can be seen that faster components of the pipeline have to wait for the next frame to become available to continue processing. Additional gaps result from scheduling and process communication.