SharedNeRF: Leveraging Photorealistic and View Dependent Rendering for Real-time and Remote Collaboration

Mose Sakashita*
Microsoft Research
United States

Balasaravanan Thoravi Kumaravel
Microsoft Research
United States

Nicolai Marquardt
Microsoft Research
United States

Andrew D. Wilson
Microsoft Research
United States

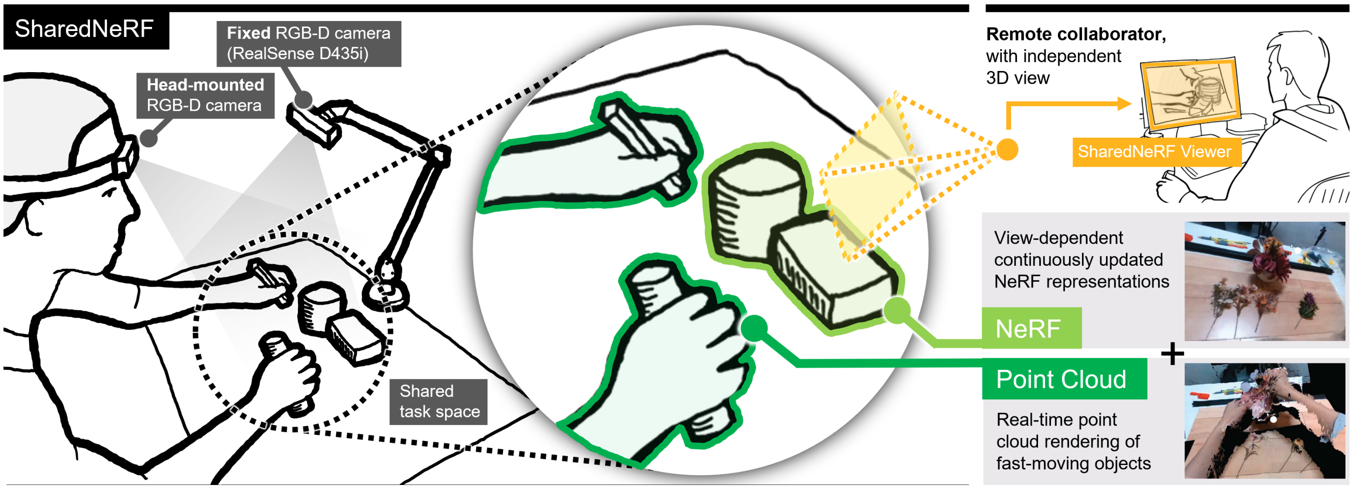
SharedNeRF
Fixed RGB-D camera (RealSense D435i)
Head-mounted RGB-D camera
Shared task space
Remote collaborator, with independent 3D view
SharedNeRF Viewer
View-dependent continuously updated NeRF representations
NeRF
+
Point Cloud
Real-time point cloud rendering of fast-moving objects

Figure 1: SharedNeRF uses a head-mounted camera and stationary camera in a local environment (left) and then renders high fidelity and view-dependent shared space where a remote viewer in another location can look at physical artifacts from different viewpoints, leveraging both NeRF and point cloud techniques (right).

ABSTRACT

Collaborating around physical objects necessitates examining different aspects of design or hardware in detail when reviewing or inspecting physical artifacts or prototypes. When collaborators are remote, coordinating the sharing of views of their physical environment becomes challenging. Video-conferencing tools often do not provide the desired viewpoints for a remote viewer. While RGB-D cameras offer 3D views, they lack the necessary fidelity. We introduce SharedNeRF, designed to enhance synchronous remote collaboration by leveraging the photorealistic and view-dependent nature of Neural Radiance Field (NeRF). The system complements the higher visual quality of the NeRF rendering with the instantaneity of a point cloud and combines them through carefully accommodating the dynamic elements within the shared space, such as hand gestures and moving objects. The system employs a head-mounted camera for data collection, creating a volumetric task space on the fly and updating it as the task space changes. In our preliminary study, participants successfully completed a flower arrangement task, benefiting from SharedNeRF's ability to render the space in high fidelity from various viewpoints.

CCS CONCEPTS

• **Human-centered computing** → *Collaborative interaction.*

KEYWORDS

NeRF; Collaboration; Spatial Interfaces;

ACM Reference Format:
Mose Sakashita, Balasaravanan Thoravi Kumaravel, Nicolai Marquardt, and Andrew D. Wilson. 2024. SharedNeRF: Leveraging Photorealistic and View Dependent Rendering for Real-time and Remote Collaboration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3613904.3642945

[*] This work was done while the first author was interning at Microsoft Research.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642945

1 INTRODUCTION

Collaboration around physical objects often requires that each collaborator closely examines the various facets of a design artifact or hardware device for detailed review or discussion [43]. While this is straightforward in co-located settings, where individuals can independently view the physical design from different perspectives and directly reference specific areas or objects, it becomes considerably more challenging in remote scenarios. Previous video-based systems have attempted to facilitate remote view sharing through the use of head-mounted cameras [10, 12, 13, 20, 29, 35, 42] or handheld devices [9, 15, 63]. These approaches, however, heavily rely on the local user with regard to obtaining a preferable viewpoint, which can necessitate substantial verbal communication to achieve the desired viewpoint for remote viewers [8, 69]. Additionally, local collaborators often struggle to offer the best views to remote viewers, a problem commonly caused by inattention or misunderstandings [39, 43]. While other video approaches offer several views through multiple cameras [16, 56, 72, 74], they lack the ability to seamlessly view the scene from free viewpoints. Other research has explored the use of controllable cameras equipped with mobility [36, 40, 59, 73] or arms [18, 28, 52] to grant remote users control over the camera's positioning within the space. Despite these advancements, the range of accessible viewpoints and the pace of interaction remain constrained by the limitations of physical robots.

Point clouds obtained from RGB-D cameras have been used for rendering 3D views to share task space in remote collaboration [1, 24, 49, 51, 63, 68, 72]. These 3D rendering techniques can provide spatial awareness of the workspace in real-time but often fail to deliver a high-fidelity or photorealistic representation of the task space, which may be important for physical tasks or design critique, as they lack the fine-grained details such as lighting, reflections, textures, or small components. The advent of the Neural Radiance Field (NeRF) techniques has revolutionized the realm of photorealistic and view-dependent rendering [41]. It empowers viewers with the ability to navigate and explore high-fidelity 3D spaces, granting them the freedom to alter their viewpoints. Despite the great benefits, NeRF representations are often optimized for static scenes. While there have been efforts to adapt NeRF to dynamic scenes [14, 38, 48, 53, 64], a prominent issue is the considerable amount of time required for training, which precludes the possibility of synchronous interactions by using NeRF techniques alone.

We present SharedNeRF, a system designed to enhance synchronous remote collaboration by leveraging the high-fidelity and view-dependent nature of the NeRF technique, as illustrated in Fig. 1. The system complements the high latency updates of the NeRFs with the low latency real-time point cloud rendering. With SharedNeRF, remote viewers can independently navigate a shared 3D task space, immersing themselves in high-fidelity views of physical artifacts while simultaneously being able to observe dynamic elements such as hand gestures or moving objects in real-time. Moreover, when alterations to the task space occur, SharedNeRF enables NeRF updates to gradually align with the most recent state of the space, affording a up-to-date collaborative environment. We demonstrate scenarios of using SharedNeRF where collaborators work on flower arrangement and computer inspection tasks. We also report initial user feedback from our preliminary study, conducted to validate the potential of our approach.

2 RELATED WORK

SharedNeRF builds upon prior research in remote collaboration involving physical objects, video systems, and spatial rendering techniques for view-sharing. We offer a comprehensive review of significant literature in these domains.

2.1 Remote Collaboration for Physical Tasks

Buxton's framework highlights the significant role of seamlessly merging both *person space* and *task space* to facilitate natural flow of interactions in remote collaboration [3]. There is also a recognized need to render *reference space* over the shared space, facilitating non-verbal cues such as deictic pointing gestures [2, 19]. While sharing precise task space is achieved within purely virtual environments (e.g., Autodesk Design Review [22], DDRIVE [6]), the scenario becomes more complex when the tasks pivot around physical objects in real-world [43, 69]. In the context of remote physical tasks, prior studies identify a shared visual space for collaborators as a central role in establishing 'common ground' [5] and ensuring successful collaboration [12, 13]. This shared visual space affords awareness of the current status of tasks [17]. Standard video tools like Teams or Zoom often fail to seamlessly share physical artifacts' visuals. Local collaborators struggle to align views with remote partners' needs [39, 43]. We explore a new technique to afford sharing of their task space in high-fidelity where their remote collaborators can independently explore the space while supporting several non-verbal cues such as head direction and gestures.

2.2 Sharing Task Space in Video Systems

There has been much exploration of video-based techniques to share task space for physical collaboration. ClearBoard [26] or VideoWhiteboard [66] supports rendering drawings and gestures over a physical board. While these approaches may be effective for whiteboard design interactions, sharing arbitrary task space with three-dimensional objects is not well supported. TeamWorkStation [25] uses a tabletop camera to share task space around a table with annotations, but this flat 2D view makes sharing different angles of artifacts difficult [43]. Other works propose techniques for sharing more than one fixed view of the task space. Multiple camera views [16, 56, 74] have been used to provide several viewpoints, but synchronizing and switching between views can be disorienting and view changes are not seamless. While head-mounted cameras [10, 12, 13, 20, 29, 35, 42, 70–72] can offer a first-person perspective, they can require more coordination effort to share a view desired by the remote user [8]. Other work uses tracking cameras [57] to follow movement and adjust angles automatically, but may sometimes misinterpret user intent or miss rapid actions. Taking snapshots of objects has been explored to share task space [21, 46], but lack the continuity and the ability of looking at an artifact from different perspectives. Handheld mobile devices [9, 15] can be versatile and portable, but holding them can be tiring, and they may not always provide a stable view. Moreover, it is not practical for a collaborator to hold a camera while engaging with the physical objects in the environment [27].

2.2.1 Enabling Independent Viewpoints in Shared Space. Remote collaboration can be augmented by giving remote collaborators independent control over their viewpoints, reducing verbal communication efforts in sharing views [8]. This has been explored using mobile platforms [36, 40, 59, 73] and adjustable arms [18, 28, 52]. For example, the Asteroids system [36] employs a set of robots on a workbench, allowing remote users to switch between them and control their position to adjust their focus within a collaborative task environment. Similarly, mobile robotic systems like VRoxy [59] and RobotAR [73] utilize an onboard camera and the robot's mobility to alter perspectives within a room-scale environment. Systems such as Heimdall [28] and TeleAdvisor [18] employ actuated arms to manipulate camera views for hands-on tasks, while Periscope [52] features a more extensive robotic arm, facilitating both local and remote users in placing a camera to a new viewpoint. While these systems may enhance collaboration, they also have limitations: specialized robots may be inaccessible, physical constraints can limit viewpoints, and remote users may experience camera feed/control latency. Our work utilizes a common setup (e.g., a tabletop camera, and head-mounted camera) for enabling remote users to explore the shared space, looking from different viewpoints with instant interactivity with a mouse and keyboard.

2.3 3D Rendering for Physical Space Sharing

Another stream of work leverages 3D rendering techniques, as opposed to 2D videos, for remotely sharing physical spaces. VRoxy [59] or Mini-Me [50], for example, pre-capture the photogrammetry of their task space and render them through a head-mounted display, but these 3D geometries of the scene are static since they are captured in advance. Other work employs RGB-D cameras and creates point cloud or mesh rendering for sharing the physical environment [1, 24, 49, 51, 63, 68]. Systems such as Volumetric Mixed Reality [24] and CoVAR [51] show RGB-D-based meshes for supporting physical collaborative tasks. MirageTable [1] and Room2Room [49] use a set of RGB-D cameras and projectors to render the shared space on a physical environment, supporting table-scale and room-scale interactions respectively. While they provide freedom for spatial exploration in the shared space, point cloud rendering techniques are also known for poor quality due to sensor noise and a large number of missing points or incomplete meshes [24]. Holoportation [47] uses RGB-D cameras for real-time 3D model reconstruction for immersive telepresence. However, these rendering methods fall short in delivering a high-fidelity, photorealistic representation of task spaces, often missing out on complex details like lighting, reflections, material properties such as texture, or small components. Loki [68] merges point cloud rendering with video streaming, utilizing both 2D video and point cloud [59, 67]. Building on the value of capturing and rendering 3D spaces in real-time demonstrated by such prior work, our system also leverages RGB-D based point clouds for its strength. Additionally, it augments them with Neural Radiance Fields that add higher-quality details to it.

2.4 Photorealistic Volumetric Representation

Recent volume rendering techniques such as Neural Radiance Fields (NeRF) [41] and Gaussian Splatting [31] allow the synthesis of photorealistic novel views without the need for special cameras or hardware. These methods can handle rendering of fine-grained details such as reflection and transparent objects, small details, and objects with fabrics or other unique textures, that are difficult to capture with point cloud or photogrammetry-based techniques. Their limitation as first presented is its long training time. To reduce the training time, Instant-NGP [45] proposes a novel input representation called multiresolution hash encoding that speeds up the training from hours to seconds or minutes. However, these NeRF representations only support static scenes in which there are no dynamic or moving objects, making them unsuited for real-time collaborative tasks. There have been advancements in NeRF for dynamic scenes, including Dy-NeRF [38], D-NeRF [53], Temporal Interpolation-based NeRF [48], DynamicNeRF[14] and NeRFPlayer [64]. These methods extend the NeRF framework to allow rendering of volumetric representation of dynamic scenes. However, to date, these techniques are not fast enough to train and show a volumetric scene frame by frame in real-time.

Luma AI [23] offers a platform for capturing and rendering volumetric scenes, enabling users to share spatial representations with others; however, its application is largely limited to asynchronous sharing. While several systems utilize Simultaneous Localization and Mapping (SLAM) for the real-time training of volumetric scenes, their design primarily focuses on static scenes and lacks support for remote interface [30, 58]. To our knowledge, SharedNeRF is the first system designed to leverage the photorealistic and view-dependent capabilities of the NeRF technique to facilitate synchronous remote collaboration, which we achieve through the integration of traditional high-frequency 3D rendering.

3 SharedNeRF DESIGN

In this section, we present the SharedNeRF system designed to support remote interactions during collaboration around physical artifacts. Iterative prototyping is a complex and dynamic process where collaborators engage in a cycle of experimenting with ideas and assimilating feedback [7, 43]. This interactive cycle is crucial, especially in collaborative settings, where collaborators observe different aspects of a design before convening to discuss improvements. During the review phase, individual reviewers can suggest

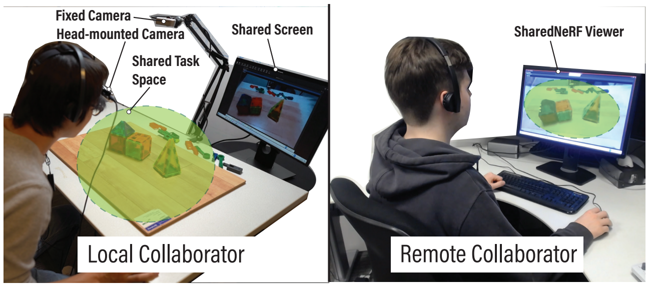

Figure 2: SharedNeRF local and remote setups. The local collaborator wears a head-mounted camera for real-time data collection and has a fixed camera on a desk for detecting movements in the task space (left). The remote collaborator uses the SharedNeRF viewer using a mouse and keyboard, navigating in the shared space (right).

changes, effectively indicating their focus or gesturing towards specific design elements. This process of feedback and discussion enables collaborators to make iterative adjustments to the design. They continue to refine and review the updated version until it aligns with the intended outcome. Such a sequence of actions in iterative prototyping and critique sessions underlines the design requirements for a system aimed at providing seamless remote collaborative experiences in such scenarios. Key functionalities of such a system encompass the capacity for a detailed examination of the initial design from multiple viewpoints, offering immediate feedback on ongoing changes, and enabling dynamic non-verbal communication for clarity. Moreover, the system should allow for the assessment of revised designs to further the iterative process.

3.1 Complementary Nature of NeRF and Point Cloud

Here we describe the pros and cons of both NeRF and traditional point cloud rendering techniques and discuss how a system can leverage the strengths and mitigate the weaknesses of each approach to facilitate the intricate interactions discussed above.

NeRF, due to its view-dependent nature, excels in rendering complex details of physical attributes that vary based on different viewpoints. It can depict phenomena such as the reflection of light through transparent materials like glass, or the fine-grained textures found in materials such as fur or even plants. This capability to achieve a high degree of realism enriches the contextual understanding of the workspace for collaborators, However, a significant drawback to NeRF is its time-consuming training process for each scene state, which can range from a few seconds to several minutes.

In contrast, traditional point cloud rendering, facilitated by RGB-D cameras, can instantly capture and represent the real-time state of a scene, including its dynamic elements. This speed ensures synchronous collaboration, where changes can be viewed and discussed in real-time. However, this method is not without its flaws; it often results in a loss of detail, creating visual artifacts such as holes or sharp edges in the rendered scene. Moreover, depth sensors used in this approach can have limited fidelity and often struggle to accurately capture transparent objects or small details such as wires or small components.

SharedNeRF leverages the complementary nature of point cloud and NeRF rendering techniques, integrating the detailed rendering capabilities of NeRF with the real-time responsiveness of point clouds to facilitate a collaborative environment that is both detailed and responsive to dynamic changes in the scene. Specifically, it utilizes NeRFs for their high-fidelity, view-dependent rendering capabilities, essential for an in-depth review. Concurrently, it integrates the real-time point cloud rendering with NeRFs, enhancing the system's capacity to handle dynamic aspects of collaboration. Fig. 2 illustrates the setup of SharedNeRF where a remote collaborator can see a 3D representation of the shared task space via a GUI viewer. In the following, we describe the details of the system design, articulating each key design component of SharedNeRF.

3.2 Real-time Dataset Collection for NeRF Training

NeRF is typically trained to utilize a fixed batch of several hundred images at most. These images are pre-processed to compute camera calibration parameters and camera poses, which are required for training. This approach is not amenable to synchronous and continuous use over extended periods, especially in environments where the scene is changing. Where the scene changes over time, there must be a mechanism to continuously gather training images and update the NeRF model as quickly as possible. Furthermore, remote participants may initially have little or no view of the physical environment. Therefore, it is important for the system to promptly present the most current state of the task space as soon as the remote connection is established.

To achieve this, SharedNeRF uses a head-mounted camera as a means to collect a dataset on the fly, as shown in Fig. 2. Many applications of NeRFs employ offline structure from motion techniques such as COLMAP [62] to compute precise camera pose over a video. Such offline approaches are too slow to use in an interactive, real-time application. Instead, SharedNeRF integrates ORB_SLAM3 [4], a real-time, frame-by-frame localization system prevalent in

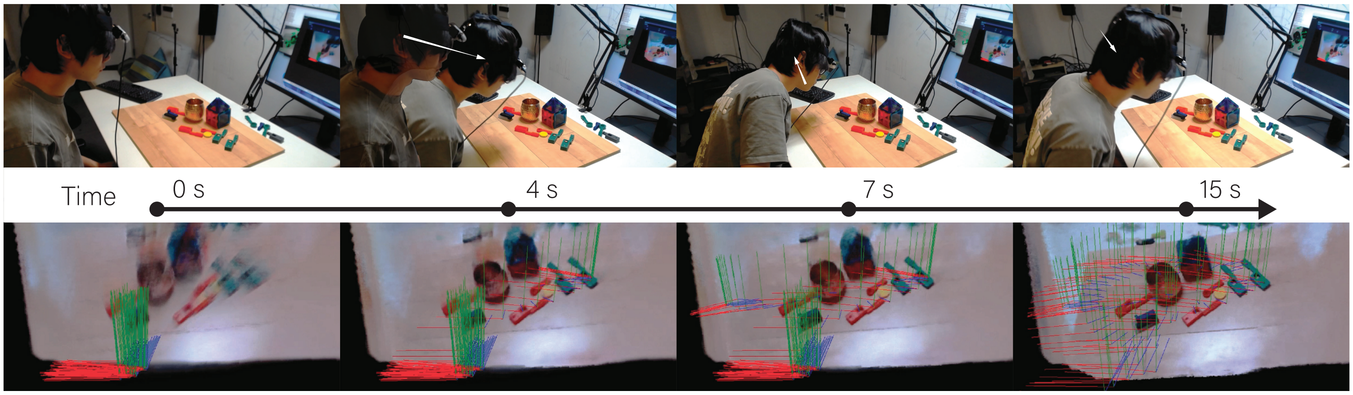

Figure 3: Sequence of optimizing the dataset by adding or discarding images in a way that increases the diversity of camera poses, showing the visualization of more diverse camera poses over time as the local collaborator moves their head. The pose of the camera associated with each training image is illustrated by red, green and blue lines depicting camera coordinate axes.

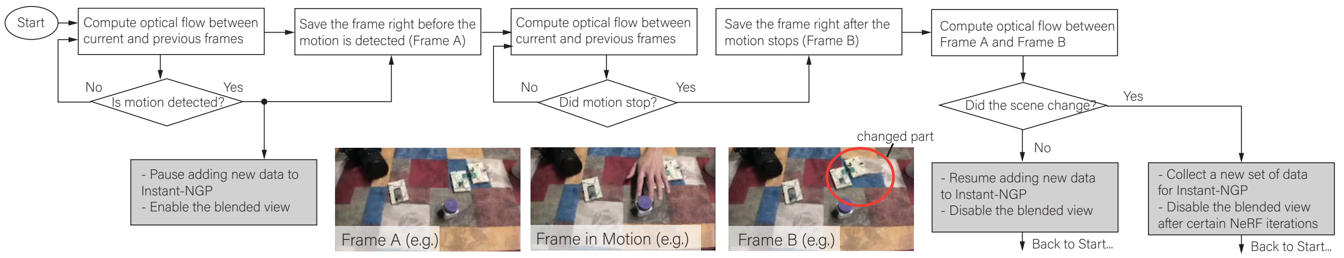
Start
Compute optical flow between current and previous frames
Is motion detected?
No
Yes
- Pause adding new data to Instant-NGP
- Enable the blended view
Save the frame right before the motion is detected (Frame A)
Compute optical flow between current and previous frames
Did motion stop?
No
Yes
Save the frame right after the motion stops (Frame B)
Compute optical flow between Frame A and Frame B
Did the scene change?
Yes
No
- Resume adding new data to Instant-NGP
- Disable the blended view
Back to Start...
- Collect a new set of data for Instant-NGP
- Disable the blended view after certain NeRF iterations
Back to Start...
changed part
Frame A (e.g.)
Frame in Motion (e.g.)
Frame B (e.g.)

Figure 4: Flow chart illustrating the algorithm to determine whether there is a moving object or a permanent change in the scene, and subsequently update the training dataset for Instant-NGP.

robotics fields. We then send the training images and poses frame by frame to Instant-NGP [45] for training a NeRF model of the scene.

3.2.1 Optimizing Dataset for Improving NeRF. Upon initiating the data collection process, the system collects an initial set of between 50 and 80 images, which we found maintained a reasonable balance between NeRF quality and data collection duration. The initial NeRF output tends to be of poor quality due to the limited head movement in the early stages, resulting in a dataset populated with images from similar camera poses.

To enhance the NeRF representation, the system optimizes the training dataset dynamically over time. Incorporating a diverse range of camera poses in the training set, as opposed to a limited variety, enables the overall quality of NeRF as well as a more extensive navigation scope within the 3D NeRF rendered scene. The system uses a greedy algorithm to maximize the diversity of camera poses, as illustrated in Fig. 3. The algorithm operates in the following manner: It first identifies the camera pose in the existing dataset that has the minimum distance to the nearest camera pose. This identified pose is then compared to the minimum distance from a new image's camera pose to the closest pose in the current dataset. If the new pose offers a greater minimum distance than the identified pose in the existing dataset, it is added to the dataset, replacing the pose that contributes the least to the diversity of camera poses. This approach supports a continual enhancement of the NeRF representation as the user moves their head as they engage with the task space.

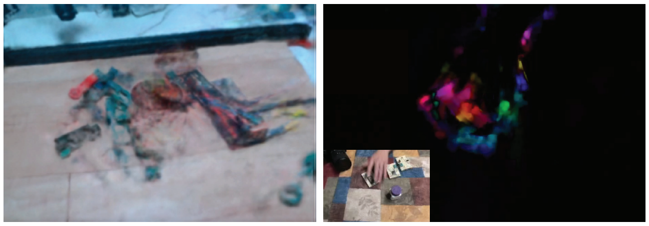

Figure 5: An example of NeRF views with poor performance when the dataset consists of images that are not consistent across the dataset (left). Optical flow is used to detect moving objects in the task space. The direction of movement is indicated by hue values, while the magnitude of the movement is represented through intensity (right).

3.3 Updating the NeRF to Reflect a Changing Task Space

The technique described above enables continuous improvement to the NeRF network by continually updating the dataset. In the SharedNeRF scenario, the local user may often manipulate the objects in the scene. This introduces the possibility of an inconsistency in the training dataset, where some training images feature a meaningful change in the scene or even the user's hands, while older images do not. Maintaining a dataset with consistent scenes is necessary, as any inconsistencies could result in rendering artifacts since NeRFs function by overfitting to the training data. These artifacts could render the representation unstable, causing it to display varied or blurry content based on the viewer's perspective (see Fig. 5 left).

Moreover, it is essential for the remote collaborator to see meaningful changes in the scene by having an up-to-date NeRF model, especially when objects within the task space are changed, moved, introduced, or removed. To manage this, the system distinguishes between two distinct states of a scene: one that identifies motion within the scene, and another that detects alterations to the scene. Recognizing these states enables us to strategically determine when to update the dataset for the NeRF by discarding an old dataset and collecting a new dataset.

The system monitors scene dynamics using a stationary camera to compute optical flow as shown in Fig. 5 (Right). To maintain consistency in the training dataset, it pauses the addition of new images to the SharedNeRF training set the moment the movement of hands or objects is detected within the scene. When the scene is static, we determine if the new data is added based on whether it improves the dataset, as described in Section 3.2.1. At the end of the detected motion, the system evaluates whether the current frame differs from the last captured frame prior to the detection of movement. This involves utilizing optical flow analysis between the frames to detect any significant change. In the event of a significant change in the scene, all images in the existing dataset are replaced with the newly incoming frames, followed by the dataset optimization process. If the scene remains unchanged, it resumes data collection without refreshing the dataset. The overall flow is illustrated in Fig. 4. Through these processes, the NeRF model can

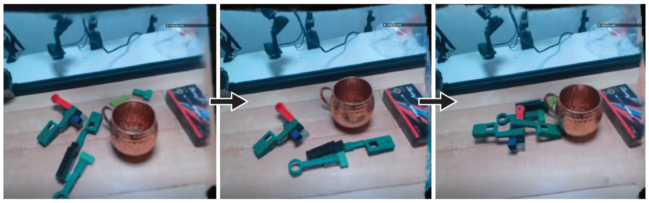

Figure 6: NeRF scene updates after each detected alternation in the task space. The last two snapshots were taken seven seconds after the change was completed by a user.

transition upon changes made in the task space over time, as shown in Fig. 6.

3.4 Blending Real-time Dynamic Elements with Point Cloud Rendering

Despite the system's ability to update the NeRF representation in response to changes in the task space, it falls short of providing remote users with real-time visibility of dynamic changes in the task space. There exists a time delay between the acquisition of new images and poses and the subsequent update of the NeRF to mirror the current state of the scene. With our current prototype system, this delay is approximately five seconds.

To bridge this delay and ensure real-time visibility of dynamics in shared spaces, we incorporate a high-frequency rendering method, traditionally used to share task space for remote interaction. Specifically, this integration uses point cloud rendering from both the head-mounted and fixed cameras after a calibration process (see Section 3.6). To merge point cloud visuals into the same coordinate system, we continually apply camera poses of the head-mounted camera obtained from SLAM every frame and set the same brightness, contrast, and color temperature values for both cameras. The blending of the point cloud and NeRF on a rendering surface is facilitated through depth maps generated by the point cloud and instant-NGP. Utilizing the z-buffer, we ensure proper occlusions, allowing for the overlay of point cloud renderings on NeRF visuals based on the depth information. As illustrated in Fig. 7, this blended visualization eliminates the black patches commonly observed in point cloud representations by supplementing them with NeRF pixels. Additionally, it handles occlusions in scenarios where dynamically moving hands, rendered as part of the point cloud, are occluded by objects rendered via NeRF as shown in Fig. 8 (Left).

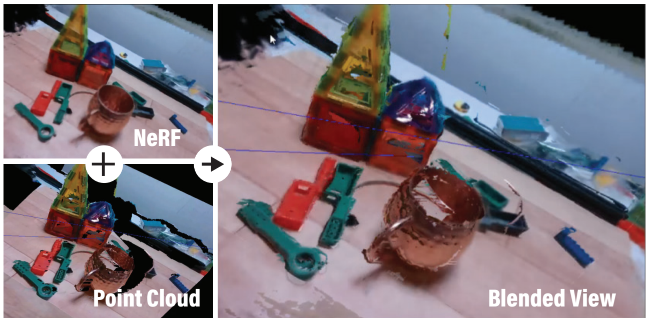

Figure 7: An example of blending NeRF and point cloud rendering. Regions of missing data in the point cloud (black regions) are filled by the NeRF. The NeRF successfully models semi-transparent and specular objects.

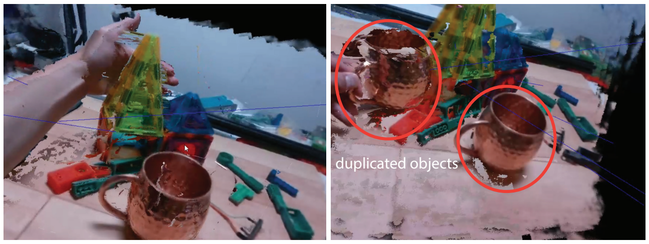

Figure 8: Parts of the dynamic point cloud, such as the user's hands, are occluded by objects modeled by the NeRF (Left). However, when an object previously included in the NeRF model is moved, the object may be duplicated (Right).

However, when local users reposition objects, it could lead to duplicated visuals (Fig. 8 (Right)), which might disorient users. To address this, we explore the utilization of a pixel-based mask to selectively determine the region in the scene that needs to be rendered as a point cloud or as a NeRF (see Fig. 9). The objective is to designate NeRF rendering for static elements of the scene and point cloud rendering for dynamic elements. One strategy to achieve this is as follows: for a given pixel, point cloud rendering is selected when the color value difference for that specific pixel exceeds a predetermined threshold; otherwise, NeRF is used, presuming no

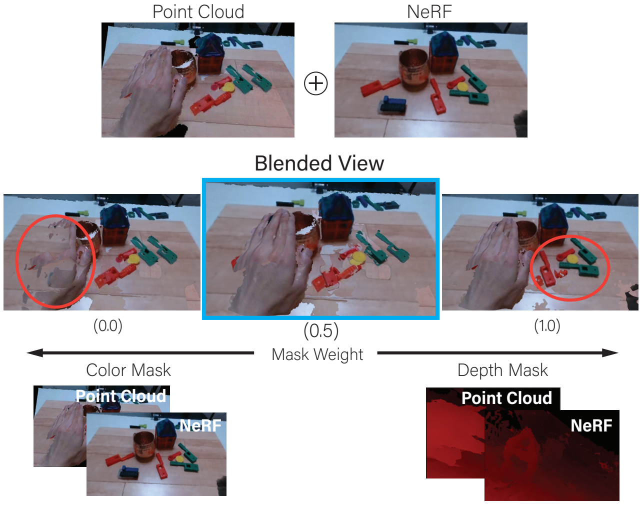

Figure 9: Demonstration of how masks can be used for selecting point cloud rendering for dynamic parts and NeRF rendering for static parts of the scene. Using only a color mask may fail when the corresponding parts of the NeRF and point cloud are similar in color, while using only a depth mask may fail when the change in depth is small. Our composition pipeline uses a weighted average to combine the two types of masks for better composition result.

change has occurred. This strategy is effective in scenarios like relocating a block from one location to another. However, it fails and results in artifacts when regions of movement and change exhibit similar colors. Fig. 9 illustrates an example, where the color of the hand and the table surface is similar, resulting in an incomplete hand rendering. Generating a mask using depth images obtained from NeRF and the point cloud can help in distinguishing objects or hands in the point cloud, for example, for the hands hovering over the scene. This approach encounters difficulties when the depth variations are minimal, such as when small objects are being moved on a table surface. To mitigate each problem found in these approaches, our system combines both masking approaches by using a weighted average, as shown in Fig. 9. For our demonstration and study, we applied a weight of 0.67 for depth and 0.33 for color, with a threshold of 0.11, empirically tuned to suit our setup.

3.5 Supporting Non-verbal Cues

In remote collaboration, rendering *reference space* helps collaborators establish an understanding of what they are referring to in a physical space [2]. In particular, hand gestures, such as pointing or moving things around, play a significant role in conducting physical tasks [33]. With the technique to blend both NeRF and point clouds, the remote viewer can see when a local user moves objects or make hand gestures, such as pointing at a certain object.

Moreover, head pose can also be a powerful indicator of a collaborator's locus of attention [44, 54, 60]. SharedNeRF incorporates a feature that renders the 3D avatar of the local user, indicating the direction of the user's head on the interface, as demonstrated in Fig. 10. The 3D avatar's pose is animated using the same head tracked pose used for dataset collection. This addition helps the remote viewer discern whether the local user is looking at a detail closely or stepping back to gain a broader perspective. The local user can also gauge where the remote viewer is pointing with a cursor or understand the area of their attention, through the shared viewer shown on a display positioned near the workspace. Alternatives include rendering these features using projection mapping or AR glasses (see Section 6.3).

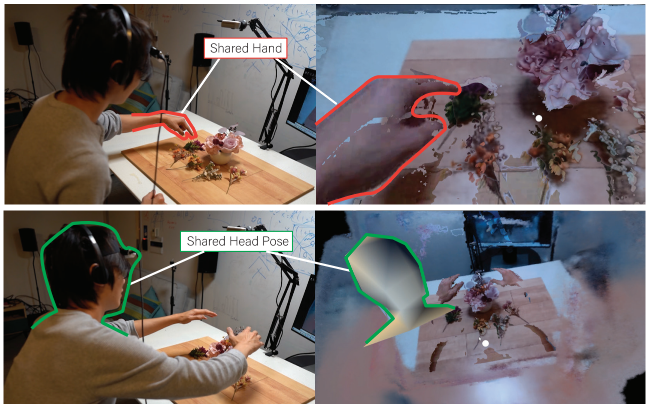

Figure 10: Hand gestures rendered as a point cloud and head direction visualized through the 3D model of an avatar.

3.6 System Architecture

Here, we detail the architecture of the SharedNeRF system, which encompasses both local and remote environments to facilitate the remote collaborative experience (see Fig. 11).

In the local environment, we employ two RealSense D435i cameras: one stationed on a tabletop to capture a broad view of the task space, and a head-mounted camera worn by a user to gather data of the surroundings from different views. This environment hosts two TCP/IP servers to stream data to the remote site. The first server leverages ORB_SLAM3 [4] to track the user's head movements, transmitting a dataset that includes color and depth images, along with the camera pose matrix, in response to HTTP requests from the remote end. The second server similarly manages the color and depth image feed from the stationary camera.

In the remote environment, we implement a GUI application, built with C++ and Direct3D, where the remote viewer navigates a 3D scene using a mouse to zoom in and out to a specific part or rotate around or translate in the scene. They can also use WASD keys on their keyboard to move their viewpoint. This application communicates with the local servers to acquire the data essential for rendering the 3D scene. The rendering pipeline, integrated within the application, renders point clouds based on the received images and camera poses and runs Instant-NGP to train a NeRF network using the data received from the ORB_SLAM3 server. It then renders the NeRF's color and depth textures. These textures, coupled with point cloud and NeRF renderings, are utilized to create a composite mask, merging them with 3D models to generate the final output for the remote viewer. A parallel process within the application utilizes optical flow [11] to identify dynamic objects and alterations in the scene, influencing the management of the training dataset and the final composition.

Our current prototype system relies on the remote system to train NeRF model with Instant-NGP [45]. Because only images and camera pose information are sent over the network, the system lends itself to using conventional video compression and transport technology. An alternative approach would be to perform training on the local user's machine or in the cloud and then send the model, typically less than 100 MB in Instant-NGP, to the remote site, perhaps only when significant changes in the scene are detected. This has the advantage of reducing the remote viewer's computational requirements and naturally allows for efficient sharing of the model among multiple users without duplication of effort. However, periodic but infrequent updates in the model may cause abrupt transitions in rendering. We leave a detailed comparison of these architectures as future work.

The stationary and head-mounted cameras must be located within the same coordinate system. We use a calibration process in which ORB_SLAM3 is used to create a map of the scene with the static camera, and the head-mounted camera is then localized within this map. The process is performed when the prototype system is started.

4 SCENARIOS USING SharedNeRF

In this section, we showcase how SharedNeRF can facilitate interactions central to physical collaborative tasks, particularly those occurring during design critique or review sessions. To this end,

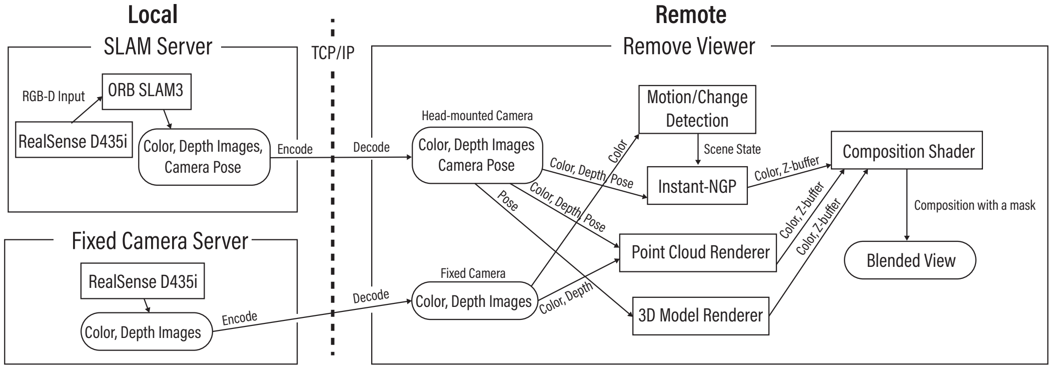

Figure 11: Overview of SharedNeRF's system architecture.

we simulate a pair of collaborative scenarios wherein participants engage remotely in guiding the local user to perform a series of complex interactions with task space objects.

Custom flower arrangement is a scenario characterized by the details of plants and flowers that pose a challenge to conventional point cloud rendering due to their complex textures and geometry. This task not only demands a realistic representation to portray the existing design but also necessitates the ability to view the arrangement from multiple viewpoints. Such a perspective is vital to better grasp the design and to understand the specific areas their partner is referring to during the discussion. Designers are also expected to make alterations to the arrangement, a process followed by a detailed examination of the modified design to facilitate further discussion and review. Another example scenario is inspecting a computer setup, a process that often involves an analysis of various hardware components and their configurations. A remote viewer may inspect the current configurations and instruct the local collaborator on how to fix a problem.

Fig. 12 and Fig. 13 illustrate a local and remote setup as well as the SharedNeRF's viewer controlled by a remote viewer for flower arrangement and computer inspection sessions, respectively. In both scenarios, SharedNeRF offers users the ability to observe the physical task space from different angles interactively. The remote viewer can observe an individual flower closely or have a wide view to grasp the overall design while being able to see where the local user is looking at or pointing (Fig. 12). Likewise, a remote inspector can identify an unplugged cable on the computer and point at it to direct the local person (Fig. 13). Moreover, the system allows for real-time feedback, where users can instantly see the dynamic movements or alternations made by others, such as picking up a flower and adding it to the design or connecting a cable to a GPU.

In addition to collaboration around physical objects with detail and complexity, SharedNeRF may be uniquely suited for applications requiring the examination and manipulation of objects with surface material properties challenging or unfeasible to capture using traditional 3D reconstruction techniques, yet can be effectively modeled by NeRFs. These include translucent materials and materials that change appearance depending on view (e.g., specularity or shininess) (see Fig. 7). For example, the quality of a milled surface may be inspected, or a glass sculpture may be viewed as intended.

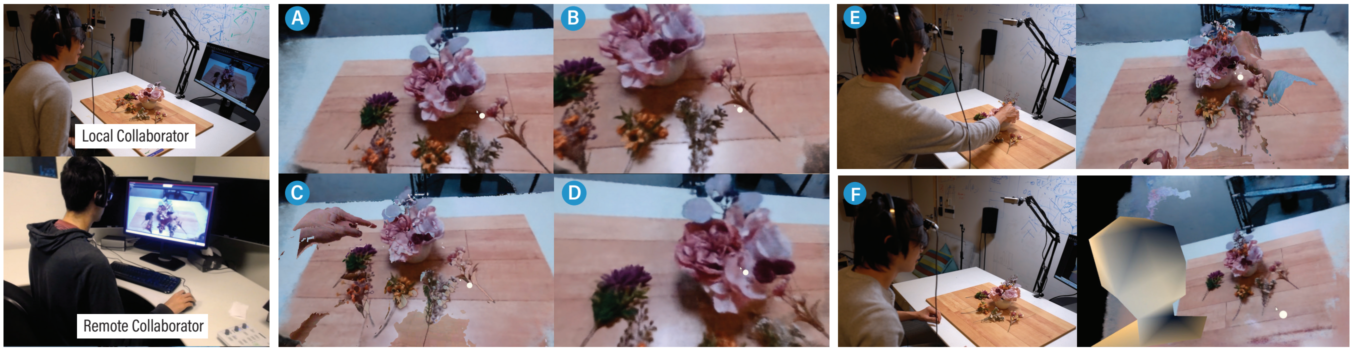

Figure 12: Demonstrating interactions during a flower arrangement session. A remote viewer observes an overview of the overall flower design (A), zooms in to closely look at these flowers on the table that are available (B), sees a local collaborator pointing at a specific flower to explain the flower (C), and the viewer changes viewing angle to see the referenced flower from a side (D). A local user then adds a flower to the existing arrangement (E) and a remote viewer sees the modified design in high fidelity, while being aware of attention of the local collaborator (F).

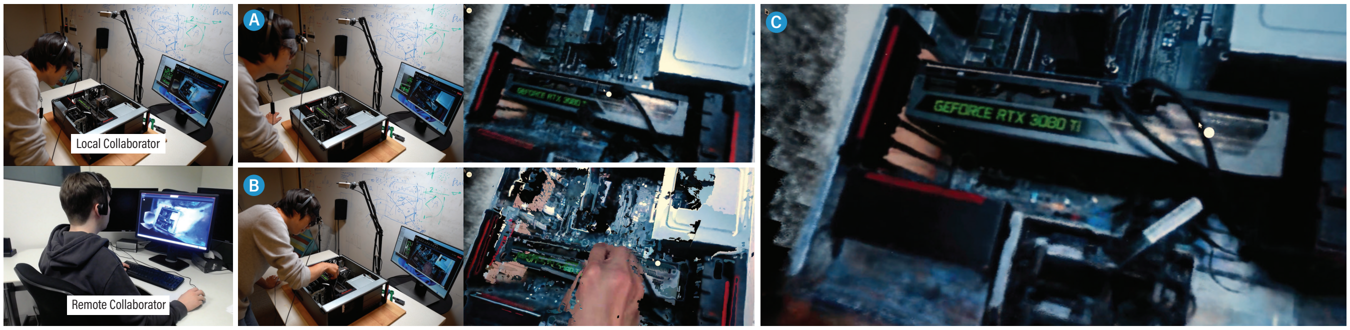

Figure 13: Demonstration of interactions during a PC inspection session using SharedNeRF. A remote viewer observes the configuration inside a desktop PC and points to an unplugged connector (A), the viewer sees the local collaborator connect the cable to the right place (B), and the remote viewer makes sure the change looks correct (C).

5 PRELIMINARY USER STUDY

We conducted a preliminary study to assess the efficacy of SharedNeRF in facilitating a remote user's exploration of a shared task space from various viewpoints of the NeRF representation, coupled with the visualization of dynamic elements via point cloud blending. In particular, we seek to investigate whether users can explore the physical space in high-fidelity and see dynamic movements in the space. This study also aimed to gauge the advantages and disadvantages of our approach relative to common remote collaboration strategies by having participants engage with two other interfaces. We recruited 7 participants (3 male, 4 female), ranging in age from 18 to 34 years old, from our institution, all of whom were compensated with $50 gift card. The participants had prior experience using videoconferencing tools such as Zoom and Teams.

5.1 Task

A similar task to the flower arrangement scenario previously described in Section 4 was used, as it is simple yet requires high-fidelity views for the intricate geometry of the flowers and plants. This task's aesthetic aspect emphasizes the need for visual clarity, aligning with the scenarios with which we aim to test our method. The objective of the task was to work with their partner, an author of the work who wears a head-mounted camera, in another location to craft a flower arrangement that meets a specific client's needs and preferences (e.g., Retirement Party, Wedding Anniversary). Before starting the remote session, the participant was given a card describing the client's information. The client's description was randomly picked for each task and was not revealed to their local partner. Upon the beginning of the remote session, the participant was first presented with a basic flower foundation by the confederate with some explanation about the design and was then asked to carefully observe the existing arrangement and make a comment on the initial design. Then the participant was asked to suggest one specific flower among five different individual flowers to add to the arrangement design as a finishing touch based on the client's request described on their card. After the confederate added the flower they suggested, the participant was asked to examine the final design once again and to make an observation comment on the final design. This process involves complex interactions, with participants actively reviewing, discussing, and suggesting changes to the flower design, and observing their collaborator referencing or altering parts of the design.

Participants engaged with three distinct interfaces throughout the task (Fig. 14): SharedNeRF, which leverages NeRF and point cloud; a first-person video captured through a head-mounted camera; and a point cloud rendered from RGB-D cameras. These conditions were selected to aid in identifying both benefits and drawbacks associated with the system's features: high-fidelity, free viewpoints, and support for dynamic elements. The first-person video condition offers high-fidelity visuals and indicates the collaborator's focus, yet lacks independent free viewpoints. In contrast, the point cloud allows for free viewpoints but compromises on visual quality. The sequence of interface engagement was counterbalanced among participants to avoid order effects.

5.2 Procedure

After completing the informed consent form, participants were informed that they would be interacting with a remote partner in another room. They were then introduced to the first condition. Following the completion of the task for each condition, they participated in a short debrief session to discuss their experience with the interface before filling out a questionnaire. After cycling through all the conditions, they completed a post-study questionnaire, which included sections on their preferences and open-ended questions, followed by a debrief interview. The session concluded with the distribution of the compensation.

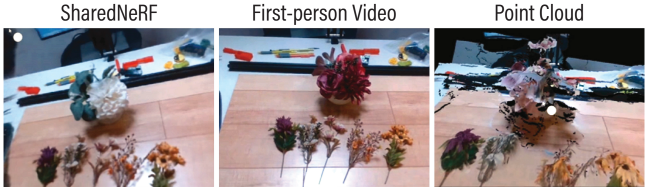

Figure 14: Participants experienced three conditions: SharedNeRF that blends NeRF and point cloud; First-person video from a head-mounted camera; Point cloud using RGB-D cameras.

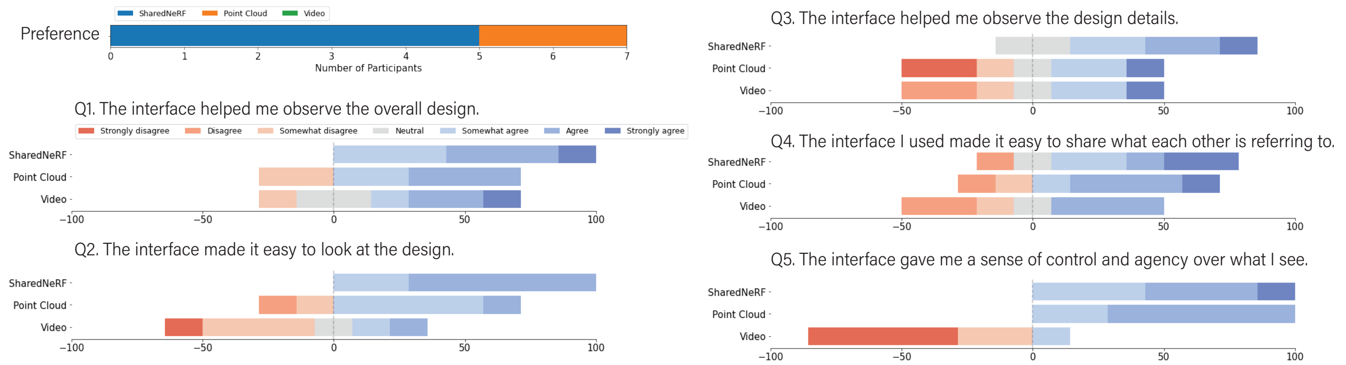

Figure 15: Results of participants' preference and responses to the questionnaire.

5.3 Results and Feedback

Independent View Control: Every participant mentioned the SharedNeRF's ability to independently alter their viewpoints over the flower arrangements, with P1 noting, *"I mostly appreciate having some independent control on the view I have over the flower arrangements..."* Similarly, P4 found that *"its option to scroll in and out really enabled me to look at the design in more detail. At times I wasn't sure about the arrangement when it was zoomed out, so it was nice to be able to zoom in on the flower I added on my own to evaluate the design."* Participants frequently highlighted the independent ability to control viewpoints from a coordination perspective. P4 noted *"The accomplishing of the micro goals such as like zooming in to look at the details when I wanted to just like on my own without having to bother (the partner)."* They enjoyed the freedom to *"do what I wanted in the scene without having to coordinate or depend on anyone else."* P6 also appreciated the ability to *"manipulate the scene with my mouse without having to tell my collaborator anything,"*. P7 also echoed on this saying *"I don't need to like describe to my partner what I want and sometimes ... I just want to look at it myself, ...instead of (asking to) move the camera here, move the camera there."*, comparing it to the first-person video condition. As shown in Fig. 15, the sense of control and agency over what is seen is reflected in the questionnaire results.

Visual Fidelity: Participants commented on the visual fidelity of the SharedNeRF. P1 felt that SharedNeRF allowed for a better sense of the design compared to the point cloud, saying *"it is easier to see the structure than the point cloud condition. .. also get a complete image ... without big black holes in the rendering (that you see in point cloud)."* P3 found it useful *"to zoom in and see the shapes of the flowers"*. P5 also echoed this point saying *"I think it did a great job of letting me observe such details."* P7 commented on the SharedNeRF condition for clear details, finding no instability problems they saw in the point cloud condition, and providing a *"really realistic view."* This positive sentiment towards visual fidelity is consistent with their questionnaire responses regarding the ability to observe both the overall design and its details (see Fig. 15). Although these comments attest to the high fidelity nature of NeRF used in SharedNeRF, participants also mentioned the fact that visual fidelity varies based on viewpoints. While P4 noted *"the center area was ... very high quality."*, P1 commented *"some angles are better than others"* and that *"I was really hoping it would be equally clear on the sides."* P6 also noted that *"I wasn't able to see the back side of the design."* P1 described a first-person view video as the interface that *"gave the clearest image."* Another limitation noted by the participants is its delay in updating NeRF representation in response to scene changes. For example, P3 noted *"it took a bit to load the the flower"* and P6 also said *"it took a little bit of time to sort of reload with the new arrangement."*

Visibility of Dynamic Elements: Participants also appreciated the benefits of being able to see point cloud rendering over NeRF for instant feedback. P4 said *"even though it did take a little bit to update (the NeRF), ... having the live updates (via point cloud) made it easier also for me to kind of figure out like what was going on."* P2 also commented *"I could see the local person's head and hands, and movements"*. Likewise, P1 mentioned being able to comprehend *"what he (the remote user) was gesturing to."* P5 and P7 noted that they could see the user's hand when brought into the scene. Questionnaire responses regarding the understanding of what each other is referring to, in Fig. 15, indicates a similar appreciation of the value of the point cloud. Despite these positive comments on the ability to see dynamic parts of the space, participants mentioned the limitations of this feature. P7 did not see the moving parts *"as clearly as static objects like flowers"*. P1 compared the SharedNeRF condition to the video condition saying *"it was not easy to see changes/moving objects as quickly as a live video stream."* P4 also noted that *"If the object didn't stay in the scene, it would cause some blur, and then the scene would return to normal (after movement is no longer detected)."*

Although the findings from this study are insightful, highlighting both the strengths and weaknesses of the proposed method, it is important to note that this was an initial, preliminary study with a limited sample size. Additionally, the local side was not examined due to the focus of our work. To gain a more comprehensive understanding of the system, future research is needed to involve more extensive, long-term studies across varied tasks.

6 DISCUSSION AND FUTURE WORK

6.1 Improving Visual Fidelity

Participants noted a desire for improved image quality, such as a sharper image when zooming in. This issue may be attributable

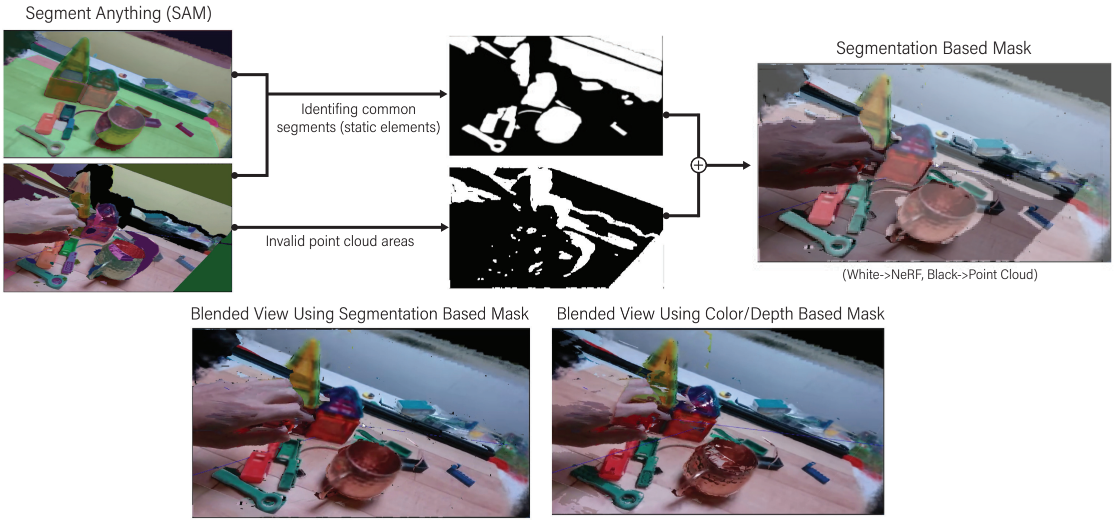
Segment Anything (SAM)
Identifing common segments (static elements)
Invalid point cloud areas
Segmentation Based Mask
(White->NeRF, Black->Point Cloud)
Blended View Using Segmentation Based Mask
Blended View Using Color/Depth Based Mask

Figure 16: Experiment of an alternative approach for creating a mask using segmentation.

to the resolution of the images utilized in our dataset. In the interest of optimizing performance, our prototype uses images with a resolution of 640 x 480 pixels for training purposes, while NeRF rendering is performed at a native resolution of 512 x 512 to ensure good performance for the viewer.

Another factor could be the size that defines the number of images used in NeRF training. As discussed in Section 3.2.1, increasing this size could increase the quality of NeRF but may decrease the rate of NeRF updates. Recent advancements in training speed and rendering speed for volumetric representations may improve both the quality and speed of such high-fidelity rendering [31]. Participants mentioned the desire to view the artifacts from the sides or back where it was not captured by the head-mounted camera. To support this, one can consider introducing more cameras for data collection and optimizing camera arrangement for improvement of the NeRF quality for these parts [34]. The following section also discusses alternative data collection methods to be explored for capturing diverse viewpoints.

While participants appreciated the dynamic elements in the scene, they suggested a need for clearer rendering in the blended view. Our weighted average approach, as detailed in Section 3.4, addresses some issues associated with basic color or depth masking. However, it sometimes results in flickering artifacts, potentially caused by noisy point clouds affecting the visual quality. We used a RealSense camera (D435i) with active stereo depth for ergonomic reasons, rather than higher-quality time of flight (ToF) based depth devices. Future improvements could include the use of a compact ToF camera. Incorporating hand tracking and rendering a rigged avatar into the blended view could further improve the clarity of dynamic person space representation. Additionally, we observed that when the color and depth of dynamic scene elements are similar in both NeRF and point cloud renderings, the quality of the mask may be reduced. As an alternative to our initial masking approach, employing more sophisticated segmentation algorithms could minimize artifacts in the blended view. To explore this, we used Segment Anything (SAM) [32] to perform segmentation on color images from both NeRF and point cloud, identifying common regions combined with invalid point cloud areas to create a mask. Our preliminary results, illustrated in Fig. 16, are encouraging, effectively reducing many artifacts and holes, such as those over a hand or inside a cup. Although these algorithms today may be too computationally demanding for real-time use, we expect them to become faster soon. Integrating this approach into our method for mask creation could significantly enhance the visual quality of the blended view.

6.2 Exploring Data Collection Methods

We utilize a single head-mounted camera to collect a dataset for NeRF training. While this approach serves to enable the demonstrated scenario, there exists a substantial design space to investigate alternative strategies for on-the-fly data collection for real-time collaboration. For example, the system operates under the assumption that both remote and local collaborators mostly work nearby, generally facing the same direction. However, collaborative environments can have a variety of configurations such as working across a table to maintain eye contact and facilitate non-verbal communication. To better accommodate these various forms of *collaborative coupling* [65], future work can explore alternative data collection methods that can offer diverse viewpoints. One such strategy could involve utilizing mechanisms like linear actuators or robotic arms to position cameras at different angles, such as opposite ends of a table, thereby optimizing the NeRF dataset based on the region of interest the remote viewer intends to focus on. Furthermore, in situations where multiple individuals are present in the local environment, equipping each person with a head-mounted camera could significantly enhance the dataset by introducing a variety of perspectives. This approach, while promising, can highlight the need to more effectively segment the training images to exclude dynamic elements, possibly drawing on more advanced techniques to remove other individuals from the training dataset for clear NeRF results.

6.3 Alternative Interfaces for Remote and Local Collaborators

In our prototype, we develop a GUI viewer that allows remote users to navigate a 3D scene using a mouse and keyboard, a standard interface in 3D editing tools or game engines. However, for tasks centered around physical objects, liberating users from the need to explicitly manage camera viewpoints can free up cognitive resources, enabling them to concentrate more fully on the primary task [55, 60, 61]. A promising alternative would be a VR interface where a remote user wearing a head-mounted display can freely change viewpoints through head movements in the SharedNeRF's rendering, and can leverage hand gestures or spatial annotations for better communicating *reference space* to the local user. Of course, the ability to more flexibly change viewpoints would encourage a remote user to look from angles where adequate data is not collected, but data collection methods discussed in Section 6.2 could be considered to address this.

For the local collaborator our demonstration and study used a shared screen setup to visualize what the remote reviewer is seeing. While this method is prevalent in daily video calls, aiding in establishing a common ground among collaborators [35], it requires the local user to shift their focus between their own task space and the shared screen, potentially fragmenting their attention. A more seamless solution would be to overlay the remote viewer's attention area directly onto the local user's physical task space. This could be achieved, for instance, through the utilization of a calibrated laser pointer or projection mapping to indicate the exact area the remote viewer is focusing on [18]. Alternatively, a local user can wear an AR head-mounted display to render 3D information about the notion of the remote collaborator over their immediate physical environment. One benefit of these approaches is that they can implicitly encourage the local user to look at the task space from viewpoints that the remote viewer would like to see, which can result in gathering a better dataset for optimizing NeRF representation for these viewpoints.

6.4 Accommodating More Interaction Modalities

The user experience of using SharedNeRF for physical tasks can be augmented by adding more interactive components to the blended space. For example, users could add annotation or virtual objects, such as CAD files over the high-fidelity NeRF representation, as explored in Magic NeRF Lens [37]. A key benefit of enabling these interactions in SharedNeRF is the fact it renders both NeRF and point cloud in real-world metrics (e.g., meter), as opposed to an arbitrary scale used in Magic NeRF Lens. This allows users to import and place CAD models with correct dimensions in the NeRF space and check if a modeled part fits with an existing physical object, for example. Fig. 17 shows the example of using SharedNeRF for a scenario where the fit of a CAD model of a camera mount is checked against the camera it is designed to work with. The combined point cloud and NeRF rendering allows the user to refer to both virtual and physical objects in a unified rendering space.

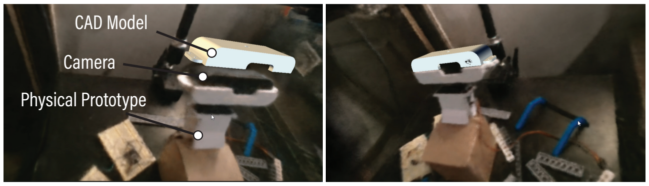

Figure 17: A CAD model can be rendered with actual dimensions in NeRF rendering and moved over the physical prototype to ensure that it fits well with the camera before it takes a long time to 3D print it.

7 CONCLUSION

SharedNeRF is designed to enhance synchronous remote collaboration during tasks centered around physical objects by leveraging high fidelity and view-dependent synthesis of a volumetric NeRF representation. The system exploits the complementary nature of NeRF representations and RGB-D point clouds, blending the rendering of both static and dynamic elements of the shared space. The system also uses algorithms to improve NeRF quality over time and update the NeRF representation upon permanent changes to the space. In our preliminary study, participants were able to complete a flower arrangement task, noting SharedNeRF's benefits of independent control over viewpoints and realistic rendering as well as its ability to visualize hands moving in the scene.

ACKNOWLEDGMENTS

We would like to thank Cyrus Vachha and Nels Numan for helping us with the production of our demo video. We also thank Lex Story for assisting with a hardware prototype. Finally, we would like to thank the study participants for their time.

REFERENCES

[1] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 199–208. https://doi.org/10.1145/2207676.2207704

[2] Bill Buxton. 2009. *Mediaspace – Meaningspace – Meetingspace*. Springer London, London, 217–231. https://doi.org/10.1007/978-1-84882-483-6_13

[3] William A. S. Buxton. 1992. Telepresence: Integrating Shared Task and Person Spaces. In *Proceedings of the Conference on Graphics Interface '92* (Vancouver, British Columbia, Canada). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 123–129.

[4] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* 37, 6 (2021), 1874–1890. https://doi.org/10.1109/TRO.2021.3075644

[5] Herbert H Clark and Catherine R Marshall. 1981. Definite knowledge and mutual knowledge. (1981).

[6] Mike Daily, Mike Howard, Jason Jerald, Craig Lee, Kevin Martin, Doug McInnes, and Pete Tinker. 2000. Distributed Design Review in Virtual Environments. In *Proceedings of the Third International Conference on Collaborative Virtual Environments* (San Francisco, California, USA) *(CVE '00)*. Association for Computing Machinery, New York, NY, USA, 57–63. https://doi.org/10.1145/351006.351013

[7] Steven P. Dow and Scott R. Klemmer. 2011. *The Efficacy of Prototyping Under Time Constraints*. Springer Berlin Heidelberg, Berlin, Heidelberg, 111–128. https://doi.org/10.1007/978-3-642-13757-0_7

[8] Romina Druta, Cristian Druta, Paul Negirla, and Ioan Silea. 2021. A Review on Methods and Systems for Remote Collaboration. *Applied Sciences* 11, 21 (2021). https://doi.org/10.3390/app112110035

[9] Omid Fakourfar, Kevin Ta, Richard Tang, Scott Bateman, and Anthony Tang. 2016. Stabilized Annotations for Mobile Remote Assistance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1548–1560. https://doi.org/10.1145/2858036.2858171

[10] Mehrad Faridan, Bheesha Kumari, and Ryo Suzuki. 2023. ChameleonControl: Teleoperating Real Human Surrogates through Mixed Reality Gestural Guidance for Remote Hands-on Classrooms. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 203, 13 pages. https://doi.org/10.1145/3544548.3581381

[11] Gunnar Farnebäck. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*, Josef Bigun and Tomas Gustavsson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 363–370.

[12] Susan R. Fussell, Robert E. Kraut, and Jane Siegel. 2000. Coordination of Communication: Effects of Shared Visual Context on Collaborative Work. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, USA) *(CSCW '00)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/358916.358947

[13] Susan R. Fussell, Leslie D. Setlock, and Robert E. Kraut. 2003. Effects of Head-Mounted and Scene-Oriented Video Systems on Remote Collaboration on Physical Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 513–520. https://doi.org/10.1145/642611.642701

[14] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. 2021. Dynamic View Synthesis from Dynamic Monocular Video. In *Proceedings of the IEEE International Conference on Computer Vision.*

[15] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. World-Stabilized Annotations and Virtual Scene Navigation for Remote Collaboration. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 449–459. https://doi.org/10.1145/2642918.2647372

[16] William W. Gaver, Abigail Sellen, Christian Heath, and Paul Luff. 1993. One is Not Enough: Multiple Views in a Media Space. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 335–341. https://doi.org/10.1145/169059.169268

[17] Darren Gergle. 2005. The Value of Shared Visual Space for Collaborative Physical Tasks. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) *(CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1116–1117. https://doi.org/10.1145/1056808.1056839

[18] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. 2012. TeleAdvisor: A Versatile Augmented Reality Tool for Remote Assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 619–622. https://doi.org/10.1145/2207676.2207763

[19] C. Gutwin and S. Greenberg. 2004. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW)* 11 (2004), 411–446.

[20] Keita Higuch, Ryo Yonetani, and Yoichi Sato. 2016. Can Eye Help You? Effects of Visualizing Eye Fixations on Remote Collaboration Scenarios for Physical Tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5180–5190. https://doi.org/10.1145/2858036.2858438

[21] Erzhen Hu, Jens Emil Sloth Grønbæk, Wen Ying, Ruofei Du, and Seongkook Heo. 2023. ThingShare: Ad-Hoc Digital Copies of Physical Objects for Sharing Things in Video Meetings. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI)*. ACM. https://doi.org/10.1145/3544548.3581148

[22] Autodesk Inc. 2023. Design Review | DWF Viewer. Retrieved Sep 4, 2023 from https://www.autodesk.com/products/design-review/overview

[23] Luma AI Inc. 2023. Luma AI. Retrieved Sep 4, 2023 from https://lumalabs.ai/

[24] Andrew Irlitti, Mesut Latifoglu, Qiushi Zhou, Martin N Reinoso, Thuong Hoang, Eduardo Velloso, and Frank Vetere. 2023. Volumetric Mixed Reality Telepresence for Real-Time Cross Modality Collaboration. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 101, 14 pages. https://doi.org/10.1145/3544548.3581277

[25] H. Ishii. 1990. TeamWorkStation: Towards a Seamless Shared Workspace. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work* (Los Angeles, California, USA) *(CSCW '90)*. Association for Computing Machinery, New York, NY, USA, 13–26. https://doi.org/10.1145/99332.99337

[26] Hiroshi Ishii and Minoru Kobayashi. 1992. ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, USA) *(CHI '92)*. Association for Computing Machinery, New York, NY, USA, 525–532. https://doi.org/10.1145/142750.142977

[27] Steven Johnson, Madeleine Gibson, and Bilge Mutlu. 2015. Handheld or Handsfree? Remote Collaboration via Lightweight Head-Mounted Displays and Handheld Devices. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Vancouver, BC, Canada) *(CSCW '15)*. Association for Computing Machinery, New York, NY, USA, 1825–1836. https://doi.org/10.1145/2675133.2675176

[28] Mitchell Karchemsky, J.D. Zamfirescu-Pereira, Kuan-Ju Wu, François Guimbretière, and Bjoern Hartmann. 2019. Heimdall: A Remotely Controlled Inspection Workbench For Debugging Microcontroller Projects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300728

[29] Shunichi Kasahara, Shohei Nagai, and Jun Rekimoto. 2014. LiveSphere: Immersive Experience Sharing with 360 Degrees Head-Mounted Cameras. In *Adjunct Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 61–62. https://doi.org/10.1145/2658779.2659114

[30] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. 2023. SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM. *arXiv preprint* (2023).

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[33] David Kirk and Danae Fraser. 2006. Comparing remote gesture technologies for supporting collaborative physical tasks, Vol. 2. 1191–1200. https://doi.org/10.1145/1124772.1124951

[34] Georgios Kopanas and George Drettakis. 2023. Improving NeRF Quality by Progressive Camera Placement for Unrestricted Navigation in Complex Environments. arXiv:2309.00014 [cs.CV]

[35] Robert E. Kraut, Susan R. Fussell, and Jane Siegel. 2003. Visual Information as a Conversational Resource in Collaborative Physical Tasks. *Hum.-Comput. Interact.* 18, 1 (jun 2003), 13–49. https://doi.org/10.1207/S15327051HCI1812_2

[36] Jiannan Li, Maurício Sousa, Chu Li, Jessie Liu, Yan Chen, Ravin Balakrishnan, and Tovi Grossman. 2022. ASTEROIDS: Exploring Swarms of Mini-Telepresence Robots for Physical Skill Demonstration. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 111, 14 pages. https://doi.org/10.1145/3491102.3501927

[37] Ke Li, Susanne Schmidt, Tim Rolff, Reinhard Bacher, Wim Leemans, and Frank Steinicke. 2023. Magic NeRF Lens: Interactive Fusion of Neural Radiance Fields for Virtual Facility Inspection. arXiv:2307.09860 [cs.GR]

[38] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. 2022. Neural 3D Video Synthesis from Multi-view Video. 5511–5521. https://doi.org/10.1109/CVPR52688.2022.00544

[39] Christian Licoppe, Paul K. Luff, Christian Heath, Hideaki Kuzuoka, Naomi Yamashita, and Sylvaine Tuncer. 2017. Showing Objects: Holding and Manipulating Artefacts in Video-Mediated Collaborative Settings. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5295–5306. https://doi.org/10.1145/3025453.3025848

[40] T. Machino, S. Iwaki, H. Kawata, Y. Yanagihara, Y. Nanjo, and K.-i. Shimokura. 2006. Remote-collaboration system using mobile robot with camera and projector. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* 4063–4068. https://doi.org/10.1109/ROBOT.2006.1642326

[41] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I* (Glasgow, United Kingdom). Springer-Verlag, Berlin, Heidelberg, 405–421. https://doi.org/10.1007/978-3-030-58452-8_24

[42] Kana Misawa and Jun Rekimoto. 2015. ChameleonMask: Embodied Physical and Social Telepresence Using Human Surrogates. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. Association for Computing Machinery, New York, NY, USA, 401–411. https://doi.org/10.1145/2702613.2732506

[43] Terrance Mok and Lora Oehlberg. 2017. Critiquing Physical Prototypes for a Remote Audience. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) *(DIS '17)*. Association for Computing Machinery, New York, NY, USA, 1295–1307. https://doi.org/10.1145/3064663.3064722

[44] Pieter Moors, Filip Germeys, Iwona Pomianowska, and Karl Verfaillie. 2015. Perceiving where another person is looking: the integration of head and body information in estimating another personâs gaze. *Frontiers in Psychology* 6 (2015).

https://doi.org/10.3389/fpsyg.2015.00909

[45] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

[46] James Norris, Holger Schnädelbach, and Guoping Qiu. 2012. CamBlend: An Object Focused Collaboration Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/2207676.2207765

[47] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-Time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 741–754. https://doi.org/10.1145/2984511.2984517

[48] S. Park, M. Son, S. Jang, Y. Ahn, J. Kim, and N. Kang. 2023. Temporal Interpolation is all You Need for Dynamic Neural Radiance Fields. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 4212–4221. https://doi.org/10.1109/CVPR52729.2023.00410

[49] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. 2016. Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (San Francisco, California, USA) *(CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1716–1725. https://doi.org/10.1145/2818048.2819965

[50] Thammathip Piumsomboon, Gun A. Lee, Jonathon D. Hart, Barrett Ens, Robert W. Lindeman, Bruce H. Thomas, and Mark Billinghurst. 2018. Mini-Me: An Adaptive Avatar for Mixed Reality Remote Collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173620

[51] Thammathip Piumsomboon, Youngho Lee, Gun Lee, and Mark Billinghurst. 2017. CoVAR: A Collaborative Virtual and Augmented Reality System for Remote Collaboration. In *SIGGRAPH Asia 2017 Emerging Technologies* (Bangkok, Thailand) *(SA '17)*. Association for Computing Machinery, New York, NY, USA, Article 3, 2 pages. https://doi.org/10.1145/3132818.3132822

[52] Pragathi Praveena, Yeping Wang, Emmanuel Senft, Michael Gleicher, and Bilge Mutlu. 2023. Periscope: A Robotic Camera System to Support Remote Physical Collaboration. *ArXiv* abs/2305.07517 (2023). https://api.semanticscholar.org/CorpusID:258676170

[53] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[54] Xun Qian, Feitong Tan, Yinda Zhang, Brian Collins, Alex Olwal, David Kim, Karthik Ramani, and Ruofei Du. 2024. ChatDirector: Enhancing Video Conferencing With Space-Aware Scene Rendering and Speech-Driven Layout Transition. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 1–12. https://doi.org/10.1145/3613904.3642110

[55] Irene Rae, Bilge Mutlu, and Leila Takayama. 2014. Bodies in Motion: Mobility, Presence, and Task Awareness in Telepresence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2153–2162. https://doi.org/10.1145/2556288.2557047

[56] Abhishek Ranjan, Jeremy P. Birnholtz, and Ravin Balakrishnan. 2006. An Exploratory Analysis of Partner Action and Camera Control in a Video-Mediated Collaborative Task. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work* (Banff, Alberta, Canada) *(CSCW '06)*. Association for Computing Machinery, New York, NY, USA, 403–412. https://doi.org/10.1145/1180875.1180936

[57] Abhishek Ranjan, Jeremy P. Birnholtz, and Ravin Balakrishnan. 2007. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1177–1186. https://doi.org/10.1145/1240624.1240802

[58] Antoni Rosinol, John J Leonard, and Luca Carlone. 2022. NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields. *arXiv preprint arXiv:2210.13641* (2022).

[59] Mose Sakashita, Hyunju Kim, Brandon Woodard, Ruidong Zhang, and François Guimbretière. 2023. VRoxy: Wide-Area Collaboration From an Office Using a VR-Driven Robotic Proxy. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 59, 13 pages. https://doi.org/10.1145/3586183.3606743

[60] Mose Sakashita, E. Andy Ricci, Jatin Arora, and François Guimbretière. 2022. RemoteCoDe: Robotic Embodiment for Enhancing Peripheral Awareness in Remote Collaboration Tasks. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW1, Article 63 (apr 2022), 22 pages. https://doi.org/10.1145/3512910

[61] Mose Sakashita, Ruidong Zhang, Xiaoyi Li, Hyunju Kim, Michael Russo, Cheng Zhang, Malte F. Jung, and François Guimbretière. 2023. ReMotion: Supporting Remote Collaboration in Open Space with Automatic Robotic Embodiment. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 363, 14 pages. https://doi.org/10.1145/3544548.3580699

[62] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[63] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. 2013. BeThere: 3D Mobile Collaboration with Spatial Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, USA, 179–188. https://doi.org/10.1145/2470654.2470679

[64] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. 2023. NeRFPlayer: A Streamable Dynamic Scene Representation with Decomposed Neural Radiance Fields. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2732–2742. https://doi.org/10.1109/TVCG.2023.3247082

[65] Anthony Tang, Melanie Tory, Barry Po, Petra Neumann, and Sheelagh Carpendale. 2006. Collaborative Coupling over Tabletop Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1181–1190. https://doi.org/10.1145/1124772.1124950

[66] John C. Tang and Scott Minneman. 1991. VideoWhiteboard: Video Shadows to Support Remote Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New Orleans, Louisiana, USA) *(CHI '91)*. ACM, New York, NY, USA, 315–322. https://doi.org/10.1145/108844.108932

[67] Theophilus Teo, Louise Lawrence, Gun A. Lee, Mark Billinghurst, and Matt Adcock. 2019. Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300431

[68] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 161–174. https://doi.org/10.1145/3332165.3347872

[69] Balasaravanan Thoravi Kumaravel and Björn Hartmann. 2022. Interactive Mixed-Dimensional Media for Cross-Dimensional Collaboration in Mixed Reality Environments. *Frontiers in Virtual Reality* 3 (2022). https://doi.org/10.3389/frvir.2022.766336

[70] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Björn Hartmann. 2019. TutoriVR: A Video-Based Tutorial System for Design Applications in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300514

[71] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Bjoern Hartmann. 2020. TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 182–195. https://doi.org/10.1145/3379337.3415827

[72] Balasaravanan Thoravi Kumaravel and Andrew D Wilson. 2022. DreamStream: Immersive and Interactive Spectating in VR. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 636, 17 pages. https://doi.org/10.1145/3491102.3517508

[73] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peppler, and Karthik Ramani. 2021. RobotAR: An Augmented Reality Compatible Teleconsulting Robotics Toolkit for Augmented Makerspace Experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 477, 13 pages. https://doi.org/10.1145/3411764.3445726

[74] Svetlana Yarosh, Kori M. Inkpen, and A.J. Bernheim Brush. 2010. Video Playdate: Toward Free Play across Distance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1251–1260. https://doi.org/10.1145/1753326.1753514