# AvatarPilot : Decoupling one-to-one motions from their semantics with weighted interpolations

Cheng Yao Wang* Cornell University
Eyal Ofek† Microsoft Research
Hyunju Kim‡ Cornell University
Payod Panda§ Microsoft Research
Andrea Stevenson Won¶ Cornell University
Mar Gonzalez Franco‖ Microsoft Research

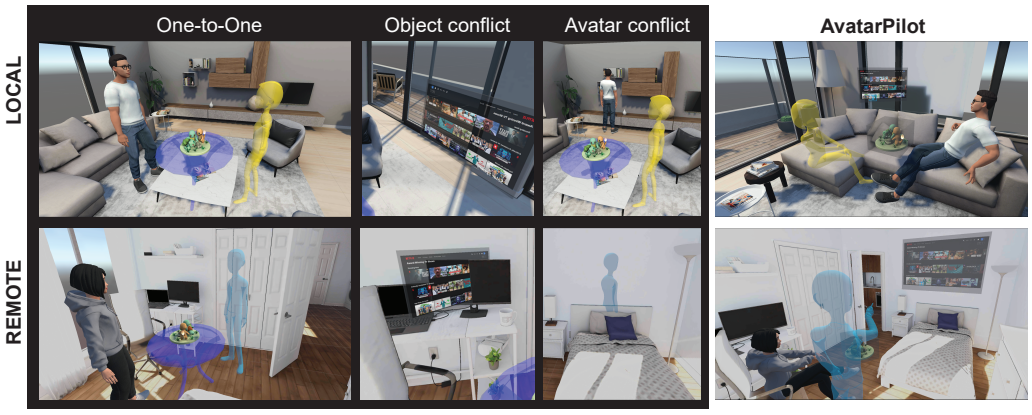


Figure 1: When we use any form of one-to-one mapping of the local and remote environment, we will likely incur object or avatar conflicts. Using Avatar Pilot, we avoid these undesirable effects while maintaining the semantics.

## Abstract

Physical restrictions of the real spaces where users are situated present challenges to remote XR and spatial computing interactions using avatars. Users may not have space in their physical environment to duplicate the physical set-up of their collaborators. Still, if avatars are relocated, one-to-one motions may no longer preserve meaning. We propose a solution: using weighted interpolations to guarantee that everybody looks or points at the same objects locally and remotely. At the same time, this preserves the meaning of gestures and postures that are not object-directed (i.e., close to the body). We extend this work to locomotion and direct interactions in near space, such as grabbing objects, exploring the limits of our social and scene understanding, and generating more flexible uses for Inverse Kinematics (IK). We discuss limitations and applications and open-source the AvatarPilot for general use.

**Index Terms:** Collaboration, Virtual Reality, Spatial Computing, Social XR

## 1 Introduction

One of the wonders of immersive computing is that it allows us to share our realities with faraway people in a more realistic way . Whether joining friends to watch a movie or attending a work meeting, activities are richer and more engaging when users can break away from 2D screens and use spatial computing [9].

*e-mail:chengyao.wang@jpmchase.com
†e-mail: eyal.ofek@gmail.com
‡e-mail: hk724@cornell.edu
§payod.panda@microsoft.com
¶asw248@cornell.edu
‖margon@google.com

Shared VR experiences allow local users to interact with remote users through virtual avatars that represent them as if they were in the same space [13]. However, implementing such experiences is challenging when users inhabit dissimilar spaces [7]. For example, if users have their feet up on a sofa at home while connecting remotely, should their avatar also be lying on a sofa in a remote location? This approach might work for casual interactions but not for formal ones like business meetings, where avatars should align with the setting, such as sitting on a conference chair [18].

This complexity increases when experiences use elements of the physical space, such as superimposing a virtual whiteboard over a physical one while maintaining congruence. A one-to-one mapping of the remote user's motion to their avatar in the local space is often inadequate and can lead to confusion. For example, if an avatar points to its right, it might not be clear if the user is pointing to their right or at an object in their space, like a whiteboard [5]. This can create impossible scenarios where avatars or shared objects appear in unrealistic positions.

To address these challenges, we propose that physical space encodes user intent through its semantics, enabling effective collaboration over physicality [14]. Instead of mapping "pointing right" to the remote collaborator's space, we map "pointing to the whiteboard" in the remote collaborator's space. We move away from hard segmentation and discrete regions by defining a master transformation function with weighted interpolation that adapts avatar pose, gaze, or grasp at any point in the space [5, 18, 7].

Our approach identifies targets in the environment that are relevant for interaction and develops an algorithm that maps remote avatar movements in dissimilar local spaces. This continuous and interpolating method allows for remapping user interactions in environments with asymmetrically arranged targets. We implemented this algorithm in our system, which enables synchronous VR remote collaboration among participants in physically asymmetrical spaces.

Previous research has addressed the realignment of physical and

remote spaces by segmenting space into discrete regions or scanning participants' rooms to find minimal empty spaces for interaction [7]. However, such methods often fail in dynamic environments or when shared spaces become impractical [6, 15]. Moreover, maintaining a high level of semantic fidelity in interactions is crucial for natural and smooth collaboration [5, 3].

To avoid unnatural scenarios, some work has changed interaction speeds or redirected users to reduce collisions [11, 2], while others have restricted avatar movements to less strange positions [12]. However, these methods can disrupt the semantics of the avatar, affecting nonverbal interactions like pointing or looking.

Our key contributions include a novel bijective transform function that decouples user motion from interaction semantics and can be updated in real-time. Our system segments asymmetrical spaces into customizable local-remote Interaction Zone pairs, preserving interaction context in remote spaces. We provide the system's codebase as an open-source repository, inviting reuse and community engagement.

## 2 AvatarPilot Implementation

The goal of AvatarPilot is to generate avatar movements that preserve the interaction context, including where users are gazing, pointing, and their spatial relation to interaction areas in the space.

To achieve this, we first need to track both the user's actions and the objects they relate to in the environment. This is as easy as traveling through the hierarchy of objects. In VR, that might mean just the list of objects in the scene. In AvatarPilot we focus only on virtual objects. Still, it can be extendable to AR, but that depends on the scene understanding algorithms and the capacity to scan and synthesize the objects in remote sites. Either way, once we can track and detect collisions with objects, we can consider all problems of user interaction in its environment as a relational problem of remapping between local and remote spaces.

Figure 2 displays how different sites and objects can be combined into one continuous mapping. $A$ and $B$ are objects on site I (local) and are matched to objects of $A^*$ and $B^*$ in site II (remote). For AvatarPilot , each interaction with collaborative virtual objects in the local user site can be associated with the corresponding one in the remote collaborator's site. Each matching pair of objects between the sites, $A$ and $A^*$, define a rigid 3D bijective transformation between two algebras $T_A$ from site I to Site II that maps the coordinate system of $A$ to $A^*$. All these transformations are computed in real time.

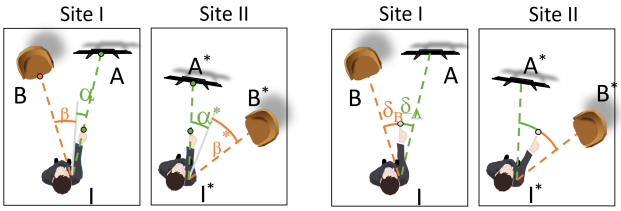


Figure 2: The proposed mapping smoothly interpolates between separate mappings generated by matching objects between sites.

For simplicity, we will first look at a 1D example of mapping hand-pointing. In figure 2, both ends have a sofa and a whiteboard. (a) The user in site I is pointing at the whiteboard $A$. We can represent the pointing angle as an angle $\alpha$ from a given direction, such as the user front (gray line). We define a 1D transformation $T_A$ to map the angle $\alpha$ to a new angle $\alpha^*$. When the avatar points at angle $\alpha^*$, its hand will point directly at the corresponding whiteboard in site II. However, $T_A$ is defined for all angle values and not just for the direction of the whiteboard. A pointing angle $\gamma$ in site I will be mapped to pointing in angle $\gamma-\alpha+\alpha^*$ in site II. In this example, defined at the user's coordinate system, the definitions of $T_A$ and $T_B$ are updated as the user or the avatar moves in their environment.

In the same way, matching the interaction area associated with sofa $B$ in site I to a corresponding one in site II will generate its 1D transformation of pointing angles $T_B$. We now define a compound mapping transformation $T$, which will be similar to $T_A$ when the user points towards $A$, similar to $T_B$ when the user points towards $B$, and do a weighted interpolation of these transforms for other pointing angles:

$$T = \sum_i (w_i * T_i) / \sum_i (w_i) \quad (1)$$

The weights $w_i$ are a function of a parameter (or parameters) that measures the distance (e.g., angular distance or Euclidean distance) from the state where $T_i$ is defined. Figure 2 (b) shows an interpolated mapping, where the weight of $T_A$ is a linear function of the angle $\delta_A$ between the current hand pointing direction and the direction to point toward $A$, and similarly for $\delta_B$ and $T_B$.

$$w_i = \begin{cases} max(1/\theta, \varepsilon) & \text{where } \theta \text{ is the angle between } \overrightarrow{gazing} \text{ and} \overrightarrow{Object_i} \\ max(1/(\theta * \|d\|), \varepsilon) & \|d\| \text{ is the distance between the pointing hand and } Object_i \\ max(1/\|d\|, \varepsilon) & \text{where } \|d\| \text{ is the distance between } avatar \text{ and } Area_i \end{cases}$$

Unlike other retargeting techniques [8, 16, 17] that are limited to redirecting deictic gestures between a single matching pair of objects, the result of the above mapping generates a seamless transition from pointing at object $A$ to $B$, and can extend to multiple matching pairs of objects, which is essential for enabling room-scale remote MR collaboration. Moreover, in addition to retargeting pointing gestures, our compound mapping techniques can retarget the user's gaze and redirect locomotion by designing different weight functions.

After getting each compounding transformation $T$ for gazing, pointing, and locomotion, we can compute IK solvers for the avatar body positions at every frame.

### 2.1 Inverse Kinematics Implementation

Once we have identified the targets of the user actions in each site, we need to create the avatar's body, head, and hand movements (Figure 3). To generate a smooth and natural motion of the avatar that can easily move between different motions, such as pointing at remote objects or walking, We used multiple variations of Inverse Kinematics (IK) algorithms, such as Final IK's 'AimIK' for generating gazing motions or Final IK's 'LookAt IK' for pointing. Each algorithm uses a different context to interpolate the full avatar pose from the tracking of the user's head and hands. Our method smoothly blends different IK results, considering the distance from objects of interest, and generates a continuous transformation for an avatar.

#### 2.1.1 Gaze and Pointing

The AimIK and LookATIK IK algorithms require a definition of a target location for the gazing or the pointing. We define the location of those targets using:

$$ik\_target_{site2} = \begin{cases} M_{hitO^*} * M_{hitO}^{-1} * ik\_target_{site1} & \text{if raycasting hit} \\ T * ik\_target_{site1} & \text{otherwise} \end{cases}$$

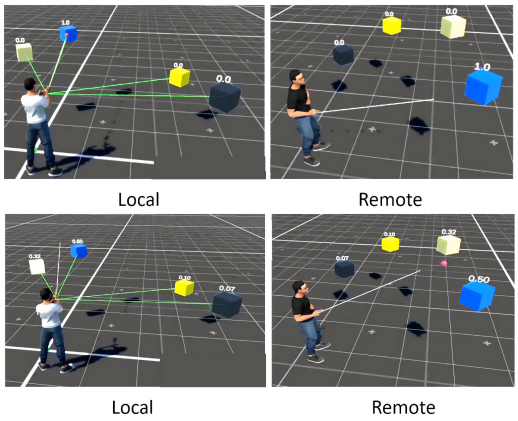


Figure 3: Left: As the user points near an object, its weight becomes dominant, and the avatar will point at the corresponding object in the remote site. Right: the mapping is defined over all the space, where each object contributes based on the distance of the pointing ray from it.

Suppose $hitO$ is the ray-cast hit-object for pointing or gazing in site I (local), and $hit^*$ is the corresponding object in site II (remote). $M^{-1}_{hitO}$ is the transformation matrix that transforms a point from the site I's world coordinate system to hit the object's coordinate system. $M_{hitO^*}$ is a transformation matrix that transforms a point from $hit^*$ object's coordinate system to the second site's coordinate system.

Since the target gaze/pointing is affected by multiple objects of interest, this generates a dynamic mapping where a gaze/pointing at the object in the local site is not only mapped directly to the corresponding object in the remote site but also shows the interpolations during transitions; the final transformation is a weighted average of the maps defined by all interest objects in the environment that is experienced very smoothly during motion transitions.

### 2.1.2 Locomotion and Avatar Position

Similarly, a user's position or a target of her walking may be mapped to location semantics that trigger walking in a remote site if necessary.

$$P\_avatar_{site2} = T * P\_avatar_{site1} \quad (2)$$

We implemented an animated locomotion module that uses a simple 8-directional strafing animation blend tree to generate locomotion poses. This module makes the character follow the horizontal direction towards the head IK target by root motion and scripted transformation.

Figure 4 illustrates the original user locomotion in the local environment and the transformed avatar locomotion in the remote environments. Despite the mismatch in the layouts of the collaboration areas, the transformed avatar locomotion maintains the spatial relationships. For example, when the user moves from the red to the blue area and then from the blue to the green area, their remote avatar also moves from the red to the blue area and then from the blue to the green area.

Locomotion isn't just enabled when a user walks in the local site. But also in cases where a user is reaching and grabbing an object of interest in her local site, her avatar has to move towards the corresponding object in the remote site. That object may be located in a different part of the site, and the avatar may need to walk toward it, avoiding obstacles in its environment. The speed at which the avatar walks is derived from the space mapping and is set to guarantee that the user and the avatar will reach the corresponding objects accurately and at the right time.

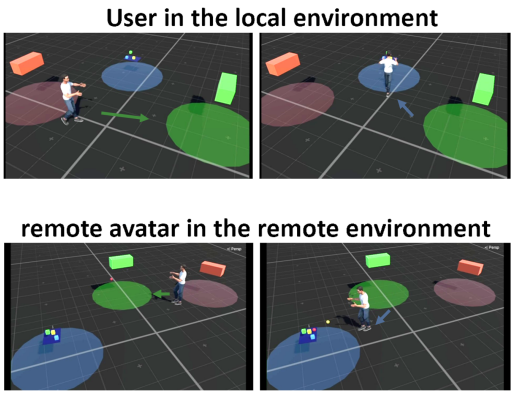


Figure 4: AvatarPilot preserves spatial relationships in dissimilar spaces by transforming the user's locomotion in the local environment to maintain the spatial relationships of the avatar's movements in the remote environments.

### 2.1.3 Grasping and Direct Hand Interaction

It is important that if an avatar is grabbing an object in a local space, the same object be grabbed in the remote space, even if the avatar or object is in a different configuration (Figure 5 Middle). For that, it is important to provide a transition from pointing to grabbing, so the hands get as close as possible to objects in the remote environment as they are in the local environment. In some cases, there will be a need for locomotion (as stated in the prior section).

Figure 2 (and supplementary video) shows how the above compounding transformations can smoothly retarget the avatar's pointing and gazing while users switch the pointing or gazing target from multiple virtual objects, resulting in a smooth and natural collaborative experience in dissimilar spaces. Having obtained the avatar's position, as well as the gaze and hand pointing IK targets, we can compute the target poses for the head and two hands, and then feed them into VRIK to generate full-body avatar movements.

## 3 Use Cases

To explore the feasibility of AvatarPilot , we developed three realistic scenarios: whiteboarding, bimanual object manipulation, and social interactions. In the whiteboarding scenario, AvatarPilot ensured that avatars could semantically point to shared applications, maintaining correct gestures despite different screen positions and angles. For bimanual object manipulation, AvatarPilot dynamically adapted to different user and object positions, allowing smooth avatar interactions across varied spatial arrangements. Finally, in social interactions, AvatarPilot demonstrated its ability to dynamically relocate avatars, maintaining consistent actions and social presence, as exemplified by a participant standing in one site while appearing seated in another.

## 4 Discussion

A common reaction of users to remapping their motions in other environments is a preference for one-to-one mapping, believing it better preserves their intention. Users often worry that a system lacking high fidelity cannot be trusted. However, recent studies suggest fidelity is more nuanced [3], with users accepting cartoon-ish avatars or filters over raw cameras for communication despite lower fidelity [10]. One-to-one mapping is traditionally seen as the best way to preserve user motion and intentions, particularly in shared or context-free environments like 2D video conferences. However, in 3D environments with different geometries, object placements,

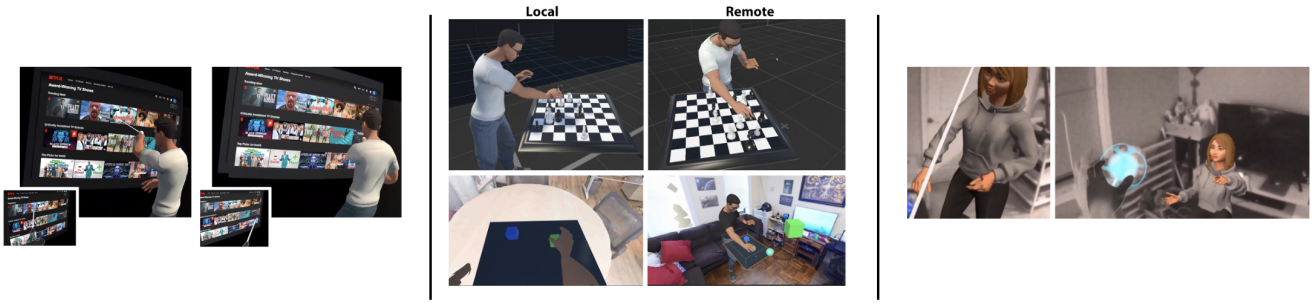


Figure 5: Use cases demonstrating AvatarPilot : (Left) Whiteboarding with correct semantic pointing, (Middle) Bimanual object manipulation with smooth avatar grasping, and (Right) Social interactions with dynamic avatar relocation.

and semantics, one-to-one mapping can lead to incorrect, misunderstood, or even offensive actions.

AvatarPilot offers an adaptive and comprehensive transformation between multiple spaces and objects, supporting interactions such as pointing, gazing, walking, and grabbing. Despite its robustness, limitations in IK and input tracking can affect the system's ability to achieve complete fidelity in body motions [3]. Combining IK with data-driven animation could further improve implementation [1]. The multi-solver approach of AvatarPilot can encounter corner cases, such as limitations in bimanual interactions or unnatural body positions when objects are too scrambled. Solutions might include moving the avatar to optimal positions or rearranging objects around the user. Prioritizing actions helps avoid conflicts; for instance, prioritizing grabbing can trigger locomotion solvers to ensure consistency across remote and local sites. Future work could explore new solver solutions.

## 5 Conclusions

Humans and their brains have evolved to handle one reality around them [4], and it is hard for a user to monitor and control their actions in multiple remote sites. Avatar Pilot is designed to dynamically control remote avatars with high semantic fidelity by creating new motions that are different from the user's original motions and maintaining the context of the original actions. It does so by detecting targets and using weighted interpolations. This way a user can engage in multiple remote locations for collaboration tasks without worrying about space dissimilarities or objects dispositions in the remote environment.

## References

[1] K. Ahuja, E. Ofek, M. Gonzalez-Franco, C. Holz, and A. D. Wilson. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021. 4

[2] M. Azmandian, M. Hancock, H. Benko, E. Ofek, and A. D. Wilson. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pp. 1968–1979, 2016. 2

[3] M. Bonfert, T. Muender, R. P. McMahan, F. Steinicke, D. Bowman, R. Malaka, and T. Döring. The interaction fidelity model: A taxonomy to distinguish the aspects of fidelity in virtual reality. *arXiv preprint arXiv:2402.16665*, 2024. 2, 3, 4

[4] M. Gonzalez-Franco and J. Lanier. Model of illusions and virtual reality. *Frontiers in psychology*, 8:273943, 2017. 4

[5] J. E. Grønbæk, J. S. Esquivel, G. Leiva, E. Velloso, H. Gellersen, and K. Pfeuffer. Blended whiteboard: Physicality and reconfigurability in remote mixed reality collaboration. In *CHI'24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024. 1, 2

[6] J. Hartmann, C. Holz, E. Ofek, and A. D. Wilson. Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019. 2

[7] M. Keshavarzi, M. Zollhoefer, A. Y. Yang, P. Peluse, and L. Caldas. Mutual scene synthesis for mixed reality telepresence. *arXiv preprint arXiv:2204.00161*, 2022. 1, 2

[8] M. Kim and S.-H. Lee. Deictic gesture retargeting for telepresence avatars in dissimilar object and user arrangements. In *The 25th International Conference on 3D Web Technology*, Web3D '20, 2020. doi: 10.1145/3424616.3424693 2

[9] A. Kitson, S. Ahn, E. Gonzalez, P. Panda, K. Isbister, and M. Gonzalez-Franco. Virtual games, real interactions: A look at cross-reality asymmetrical co-located social games. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024. 1

[10] P. Panda, M. J. Nicholas, M. Gonzalez-Franco, K. Inkpen, E. Ofek, R. Cutler, K. Hinckley, and J. Lanier. Alltogether: Effect of avatars in mixed-modality conferencing environments. In *Proceedings of the 1st Annual Meeting of the Symposium on Human-Computer Interaction for Work*, pp. 1–10, 2022. 3

[11] T. C. Peck, H. Fuchs, and M. C. Whitton. Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE transactions on visualization and computer graphics*, 15(3):383–394, 2009. 2

[12] T. Pejsa, J. Kantor, H. Benko, E. Ofek, and A. Wilson. 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*, pp. 1716–1725, 2016. 2

[13] M. Slater and A. Steed. Meeting people virtually: Experiments in shared virtual environments. In *The social life of avatars: Presence and interaction in shared virtual environments*, pp. 146–171. Springer, 2002. 1

[14] M. Sra, A. Mottelson, and P. Maes. Your place and mine: Designing a shared vr experience for remotely located users. In *Proceedings of the 2018 designing interactive systems conference*, pp. 85–97, 2018. 1

[15] J. Wentzel, D. Kim, and J. Hartmann. Same place, different space: Designing for differing physical spaces in social virtual reality. 2021. 2

[16] J. W. Woodworth, D. Broussard, and C. W. Borst. Redirecting desktop interface input to animate cross-reality avatars. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 843–851, 2022. doi: 10.1109/VR51125.2022.00106 2

[17] L. Yoon, D. Yang, C. Chung, and S.-H. Lee. A full body avatar-based telepresence system for dissimilar spaces. *arXiv preprint arXiv:2103.04380*, 2021. 2

[18] L. Yoon, D. Yang, J. Kim, C. Chung, and S.-H. Lee. Placement retargeting of virtual avatars to dissimilar indoor environments. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1619–1633, 2020. 1