The Visual Computer
https://doi.org/10.1007/s00371-023-03220-2

ORIGINAL ARTICLE

SMigraPH: a perceptually retained method for passive haptics-based migration of MR indoor scenes

Qixiang Ma[1] · Lili Wang[1] · Wei Ke[2] · Sio-Kei Im[2]

Accepted: 28 November 2023

Abstract

To enhance users' immersion in the mixed reality (MR) cross-scene environment, it is imperative to make geometric modifications to arbitrary multi-scale virtual scenes, including adjustments to layout and size, based on the appearance of diverse real-world spaces. Numerous studies have been conducted on the layout arrangement of pure virtual scenes; however, they often neglect the issue of incongruity between virtual and real environments. Our objective is to mitigate the incongruity between virtual and real scenes in MR, establish a rational layout and size for any virtual scene within an enclosed indoor environment, and leverage tangible real objects to achieve multi-class passive haptic feedback. To achieve these goals, we propose **SMigraPH**, a perceptually retained indoor scene migration method with passive haptics in MR. Firstly, we propose a scene abstraction technique for constructing mathematical representations of both virtual and real scenes, capturing geometric information and topological relationships, while providing a mapping strategy from the virtual to the real domain. Subsequently, we develop an optimization framework called *v2rSA* that integrates rationality, relationship preservation, haptic reuse, and scale fitting constraints in order to iteratively generate final layouts for virtual scenes. Finally, we render scenarios on optical see-through MR head-mounted displays (HMDs) to enable users to engage in realistic scene exploration and interaction with haptic feedback. We have conducted experiments and a user study on our proposed method, which demonstrates significant improvements in surface registration accuracy, haptic interaction efficiency, and fidelity compared to the state-of-the-art indoor scene layout arrangement method **MakeItHome** as well as the random placement approach **RandomIn**. The results of our approach closely resemble those achieved through manual placement using the **Human** method.

Keywords Mixed reality · Scene migration · Scene generation · Passive haptics · Optimization

1 Introduction

As a general use case for MR, envision an arbitrary real and virtual indoor scene. The user is situated in a typical enclosed real indoor environment while exploring a non-pre-designed virtual indoor scene. There are bound to be disparities between virtual and real scenes in terms of size, layout, and even style, if we directly render the unmodified virtual scene. To overcome this intricate challenge, it is necessary to manipulate the virtual indoor scene based on the real environment before implementing MR, reaching a seamless fusion display effect after scene migration. The primary objective of addressing the MR scene migration problem is to enable users in the physical world to remotely and interactively browse another virtual space with immersive experiences while ensuring optimal realism in the migrated virtual scene.

Previous scene layout arrangement methods have primarily focused on placing virtual furniture in empty spaces, with notable examples being [24, 44]. The approach proposed by [24] represents the scene as a triplet with multiple sub-elements, establishing various cost functions for each

✉ Lili Wang
wanglily@buaa.edu.cn

Qixiang Ma
sycamore_ma@buaa.edu.cn

Wei Ke
wke@mpu.edu.mo

Sio-Kei Im
marcusim@mpu.edu.mo

[1] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100083, China

[2] Faculty of Applied Sciences, Macao Polytechnic University, Macau SAR 999078, China

Published online: 30 December 2023

Fig. 1 As depicted in the first column, the real scene is set in a conference room, while the virtual living room scene is displayed as a single channel following optimization using **SMigraPH**. In the second column, users can seamlessly roam within the dual-rendering environment of both virtual and real scenes, with precise geometric alignment between the virtual table and its real counterpart. The third column showcases the user's perspective, providing an enhanced passive haptic feedback experience during interactions with shelves

criterion and conducting mathematical optimization. This method enables the generation of practical furniture layouts in empty scenes that address both functional and visual criteria. However, these results still fail to meet the requirements of MR scene migration. Traditional scene layout arrangement methods present challenges such as a mismatch or an incongruity between virtual and real scenes after migration and an inability to provide passive haptics for virtual objects. These approaches lack a mapping strategy from virtual objects to the real scene during scene abstraction and do not consider passive haptic feedback constraints provided by physical entities. To overcome these limitations, we draw inspiration from their ideas while incorporating additional rules and constraints to ensure proper alignment between virtual and real scenes in terms of layout. Consequently, our approach facilitates more realistic haptic feedback during interaction.

In this paper, we propose **SMigraPH** to effectively relocate all objects from the virtual scene into the real scene, enabling users in the real world to remotely explore alternative virtual spaces. Additionally, to enhance the realism of virtual objects, we introduce a haptic reuse strategy that improves object interactivity. Firstly, we extend the scene abstraction method based on [44] for extracting geometric approximations and relationships of objects in both virtual and real scenes. Herein, objects are defined as a sequence of oriented bounding boxes (OBBs) with multiple categories, and a mapping strategy from virtual to real domain are created. Secondly, we propose an optimization framework called *v2rSA* to iteratively generate final layouts for virtual scenes and seamlessly migrate them into the real environment. *v2rSA* incorporates rationality, relationship, haptic reuse, and scale fitting constraints. This approach improves geometric surface registration between virtual objects and the real environment while ensuring a coherent spatial arrangement. Finally, through an iterative process, we obtain the final layout of the virtual scene and employ an optical see-through MR HMD to achieve disocclusion rendering of dual-channel scenes.

We compared our method with **RandomIn**, **MakeItHome**, and **Human**. The results demonstrated that: 1) our method exhibited superior layout quality compared to **RandomIn** and **MakeItHome**, approaching the level of **Human**; 2) both our method and **Human** significantly reduced virtual to real cross-scenes chamfer distances in comparison with **RandomIn** and **MakeItHome**. Furthermore, we conducted a user study to evaluate the performance of virtual-real scene coupling. Compared to the state-of-the-art approach **MakeItHome**, our method showed an increase in correct haptic interactions while reducing task completion time and perceptive loss rate. Figure 1 illustrates an interaction task scenario showcasing our method's capability for achieving realistic roaming and haptic interaction within a dual-pass rendering environment.

In summary, we present our main contributions as follows:

- We introduce the novel **SMigraPH** pipeline in MR, which is the first to specifically address the perceptually retained migration of indoor scene layouts from multi-scale virtual scenes to flexible real scenes.
- We propose a mapping strategy with categorized passive haptics from virtual objects to the real scene that extends the scene abstraction method by incorporating geometric approximations and object relationships from both virtual and real scenes.
- We establish an optimization framework called *v2rSA* with a cost function that incorporates rationality, relationship, haptic reuse, and scale fitting constraints for effectively manipulating virtual objects into the real scene.

2 Related work

Our approach is heavily inspired by the prior research on indoor scene synthesis or layout arrangement, as extensively discussed in a comprehensive survey [46] that categorizes these approaches based on input, internal representation, prior knowledge, and optimization techniques. Interested readers can explore diverse strategies for virtual scene generation from this source.

2.1 Automatic indoor scene layout arrangement

It mainly focuses on placing some virtual furniture and arranging them reasonably in empty virtual space to generate or synthesize a virtual indoor scene for users to browse. This problem begins with [42], which uses physical systems as restrictions to realize the rapid scene synthesis for a large number of virtual props, where ensures objects non-intersect and reasonable free scene space. As different methods appear, the prior knowledge acquisition of the problem is derived as hard-coding, activity-driven, and example-based. [24, 44] use the prior to mathematically optimize the designed cost function of the layout. Later, data-driven methods appeared. [10] used Bayesian network and Gaussian mixtures to make a new scene from a few input examples, and [16] used undirected factor graphs learned from RGB-D images to insert objects progressively into an empty scene. Recently, some deep learning oriented methods [20, 39, 40] have been proposed to train the scene layout from the data set and find the optimal solution for the indoor scene layout after iteration. [28] take the human activity region as human-centric into account. One of the above [24] has added user's interaction during the iterative process or explicit iteration leading to multiple suggestions of output scenes, which makes synthetic indoor scene personalized. Indoor scene arrangement requires a stage to transform the initial input into an internal representation. With the help of deep learning, many articles have cleverly proposed internal representation based on graphs [40] by GCN [17] and activities [28] by human activity semantics. Some articles continue to use classic methods like projection-based representation [39, 44], which treats the 3D scene data as a 2D top view to facilitate neural networks to predict the distribution or facilitate mathematical methods to optimize the arrangement of layout. In this article, considering the mixed representation of the two scenes, the subsequent optimization processes might become complicated and time-consuming. We use the projection-based representation to perform mathematical optimization instead of deep learning method to open the scene migration problem in MR, making it easier to be explained and solved.

2.2 Passive haptic and haptic retargeting

Passive haptics was first proposed in [13], using the combination of visual virtual objects and physically real objects to enhance the haptic perception or spatial awareness in virtual environment. Some works have designed real objects for virtual props in VR to realize force feedback systems, such as [45] providing a weight-shifting physical bar to achieve passive haptics for VR objects with different weights or sizes. Some articles use low-fidelity native real objects to provide haptic feedback. [2] used a single removable physical prop to provide passive haptics. [23] extended the original physical interaction interface like HMD handle controller with additional virtual buttons. [41] used real indoor objects to map with geometric marks in a virtual environment, providing passive haptic to enhance the user's awareness in VR redirection problem. [2, 23, 41] can also be called haptic retargeting. In AR or MR systems, the supplementary of real assets provides users with visual guidance. If haptic information cannot be accurately compensated, the user's experience is still unnatural and weird, where users might freely pass through the rendered virtual geometry, or hit the real wall. In recent years, many articles have applied passive haptics systems in augmented reality, such as [14] using pin array, and mid-air devices like [34] using ultrasounds, which can make users feel pressure or thermal stimulation. Some passive haptics methods require additional wearable devices, such as [31]. Some other passive haptics methods can use the alignment of daily physical objects and virtual objects to achieve haptic and visual correspondence, such as [19]. For more information, please read survey [3]. In our method, the designed cost function in the optimization stage is based on the idea of passive haptics and haptics retargeting in MR, regarding haptic reuse and scaling fitting as high-level constraints, which is inspired by [19, 41].

2.3 Related technique in our method

The scene migration in MR differs from the traditional furniture layout arrangement, where objects in the real scene need to be extracted as the input of the optimization framework. In the field of computer vision and automation, there are deep learning-based object detection methods, such as 3D object detection [1, 27] in automatic driving. [27] implements PointNets that detect 3D objects from RGB-D images, and [1] implements end-to-end YOLO network to perform real-time oriented bounding box detection from 3D point clouds. Some methods focus on indoor detection like [12, 22] proposed network models to realize the 3D furniture detection under the single-shot points cloud. Their benchmarks are the popular ScanNet V2 [6] and SUN RGB-D [47]. Some articles dealing with indoor point cloud segmentation can not be ignored, such as [35, 37], which are oriented toward proper

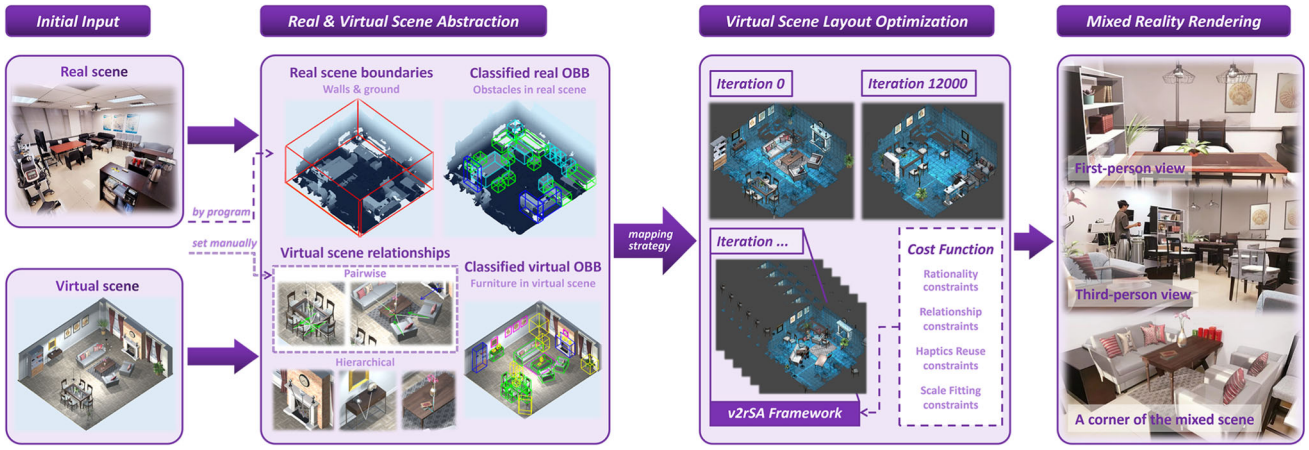


Fig. 2 The pipeline of our **SMigraPH** scene migration method

scale levels of segmentation, such as wall, floor or object levels. We also draw on works of point cloud registration for indoor scenes or objects, such as [7, 33], to verify the rationality of our scene migration method. [21] even creatively sent the segmented semantic indoor scene into AR, and some virtual digital content is placed correspondingly. Our method uses Microsoft HoloLens 2 [38] to scan the geometric mesh of the real indoor scene, and chose a more straightforward way, using the PCL [30] point cloud processing library for the scanned geometric information to perform points cloud clustering as input bounding boxes (Sect. 3.1). After extracting the real scene data as a migration target, we need an optimization process to obtain the optimal virtual scene layout. The Metropolis–Hastings algorithm [5] uses Markov chain Monte Carlo (MCMC) sampling method [11] in statistics, which is the core of the simulated annealing [9] algorithm for accept-reject sampling. Similarly, some previous works on virtual scene synthesis have extended methods like using Reversible Jump MCMC (RJMCMC) [36] or Locally Annealed RJMCMC (LA-RJMCMC) [43] on classic MCMC. Their sampling and transfer strategies are friendly when the number of virtual objects is variable or when the scene is open.

3 Method

In this section, we present the pipeline of our scene migration approach. As illustrated in Fig. 2, it is divided into three stages:

- **Real and virtual scene abstraction.** We construct abstractions for both the real and virtual scenes by incorporating the input of the virtual scene and geometry meshes obtained from scanning the real scene using MR HMDs. These abstractions include information about the boundaries of the real scene, OBBs of objects in both scenes, as well as layout relationships among objects in the virtual scene. In this process, we propose a mapping strategy (detailed in Sect. 3.1.3) that links virtual objects to their corresponding positions in the real scenes. Further details regarding this process can be found in Sect. 3.1.
- **Virtual scene layout optimization.** In order to seamlessly migrate virtual scene layouts into reality, we introduce an optimization framework called v2rSA (detailed in Sect. 3.2). This framework defines a cost function comprising rationality, relationship, haptic reuse, and scale fitting constraints to ensure reasonable placement of virtual objects while achieving fine geometric registration between both scenes. For more information on this topic, please refer to Sect. 3.3.
- **Real-virtual mixed rendering.** Leveraging our optimized virtual scene layout, we render virtual objects within optical see-through HMDs alongside the real environment, offering users an immersive visual experience that combines elements from both worlds while preserving passive haptics from reality. Dual-pass disocclusion rendering is employed where visibility between virtual and real objects is determined by a z-buffer algorithm utilizing scanned geometry from the real environment and optimized representation of the virtual scene.

As depicted in Fig. 3, the user can perceive the virtual layout suspended on the physical geometric surface through HMDs and freely explore within the mixed reality environment, thereby gaining an immersive understanding of the real space. Additionally, interactive physical surfaces provide passive haptic feedback to virtual objects.

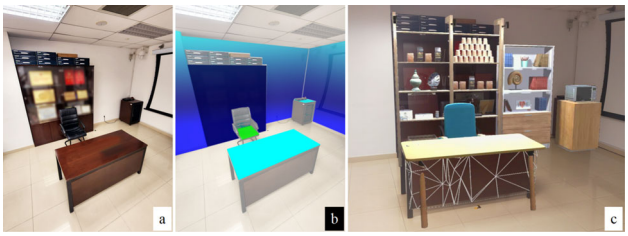


Fig. 3 **a** The original real scene. **b** The interactive surfaces where vertical haptic planes are highlighted in blue and horizontal haptic planes are highlighted in cyan or green. **c** Dual-pass mixed scene rendering by HoloLens 2 with disocclusion

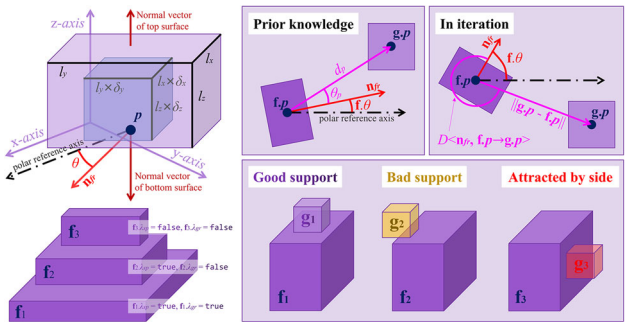


Fig. 4 The left column displays the parameters contained in an OBB, where the polar reference axis is aligned parallel to the x-axis for measuring θ. The upper right section shows factors of the pairwise relationship constraint in both prior and optimization iterations (from a top view). The lower right section shows cases of good support (violet), bad support (yellow), and illegal attraction (red) achieved through hierarchical relationship constraints

3.1 Virtual and real scenes abstraction

Virtual and real scenes are abstracted in this section to facilitate the process of scene migration. We denote virtual scenes as $\mathbb{S}_{\mathbb{V}}$ and real scenes as $\mathbb{S}_{\mathbb{R}}$. A virtual scene is represented by a tuple $\mathbb{S}_{\mathbb{V}} = (\mathbb{F}, \mathbb{R})$, where $\mathbb{F}$ denotes the set of virtual objects, such as indoor furniture, and $\mathbb{R} \subseteq \mathbb{F}^2 - \mathbb{F} \circ \mathbb{F}$ represents the relationships among these objects, such as the proximity between a chair and a table. Similarly, a real scene is abstracted as tuple $\mathbb{S}_{\mathbb{R}} = (\mathbb{O}, \mathbb{B})$, where $\mathbb{O}$ refers to the set of real objects present in the scene, and $\mathbb{B}$ represents its boundaries including walls and ground. An illustration depicting this abstracted information can be seen in Fig. 7. An illustration depicting this abstracted information can be seen in Sect. 4.

3.1.1 Boundaries extraction for real scene

To accurately migrate the virtual scene to reality, it is crucial to determine the boundary of the real scene confining virtual objects within appropriate indoor space. The boundary can be classified into two categories: $\mathbb{B} = \mathbb{B}_{wall} \cup \mathbb{B}_{ground}$, where $\mathbb{B}_{wall}$ and $\mathbb{B}_{ground}$ denote walls and ground surfaces, respectively. To obtain this boundary information, we utilize MR HMDs to scan the real scene mesh and sample it into a 3D point cloud representation. Both walls and ground are then segmented as planar structures using the Random Sample Consensus algorithm (RANSAC) [32] applied on the acquired 3D point cloud data.

3.1.2 Bounding boxes extraction

To represent sets of virtual objects $\mathbb{F}$ and real objects $\mathbb{O}$, we generate an OBB for each object. Each OBB comprises multiple parameters (illustrated in the first column of Fig. 4) and can be expressed as follows:

$$\text{OBBs} := \begin{cases} \mathbf{f} = (\boldsymbol{p}, \theta, \boldsymbol{l}, \boldsymbol{\delta}, \mathbf{n}_{fr}, \lambda_{sp}, \lambda_{gr}) \,, & \mathbf{f} \in \mathbb{F} \\ \mathbf{o} = (\boldsymbol{p}, \boldsymbol{l}, \boldsymbol{\delta} \equiv (1,1,1)) \,, & \mathbf{o} \in \mathbb{O} \end{cases} \tag{1}$$

where $\boldsymbol{p}(x, y, x)$ represents the centers of the bottom surface of OBBs, $\theta \in [0, 2\pi)$ denotes the orientation indicating the Euler angle of the object rotated relative to its initial position along the z-axis, $\boldsymbol{l}(l_x, l_y, l_z)$ refers to the original size, $\boldsymbol{\delta}(\delta_x, \delta_y, \delta_z)$ represents the scale factor, and $\mathbf{n}_{fr}$ signifies the normal vector of the front surface. The Boolean parameter λ_{sp} indicates whether this object can support other objects on top of it, while λ_{gr} indicates whether this object can lay on the ground. For example, a desk has both λ_{sp} and λ_{gr} set to true, whereas a plate has only λ_{sp} set to true but not λ_{gr}.

Pre-modeling OBBs in $\mathbb{F}$ can be directly obtained from the geometry mesh of $\mathbb{S}_{\mathbb{V}}$. The front vector $\mathbf{n}_{fr}$ and Booleans $\lambda_{i,sp}$, $\lambda_{i,gr}$ are also assigned initial values during the pre-modeling phase. By clustering the 3D point cloud of $\mathbb{S}_{\mathbb{R}}$, we can acquire the OBB in $\mathbb{O}$. The KD-tree [18] is utilized to segment the point cloud into clusters based on specific thresholds for Euclidean distance and minimum number of closure points. Subsequently, an OBB is constructed using PCA [30] for each cluster. Prior to implementing optimization, users have the flexibility to adjust both the number of OBBs in $\mathbb{O}$ through our designed MR build-in user interface as well as fine-tune their pose and size.

3.1.3 OBBs classification and mapping strategy

As daily objects are orthogonal to the ground, users typically interact with them in two scenarios: either by engaging with their vertical or horizontal surfaces. In order to provide virtual objects with a diverse haptic experience, we classify them into three haptic types denoted as $\mathbb{F} = \mathbb{F}_{hor} \cup \mathbb{F}_{ver} \cup \mathbb{F}_{no}$, where these categories represent distinct classifications of objects based on their haptic properties: **horizontal haptic objects** (e.g., desks, sofas), **vertical haptic objects** (e.g., bookshelves, paintings), and **non-haptic objects** (e.g., potted plants, lamps). We manually assign the appropriate haptic

type to each virtual object by its geometry attributes during the scene pre-modeling process and inherit haptic type to its OBB. In general, objects with planar interaction surfaces will be classified as either horizontal or vertical haptic objects, while ornamental objects or significantly inclined objects are considered as non-haptic objects.

To approximate the semantics of real scene 3D point clouds, we employ a straightforward yet practical approach to classify $\mathbb{O}$ for facilitating subsequent mapping from $\mathbb{F}$ to $\mathbb{O}$. Real objects are categorized into three groups based on the height of their OBBs' top surface: $\mathbb{O} = \mathbb{O}_{low} \cup \mathbb{O}_{mid} \cup \mathbb{O}_{high}$, where these categories represent **low real objects** ($< 0.6m$), **medium real objects** ($< 1.25m$), and **high real objects** ($\geq 1.25m$), respectively.

To align $\mathbb{S}_{\mathbb{V}}$ with $\mathbb{S}_{\mathbb{R}}$ during the optimization step, we establish a mapping strategy from the virtual domain (classified $\mathbb{F}$, to be optimized) to the real domain (classified $\mathbb{O}$ and $\mathbb{B}$, as attractors).

- We map $\mathbb{F}_{hor}$ to either $\mathbb{O}_{low}$ or $\mathbb{O}_{mid}$. For example, if an object in $\mathbb{F}_{hor}$ represents a virtual chair or desk that is interacted with horizontally, then corresponding real objects could be a sofa for $\mathbb{O}_{low}$ or a table for $\mathbb{O}_{mid}$.
- We map $\mathbb{F}_{ver}$ to $\mathbb{O}_{high}$. For example, if an object in $\mathbb{F}_{ver}$ represents a virtual closet that is interacted with vertically, then the corresponding real object could be a bookshelf for $\mathbb{O}_{high}$.
- Additionally, objects in $\mathbb{F}_{ver}$ can be mapped to $\mathbb{B}_{walls}$. For instance, when the number of objects in $\mathbb{O}_{high}$ is insufficient for mapping, the front surface of the virtual cabinet in $\mathbb{F}_{ver}$ can be precisely positioned on the wall (as depicted in Fig. 5), as users typically engage in haptic interaction with its frontal area.
- The objects in $\mathbb{F}_{no}$ can be mapped to, based on their respective λ_{gr}, vacant areas within $\mathbb{B}_{ground}$ or positioned on the upper surface of other objects in $\mathbb{F}_{\lambda_{sp}=true}$. For instance, a small virtual potted plant (with true λ_{gr}) lacking haptic feedback may be situated in an unoccupied space on the ground, while a microwave oven (with false λ_{gr}) could be placed atop the counter.

Regarding the mapping, we do not view it as an exclusive quantitative peer mapping. We do not assume that the number of virtual haptic objects and the number of corresponding real objects are roughly close. For example, even if the physical scene contains no or very few high real objects that provide haptic feedback, we can use the constraints of the wall to map vertical tactile objects to reasonable locations. Similarly, when the physical scene does not contain low or medium real objects, the mapping strategy of horizontal haptic objects will not be interrupted abnormally, and the secondary constraints in Sect. 3.3 will ensure that they eventually appear in a relatively reasonable position globally.

3.1.4 Pairwise and hierarchical relationships

In this section, we extract the topological relationship among virtual objects in $\mathbb{F}$. The relationships comprise two subrelations: $\mathbb{R} = \mathbb{R}_p \cup \mathbb{R}_h$, where $\mathbb{R}_p$ denotes pairwise relationships and $\mathbb{R}_h$ denotes hierarchical relationships. Specifically, $\mathbb{R}_p$ refers to the topological structure of relative distance and orientation between two virtual objects in $\mathbb{F}$; for instance, a chair must face a desk while maintaining proximity. On the other hand, $\mathbb{R}_h$ defines scenarios where one virtual object supports another; for instance, a laptop should be positioned directly on the upper surface of a desk.

We manually set pairwise relations between virtual objects and automatically generate hierarchical relations among them. Yu et al. [44] provide a user interface for users to click on corresponding objects in the virtual scene to establish pairwise relations. Similarly, we write a script to store pairs of virtual objects and record their initial relative distance and orientation. Currently, we have obtained the set $\mathbb{R}_p = \{(< \mathbf{f_i}, \mathbf{f_j} >, d_p, \theta_p) \mid \cdot\}$ where $\mathbf{f_i}, \mathbf{f_j} \in \mathbb{F}$, d_p represents initial distance and θ_p denotes initial relative orientation between $\mathbf{f_i}$ and $\mathbf{f_j}$. The enumerations of $< \mathbf{f_i}, \mathbf{f_j} >$ are not necessarily full of the complete set of $\mathbb{F}^2 - \mathbb{F} \circ \mathbb{F}$. If we don not provide the pair of the virtual object in the input script (e.g., laptop is irrelevant to bookshelf), then there will be no relation about them recorded in $\mathbb{R}_p$. We can obtain the set $\mathbb{R}_h$ automatically by principle $\mathbb{R}_h = \{< \mathbf{f}, \mathbf{g} > \mid \text{proj}(\mathbf{f} \in \mathbb{F}_{hor}) \cap \text{proj}(\mathbf{g} \in \mathbb{F}) \neq \varnothing, \mathbf{f}.\lambda_{sp} \wedge \neg \mathbf{g}.\lambda_{gr}\}$, where operator $\text{proj}(\cdot) = \text{proj}_{z=0}(\cdot)$ denotes projecting an object onto plane with equation $z = 0$ from top view; also note that $\text{proj}(\mathbf{f}) \cap \text{proj}(\mathbf{g})$ means intersection between projections of two different virtual objects $\mathbf{f}$ and $\mathbf{g}$.

3.2 Virtual scene layout optimization framework

Inspired by previous works [24, 44], we employ the abstracted data and mapping strategy discussed in Sect. 3.1 to achieve an optimized virtual scene layout through mathematical optimization of a high-dimensional cost function. Our migration approach enables manipulation such as relocation, rotating, or scaling of virtual objects within the real scene, aiming to minimize this cost through an iterative process of mapping them to different locations. To facilitate this mapping from the virtual domain to the real domain, we propose an optimization framework called *v2rSA* which utilizes simulated annealing (SA) [9] for approximating the global optimal solution iteratively. The cost function $C(\phi)$ is defined in detail in Sect. 3.3. For brevity, we represent the independent variable input of the cost function as $\phi = \mathbb{S}_{\mathbb{V}} \cup \mathbb{S}_{\mathbb{R}} = \{(\mathbb{F}, \mathbb{R}), (\mathbb{O}, \mathbb{B})\}$.

The proposal optimization for a current virtual scene layout ϕ, represented as a step $\phi \rightarrow \phi'$, is considered in each iteration of *v2rSA*. The acceptance probability for this proposal step, denoted as $\alpha(\phi' \mid \phi)$ [44], is determined based on

Eq. (2):

$$\alpha(\phi' \mid \phi) = \min\left(\exp(\frac{1}{t}(C(\phi) - C(\phi'))), 1\right) \tag{2}$$

where t is the temperature decreasing over iterations.

To displace, rotate, scale virtual objects, or swap their layout for each proposal step $\phi \to \phi'$, we employ the following strategies:

$$\underset{\mathbf{f},\mathbf{g}\in\mathbb{F}}{\phi \to \phi'} = \begin{cases} \mathbf{f}.\boldsymbol{p} \to \mathbf{f}.\boldsymbol{p} + \Delta p, & \Delta p \sim [\mathcal{N}(0,\sigma_p^2), \mathcal{N}(0,\sigma_p^2), 0]^\top, \\ & 0 \le \beta < 4 \\ \mathbf{f}.\boldsymbol{p} \to \mathbf{f}.\boldsymbol{p} + \Delta p, & \Delta p \sim [0, 0, \mathcal{N}(0,\sigma_p^2)]^\top, \\ & \neg\mathbf{f}.\lambda_{gr} \wedge \beta = 4 \\ \mathbf{f}.\theta \to \mathbf{f}.\theta + \Delta\theta, & \Delta\theta \sim \mathcal{N}(0,\sigma_\theta^2), \\ & 5 \le \beta < 8 \\ \mathbf{f}.\boldsymbol{\delta} \to \mathbf{f}.\boldsymbol{\delta} + \Delta\boldsymbol{\delta}, & \Delta\boldsymbol{\delta} \sim \mathbf{T}^\beta[\mathcal{N}(0,\sigma_\delta^2), 0, 0]^\top, \\ & \beta = 8, 9, 10 \\ \mathbf{f} \leftrightarrow \mathbf{g}, & \beta = 11 \end{cases} \tag{3}$$

where $\Delta\boldsymbol{p}$, $\Delta\theta$ and $\Delta\boldsymbol{\delta}$ represent the movements in position, rotation, and scale of $\mathbf{f}$, respectively. These variables follow a Gaussian distribution denoted as $\mathcal{N}(\mu,\sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, whose mean is set to $\mu = 0$ and variances $\sigma_p^2, \sigma_\theta^2, \sigma_\delta^2$ are proportional to temperature t. In each proposal step, both $\mathbf{f}, \mathbf{g}$ are randomly selected from set $\mathbb{F}$. The type of movements is determined by operator β, which is obtained from a uniform distribution $\beta \sim \lfloor\mathcal{U}(0, 11.99)\rfloor$. Moreover, the circulant matrix $\mathbf{T} = (\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{x}})$ is used to select the components δ_x, δ_y, or δ_z for scaling movements. The *v2rSA* optimization framework adjusts only the abstract parameters in one or two dimensions (e.g., specific dimensions of position, scalar angle, and scale) during each proposal iteration. According to [26], the solution space is finite on probability, ensuring convergence of the results. Furthermore, the global optimization capabilities of SA assist in avoiding local optima.

3.3 Cost function

Yu and Merrell et al. [24, 44] optimized the layout of the virtual scene in an empty space, focusing primarily on two constraints in the cost function: rationality and relationship. However, to achieve passive haptic feedback during scene migration in MR, we have modified the traditional rationality and relationship constraints while introducing two additional constraints: haptic reuse and scale fitting. By weighting and integrating all constraint cost items, we can obtain the global cost function $C(\phi)$ with Eq. (4):

$$\begin{aligned} C(\phi) =& w_{no} w_{in} \mathbf{C}\mathbf{w}^\top \\ \mathbf{C} =& \left(C_{ac}(\phi), C_p(\phi), C_{hs}(\phi), C_h(\phi), C_{sf}(\phi)\right) \\ \mathbf{w} =& \left(w_{ac}, w_p, w_{hs}, w_h, w_{sf}\right) \end{aligned} \tag{4}$$

where the global weight w_{no} avoids overlapping between virtual objects or between non-corresponding virtual-real objects, and w_{in} ensures virtual objects remain within the boundaries of the real space. The cost item $C_{ac}(\phi)$ serves as a rationality constraint, with $C_p(\phi)$ and $C_{hs}(\phi)$ serving as relationship constraints, $C_h(\phi)$ serving as a haptic reuse constraint, and $C_{sf}(\phi)$ serving as a scale fitting constraint. In our implementation, we manually set the weight vector $\mathbf{w}$ for each item to (0.05, 0.15, 0.15, 0.35, 0.3).

3.3.1 Rationality constraint

The rationality constraint ensures that the mixed scene adheres to physical laws in order to appear plausible and rational. Within this constraint, we consider three sub-terms: **non-overlapping**, **indoor restriction**, and **accessibility**.

Non-overlapping constraint ensures that there is no intersection between virtual objects and that virtual objects do not overlap with irrelevant real objects. To represent this constraint, we introduce the global weight w_{no} in Eq. (5). Overlapping would result in a significantly higher cost function value during downward optimization iterations.

$$w_{no} = \begin{cases} 1.0, & \sum\limits_{\mathbf{f},\mathbf{g}\in\mathbb{F}} \text{body}(\mathbf{f}) \cap \text{body}(\mathbf{g}) = \varnothing, \\ & \sum\limits_{\mathbf{f}\in\mathbb{F}_{hor}} \sum\limits_{\mathbf{o}\in\mathbb{O}_{high}} \text{body}(\mathbf{f}) \cap \text{body}(\mathbf{o}) = \varnothing \\ & \sum\limits_{\mathbf{f}\in\mathbb{F}_{ver}} \sum\limits_{\mathbf{o}\in\mathbb{O}_{low}\cup\mathbb{O}_{mid}} \text{body}(\mathbf{f}) \cap \text{body}(\mathbf{o}) = \varnothing \\ \mathbf{INF}, & otherwise \end{cases} \tag{5}$$

where the operator body$(\cdot)$ represents the geometric body of the object. **INF** is a manually set constant with a large value, typically 1,000,000. When $w_{no} = 1.0$, there is no overlap between any pair of virtual objects $\mathbf{f}, \mathbf{g} \in \mathbb{F}$, and irrelevant virtual-real objects pairs also do not exhibit overlap, such as a virtual horizontal haptic object from set $\mathbb{F}_{hor}$ and a high real object from set $\mathbb{O}_{high}$, or a virtual vertical haptic object from set $\mathbb{F}_{ver}$ and a low or medium real object from set $\mathbb{O}_{low} \cup \mathbb{O}_{mid}$.

Indoor restriction implies that virtual objects must remain within the boundaries of the physical space. To represent this concept, we introduce a global weight denoted as w_{in} in Eq. (6):

$$w_{in} = \begin{cases} 1.0, & \sum\limits_{\mathbf{f}\in\mathbb{F}_{hor}\cup\mathbb{F}_{no}} \text{proj}(\mathbf{f}) \cap \overline{\text{proj}(\mathbb{B}_{ground})} = \varnothing, \\ & \sum\limits_{\mathbf{f}\in\mathbb{F}_{ver}} \text{proj}(\mathbf{f}) \cap \text{proj}(\mathbb{B}_{ground} \oplus \epsilon) \neq \varnothing \\ \mathbf{INF}, & otherwise \end{cases} \tag{6}$$

where in the condition $w_{in} = 1.0$, virtual objects with horizontal haptic feedback $\mathbb{F}_{hor}$ or without any haptic feedback

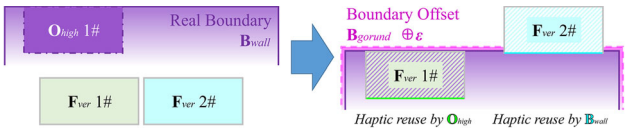


Fig. 5 The vertical haptic objects can receive passive haptic feedback by interacting with high real objects or walls, while the interactive front surface is visually emphasized in green and cyan. For instance, when the real high object $\mathbb{O}_{high}$1# is occupied by another virtual object $\mathbb{F}_{ver}$1#, a bookshelf $\mathbb{F}_{ver}$1# belonging to category $\mathbb{F}_{ver}$ can be securely attached to the surface of wall $\mathbb{B}_{wall}$ but remains outside of $\mathbb{B}_{ground}$. Nevertheless, this arrangement still enables seamless haptic interaction

$\mathbb{F}_{no}$ will be positioned within the physical space. As depicted in Fig. 5, it is worth noting that the virtual object with vertical haptic feedback $\mathbb{F}_{ver}$ can also adhere tightly to a wall surface, while still potentially extending beyond the ground plane $\mathbb{B}_{ground}$ due to accessibility constraints in Eq. (7). The dilation operator $\oplus$ is employed to expand the projection of the ground as proj($\mathbb{B}_{ground} \oplus \epsilon$) by incorporating a small neighborhood defined as ϵ.

Accessibility refers to the requirement of leaving a certain amount of space around an object to enable people to interact with it. For instance, users should be able to access a dining table with sufficient surrounding space in order to walk around or sit down by it. This principle holds true in the physical world and should also be upheld in the MR scene migration application that combines numerous virtual-real objects. In MR, as the virtual scene is integrated into the real environment, objects within the scene tend to become more densely packed, thereby necessitating higher standards for accessibility. To address this constraint, we introduce $C_{ac}(\phi)$ comprising five sub-terms in Eq. (7), aiming to optimize object layout by maximizing looseness.

$$C_{ac}(\phi) = -\boldsymbol{\alpha_{ac}} \begin{pmatrix} \sum\limits_{\mathbf{f},\mathbf{g}\in\mathbb{F}} (\|\mathbf{f}.\boldsymbol{p} - \mathbf{g}.\boldsymbol{p}\| - \bar{L}(\mathbf{f}) - \bar{L}(\mathbf{g}))^{\gamma_{ac1}} \\ \sum\limits_{\mathbf{f}\in\mathbb{F}_{hor}\cup\mathbb{F}_{no}} \sum\limits_{\mathbf{b}\in\mathbb{B}_{walls}} (\|\mathbf{f}.\boldsymbol{p} - \mathbf{b}.\boldsymbol{p}\| - \bar{L}(\mathbf{f}))^{\gamma_{ac2}} \\ \sum\limits_{\mathbf{f}\in\mathbb{F}_{no}} \sum\limits_{\mathbf{o}\in\mathbb{O}} (\|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\| - \bar{L}(\mathbf{f}) - \bar{L}(\mathbf{o}))^{\gamma_{ac3}} \\ \sum\limits_{\mathbf{f}\in\mathbb{F}_{hor}} \sum\limits_{\mathbf{o}\in\mathbb{O}_{high}} (\|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\| - \bar{L}(\mathbf{f}) - \bar{L}(\mathbf{o}))^{\gamma_{ac4}} \\ \sum\limits_{\mathbf{f}\in\mathbb{F}_{ver}} \sum\limits_{\substack{\mathbf{o}\in\mathbb{O}_{low} \\ \cup\mathbb{O}_{mid}}} (\|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\| - \bar{L}(\mathbf{f}) - \bar{L}(\mathbf{o}))^{\gamma_{ac5}} \end{pmatrix} \tag{7}$$

where $\bar{L}(\cdot)$ is an offset operator for determining objects' distances based on their average side length obtained from their bounding boxes ($\mathbf{f} \in \mathbb{F} \cup \mathbb{O}$), expressed as $\bar{L}(\mathbf{f}) = \mathbf{f}.\mathbf{l} \circ \bar{\mathbf{f}}.\boldsymbol{\delta}$. The weight vector $\boldsymbol{\alpha_{ac}}$ and index adjustment factors γ_{ac} are manually set for each sub-item. In our implementation, we ensure that $\Sigma_{i=1}^{5}\alpha_{aci} \in \boldsymbol{\alpha_{ac}} = 1$, and $\gamma_{aci} \in [1.0,\ 1.5]$. The first term measures the sum of the distances between any two virtual objects, while the second term quantifies the distance from virtual objects without vertical haptics to walls. The third term represents the distance from non-haptic objects to real objects, and the fourth term denotes the distance from horizontal haptic objects to high real objects. Lastly, the fifth term accounts for the distance between vertical haptic objects and low or medium real objects. However, maintaining centers' distances alone cannot guarantee accessibility; therefore, it is necessary to consider object sizes as well. For relatively large objects, the distance between the center points of them should be increased slightly by $\bar{L}(\cdot)$. Since $C_{ac}(\phi)$ represents a weaker constraint (explained in Sect. 3.3.5), $\bar{L}(\cdot)$ can provide a rough indication of an object's size.

3.3.2 Relationship constraint

The relationship constraint refers to the requirement that virtual objects exhibiting topological relations must maintain a specific distance or orientation between each other, or provide support to one another. This constraint encompasses two sub-constraints: **pairwise distance and orientation**, as well as **hierarchical support**.

Pairwise distance and orientation denoted as $C_p(\phi)$, serve as a stronger constraint to ensure the consistency of both distance and orientation between pairwise objects in $\mathbb{R}_p$ with those in the original virtual scene (as shown in Fig. 4). The constraint $C_p(\phi) = w_{pd}C_{pd}(\phi) + (1 - w_{pd})C_{po}(\phi)$ can be decomposed into two sub-terms: the distance constraint $C_{pd}(\phi)$ defined in Eq. (8) and the orientation constraint $C_{po}(\phi)$ defined in Eq. (9). In our implementation, we manually set the weight parameter to $w_{pd} = 0.3$.

$$C_{pd}(\phi) = \sum_{\mathbf{r}=(<\mathbf{f},\mathbf{g}>,d_p,\theta_p)\in\mathbb{R}_p} \max(|\ \|\mathbf{g}.\boldsymbol{p} - \mathbf{f}.\boldsymbol{p}\| - d_p\ | - d_\epsilon, 0)^{\gamma_{pd}} \tag{8}$$

$$C_{po}(\phi) = \sum_{\substack{\mathbf{r}=(<\mathbf{f},\mathbf{g}>,d_p,\theta_p) \\ \in\mathbb{R}_p}} \left(\frac{1 - \cos(D\langle\mathbf{n}_{fr}, \overrightarrow{\mathbf{f}.\boldsymbol{p}\mathbf{g}.\boldsymbol{p}}\rangle - \theta_p)}{2}\right)^{\gamma_{po}} \tag{9}$$

Here, γ_{pd} and γ_{po} are manually adjusted index factors within the range of [1.5, 2.0]. We introduce an offset d_ϵ in $C_{pd}(\phi)$ to ensure that the relative distance between pairwise objects during optimization falls within the range of $[d_p - d_\epsilon,\ d_p + d_\epsilon]$, where we set d_ϵ to be 0.2 (m). In $C_{po}(\phi)$, we define the operator $D\langle\cdot,\cdot\rangle$ to represent the angle between two vectors. We model the orientation cost using a cosine function that captures the angle difference between θ_p and $D\langle\mathbf{n}_{fr}, \overrightarrow{\mathbf{f}.\boldsymbol{p}\mathbf{g}.\boldsymbol{p}}\rangle$, aiming to minimize this difference and increase acceptance probability within the relaxed range. A sharp angle difference incurs a significant penalty.

Hierarchical support $C_{hs}(\phi)$ promotes the cohesion of objects with a hierarchical relationship in $\mathbb{R}_h$, ensuring that they either provide support or are supported by each other. The centers of their contact surfaces should have minimal

Manhattan distance (represented by 1-norm) to maintain close proximity:

$$C_{hs}(\phi) = \sum_{\mathbf{r}=<\mathbf{f},\mathbf{g}>\in\mathbb{R}_h} \alpha_{hs1}\|\mathbf{f}.\boldsymbol{p} + \mathbf{f}.\boldsymbol{l} \circ \mathbf{f}.\boldsymbol{\delta} \circ \hat{\mathbf{z}} - \mathbf{g}.\boldsymbol{p}\|_1{}^{\gamma_{hs}} + \sum_{\exists \mathbf{g} \wedge \neg \mathbf{g}.\lambda_{gr}\, \tilde{\mathbf{r}}=<\mathbf{o}\in\mathbb{O}_{mid},\mathbf{g}>,\tilde{\mathbf{r}}\notin\mathbb{R}_h} \min(\alpha_{hs2}\|\mathbf{o}.\boldsymbol{p} + \mathbf{o}.\boldsymbol{l} \circ \mathbf{o}.\boldsymbol{\delta} \circ \hat{\mathbf{z}} - \mathbf{g}.\boldsymbol{p}\|_1)^{\gamma_{hs}} \tag{10}$$

$$\alpha_{hs1} = \min_{\mathbf{r}=<\mathbf{f},\mathbf{g}>\in\mathbb{R}_h} \left(\frac{\|\text{proj}(\mathbf{g})\|_\text{A}}{\|\text{proj}(\mathbf{f}) \cap \text{proj}(\mathbf{g})\|_\text{A}}, \mathbf{INF} \right)$$
$$\alpha_{hs2} = \min_{\substack{\exists \mathbf{g} \wedge \neg \mathbf{g}.\lambda_{gr} \\ \tilde{\mathbf{r}}=<\mathbf{o}\in\mathbb{O}_{mid},\mathbf{g}>,\tilde{\mathbf{r}}\notin\mathbb{R}_h}} \left(\frac{\mathbf{INF}\|\text{proj}(\mathbf{o}) \cap \text{proj}(\mathbf{g})\|_\text{A}}{\|\text{proj}(\mathbf{g})\|_\text{A}}, 1 \right) \tag{11}$$

The first term is designed to prioritize the movement of the supported object toward the upper surface of the supporter in order to prevent any illegal lateral attraction toward the side of the supporter (as shown in Fig. 4). The second term is introduced to account for situations where an object may be attracted by another irrelevant physical supporter in $\mathbb{O}_{mid}$, excluding those within $\mathbb{R}_h$, when a prior supporter is occupied. To ensure a sharp increase in cost when there is no intersection between objects' horizontal projections, we incorporate weight $\alpha_{hs1,2}$ (Eq. 11) into our formulation. We define A-norm $\|\cdot\|_\text{A}$ for calculating the area of projection.

3.3.3 Haptic reuse constraint

Haptic reuse constraint is a crucial aspect in the migration of MR scenes, as it allows virtual objects to receive passive haptic feedback through physically interactive surfaces of real objects (as shown in Fig. 3). We consider this as a strong constraint, denoted by $C_h(\phi)$, which comprises three sub-constraints: **horizontal haptic reuse** $C_{hh}(\phi)$, **vertical haptic reuse** $C_{hv}(\phi)$, and **haptic orientation** $C_{ho}(\phi)$. These sub-constraints can be combined with appropriate weights to obtain the overall constraint.

$$C_h(\phi) = w_{hh}C_{hh}(\phi) + w_{hv}C_{hv}(\phi) + (1 - w_{hh} - w_{hv})C_{ho}(\phi) \tag{12}$$

where the weights $(w_{hh}, w_{hv}) = (0.4, 0.4)$ were set in our implementation, while the index adjustment factors $\gamma_{hh}, \gamma_{hv}, \gamma_{ho}$ in the following equations were manually adjusted within the range of [3.0, 4.0].

Horizontal haptic reuse leverages real objects to optimize the positioning of virtual objects, enabling a more realistic haptic experience in the horizontal direction. This constraint is represented by the cost function $C_{hh}(\phi)$ in Eq. (13), aiming to minimize the distance between the centers of virtual objects with horizontal haptic reuse property ($\mathbf{f} \in \mathbb{F}_{hor}$) and their corresponding real objects ($\mathbf{o} \in \mathbb{O}_{low} \cup \mathbb{O}_{mid}$):

$$C_{hh}(\phi) = \begin{cases} ll \sum\limits_{\mathbf{f}\in\mathbb{F}_{hor}} \min\limits_{\mathbf{o}\in \mathbb{O}_{low}\cup\mathbb{O}_{mid}} \|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\|^{\gamma_{hh}}, & m \le n \\ \sum\limits_{\mathbf{o}\in \mathbb{O}_{low}\cup\mathbb{O}_{mid}} \min\limits_{\mathbf{f}\in\mathbb{F}_{hor}} \|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\|^{\gamma_{hh}}, & m > n \end{cases} \tag{13}$$

where the dimension $m = \dim \mathbb{F}_{hor}$ represents the cardinality of the set $\mathbb{F}_{hor}$, while $n = \dim(\mathbb{O}_{low} \cup \mathbb{O}_{mid})$ denotes the number of objects in the combined sets $\mathbb{O}_{low}$ and $\mathbb{O}_{mid}$. When $m \le n$, there are fewer virtual horizontal haptic objects than low and medium real objects. In this case, our objective is to find a corresponding real object $\mathbf{o}$ for each virtual object $\mathbf{f}$ and optimize virtual objects' positions accordingly. Conversely, when $m > n$, our objective is to find a corresponding virtual object $\mathbf{f}$ for each real object $\mathbf{o}$ so that the real objects can receive a portion of the virtual objects in $\mathbb{F}_{hor}$.

Vertical haptic reuse is governed by the cost function $C_{hv}(\phi)$ in Eq. (14), which slightly differs from $C_{hh}(\phi)$. As illustrated in Fig. 5, the mapping strategy allows vertical haptic objects $\mathbf{f} \in \mathbb{F}_{ver}$ to leverage the presence of walls $\mathbf{b} \in \mathbb{B}_{walls}$ for haptic perception. Consequently, the optimization objective for positioning $\mathbf{f}$ encompasses not only $\mathbf{o} \in \mathbb{O}_{high}$ but also includes considerations for the wall $\mathbf{b}$.

$$C_{hv}(\phi) = \begin{cases} \sum\limits_{\mathbf{f}\in\mathbb{F}_{ver}} \min\limits_{\substack{\mathbf{o}\in\mathbb{O}_{high}, \\ \mathbf{b}\in\mathbb{B}_{walls}}} \left(\max(\|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\|, \|\mathbf{f}.\boldsymbol{p} - \mathbf{b}.\boldsymbol{p}\| - L_\infty(\mathbf{f}), 0) \right)^{\gamma_{hv}}, & m \le n \\ \sum\limits_{\mathbf{o}\in\mathbb{O}_{high}, \atop \mathbf{b}\in\mathbb{B}_{walls}} \min\limits_{\mathbf{f}\in\mathbb{F}_{ver}} \left(\max(\|\mathbf{f}.\boldsymbol{p} - \mathbf{o}.\boldsymbol{p}\|, \|\mathbf{f}.\boldsymbol{p} - \mathbf{b}.\boldsymbol{p}\| - L_\infty(\mathbf{f}), 0) \right)^{\gamma_{hv}}, & m > n \end{cases} \tag{14}$$

where $m = \dim \mathbb{F}_{ver}$ and $n = \dim(\mathbb{O}_{high} \cup \mathbb{B}_{walls})$. When $m \le n$, the number of vertical haptic objects is fewer compared to high real objects and walls, and it is expected that each $\mathbf{f}$ can be mapped to the corresponding $\mathbf{o} \in \mathbb{O}_{high}$ or $\mathbf{b} \in \mathbb{B}_{walls}$, optimizing the position of $\mathbf{f}$. However, when $m > n$, it is desired to have a corresponding $\mathbf{f}$ for each $\mathbf{o}$ or $\mathbf{b}$, allowing for optimization of their positions. To optimize the position of $\mathbf{f}$ within the space of $\mathbb{S}_\mathbb{R}$ while leveraging $\mathbf{b}$ in haptics, we introduce an offset $L_\infty(\cdot)$ to Eq. (14). The offset operator $L_\infty(\cdot)$ can be defined as $L_\infty(\mathbf{f} \in \mathbb{F}) = \|\mathbf{f}.\boldsymbol{l} \circ \mathbf{f}.\boldsymbol{\delta}\|_\infty$, aiming to quantify the length of the longest side of the virtual object. This causes a slight drift in the center position

of **f** instead of precise alignment with the wall **b**. The constraint $C_{ac}(\phi)$ (Eq. 7) ensures that the drift direction of **f** is toward the outer region of real scene $\mathbb{S}_{\mathbb{R}}$. Meanwhile, the global weight w_{in} (Eq. 6) and following **haptic orientation** constraint (Eq. 15) guarantee that the front face of **f** remains closely attached to the wall without separation.

Haptic orientation with $C_{hh}(\phi)$ and $C_{hv}(\phi)$ allows the virtual object **f** to strategically position itself in order to leverage real objects or walls for passive haptic feedback and accurately represent the interaction surface. Since the orientation of $\mathbf{f} \in \mathbb{F}_{ver}$ is unrestricted in previous haptic reuse processes, certain virtual vertical haptic objects like bookshelves or paintings may face away from the scene or toward a wall. Therefore, we introduce a constraint $C_{ho}(\phi)$ to ensure that interactive vertical haptic objects are oriented toward the interior of the scene:

$$C_{ho}(\phi) = \sum_{\mathbf{f}\in\mathbb{F}_{ver}} \left(\frac{1 - \cos(D\langle \mathbf{n}_{fr}, \overrightarrow{\mathbf{f}.\boldsymbol{p}\mathbb{B}_{ground}.\boldsymbol{p}} \rangle)}{2} \right)^{\gamma_{ho}} \tag{15}$$

where we utilize the cosine of the angle between $\mathbf{n}_{fr}$ and vector $\overrightarrow{\mathbf{f}.\boldsymbol{p}\mathbb{B}_{ground}.\boldsymbol{p}}$ to quantify the cost (similar to Eq. 9), ensuring that the object's front vector approximately point at the center of the ground.

3.3.4 Scale fitting constraint

Constrained by the cost function $C_{sf}(\phi)$ (Eq. 16), scale fitting aims to adjust the current size of the virtual object in order to wrap it around its corresponding real object tightly. During migration, when haptic mapping is utilized, the virtual and real objects may still have different geometric wrap sizes despite having identical positions. If we directly place the virtual objects without adjusting their scale, virtual objects would be pierced in the $\mathbb{S}_{\mathbb{V}}$ and $\mathbb{S}_{\mathbb{R}}$ mixed scene output, resulting in a lack of tight fit between virtual objects and real surfaces, thereby compromising passive haptic fidelity.

$$C_{sf}(\phi) = \begin{cases} \mathbf{INF}, & \exists \mathbf{o} \in \mathbb{O},\ \exists \mathbf{f} \in \mathbb{F},\ \Delta_{max}(\mathbf{f}) > \delta_\tau \\ & \text{body}(\mathbf{o}) \cap \text{body}(\mathbf{f}) \neq \varnothing \\ \sum\limits_{\substack{\mathbf{o}\in\mathbb{O},\ \mathbf{f}\in\mathbb{F}, \\ \text{body}(\mathbf{o})\cap\text{body}(\mathbf{f})\neq\varnothing}} & \boldsymbol{\alpha_{sf}} \left(\dfrac{\|\text{body}(\mathbf{o}) - \text{body}(\mathbf{f})\|_{\mathbb{V}}^{\gamma_{sf}}}{\|\text{body}(\mathbf{f}) - \text{body}(\mathbf{o})\|_{\mathbb{V}}^{\gamma_{sf}}} \right), \\ & otherwise \end{cases} \tag{16}$$

where the V-norm $\|\cdot\|_{\mathbb{V}}$ is defined to calculate the volume of geometric bodies resulting from Boolean operations on objects. In the first condition of $C_{sf}(\phi)$, the operator $\Delta_{max}(\cdot)$ quantifies the current deformation of an object by representing its most significant scale ratio (as shown in Eq. (17)). If the deformation extent exceeds a threshold value δ_τ, the cost will increase to **INF**. It ensures that objects with significantly different sizes cannot be mapped together through haptic reuse using scaling techniques. For instance, optimizing a chair into a long sofa would result in excessive stretching and is therefore prevented. We manually set the threshold value δ_τ to 30% based on references [8].

Fig. 6 The scale fitting of a virtual blue sofa mapped to a real black armchair. **a** An armchair instance for a low real object. **b** Spatial mesh representation of the armchair. **c** Loose fitting in terms of scale. **d** Tight fitting in terms of scale. **e** Excessive deformation resulting from scale fitting. **f** Optimal fit in terms of scale

$$\Delta_{max}(\mathbf{f} \in \mathbb{F}) = \frac{\max(\mathbf{f}.\delta_x,\ \mathbf{f}.\delta_y,\ \mathbf{f}.\delta_z,\ 1)}{\min(\mathbf{f}.\delta_x,\ \mathbf{f}.\delta_y,\ \mathbf{f}.\delta_z,\ 1)} - 100\% \tag{17}$$

In the second condition, when the virtual-real object pairs exhibit similar shapes and sizes, $C_{sf}(\phi)$ quantifies the cumulative cost of non-overlapping volumes between the virtual and real objects in each pair, thereby reducing their size differences. The weight parameters $\boldsymbol{\alpha_{sf}} = (\alpha_{sf1}, \alpha_{sf2})$ were manually assigned as (0.7, 0.3), while the trim factor was set to $\gamma_{sf} \in [2.0, 3.0]$. Figure 6 illustrates various instances of scale fitting with different deformations.

3.3.5 The priority of the constraints

The weight vector **w** is determined empirically to stipulate the relative importance of each sub-constraint. Readers can fine-tune this vector according to their specific environment, but such adjustments should be kept to a minimum. The index adjustment factors γ are employed in the overall cost function inside each term of the cost function to establish the priority among sub-constraints. We present this priority as a partial order in Eq. (18):

$$\begin{aligned} w_{no} &\succeq w_{in} \succeq C_h(\phi) \succeq C_{sf}(\phi) \succeq (C_p(\phi), C_{hs}(\phi)) \\ &\succeq C_{ac}(\phi) \end{aligned} \tag{18}$$

where we use this partial order relation to guide the value of γ: higher partial order relations correspond to larger γ values and faster iterative convergence of sub-cost functions. The global weight w_{no} holds the highest priority to prevent piercing after optimization, while w_{in} ensures objects remain inside the room. This is followed by strong constraint haptic reuse C_h, scale fitting C_{sf}, and weaker constraints such as C_p, C_{hs}, and C_{ac}. For instance, we expect **scaling fitting** to begin only after virtual object positioning has been achieved through **haptic reuse** constraint. Equation 18 indicates that $\gamma_{h(\cdot)}$ should be greater than γ_{sf} for constrained optimization in sequential order.

4 Experiment

We evaluated our approach on three virtual scenes (VS1, VS2, VS3) named *LivingRoom*, *Clinic*, and *Bakery*, as well as two real scenes (RS1, RS2) referred to as *Office* and *ConferenceRoom*. We conducted six scenario combinations by migrating three virtual scenes for each pair of real scenes. The raw input data and scene abstraction details are presented in Fig. 7. The parameters of the scenes are provided in Table 1, where there are differences in the number of objects and scene size between the virtual and real scenes. The hardware utilized in the experiment comprises the following:

- HoloLens 2, employed for scanning real (physical) scenes, adjusting real OBBs through a programmed user interface, and rendering the mixed scene.
- A laptop (equipped with Intel i9 12900k processor, 32GB RAM, and NVIDIA RTX A2000 graphics card), utilized for executing the *v2rSA* optimization framework and performing scene migration computations.

In Table 2, we documented the performance of our method across various scenario combinations, including the iterations number and time costs during relative cooling in *v2rSA* optimization, along with their corresponding initial temperature t_0 as defined in Eq. (2). The optimization procedure does not incorporate parallel acceleration.

We have compared our method with three comparison methods, namely **RandomIn**, **MakeItHome** [44], and **Human**, regarding quality and surface registration. **RandomIn** is a programming-based implementation where virtual objects are randomly placed in the real scene without considering constraints from other objects. **MakeItHome**, as a classic scene layout method based on mathematical constraints, does not consider haptics mapping strategy from real objects in the scene migration experiment but roughly retains the semantic relationship between virtual objects and the layout between virtual objects and the real scene boundary. The **Human** method involves the manual placement of virtual objects by designers to create a migrated scene, serving as a benchmark for evaluating our method.

4.1 Quality

To demonstrate the quality of our method, we included the results obtained by three other methods for comparison in each of the six scenario combinations. Figure 8 illustrates the migration results achieved by our proposed **SMigraPH** method, as well as **RandomIn**, **MakeItHome** [44], and **Human**. When the scales of virtual and real scenes are similar, such as in VS1-RS1 and VS3-RS2 scenarios, both **SMigraPH** and **Human** exhibit a distinct advantage over the other two methods. In contrast, **RandomIn** and **MakeItHome** fail to maintain a semantic mapping from the virtual domain to the real domain, resulting in a disordered scene migration.

Our approach facilitates the creation of numerous remarkable interactive haptic mappings. The virtual hospital bed in VS2-RS2 is positioned on a physical, spacious table, ensuring precise alignment that allows individuals to even recline upon it. Similarly, the virtual sofa in VS1-RS1 can be comfortably sat upon, simulating the tactile properties of an actual sofa, which just like the material of the real sofa has been changed to virtual white. The virtual tableware and other physical small objects in VS3-RS2 can coexist on the real table simultaneously. **SMigraPH** generates mixed scenes with ample vacant space, even when there are numerous virtual objects present, as seen in VS2-RS1 and VS3-RS1. In contrast, the virtual dining tables are roughly distributed in the vicinity of the others using **MakeItHome** in VS3-RS2. Due to the constraints of daily physical environments, walls typically intersect perpendicularly. Our method should also be effective in certain non-orthogonal physical spaces, such as rooms with acute or obtuse angles in the top view. In extreme cases, vertical haptic objects may acquire passive haptic feedback via walls and potentially appear congested in sharp corners, rendering them inconvenient for users to access.

Regarding congruence, **MakeItHome** occasionally achieves partial surface registration by chance, as exemplified in the case of the glass table in VS1-RS1. However, due to optimization and scanning accuracy limitations, the plane registration performance of **SMigraPH** is slightly inferior compared to that of **Human**, as observed in Table 3 across most scenario combinations. Moreover, our method does not extensively consider the mapping strategy for $\mathbb{F}_{no}$, resulting in less reasonable potted plants in VS1-RS1 compared to **Human**. When migrating a large virtual scene to a smaller real scene, the density increases, and some semantic mappings become unreasonable. To address this issue, **Human** may compromise on object deformations, such as the two small sofas no longer being placed around the hospital bed like in our method for VS2-RS1. Due to inertial

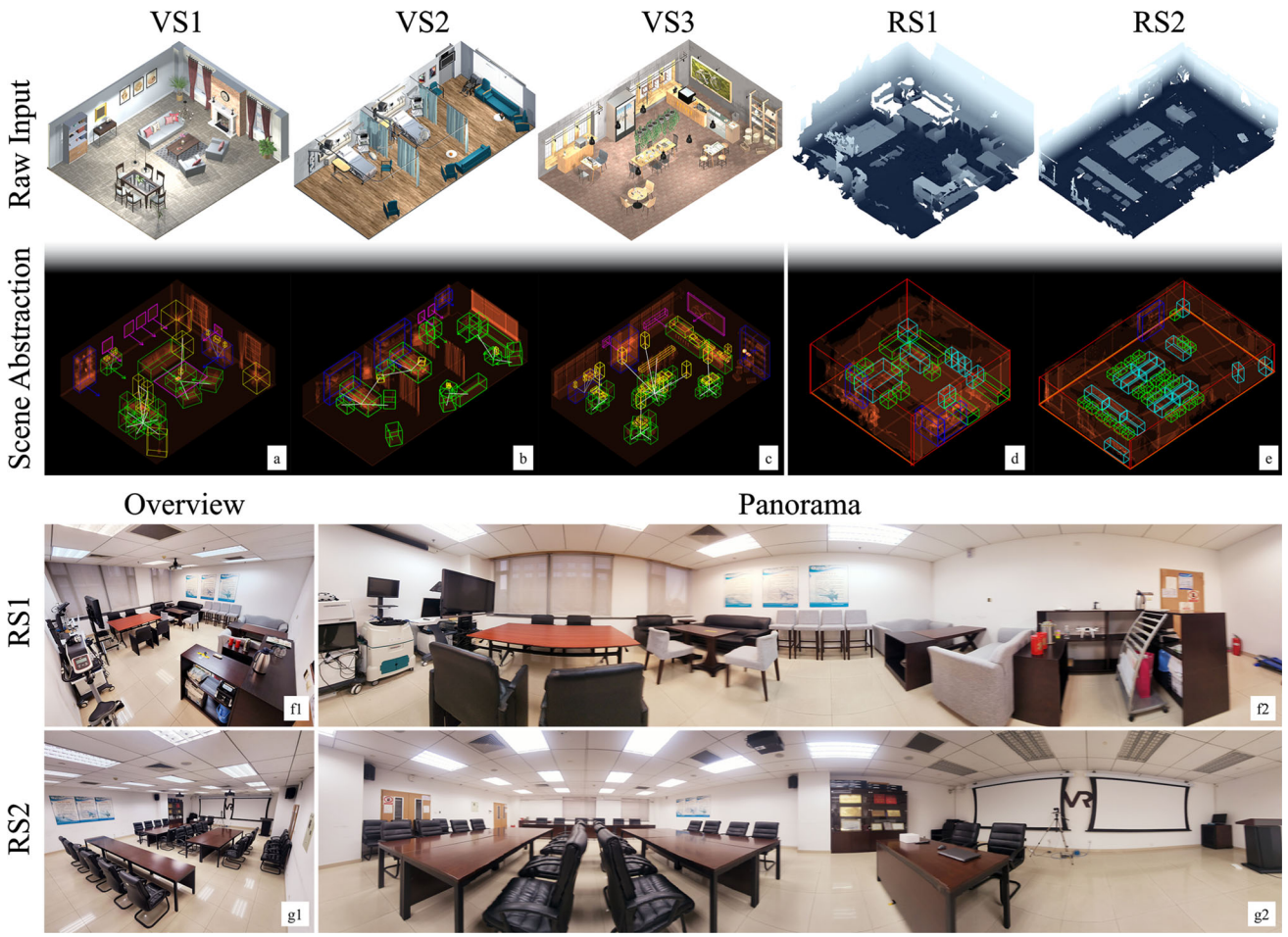


Fig. 7 The columns from **a**–**e** display virtual scenes VS1, VS2, and VS3, as well as real scenes RS1 and RS2. The raw input for each scene is presented in the top row. The second row represents the scene abstraction data using different colors: green for $\mathbb{F}_{hor}$ or $\mathbb{O}_{low}$, blue for $\mathbb{F}_{ver}$ or $\mathbb{O}_{high}$, yellow for $\mathbb{F}_{no}$, cyan for $\mathbb{O}_{mid}$, red for $\mathbb{B}_{walls}$, orange for $\mathbb{B}_{ground}$, white denotes $\mathbb{R}_p$, and gray signifies $\mathbb{R}_h$. Subsequently, rows (f, g) show scene overview (f1, g1) and panorama (f2, g2) of *Office* and *ConferenceRoom*

Table 1 Scene abstraction parameters

	Size	Area (m^2)	Number of virtual objects			Number of relationships		Number of real objects		
			VerH[a]	HorH[b]	NoH[c]	Pair	Hier.	Low	Mid	High
VS1 *LivingRoom*	6.8 m*8 m	54.4	2	12	14	13	8	–		
VS2 *Clinic*	5.9 m*13 m	76.7	3	16	5	12	3			
VS3 *Bakery*	7.4 m*11.3 m	83.6	3	21	38	22	25			
RS1 *Office*	5.8 m*6 m	34.8	–			–		10	8	3
RS2 *ConferenceRoom*	8.2 m*12.6 m	103.32						23	16	1

[a]vertical haptic objects, [b]horizontal haptic objects, [c]non-haptic objects

Table 2 Optimization performance and process parameters of *v2rSA*

	VS1-RS1	VS1-RS2	VS2-RS1	VS2-RS2	VS3-RS1	VS3-RS2
t_0	500	900	600	1000	800	1200
Iteration num.	12,000	19,000	16,000	25,000	20,000	30,000
Time cost (s)	116	178	161	213	362	398

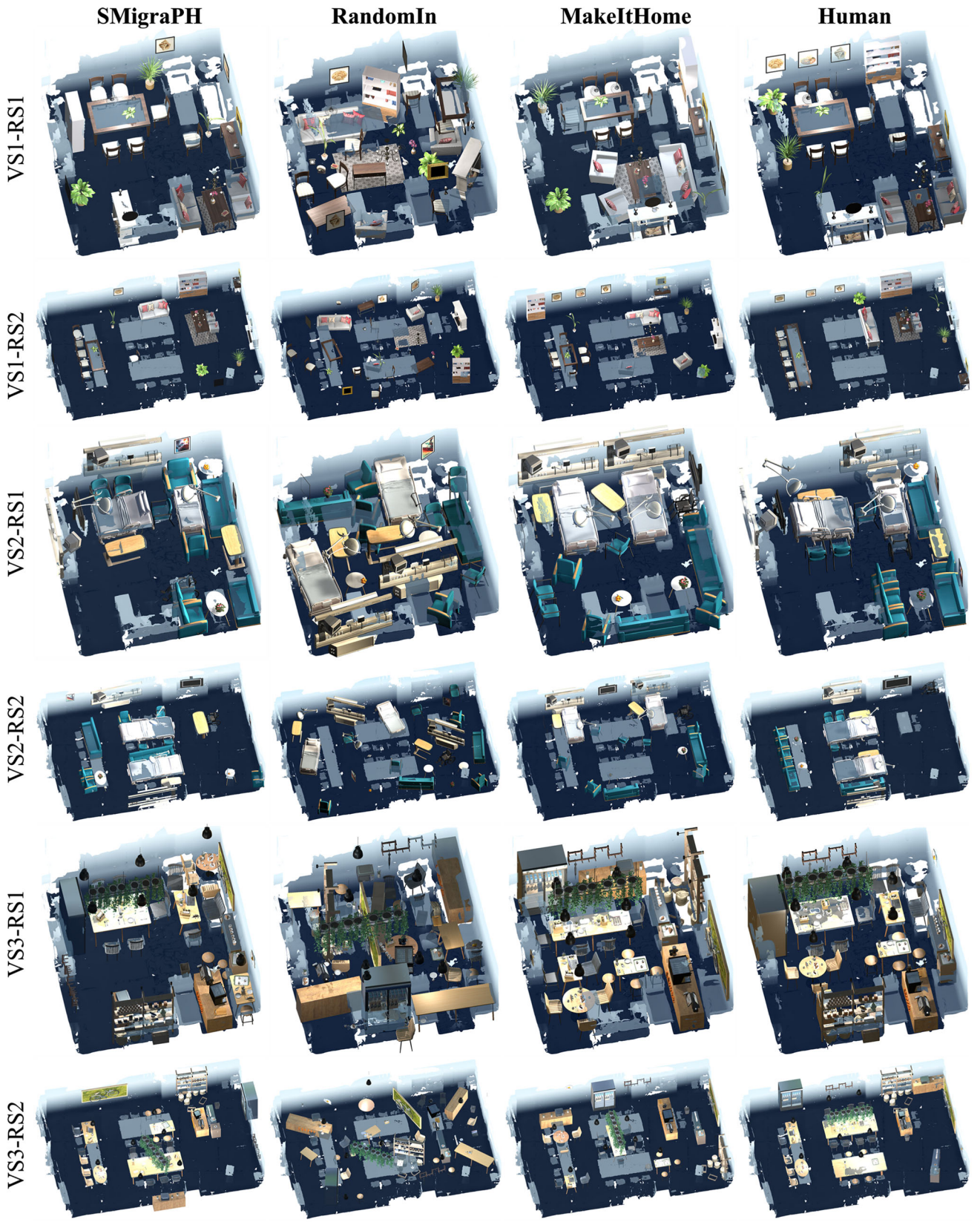


Fig. 8 The scene layout overview after migration based on our approach (the first column) is compared with three other methods (following columns) in this figure. The results of six scenario combinations are presented from top to bottom rows. To demonstrate the congruence between virtual objects and real scenes using different methods, the virtual layout optimization results are integrated with the corresponding real scene meshes. Pseudo-colors are applied to enhance the meshes of the real scene, gradually transitioning from light to dark blue as depth increases relative to the camera viewport

Table 3 Analysis of surface registration quantified in chamfer distance (cm)

		VS1, small room (*LivingRoom*, 6.8 m * 8 m)			VS2, medium room (*Clinic*, 5.9 m * 13 m)			VS3, large room (*Bakery*, 7.4 m * 11.3 m)		
		CD Avg ± std. dev.	$(CD_{\mathbf{SMi}} - CD_i)/CD_i$	p	CD Avg ± std. dev.	$(CD_{\mathbf{SMi}} - CD_i)/CD_i$	p	CD Avg ± std. dev.	$(CD_{\mathbf{SMi}} - CD_i)/CD_i$	p
RS1, small room (*Office*, 5.8 m*6 m)	**SMi**	23.19 ± 1.25			29.54 ± 1.71			30.56 ± 5.55		
	Ran	47.71 ± 3.43	**−51.4%**	***< 0.001****	54.04 ± 4.02	**−45.3%**	***< 0.001****	60.45 ± 5.90	**−49.4%**	***< 0.001****
	Mak	30.23 ± 1.24	*−23.3%*	***< 0.001****	35.87 ± 2.48	*−17.6%*	*0.003*	44.43 ± 4.22	**−31.2%**	*0.004*
	Hum	21.80 ± 1.19	*6.4%*	0.145	26.33 ± 0.98	*12.2%*	0.012	38.11 ± 3.21	*−19.8%*	0.046
RS2, large room (*ConferenceRoom*, 8.2 m * 12.6 m)	**SMi**	31.21 ± 5.08			39.25 ± 6.34			44.73 ± 3.12		
	Ran	50.13 ± 5.59	**−37.7%**	*0.001*	63.71 ± 4.31	**−38.4%**	***< 0.001****	77.63 ± 10.10	**−42.4%**	***< 0.001****
	Mak	45.33 ± 3.82	**−31.1%**	*0.002*	55.15 ± 4.32	**−28.8%**	*0.003*	61.21 ± 4.04	**−26.9%**	***< 0.001****
	Hum	28.55 ± 2.47	*9.3%*	0.373	35.34 ± 2.85	*11.1%*	0.293	41.29 ± 3.85	*8.3%*	0.203

Italics percentage: reduces italics p: there is a difference with a confidence of less than 0.005; Bold percentage: reduces by more than 25%; Bold italics p: there is a significant difference

thinking, humans tend to place the same class of objects together to reduce mental consumption during task load, but this will cause semantic confusion in the local area (virtual armchair corresponds to low cabinets) and more crowded. (Three virtual armchairs are placed side by side). Instead, our program assigned the armchair to another physical desk and found a virtual wheelchair to replace it. In this confined case, **Ours** can be considered to have advantages over **Human**. Additionally, **Human** may sacrifice passive haptics reuse of certain objects (e.g., many tables in VS3-RS1) to achieve semantically coherent scenes after migration. Please note that although **Human** has spent a lot of effort and cannot compete with **Ours** in terms of automation, in this work we only use **Human** as a benchmark close to ground truth and as a direction for quality optimization (like what they did in [15]). We do not intend to surpass **Human** in all metrics.

4.2 Analysis of surface registration

In terms of scenes congruence between $\mathbb{S}_\mathbb{V}$ and $\mathbb{S}_\mathbb{R}$, the matching condition between the geometries of two scenes after implementing different migration methods is analyzed by employing the chamfer distance [4] of two point clouds to quantify the surface registration. As shown in Eq. (19), the chamfer distance is a commonly used loss metric in 3D reconstruction [25]:

$$CD(\mathbf{S_V}, \mathbf{S_R}) \equiv \frac{1}{|\mathbf{S_V}|} \sum_{v \in \mathbf{S_V}} \min_{r \in \mathbf{S_R}} \|v - r\| \tag{19}$$

where the set $\mathbf{S_V}$ represents the point cloud of virtual objects in $\mathbb{S}_\mathbb{V}$, while the set $\mathbf{S_R}$ denotes the point cloud of the real scene $\mathbb{S}_\mathbb{R}$. This equation signifies the cumulative mean of the minimum distance between all points v in $\mathbf{S_V}$ and any given point in $\mathbf{S_R}$.

Each method in a different scenario combination produces five iterative results, and we collect the calculated chamfer distance into Table 3. For each data block of scenario combination, the first column presents the average chamfer distance along with its standard deviation. The second column illustrates the rate of increased chamfering loss when **SMigraPH** compared to the other three methods. In the third column, a t-test is conducted assuming there is no statistical difference between the chamfer distances calculated by the other three methods and our **SMigraPH** method.

In most cases, the metrics of our method closely resemble those exhibited in hand-placed scenes by **Human** and exhibit a statistically significant difference compared to the **RandomIn** and **MakeItHome** methods. This is particularly evident in dense virtual scenes and large real scenes where our method's chamfering loss is significantly reduced. While **MakeItHome** can provide a more reasonable layout for pure virtual scenes in empty spaces, it overlooks the haptic reuse

offered by real scenes, resulting in large variation compared to the chamfer distance of **SMigraPH**. This is even more so for **RandomIn**.

In certain cases, the utilization of **RandomIn**, as demonstrated in VS1-RS2, may serendipitously result in the coupling of specific virtual objects with real objects, thereby leading to a marginal reduction in chamfer distance that closely approximates **MakeItHome**. Furthermore, in the VS3-RS1 combination, which involves mapping a large virtual scene to a smaller real scene, our method achieves a chamfer distance that is 19.8% lower than that achieved by **Human**. As discussed in Sect. 4.1, this phenomenon may arise due to the trade-off made by **Human**, sacrificing passive haptics reuse in order to achieve semantically coherent scenes.

In general, our approach significantly reduces chamfering error (highlighted in bold or italics in Table 3). Figure 9 illustrates point cloud chamfer distances and their Weibull distributions for combinations VS1-RS1 and VS3-RS2 using four different methods, with the reference points set shown in Fig. 10. It can be observed that our method exhibits a higher concentration of points within the low loss region (depicted in deep violet) compared to traditional control methods where more points within high loss region (depicted in white or red). Both our method and **Human** demonstrate effective cross-scene surface registration.

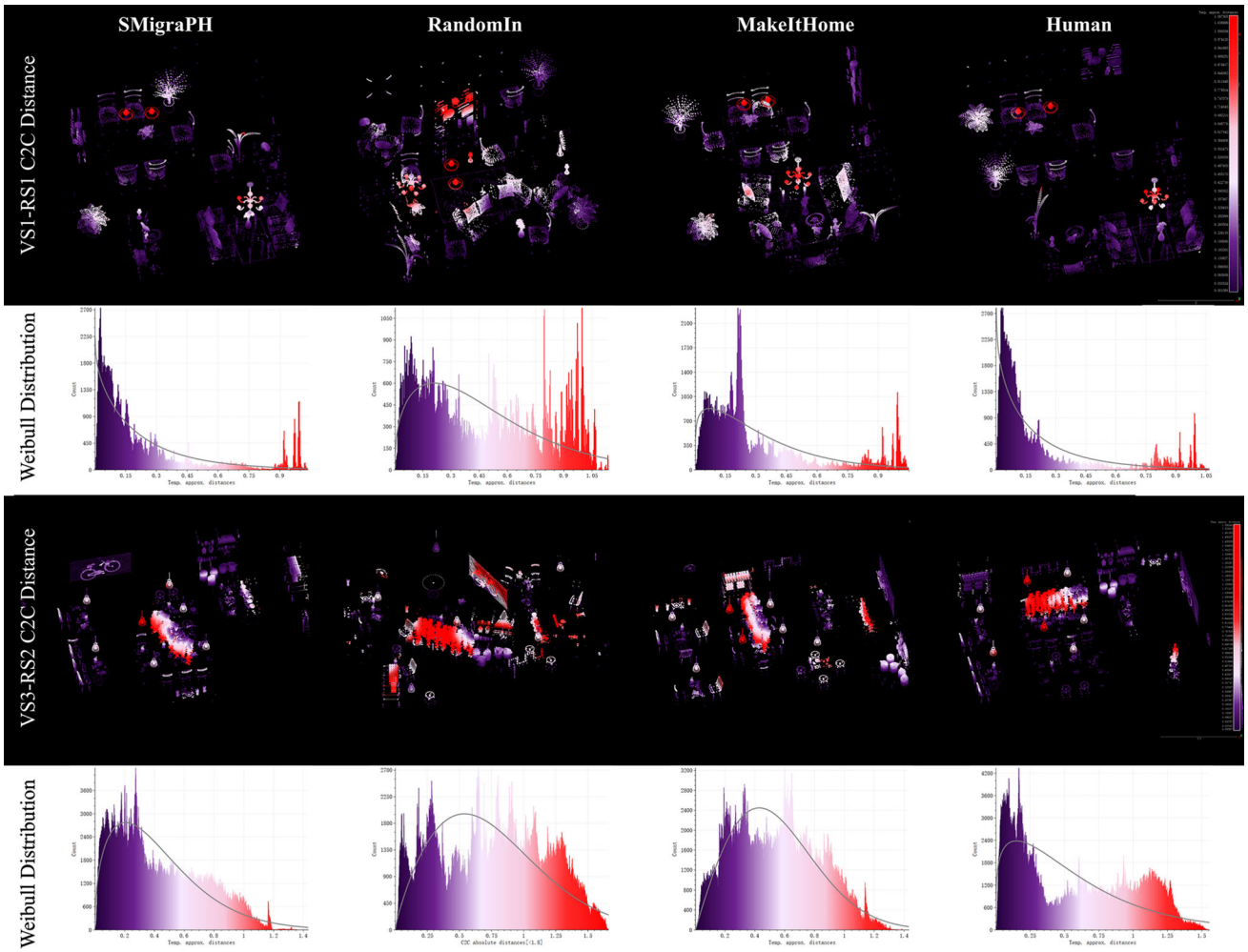


Fig. 9 A chamfer distance diagram depicting the C2C comparison (with Weibull distribution [29] in each second row) from VS obtained by different methods to RS. The first two rows illustrate the combination of VS1-RS1, while the subsequent two rows demonstrate the combination of VS3-RS2. The distances of points in the VS relative to the reference, ranging from near to far, are progressively indicated by a color gradient of violet-white-red. The accuracy of surface registration in the corresponding method of a C2C increases with a higher proportion of dark violet parts

Fig. 10 The references for calculating point cloud to cloud (C2C) chamfer distance encompass the point clouds and mesh data of two distinct indoor real scenes RS1 and RS2

5 User study

To further substantiate the validity of our proposed approach **SMigraPH**, in conjunction with three other comparative methods (**RandomIn**, **MakeItHome**, and **Human**), we have conducted a user study based on haptic interaction to acquire both objective and subjective metrics.

5.1 Haptic interaction study design

Participants. We recruited 36 participants, comprising 20 male and 16 female users aged between 23 and 45 years old. Among them, 9 users had no prior experience with MR devices, while the remaining 27 users possessed varying degrees of experience or proficiency in using MR devices. Each participant was randomly assigned to either to a combination of virtual and real scenes.

Task 1 (T1). Participants were instructed to navigate through the entire mixed scene and interact with labeled virtual objects, both vertical haptic (VerH) and horizontal haptic (HorH) objects, as quickly as possible. We required participants not to collide or pierce any virtual objects. Once an object was interacted with by the participant, its label would be automatically removed by the program. T1 terminated when all labels had been removed. The purpose of T1 is to evaluate the overall rationality and interactive efficiency of the migrated scene by measuring the task's completion time.

Task 2 (T2). We consecutively assigned random labels to eight virtual objects, including vertical haptic (VerH), horizontal haptic (HorH), and non-haptic objects (NoH). We instructed participants to evaluate them based on their haptic effects, topological relationships, and placement rationality. Subsequently, the user was required to provide a comprehensive assessment of each object's qualities to the experimenter. T2 terminated once all eight evaluations had been reported. The purpose of T2 is to evaluate the local rationality, passive haptics feasibility, and subjective fidelity of the migrated scene by recording score reports and calculating perception loss rates.

Procedure. For both T1 and T2, each participant assigned to a specific combination of scenes will undergo testing using the aforementioned four methods. Once the user enters the real scene, all four methods' virtual environments (VE) will be presented to participants in a randomized order for completing two tasks. A 10-minute break will be provided between each VE, with no time interval between T1 and T2 within each VE. All the migrated VE layout of four methods are pre-optimized or prepared in advance. Besides, participants are allowed to fine-tune the generated real OBBs through the HMD user interface, following which our method will seamlessly re-optimize in the background. Due to potential unreasonable placement of certain virtual objects, users may face difficulty locating them all before concluding T1; participants have discretion to terminate T1 prematurely, and their final time consumption on this task will be recorded as a ratio of **actual elapsed time** to **task completion rate**. [1] In T2, if an object labeled for assessment is not found, participants can choose whether or not to continue with the task; any unreported score for that object would be considered as a **complete loss** (as shown in Table 4).

Metrics and statistical analysis. The performance of T1 is evaluated based on the objective metric of **time consumption** measured in seconds. As participants are required to interact efficiently with the mixed scene, a decline in the effectiveness of haptic feedback, environmental conditions, or scene density for the corresponding scene migration method generally leads to an increase in time consumption. The performance of T2 is evaluated based on the subjective metric of **fidelity loss rate**. Within the specific scenes combination, the calculation of loss rate for each method is derived from the user's comprehensive assessment score report. The assigned integer scores for objects in this metric range from 0 to 3, representing no loss, slight loss, moderate loss, and complete loss, respectively. The loss rate is determined as a weighted average of all reported scores relative to complete loss. A higher loss rate indicates poorer subjective perceptions toward virtual objects by users. Two metrics for four methods across six scenes combinations are presented in Fig. 11 and Table 4. For each metric, we compared the values obtained from our method with those from **RandomIn**, **MakeItHome**, and **Human**. The sample population was derived from the same batch of participants. In T1, a total sample size of 36 was considered for time consumption (TC). In T2, the fidelity loss rate (LR) was calculated based on score reports of 288 original samples (8 per person), resulting in 12 LR samples across 6 scenes combinations. Therefore, it is reasonable to assume that these data follow a normal distribution. We conducted

[1] For instance, if a participant spends 120 s interacting with 20 out of 24 objects, then their final time consumption would amount to $120/(20/24) = 144(s)$.

Table 4 The subjective fidelity loss rate of T2, with comprehensive assessment score reports

		The number of reports for virtual objects corresponding to some score (haptic objects + no haptic objects)										Total number of reports for virtual objects (H. Objs. + N. H. Objs.)			Fidelity loss rate (haptic objects + no haptic objects) $\Sigma(avgNum_i * Score_i)/(totNum * Score_{CI})$		
		No loss ($Score_{NI} = 0$)			Slight loss ($Score_{SI} = 1$)			Moderate loss ($Score_{MI} = 2$)			Complete loss ($Score_{CI} = 3$)						
		VS1	VS2	VS3	VS1	VS2	VS3	VS1	VS2	VS3	VS1	VS2	VS3	VS1	VS2	VS3	VS1	VS2	VS3
SMi	RS1	14 + 10	24 + 6	11 + 8	10 + 9	6 + 6	11 + 1	4 + 0	4 + 1	6 + 2	0 + 1	0 + 1	7 + 2	28 + 20	34 + 14	35 + 13	**21.4%** +**20%**	***13.7%*** + ***26.2%***	41.9% + 28.2%
	RS2	17 + 17	20 + 11	18 + 11	4 + 4	8 + 5	6 + 6	4 + 0	4 + 0	4 + 0	1 + 1	0	2 + 1	26 + 22	32 + 16	30 + 18	**19.2%** + **10.6%**	**16.7%** + **10.4%**	***22.2%*** + ***16.7%***
Ran	RS1	0 + 6	0	1 + 2	1 + 1	4 + 2	1 + 2	7 + 3	6 + 1	7 + 4	15 + 15	21 + 14	20 + 11	23 + 25	31 + 17	29 + 19	*86.9%* + *69.3%*	*84.9%* + *90.1%*	*86.2%* + *75.4%*
	RS2	0 + 3	1 + 0	0	1 + 3	1 + 3	2 + 1	7 + 5	6 + 4	7 + 2	19 + 10	27 + 6	23 + 13	27 + 21	35 + 13	32 + 16	*88.9%* + *68.3%*	*89.5%* + *74.3%*	*88.5%* + *91.7%*
Mak	RS1	2 + 6	2 + 3	2 + 0	8 + 4	10 + 5	7 + 0	11 + 2	13 + 3	8 + 8	7 + 8	9 + 3	14 + 9	28 + 20	34 + 14	31 + 17	*60.7%* + *53.3%*	*61.8%* + *47.6%*	*69.8%* + *84.3%*
	RS2	6 + 8	4 + 5	4 + 5	3 + 7	10 + 3	12 + 1	7 + 6	15 + 3	15 + 3	8 + 3	8 + 0	6 + 2	24 + 24	37 + 11	37 + 11	*56.9%* + *38.9%*	*57.6%* + *27.3%*	*54.1%* + *39.4%*
Hum	RS1	17 + 13	25 + 9	5 + 7	9 + 7	7 + 3	13 + 4	2 + 0	3 + 0	14 + 1	0	0 + 1	3 + 1	28 + 20	35 + 13	35 + 13	**15.5%** + **11.7%**	**12.4%** + **15.4%**	***47.6%*** + ***23.1%***
	RS2	17 + 18	20 + 12	18 + 12	3 + 4	8 + 5	7 + 6	3 + 0	3 + 0	3 + 0	1 + 2	0	1 + 1	24 + 24	31 + 17	29 + 19	**16.7%** + **13.9%**	**15.1%** + **9.8%**	**18.4%** + **15.8%**

Italics: has a loss exceeding 75% in one component; Bold: has a loss of less than 25% in both components; Bold italics: has a loss of less than 25% in one component

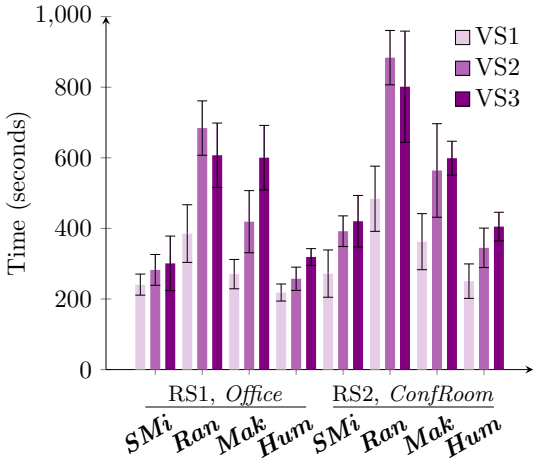


Fig. 11 The objective time consumption of T1

a t-test to examine statistical differences, and the resulting p-values are presented in Table 5.

5.2 Results and discussion

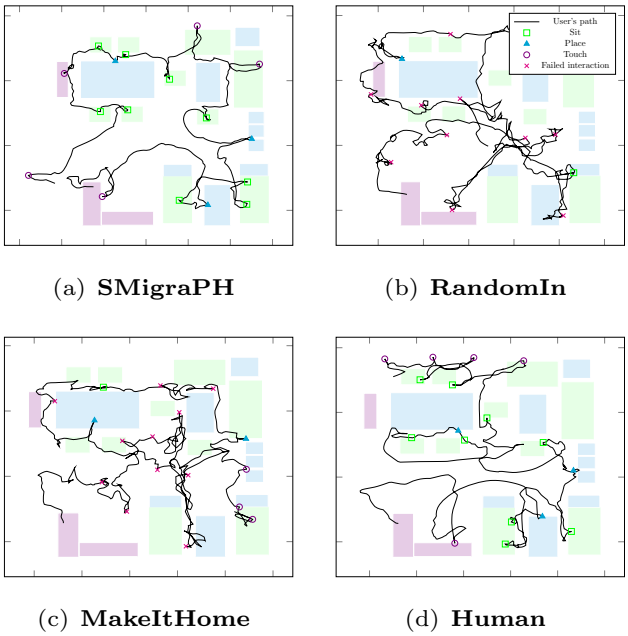
Objective time consumption. As depicted in Fig. 11, our method demonstrates less time consumption compared to the **RandomIn** and **MakeItHome** approaches across most scenario combinations and even exhibits similar interaction efficiency to that of **Human** in certain scenario combinations. Moreover, as shown in Table 5, the task completion time of our method significantly differs from that of **RandomIn** and **MakeItHome**, but closely approximates that of **Human**. In a large real scene such as the RS2, *ConferenceRoom*, where user motion trajectories increase, task time consumption tends to elongate. When the virtual scene's load is increased such as in the VS3, the time consumption disparity between our method and traditional methods becomes more pronounced, since the cases where the virtual layouts are completely lost are greatly reduced due to our constraints of haptic reuse and scale fitting.

Subjective fidelity loss. After collecting reports of perception fidelity loss for both haptic and non-haptic objects, we calculated the loss rates by weighted percentage, resulting in the last column of Table 4. Subjective reports of non-haptic objects tend to be less costly as they do not require real objects for placement, while restrictions on haptic objects are more stringent. Overall, our method yielded fewer perceptive losses with less than 25% (in bold) appearing in several scenario combinations across all object categories. The **Human** method resulted in smaller losses with most achieving a rate of less than 25% ($<$ 25% in single haptic or non-haptic objects category are marked as bold italics). **RandomIn** performed poorly with reported loss rates exceeding 75% under all combinations (in italics). The traditional **MakeItHome**

Table 5 Statistical analysis of TC and LR

Method	TC Avg ± std. dev.	$(TC_{SM} - TC_i) / TC_i$	p	Samp. size	LR Avg ± std. dev.	$(LR_{SM} - LR_i) / LR_i$	p	Samp. size
SMi	318.09 ± 87.64			36	0.206 ± 0.083			288 → 12
Ran	641.05 ± 199.14	**−50.4%**	***< 0.001****		0.828 ± 0.081	**−75.1%**	***< 0.001****	
Mak	469.24 ± 153	**−32.2%**	***< 0.001****		0.543 ± 0.144	**−62.1%**	***< 0.001****	
Hum	299.24 ± 75.01	6.3%	0.16		0.179 ± 0.095	14.9%	0.07	

Bold percentage: reduces by more than 25%; Bold italics p: there is a significant difference



(a) **SMigraPH** (b) **RandomIn**

(c) **MakeItHome** (d) **Human**

Fig. 12 A participant's spatial trajectories under four methods in migration VS1-RS1. The low, medium, and high real objects are represented by green, cyan, and violet squares respectively

method does not utilize haptic reuse resulting in suboptimal loss rates. As shown in Table 5, compared to **RandomIn** and **MakeItHome**, our method reduced loss rates by over half with significant differences while showing minor increases compared to **Human**.

Spatial trajectories. The spatial trajectories presented in Fig. 12 illustrate a median participant's movements during T1 under four different methods (in scene migration from *LivingRoom* to *Office*), with various interaction points marked (the participant was asked to report the interaction type with each virtual object during this process). The virtual scene layout results obtained by our method and **Human** align more closely with real scene layouts, resulting in more regular participant trajectories. Conversely, the other two methods occupy excessive free space within the physical room due to haptics loss, resulting in more chaotic participant trajectories and an increased number of failed interactions.

Algorithm convergence and weight vector. With appropriate weight vector settings and the theoretical underpinning of the optimization framework described in Sect. 3.2, we can achieve convergence of the virtual scene to an approximate global optimal solution within a finite time and a limited number of iterative steps. The total iteration number is generally directly proportional to the complexity of the scene, as illustrated in Table 2. As described in Sect. 3.3.5, the assignment of the weight vector $\mathbf{w}$ for the cost function should fall within a reasonable range, in accordance with the empirical findings. For instance, altering the values of (w_p, w_h) from (0.15, 0.35) to (0.35, 0.15) may result in the deliberate hovering and clustering of certain virtual objects, rather than seeking out other physical entities for object registration. In each constraint, when we conduct a mapping on a small scene like RS1, we found that if W_{no} and W_{in} are not guaranteed to be in the leading position of the partial order relation, many objects will be tightly attracted or just be intersected, and directly be crowded out by C_{ac} (the last constraint in partial order) locating on the outside of the wall. For another example, if we set γ_{sf} in Sect. 3.3.4 to 1.0, which is smaller than γ_{pd}, some objects will no longer consider deformation, but directly adsorbed on some imperfect objects, such as a virtual small table to a physical small sofa, rather than a physical large table.

Limitations. There are certain limitations inherent in our method. Due to the adoption of approximate classification during the scene abstraction process, disregarding the original precise semantics of real objects, haptic semantic mismatches may arise in some migration examples. For examples, a user might be mistakenly sitting on a coffee table; or a high stool being misinterpreted as a passive haptic provider for virtual bookshelves. Furthermore, it is important to note that our *v2rSA* framework relies on SA typical imprecise optimization and does not consider proposal steps along the z-axis that much during the iterative transfer process. Consequently, when ordinary virtual scenes migrate to real scenes with varying heights, the virtual objects such as chandeliers, range hoods, curtains, and murals may appear misplaced. In addition, certain inclined or irregular objects possessing large beveled surfaces may exhibit inadequate

passive haptic feedback, including vases, lamps, and potted plants, for instance.

6 Conclusion

We propose **SMigraPH**, a passive haptics-enabled method for migrating indoor virtual scenes to real scenes. Through experiments and user studies, our approach demonstrates superior capability in aligning the migrated virtual scene with the real scene surface, resulting in enhanced efficiency, accuracy, and subjective fidelity during haptic interactions. Notably, our method pioneers the consideration of the mapping strategy between the virtual and real scenes, offering insights and potential advancements for future MR scene migration applications.

There are certain limitations inherent in our method. Due to the adoption of approximate classification during the scene abstraction process, disregarding the original precise semantics of real objects, haptic semantic mismatches may arise in some migration examples. For examples, a user might be mistakenly sitting on a coffee table; or a high stool being misinterpreted as a passive haptic provider for virtual bookshelves. Furthermore, it is important to note that our *v2rSA* framework relies on SA typical imprecise optimization and does not consider proposal steps along the z-axis that much during the iterative transfer process. Consequently, when ordinary virtual scenes migrate to real scenes with varying heights, the virtual objects such as chandeliers, range hoods, curtains, and murals, may appear misplaced.

The limitations mentioned in Sect. 5.2 can be addressed by acquiring additional real scene datasets with semantic information and establishing a more intricate virtual-real mapping relationship, which could serve as a potential avenue for future research. Additionally, future works could address irregular room configurations, to ensure that virtual scenes can be better mapped to more complex real scenes, such as non-orthogonal spaces or curved spaces with non-planar walls. For the mapping of inclined objects, more sophisticated abstract modeling methods could be introduced to deal with entities with large beveled planes. Furthermore, the field of scene migration has also given rise to various open research challenges, including scene texture migration and cross-scene lighting layout, which hold promise as valuable areas for future breakthroughs.

Acknowledgements We sincerely thank the reviewers for their constructive suggestions and comments. This work is supported by the National Natural Science Foundation of China through Project 61932003, by Beijing Science and Technology Plan Project Z22110000 7722004, and by National Key R&D plan 2019YFC1521102.

Data availability The data in this study will be made available upon reasonable request to the corresponding author.

Declarations

Conflict of interest The authors declare no competing interests that are relevant to the content of this article, and there are no potential conflicts of interest, whether financial or non-financial. The funders did not have any involvement in the study or manuscript preparation.

References

1. Ali, W., Abdelkarim, S., Zidan, M. et al.: Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
2. Azmandian, M., Hancock, M., Benko, H., et al.: 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)
3. Bermejo, C., Hui, P.: A survey on haptic technologies for mobile augmented reality. ACM Comput. Surv. (CSUR) **54**(9), 1–35 (2021)
4. Butt, M.A., Maragos, P.: Optimum design of chamfer distance transforms. IEEE Trans. Image Process. **7**(10), 1477–1484 (1998)
5. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. Am. Stat. **49**(4), 327–335 (1995)
6. Dai, A., Chang, A.X., Savva, M., et al.: Scannet: richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5828–5839 (2017)
7. Dong, K., Gao, S., Xin, S., et al.: Probability driven approach for point cloud registration of indoor scene. Vis. Comput. 1–13 (2022)
8. Dong, Z.C., Wu, W., Xu, Z., et al.: Tailored reality: perception-aware scene restructuring for adaptive VR navigation. ACM Trans. Graph. (TOG) **40**(5), 1–15 (2021)
9. Du, K.L., Swamy, M.: Simulated annealing. In Search and Optimization by Metaheuristics, pp. 29–36. Springer (2016)
10. Fisher, M., Ritchie, D., Savva, M., et al.: Example-based synthesis of 3d object arrangements. ACM Trans. Graph. (TOG) **31**(6), 1–11 (2012)
11. Geyer, C.J.: Practical markov chain monte carlo. Stat Sci. 473–483 (1992)
12. Gwak, J., Choy, C., Savarese, S.: Generative sparse detection networks for 3d single-shot object detection. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, pp. 297–313. Springer (2020)
13. Insko, B.E.: Passive haptics significantly enhances virtual environments. The University of North Carolina at Chapel Hill (2001)
14. Jang, S., Kim, L.H., Tanner, K., et al.: Haptic edge display for mobile tactile interaction. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 3706–3716 (2016)
15. Jin, S., Lee, S.H.: Lighting layout optimization for 3d indoor scenes. In Computer Graphics Forum, Wiley Online Library, pp. 733–743 (2019)
16. Kermani, Z.S., Liao, Z., Tan, P., et al.: Learning 3d scene synthesis from annotated RGB-D images. Comput. Graph. Forum **35**(5), 197–206 (2016)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. (2016) arXiv preprint arXiv:1609.02907

18. Lari, Z., Habib, A., Kwak, E.: An adaptive approach for segmentation of 3d laser point cloud. In ISPRS Workshop Laser Scanning, pp. 29–31 (2011)
19. Lee, W., Foam, J.P.A.: A tangible augmented reality for product design. In ISMAR (The 4th IEEE and ACM International Symposium on Mixed and Augmented Reality), pp. 106–109 (2005)
20. Li, M., Patil, A.G., Xu, K., et al.: Grains: generative recursive autoencoders for indoor scenes. ACM Trans. Graph. (TOG) **38**(2), 1–16 (2019)
21. Liu, J., Li, Y., Goel, M.: A semantic-based approach to digital content placement for immersive environments. Vis. Comput. 1–15 (2022)
22. Liu, Z., Zhang, Z., Cao, Y., et al.: Group-free 3d object detection via transformers. (2021) arXiv preprint arXiv:2104.00678
23. Matthews, B.J., Thomas, B.H., Von Itzstein, S., et al.: Remapped physical-virtual interfaces with bimanual haptic retargeting. In 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), IEEE, pp. 19–27 (2019)
24. Merrell, P., Schkufza, E., Li, Z., et al.: Interactive furniture layout using interior design guidelines. ACM Trans. Graph. (TOG) **30**(4), 1–10 (2011)
25. Mescheder, L., Oechsle, M., Niemeyer, M., et al.: Occupancy networks: Learning 3d reconstruction in function space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4460–4470 (2019)
26. Pearson, K.: The problem of the random walk. Nature **72**(1865), 294–294 (1905)
27. Qi, C.R., Liu, W., Wu, C., et al.: Frustum pointnets for 3d object detection from RGB-D data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 918–927 (2018a)
28. Qi, S., Zhu, Y., Huang, S., et al.: Human-centric indoor scene synthesis using stochastic grammar. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5899–5908 (2018b)
29. Rinne, H.: The Weibull Distribution: A Handbook. Chapman and Hall/CRC, Boca Raton (2008)
30. Rusu, R.B., Cousins, S.: 3d is here: point cloud library (pcl). In 2011 IEEE International Conference on Robotics and Automation, IEEE, pp. 1–4 (2011)
31. Salazar, S.V., Pacchierotti, C., de Tinguy, X., et al.: Altering the stiffness, friction, and shape perception of tangible objects in virtual reality using wearable haptics. IEEE Trans. Haptics **13**(1), 167–174 (2020)
32. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point-cloud shape detection. In Computer Graphics Forum, Wiley Online Library, pp. 214–226 (2007)
33. Song, Y., Shen, W., Peng, K.: A novel partial point cloud registration method based on graph attention network. Vis. Comput. **39**(3), 1109–1120 (2023)
34. Spelmezan, D., González, R.M., Subramanian, S.: Skinhaptics: ultrasound focused in the hand creates tactile sensations. In 2016 IEEE Haptics Symposium (HAPTICS), IEEE, pp. 98–105 (2016)
35. Sun, Y., Miao, Y., Chen, J., et al.: Pgcnet: patch graph convolutional network for point cloud segmentation of indoor scenes. Vis. Comput. **36**, 2407–2418 (2020)
36. Talton, J.O., Lou, Y., Lesser, S., et al.: Metropolis procedural modeling. ACM Trans. Graph. (TOG) **30**(2), 1–14 (2011)
37. Tang, K., Chen, Y., Peng, W., et al.: Reppvconv: attentively fusing reparameterized voxel features for efficient 3d point cloud perception. Vis. Comput. 1–12 (2022)
38. Ungureanu, D., Bogo, F., Galliani, S., et al.: Hololens 2 research mode as a tool for computer vision research. (2020) arXiv preprint arXiv:2008.11239
39. Wang, K., Savva, M., Chang, A.X., et al.: Deep convolutional priors for indoor scene synthesis. ACM Trans. Graph. (TOG) **37**(4), 1–14 (2018)
40. Wang, K., Lin, Y.A., Weissmann, B., et al.: Planit: planning and instantiating indoor scenes with relation graph and spatial prior networks. ACM Trans. Graph. (TOG) **38**(4), 1–15 (2019)
41. Wang, L., Zhao, Z., Yang, X., et al.: A constrained path redirection for passive haptics. In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), IEEE, pp. 650–651 (2020)
42. Xu, K., Stewart, J., Fiume, E.: Constraint-based automatic placement for scene composition. In Graphics Interface, pp. 25–34 (2002)
43. Yeh, Y.T., Yang, L., Watson, M., et al.: Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. ACM Trans. Graph. (TOG) **31**(4), 1–11 (2012)
44. Yu, L.F., Yeung, S.K., Tang, C.K., et al.: Make it home: automatic optimization of furniture arrangement. ACM Trans. Graph. (TOG) Proc. ACM SIGGRAPH **30**(4), 86 (2011)
45. Zenner, A., Krüger, A.: Shifty: a weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. IEEE Trans. Visual Comput. Graph. **23**(4), 1285–1294 (2017)
46. Zhang, S.H., Zhang, S.K., Liang, Y., et al.: A survey of 3d indoor scene synthesis. J. Comput. Sci. Technol. **34**(3), 594–608 (2019)
47. Zhou, B,. Lapedriza, A., Xiao, J., et al.: Learning deep features for scene recognition using places database (2014)

Qixiang Ma is currently pursuing a PhD degree in the School of Computer Science and Engineering at Beihang University. His research focuses on computer graphics, mixed reality, and physical simulation.

Lili Wang received her PhD degree from Beihang University where she is now a professor with the School of Computer Science and Engineering, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, augmented reality, mixed reality, real-time rendering, and realistic rendering.

Wei Ke received the PhD degree from the School of Computer Science and Engineering, Beihang University. He is currently a professor with the Computer Applied Technology Program, Macao Polytechnic University. His current research projects involve the design and implementation of open platforms for applications of computer vision and pattern recognition, including programming tools, environments, and frameworks. His research interests include programming languages, image processing, computer vision, and tool support for component-based engineering and systems.

Sio-Kei Im received the degree in computer science and the master's degree in enterprise information system from the King's College, University of London, UK, in 1998 and 1999, respectively, and the PhD degree in electronic engineering from the Queen Mary University of London (QMUL), UK, in 2007. He was a lecturer with the Computing Program, Macao Polytechnic Institute (MPI), in 2001. In 2005, he became the Operations Manager of the MPI-QMUL information systems research center jointly operated by MPI and QMUL, where he carried out signal processing work. He was promoted to a professor with the MPI, in 2015. He was a visiting scholar with the School of Engineering, University of California, Los Angeles (UCLA) and an Honorary Professor of The Open University of Hong Kong.