Dynamic Optimal Space Partitioning for Redirected Walking in Multi-user Environment

SANG-BIN JEON, SOON-UK KWON, JUNE-YOUNG HWANG, YONG-HUN CHO, HAYEON KIM, JIN-HYUNG PARK, and IN-KWON LEE, Dept. of Computer Science, Yonsei University, Rep. of Korea

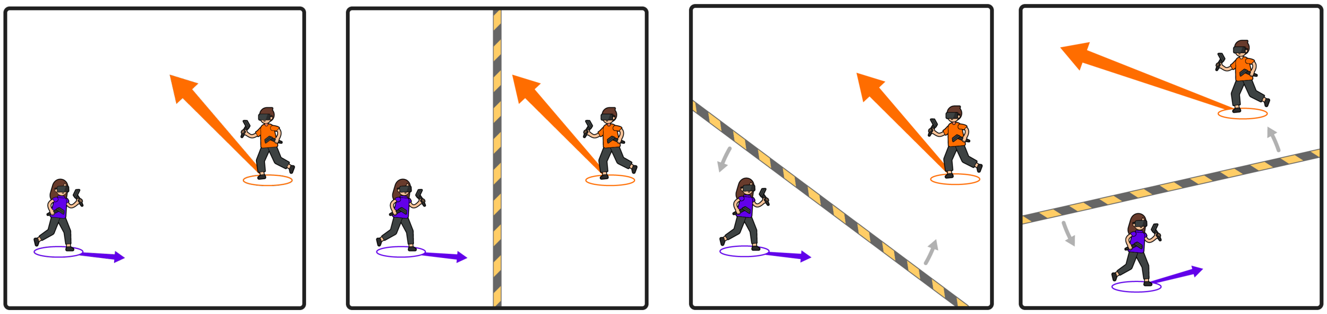

(a) (b) (c)

Fig. 1. Illustration of the environments where two or more users experience different (or the same) virtual environments in the shared physical space. (a) Space sharing strategy: Two or more users share the entire physical space. (b) Uniform Partitioning (conventional method): Divides a shared physical space into equal and static sub-spaces separated by shutters and allocates a sub-space to each user. (c) Optimal Space Partitioning (OSP): Based on the spatial information of the physical and virtual environment, a trained model predicts the movement of each user and optimally divides the shared physical space with dynamic shutters.

In multi-user Redirected Walking (RDW), the space subdivision method divides a shared physical space into sub-spaces and allocates a sub-space to each user. While this approach has the advantage of precluding any collisions between users, the conventional space subdivision method suffers from frequent boundary resets due to the reduction of available space per user. To address this challenge, in this study, we propose a space subdivision method called Optimal Space Partitioning (OSP) that dynamically divides the shared physical space in real-time. By exploiting spatial information of the physical and virtual environment, OSP predicts the movement of users and divides the shared physical space into optimal sub-spaces separated with shutters. Our OSP framework is trained using deep reinforcement learning to allocate optimal sub-space to each user and provide optimal steering. Our experiments demonstrate that OSP provides higher sense of immersion to users by minimizing the total number of reset counts, while preserving the advantage of the existing space subdivision strategy: ensuring better safety to users by completely eliminating the possibility of any collisions between users beforehand. Our project is available at https://github.com/AppleParfait/OSP-Archive.

Authors' address: Sang-Bin Jeon, ludens0508@yonsei.ac.kr; Soon-Uk Kwon, kwonars@yonsei.ac.kr; June-Young Hwang, tlavotl1@yonsei.ac.kr; Yong-Hun Cho, maxburst88@gmail.com; Hayeon Kim, qoocrab@gmail.com; Jinhyung Park, jh9604@yonsei.ac.kr; In-Kwon Lee, iklee@yonsei.ac.kr, Dept. of Computer Science, Yonsei University, Yonsei-Ro 50, Seodaemungu, Seoul, Rep. of Korea, 03722.

0730-0301/2022/7-ART90 $15.00
https://doi.org/10.1145/3528223.3530113

CCS Concepts: • **Computing methodologies** → **Virtual reality**.

Additional Key Words and Phrases: multi-user redirected walking, space partitioning, reinforcement learning

ACM Reference Format:
Sang-Bin Jeon, Soon-Uk Kwon, June-Young Hwang, Yong-Hun Cho, Hayeon Kim, Jinhyung Park, and In-Kwon Lee. 2022. Dynamic Optimal Space Partitioning for Redirected Walking in Multi-user Environment. *ACM Trans. Graph.* 41, 4, Article 90 (July 2022), 14 pages. https://doi.org/10.1145/3528223.3530113

1 INTRODUCTION

Users with Head-mounted Display (HMD) can experience a high level of presence while exploring infinite virtual worlds with real walking [Usoh et al. 1999]. However, it is practically impossible for users to explore infinitely large virtual spaces while they exist in finite physical space [Dong et al. 2021b; Li et al. 2021a; Serrano et al. 2020]. To overcome this challenge, Redirected Walking (RDW) is introduced as an alternative by transforming virtual environments or manipulating the ratio of the locomotion mapping between the physical and virtual space [S. Razzaque and Whitton 2001; Steinicke et al. 2010; Suma et al. 2011; Yang et al. 2019]. This guides users to explore infinite virtual space from a finite physical space without noticing the spatial discrepancies. Hence, RDW provides a higher sense of presence and immersion to users in virtual environments compared to other locomotion interfaces such as teleportation, joystick, or in-place walking [Langbehn et al. 2018a; Steinicke et al. 2009a,b].

Recently, RDW in multi-user environments has been a core domain in many RDW studies. A Redirection Technique (RET) in RDW can guide multiple users to their own physical path to prevent any collisions with the boundaries of the physical space, obstacles, or other users. Thus, multiple users can explore infinite virtual spaces concurrently while sharing the same physical space. However, the difficulty of enforcing an RDW method rises as more users share the same physical space [Nilsson et al. 2018]. This is mainly because the more users are densely located in the shared physical space, the more challenging it is to detect unexpected collisions or take preventive measures [Azmandian et al. 2017; Bachmann et al. 2013]. Moreover, user reset poses much more danger to users than boundary reset; user reset is a reset process that prevents physical collisions between users, whereas boundary reset is a reset process that prevents collisions against the boundaries of walkable area [Dao et al. 2021; Holm 2012]. The reason for this lies in the insecurity of user reset. When each user is asked to execute a user reset, one of the users can fail to properly execute a user reset for various reasons, such as over-immersion. Therefore, users who have correctly performed a user reset may still experience unexpected collisions due to their counterparts. This phenomenon is similar to car accidents that law-abiding drivers encounter when they face errant vehicles on the road. Therefore, completely eliminating the chances of collisions between users, such as placing guardrails on the road, is an effective approach unless mutual interaction between users is required.

For multi-user RDW, Azmandian et al. [2017] first proposed the concept of *space sharing strategy* (Figure 1(a)) and *space subdivision strategy*. In the *space sharing strategy*, all users share the entire physical space, while in the *space subdivision strategy*, the given physical space is divided into sub-spaces where each sub-space is dedicated to each user. The uniform partitioning (UP) method, which is proposed as one of the methods of *space subdivision strategy*, uniformly divides the shared physical space into equal areas of dedicated walkable sub-spaces for users (Figure 1(b)). Unlike the *space sharing strategy*, this approach completely precludes the possibility of any collisions between users by placing static shutters.

However, a weak point of uniform partitioning is that it provokes more boundary resets than when using the *space sharing strategy* since the area of physical sub-space given to each user is substantially reduced (Figure 1(b)). Moreover, uniform partitioning does not consider the asynchronous path patterns of users when dividing the given space. If users are engaged in different VR content or different scenarios within the same content, the path pattern of each user may be significantly different from each other, and this must be taken into account when allocating a sub-space to each user. Therefore, simply dividing the physical space into equal sub-spaces may lead to severe spatial underutilization. Due to this drawback, recent multi-user RDW studies have only focused on *space sharing strategy* [Bachmann et al. 2019; Dong et al. 2020, 2021a, 2019; Lee et al. 2020; Messinger et al. 2019], and no further discussion or progress was made for *space subdivision strategy.*

In response to this, we propose Optimal Space Partitioning (OSP) that dynamically divides the shared physical space with a model trained with deep reinforcement learning (Figure 1(c)). Based on the spatial information of the physical and virtual environment, OSP divides the shared physical space into optimal sub-spaces by placing dynamic shutters in real-time. Within each sub-space, a user is redirected by using well-known reactive algorithms such as S2C [Razzaque 2005], APF-SC [Messinger et al. 2019], and ARC [Williams et al. 2021a]. To compare the performance of OSP and other methods described in Figure 7, we conducted simulation tests and user studies. Under all experimental conditions, OSP outperformed other methods, including the conventional space subdivision method (uniform partitioning) and space sharing methods. Based on these results, we confirmed that OSP ensures better safety with higher user immersion than previous methods. The contributions of our work include:

1. In a multi-user RDW environment, we propose a reinforcement-learning-based approach to solve the problem of dynamically dividing the shared physical space into optimal sub-spaces.
2. By considering the path pattern of each user, OSP can efficiently divide the given space to achieve multiple objectives at the same time: minimize the total number of resets, ensure better safety for users by precluding any user resets beforehand, and hence, provide higher sense of immersion to users.

2 BACKGROUND

2.1 Redirection Technique

Many well-organized survey studies have described the evolution of RDW [Nilsson et al. 2018; Suma et al. 2012a]. RDW is a technology that allows users to explore virtual environments wider than their physical space. After RDW was first proposed by Razzaque et al. [2001], RDW has been a core domain in virtual reality over the past 20 years, which led to the development of various RETs.

RETs can be classified into subtle techniques and overt techniques [Suma et al. 2012a]. Subtle techniques redirect users by applying redirection gains – a ratio of the locomotion mapping between the physical space and the virtual space – to the extent that users cannot notice. On the other hand, overt techniques redirect users by using explicit notifications such as UI displays or sound effects to guide users. In particular, the 'Reset' technique [Williams et al. 2007] among overt techniques is a redirection method that instructs users who have reached the boundary of walkable physical space to rotate in place (in most cases). Meanwhile, various studies have also discovered methods for combining RDW with the change blindness phenomenon [Simons and Levin 1997] in which users cannot notice the deformation of the virtual space [Steinicke et al. 2010; Suma et al. 2011, 2012b; Vasylevska et al. 2013]. Recently, researchers have proposed novel methods that combine RDW with new concepts, such as view manipulation techniques [Langbehn et al. 2018b; Sun et al. 2018] that use eye blinking and eye movement (saccade). Also, smooth mapping [Dong et al. 2019] that distorts the paths in a virtual environment or visibility polygon [Williams et al. 2021b] has been suggested.

2.2 Detection Threshold

Since overt techniques such as reset have negative effects on user presence [Suma et al. 2012a], many RDW studies focus on minimizing the number of resets by using subtle techniques to their full extent. However, users may experience cybersickness or feel a lower sense of immersion if a subtle technique is used with a redirection

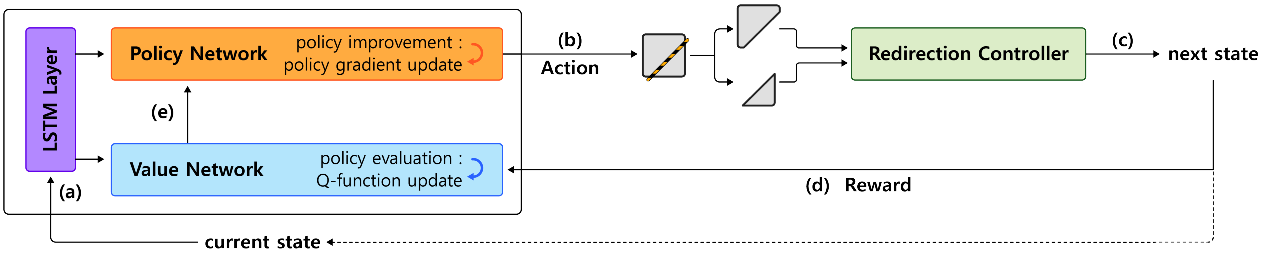


Fig. 2. An overview of the training procedure of the proposed OSP model. The procedure is as follows: (a) The LSTM layer consists of one recurrent layer and processes the cumulative spatial information at the current state. (b) The policy network determines the optimal action at the current state that can maximize the output of the expected cumulative reward function. By taking an action, shutters are placed within the shared physical space to divide the space into sub-spaces. (c) The simulator operates the designated redirection controller (i.e., S2C, APF-SC, or ARC) in each sub-space and updates the current state to its next state. (d) When the simulator calculates the output of the reward function for the next state, the value network updates the action-value function (Q-function) to maximize the cumulative reward. (e) The value network updates the parameters of the policy network to increase the probability of choosing the optimal action that can maximize the cumulative reward. At this point, a single step within an episode is completed, and the entire process will be repeated.

gain that exceeds the detection threshold; the detection threshold is a threshold of a user for not noticing whether or not the redirection gains are currently being applied in their environments. In light of this phenomenon, Steinicke et al. [2009a; 2009b] generalized subtle techniques to RETs that use translation, rotation, and curvature gain and analyzed the detection threshold of each gain. Following this work, many recent studies investigated the detection threshold for RDW in various situations [Bölling et al. 2019; Cho et al. 2021; Kim et al. 2021; Langbehn et al. 2017; Li et al. 2021b; Williams and Peck 2019; Zhang et al. 2018].

2.3 Redirection Controller

Utilizing a single RET is insufficient to enable users to fully immerse themselves in exploring infinite virtual environments. This led various studies to suggest redirection controllers combining overt and subtle techniques.

Redirection controllers can be classified into reactive controllers and predictive controllers. Reactive controllers such as S2C [Razzaque 2005], S2O [Razzaque 2005], APF-RDW [Bachmann et al. 2019], and ARC [Williams et al. 2021a] consist of generalized algorithms that guide users to a certain place or guide with a specific pattern. On the other hand, predictive controllers such as FORCE [Zmuda et al. 2013], MPCRed [Nescher et al. 2014], S2OT [Lee et al. 2019], and SRL [Strauss et al. 2020] are driven by algorithms that predict future paths of users and execute optimal RETs.

Following the work of Azmandian et al. [2017], many studies have investigated methodologies that handle environments where two or more users exist in the same physical space. Dong et al. [2019] improved the smooth mapping [2017] method which distorts the paths in the virtual environment. Then, smooth mapping was combined with RETs and demonstrated in multi-user environments. Bachmann et al. [2019] proposed APF-RDW, which expanded the concept of Artificial Potential Field (APF) [Krogh 1984] from robotics to multi-user RDW environments. Messinger et al. [2019] analyzed the APF-RDW framework on physical rooms with various shapes and sizes. Based on their findings, they proposed APF-RDW with Scaling (APF-SC). Lee et al. [2020] proposed MS2OT as a predictive controller that uses a model trained using reinforcement learning and Dueling Double Deep Q Network (D3QN). Dong et al. [2020] proposed a redirection controller that uses APF-RDW while considering the future location of the users by using a heuristic method. Then, Dong et al. [2021a] proposed another redirection controller that can redirect users not to be concentrated in certain spots in the shared physical space by employing the concept of density in multi-user RDW environments. While all of these studies on *space sharing strategy* have focused on minimizing the total number of resets (boundary resets and user resets), they did not attempt to develop a method for completely eliminating user resets or focused on the insecurity of the user reset process.

Outside of the RDW research community, Marwecki et al. [2018] proposed a partitioning method in a multi-user environment by using real walking. In this method, each user's application directly intervenes in the scenario to instruct users to move to their next region in the virtual space. This ultimately makes the users swap their sub-spaces with each other in the physical space. However, this introduces another constraint in the storyline of the VR content and poses a danger to users since it does not take physical collisions into account.

2.4 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning method that trains an agent in a certain environment to observe the state s_t at time t, and take the optimal action a_t selected from the action space to maximize the output of the expected cumulative reward function [Sutton et al. 1998]. The expected cumulative reward function $\mathbb{E}$ is as follows:

$$\mathbb{E}\left[\sum_{t=1}^{t=\infty} \gamma^{t-1} r_t\right] = \mathbb{E}\Big[r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots + \gamma^{\infty} r_{\infty}\Big], \tag{1}$$

where γ represents a discount factor, and r_t represents the reward at time t that maximizes the cumulative reward.

A reinforcement learning problem can be formulated as a Markov Decision Process (MDP) [Puterman 2014] and be solved by using either one of the two main approaches in a model-free environment:

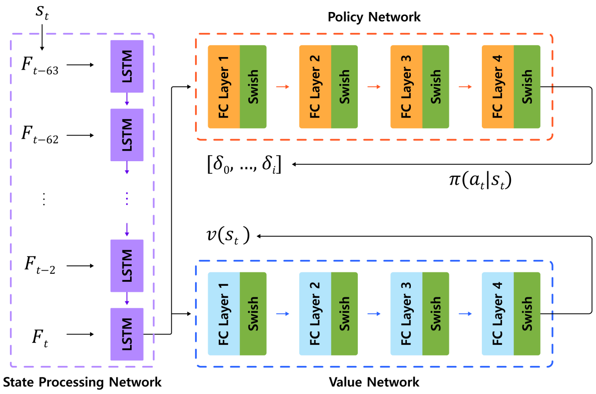


Fig. 3. The architecture of the proposed network for training the OSP model. We use an artificial neural network to optimize a policy π that can maximize the expected cumulative reward defined in Equation (1). The policy network and the value network have four fully connected layers (FC), respectively, which are both preceded by a shared LSTM layer. A Swish layer attached to each FC layer indicates an activation function.

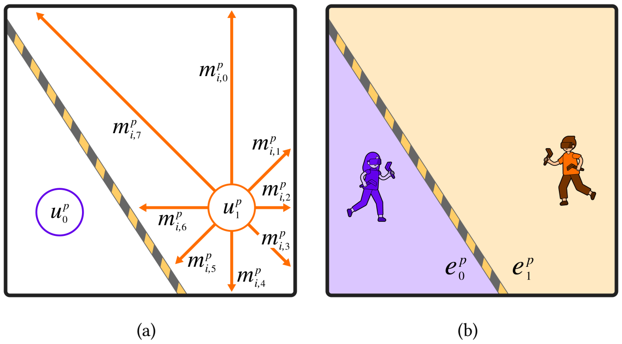


Fig. 4. Illustrations of m and e, which are components of the spatial frame information. (a) Vector $m^p_{i,j}$ consists of eight distance values, where each distance is the length between the i^{th} user's physical 2D position u^p_i and the nearest physical boundary, obstacle, or shutter in one of the eight cardinal directions. (b) Vector e represents the area of each physical sub-space.

value-based learning and *policy-based learning*. Value-based learning approximates the optimal policy π by obtaining the maximum *Q-value* from the value function [Watkins and Dayan 1992], whereas in policy-based learning, the policy π is directly modeled as $\pi(s, a, \theta)$ with a policy parameter θ and is optimized by θ with policy gradient [Williams 1992]. Meanwhile, training a model using both approaches also shows excellent performance. In this study, we use Actor-Critic learning [Konda and Tsitsiklis 2000] in which the Actor (policy network) determines the optimal action to be taken at the current state, and the Critic (value network) evaluates the value of the next state.

2.5 Deep Reinforcement Learning for RDW

The basic principle of predictive controllers is to predict future situations and redirect users to their optimal positions. Lee et al. [2019] substituted an RDW problem with MDP and first proposed a reinforcement-learning-based approach for developing predictive controllers. Following this work, various RDW studies have suggested reinforcement-learning-based approaches to minimize the number of resets or provide passive haptic feedback [Chang et al. 2021; Chen et al. 2021; Lee et al. 2020; Shibayama and Shirakawa 2020; Strauss et al. 2020]. Among these works, MS2OT [Lee et al. 2020], which expanded S2OT to multi-user environments, attempts to minimize the number of resets by designating optimal redirection target or pre-reset to each user.

2.6 Curriculum Learning

Curriculum learning is a technique that trains a model with easy-to-learn data first and then gradually trains it with difficult data [Bengio et al. 2009]. This facilitates the model to converge to global minima more quickly and reduces the likelihood of optimizing towards local minima. To utilize this technique in our study, we defined the level of difficulty for learning each training data by how the path patterns between users are different from each other. Then, we trained our OSP model using curriculum learning. Please refer to Section 3.5 for details.

3 TRAINING

Our goal is to train the OSP model to learn a policy that completely eliminates the need for any user reset and minimizes the total number of resets. To achieve this goal, the agent should pursue a policy that dynamically divides the shared physical space into optimal and walkable sub-spaces every moment, in accordance with the current status of the physical and virtual environment.

We formulated this problem as an MDP and used Soft Actor-Critic (SAC) [Haarnoja et al. 2018], an off-policy algorithm that shows better sampling efficiency compared to PPO [Schulman et al. 2017], which has been widely used in previous RL-based RDW studies [Chang et al. 2021; Chen et al. 2021; Shibayama and Shirakawa 2020; Strauss et al. 2020]. Moreover, we defined the state, action, reward function, and deep neural network in the Unity ML-Agents Toolkit (ver. 0.25.0) [Juliani et al. 2018] to train our OSP model (Figure 2).

3.1 State

Predicting a future state by solely depending on the observations of the current state can be considered as a Partially Observable Markov Decision Process (POMDP) [Kaelbling et al. 1998], which may not fully reflect the actual status of the current environment [Hausknecht and Stone 2015; Li et al. 2020]. Therefore, it is essential to accumulate information of the physical and virtual environment from its past to the present to predict future situations. Therefore, we injected a recurrency in our network by using LSTM [Hochreiter and Schmidhuber 1997] cells so that the agent can better understand the dynamics that occur in the environments (Figure 3). To develop our model capable of reasoning on asynchronous path patterns of each user, we design the input state with windows of spatial frame information F from time $t - 63$ to time t. Thus, the input state s_t

Algorithm 1: Optimal Space Partitioning (OSP)

```
Input:
    $n$: the number of users ($i = 0, ..., n-1$),
    $\zeta$: width of shutter,
    $\epsilon$: radius of user's body,
    redirection controller
begin
    $t \leftarrow 0$;
    $\mathbb{E}(s_0) \leftarrow 0$;        ▷ initialize cumulative reward function
    repeat
        observe state $s_t$ (see Equation (2));
        $u^p_{i,t} \leftarrow$ user $i^{th}$'s position in physical space at time $t$;
        $Q_m \leftarrow \min^{n-1}_{i,j=0,\ i \neq j} \|u^p_i - u^p_j\|$;
        $q_t \leftarrow \frac{Q_m - \zeta}{2} - \epsilon$;
        $\delta_{i,t} \leftarrow$ translation vector from action space,
            $\|\delta_{i,t}\| \leq q_t$;
        $\psi_{i,t} \leftarrow$ translate $u^p_{i,t}$ by $\delta_{i,t}$;
        $V \leftarrow$ Voronoi diagram of $\{\psi_{i,t}\}$;
        $VE(V) \leftarrow$ set of Voronoi edges in $V$;
        shutter $\leftarrow VE(V)$;
        $u^p_{i,t} \leftarrow$ update $u^p_{i,t}$ by the given redirection controller;
        compute $\mathbb{E}(s_{t+1})$;
        $t \leftarrow t + 1$;
    until time step $t$ reaches its maximum;
end
```

can be expressed as follows:

$$s_t = (F_{t-63},\ F_{t-62},\ ...,\ F_{t-1},\ F_t),$$
$$F_t = (f_{t-449},\ f_{t-448},\ ...,\ f_{t-1},\ f_t), \tag{2}$$

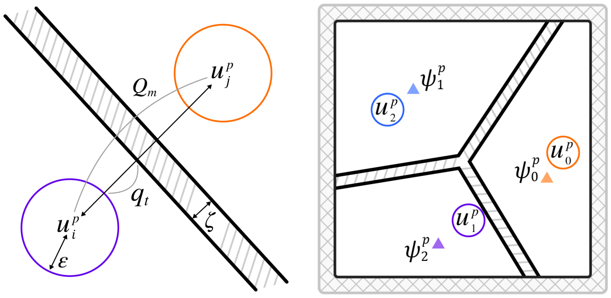
where the spatial frame information f_t consists of f^p_t and f^v_t. The f^p_t represents the spatial information acquired from the physical space where users actually exist, and f^v_t represents the spatial information acquired from the virtual space where virtual avatars exist. f_t is represented as follows:

$$f_t = \left[f^p_t,\ f^v_t\right] = \left[\left(u^p_{i,t},\ o^p_{i,t},\ m^p_{i,j,t},\ e^p_{i,t}\right),\ \left(u^v_{i,t},\ o^v_{i,t},\ m^v_{i,j,t}\right)\right], \quad i = 0, ..., n-1, \quad j = 0, ..., 7, \tag{3}$$

where n is the number of users and $u^p_{i,t}$, $o^p_{i,t}$, $u^v_{i,t}$, and $o^v_{i,t}$ are the i^{th} user's physical 2D position, physical 1D orientation (for up-axis), virtual 2D position, and virtual 1D orientation, respectively. To include the information of spatial proximity, we design $m^p_{i,j,t}$, which consists of the minimum distances from $u^p_{i,t}$ to the boundaries of the physical space, obstacles, or shutters in eight cardinal directions (Figure 4(a)). Similarly, $m^v_{i,j,t}$ denotes the minimum distances from $u^v_{i,t}$ to other users, obstacles, and boundaries of the virtual space. Moreover, the state also includes the area $e^p_{i,t}$ of the physical subspace allocated to the i^{th} user (Figure 4(b)).

3.2 Action

A Voronoi diagram divides a given plane into sub-regions (Voronoi regions) constructed by the nearest points of each object (Voronoi seed point) in the plane, where the boundaries of each region are called Voronoi edges [Aurenhammer 1991; Okabe et al. 2009; Voronoi 1908]. Therefore, our problem can be simply formulated by defining the position of each user as a Voronoi seed point and allocating the Voronoi regions constructed by those points to each user. Hence, the shutters can be considered as Voronoi edges. However, the OSP adopts a more sophisticated process when locating seed points. The OSP searches for seed points *near* each user, rather than directly using the position of each user as seed points, in order to construct Voronoi regions that can maximize the cumulative reward and enable optimal partitioning. The reasons for employing the concept of a Voronoi diagram for designing our OSP framework are as follows: first, we can intuitively divide the given space within efficient computation time, $O(n\log(n))$ [Fortune 1987]. Second, this algorithm always produces sub-regions with convex shapes. In general, this reduces the number of resets during RDW than when the same areas with concave shapes are given [Lee et al. 2020; Messinger et al. 2019].



(a) (b)

Fig. 5. Illustrations of the mechanism of performing an action at time t. (a) Calculate the distance range of stable location that each Voronoi seed point can be placed. (b) Construct a Voronoi diagram based on the seed points determined by taking the optimal action. Shutters are placed by overlaying Voronoi edges.

Algorithm 1 shows the procedure of training our OSP model to perform optimal partitioning by locating optimal seed points. In other words, the action of the OSP model at time t is to pinpoint Voronoi seed points $\psi_{i,t},\ (i = 0, ..., n-1)$ that can lead to optimal partitioning of the shared physical space for n users. Note that if the Voronoi seed points are selected arbitrarily, multiple users can be assigned to the same region. To prevent such failures, we enforced bound constraints when pinpointing $\psi_{i,t}$.

To implement this process, we define q_t, which represents the radius of a circular region where $\psi_{i,t}$ can be located, and $u^p_{i,t}$ is the center of the region (Figure 5(a)). Moreover, we define G_m as the minimum distance between users, ϵ as the radius of a user body ($\epsilon = 0.5m$), and ζ as the width of the shutters ($\zeta = 0.5m$). As a result, $\psi_{i,t}$ is placed $\delta_{i,t}$ ($\|\delta_{i,t}\| \leq q_t$) distance away from $u^p_{i,t}$. Note that $\delta_{i,t}$ is the output of the policy network (Figure 3), which is determined by the general behavior of regular reinforcement learning; in the early phase of training, the output mainly reflects exploration while more weight is given to exploitation as it proceeds to the later

Table 1. The hyperparameters used for training our OSP model.

Hyperparameter	Value	Description
activation function	Swish	$f(x) = x \cdot sigmoid(\beta x)$, where $\beta = 1.0$ [Ramachandran et al. 2017]
γ	0.99	A discount factor that balances the weight between the reward acquired in the current state and the reward that is expected to be acquired in the future.
τ	0.005	A hyperparameter that determines how aggressively the target network used for bootstrapping value estimation in SAC, is updated.
batch size	2048	The number of experiences in each iteration of gradient descent.
learning rate	3×10^{-4}	The initial learning rate for gradient descent.
layers	4	The number of fully connected layers in the policy network and the value network, respectively. Each layer has 128 hidden units.
max steps	1.62×10^{8}	The total number of steps that must be taken in the environment before terminating the training process.
sequence_length	64	The length of input sequence that enters the LSTM layer.
hidden_size	128	The number of features in the hidden state.

phase of training. Then, OSP constructs a Voronoi diagram by using $\psi_{i,t}(i = 0, ..., n-1)$ as seed points and overlays shutters on Voronoi edges determined by the diagram (Figure 5(b)). Finally, the RDW module updates each $u^{p}_{i,t}$ by using a designated redirection controller in each sub-space. By this, the state changes from s_t to s_{t+1}, and the parameters of the OSP model are updated by the cumulative reward computed from Equation (1).

3.3 Reward Function

For every step in the learning process, the agent receives a positive or negative reward in return for their action. Through these steps, OSP collects a reward r_t that maximizes the cumulative reward at time t, hence learning to perform optimal partitioning. This process starts with the agent observing the environment as s_t and selecting its action a_t. This results in a transition from the current state s_t to its next state s_{t+1}. The reward function $R(s_t,\ a_t,\ s_{t+1})$ used within this process to calculate the reward for each step is defined as below:

$$R(s_t,\ a_t,\ s_{t+1}) = \begin{cases} R_q, & \text{when reset occurs} \\ w_b R_b + R_0, & \text{otherwise,} \end{cases} \tag{4}$$

where R consists of three sub-functions R_q, R_b, and R_0. The function R_q is activated when one or more users in the environment are asked to execute a reset process. Otherwise, the agent receives rewards that is a weighted combination of R_b and R_0. A positive reward R_b is the summation of the total distance traveled by all users in the physical space during one step. This reward encourages users to travel a longer physical path. Likewise, we train the agent to favor a policy that generates a partitioning that results in the most efficient redirection for each user while restraining any resets. In general, the cumulative reward often diverges since it is common for the agent to receive a lot of penalties during the early phase of training. Thus, we design $R_0 = 10$ as a constant reward to easily observe the learning progress by steering the cumulative reward converge to a positive value.

The function R_q is computed by a weighted summation of three components as described below:

$$R_q(s_t,\ a_t,\ s_{t+1}) = w_k R_k + w_h R_h + w_d R_d, \tag{5}$$

where each coefficient is a constant that was empirically determined from trial and error in producing the best results ($w_k = 2\times 10^2$, $w_h = 1.4 \times 10^2$, $w_d = 1 \times 10^1$, $w_b = 0.9 \times 10^0$).

First of all, R_k acts as a penalty factor when one or more users execute “Boundary Reset", which occurs when users reach the boundaries of the physical space or an internal obstacle.

$$R_k(s_t,\ a_t,\ s_{t+1}) = \begin{cases} -1, & \text{when “Boundary Reset" occurs} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Then, R_h is a penalty factor when one or more users execute “Shutter Reset", which occurs when the boundaries of shutters touch a user.

$$R_h(s_t,\ a_t,\ s_{t+1}) = \begin{cases} -1, & \text{when “Shutter Reset" occurs} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Note that we assign a smaller value to the coefficient w_h than w_k, which has the effect of steering users toward the shutters rather than the boundaries of the physical space or internal obstacles. Although the boundary reset and the shutter reset both hinder users' immersion, it is better to avoid boundary reset rather than shutter reset. While the shutters in our framework are genuinely imaginary figures, the boundaries of the physical space or obstacles pose a real threat to users if the users fail to carry out boundary reset. Therefore, we designed the algorithm to prefer shutter reset rather than boundary reset. Lastly, it is crucial to train our algorithm not to be biased towards certain users to experience more number of resets than others. To implement this idea, we define a penalty factor R_d that is computed by the sum of the deviation of all users that experience more resets than the average value of the group. The equation of R_d is as follows:

$$\begin{gathered} R_d(s_t,\ a_t,\ s_{t+1}) = -\sum_{i=1}^{n} d_i, \qquad \text{if } d_i > 0, \\ \text{where } \ d_i = T_i \ - \ \frac{1}{n}\sum_{j=1}^{n} T_j, \end{gathered} \tag{8}$$

where T_i is the total number of resets that the i^{th} user experience, and d_i is the deviation between T_i and the average number of total resets of all users present in the shared physical space. Note that

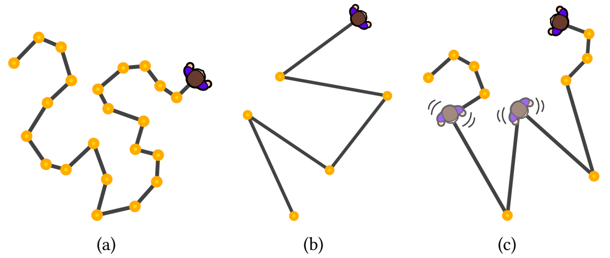


Fig. 6. The simulator can generate three types of virtual paths: (a) small exploration, (b) large exploration, and (c) mixed exploration (including idling). The simulator randomly generates the position of the next target on each path.

we only summate the positive values of d_i when computing the penalty R_d to discourage cases where some users encounter more frequent resets than others. By this, we create an unbiased environment where all users can equally immerse themselves in the virtual environment as much as possible.

3.4 Training Environment

We train our model with Intel i7-11700 CPU, NVIDIA Geforce RTX 3090 GPU, and 64GB of RAM. The major hyperparameters used in our training are shown in Table 1. The values for other parameters were set to their default values used in Unity ML-Agents Toolkit.

The environment where our agent is defined operates in 90 fps, where one frame corresponds to one time step. Also, we defined one episode to consist of 16,200 steps. At the start of each episode, the simulator randomly initializes the position and orientation of each simulated user in the physical space and the virtual space. In the virtual space, each simulated user is asked to move straight towards a target that is generated successively. The velocity and angular velocity of each simulated user are $1m/s$ and $90°/s$, respectively.

The simulator generates a series of targets according to the current path pattern of the user (Figure 6). We defined *small exploration* and *large exploration* by using path segments defined in a previous study [Azmandian et al. 2016]. In addition, we define the *mixed exploration* as a path pattern where the path pattern for each segment is randomly changed among small exploration, large exploration, and idling after the simulated user acquires a certain number of targets along its way.

For all simulated users, we apply the same detection threshold that was calculated in previous works that analyzed redirection gains [Hodgson and Bachmann 2013; Hodgson et al. 2014; Steinicke et al. 2009a, 2008a, 2009b, 2008b]. The threshold of curvature, translation, and rotation gain applied in our study are $7.5m$, $[0.86, 1.26]$, and $[0.67, 1.24]$, respectively. When applying reactive controllers in each sub-space, we followed the specifications described in previous works [Messinger et al. 2019; Williams et al. 2021a]. Exceptionally, for OSP-S2C, we use the Reset-to-center (R2C) [Thomas and Rosenberg 2019] algorithm as its reset technique instead of using 2:1-Turn [Williams et al. 2007] since 2:1-Turn may cause a series of resets at the corner of the room in the worst case. Then, we trained each model under different environmental conditions (the number of users, room size, empty room or non-empty room, and the reactive controller used within each sub-space) to verify if our agent can take the optimal action in each environment.

Table 2. A summary of the training curriculum for OSP model. When transitioning from the 3rd to the 4th quarter, the simulator expands its range of randomization at which it starts changing the path pattern: from $3 \leq x \leq 5$ to $2 \leq x \leq 6$, where x denotes the number of targets the user collected from the beginning when the path pattern starts changing. By this, we make the simulator to change the user's path pattern more often.

Quarter	Path pattern setting
1st quarter	large exploration
2nd quarter	mixed exploration (small and large)
3rd quarter	mixed exploration (small, large, and idling)
4th quarter	mixed exploration (small, large, and idling)

3.5 Curriculum Step

In this study, we assumed that the level of difficulty for learning each training data is determined by how the path patterns between users are different from each other. The path pattern of each user may be significantly different depending on their status within the VR content. Unlike uniform partitioning (Figure 1(b)), the objective of OSP is to consider the path pattern of all users in the shared physical space and predict the future state to divide the space accordingly (Figure 1(c)). However, we have empirically identified that the cumulative reward diverges if we set the path pattern of all simulated users to mixed exploration in the entire training curriculum. In contrast, the reward converged to optima when the path patterns of all simulated users were set to either small or large exploration. To efficiently train our model on mixed exploration, we divided the total training episodes (10,000 episodes) into four quarters and made the simulator generate the path patterns as shown in Table 2.

4 EXPERIMENTAL EVALUATION

4.1 Simulation Test Design

For experimental evaluations of OSP, we first conducted simulation tests. We analyzed the performance of each RDW method by setting the independent variables of our experiment as the number of users, room size, and the type of RDW method that is either a *space subdivision strategy* or the *space sharing strategy*. We split our experiment into two experiments; in **E1**, we fixed the path pattern to large exploration, and in **E2**, we fixed the path pattern to mixed exploration.

The simulator creates four physical empty square rooms: *tiny* (6×6 m^2), *small* (10×10 m^2), *medium* (20×20 m^2), and *large* (30×30 m^2). Likewise, the virtual rooms are empty square rooms with a dimension of 100×100 m^2 in all experimental conditions. While we involved 2-10 simulated users in each case, we placed up to two users in the tiny room, up to three users in the small room, and up to ten users in the medium and large room, to respect the well-known capacity of each room size, as suggested in a previous study [Messinger et al. 2019]. Then, we selected eight different RDW methods as shown in Figure 7 and compared their performance. For uniform partitioning, we used Lioyd relaxation [Du et al. 2006, 1999]

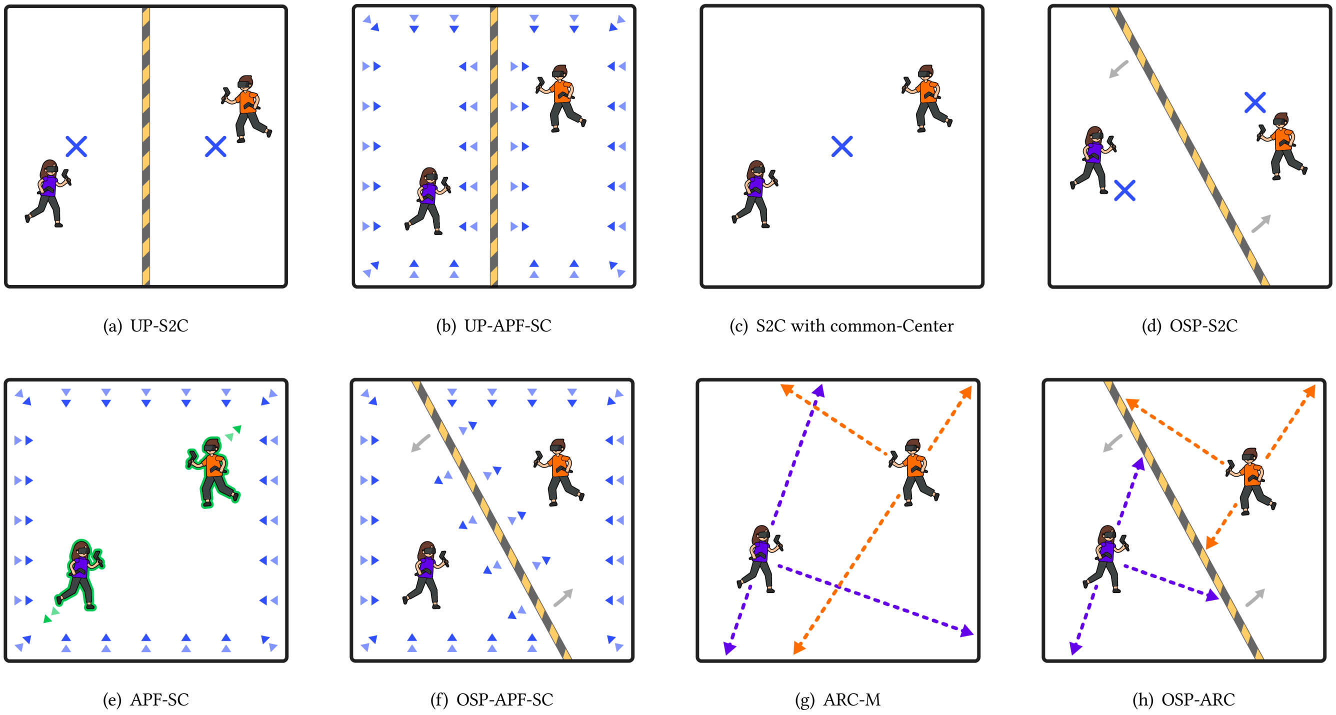

(a) UP-S2C
(b) UP-APF-SC
(c) S2C with common-Center
(d) OSP-S2C
(e) APF-SC
(f) OSP-APF-SC
(g) ARC-M
(h) OSP-ARC

Fig. 7. Illustrations of RDW methods referred to in this work. The methods that involve shutters, such as UP-S2C (UP with S2C) and UP-APF-SC (UP with APF-SC), belong to the *space subdivision strategy* (a,b,d,f,h), and other methods belong to the *space sharing strategy* (c,e,g). For RDW methods where S2C operates within each sub-space (a,c,d), the steering target of each sub-space is the centroid of the sub-space. For RDW methods that APF-SC operates within each sub-space (b,f), the potential fields are not only calculated for each boundary but also for the shutters. Moreover, ARC is a single-user RDW controller designed for passive haptic while it does not consider a multi-user environment or dynamic obstacles as its context. Still, ARC can be used within each sub-space without adjusting its original algorithm. Therefore, we use ARC-M (ARC with Multi-user) as a method that assumes other users as dynamic obstacles in the shared physical space, and compare its performance with OSP-ARC.

to divide the given physical space into convex-shaped sub-spaces with equal areas as much as possible, in accordance with the number of users sharing the space. Since OSP learned all of the path patterns described in Figure 6 with curriculum learning, we used the same, trained OSP model when testing its performance in **E1** and **E2**. Note that the definition of mixed exploration used in these experiments is the same as defined in the last step of our curriculum (Table 2).

Furthermore, we trained the models in a non-empty physical and virtual environment (Figure 1 in the supplementary material) and conducted another experiment, **E3**. In **E3**, we analyzed the performance of each RDW method on mixed exploration by comparing the total number of resets T and the mean distance between resets in the virtual environment ($MDbR$). Due to the limitation of using R2C [Thomas and Rosenberg 2019] in virtual space with obstacles, we used 2:1-Turn for S2C and OSP-S2C, respectively, as suggested in a previous study [Williams et al. 2021a] that analyzed the mechanism of S2C in environments with obstacles. In experiments **E2** and **E3** which covered mixed exploration, we added $MDbR$ to one of their primary metrics since the path pattern included the idling pattern.

For each experimental condition, we ran 100 simulations to collect and prepare our dataset. It takes 16,200 time steps for one trial to end. On this dataset, we conducted a parametric method – N-way analysis of variance test (ANOVA). In order to meet the precondition of using an ANOVA test, we confirmed that all of the datasets collected for each experimental condition pass the Shapiro-Wilk test and the Brown-Forsythe test ($p > .05$). Under each condition, we analyzed the effect of independent variables on dependent variables such as T and $MDbR$, respectively. Then, we ranked the RDW methods in the order of their performance, based on the results from multiple post-hoc comparison analysis using Tukey's HSD test with p-value from Bonferroni correction (p_c). Then, we used multiple linear regression analysis to measure the influence of each independent variable on the performance of each RDW method.

4.2 Result of E1: performance on large exploration

4.2.1 Total resets. We performed N-way ANOVA tests on the data collected from **E1** by setting T as the dependent variable. Among these tests, we visualized the results of a 9 (the number of users) × 8 (RDW method) × 2 (room size) three-way ANOVA test in Figure 8. Please refer to our supplementary material for details.

4.2.2 Regression. We conducted multiple linear regression analyses to investigate the relationship between the RDW method and T in

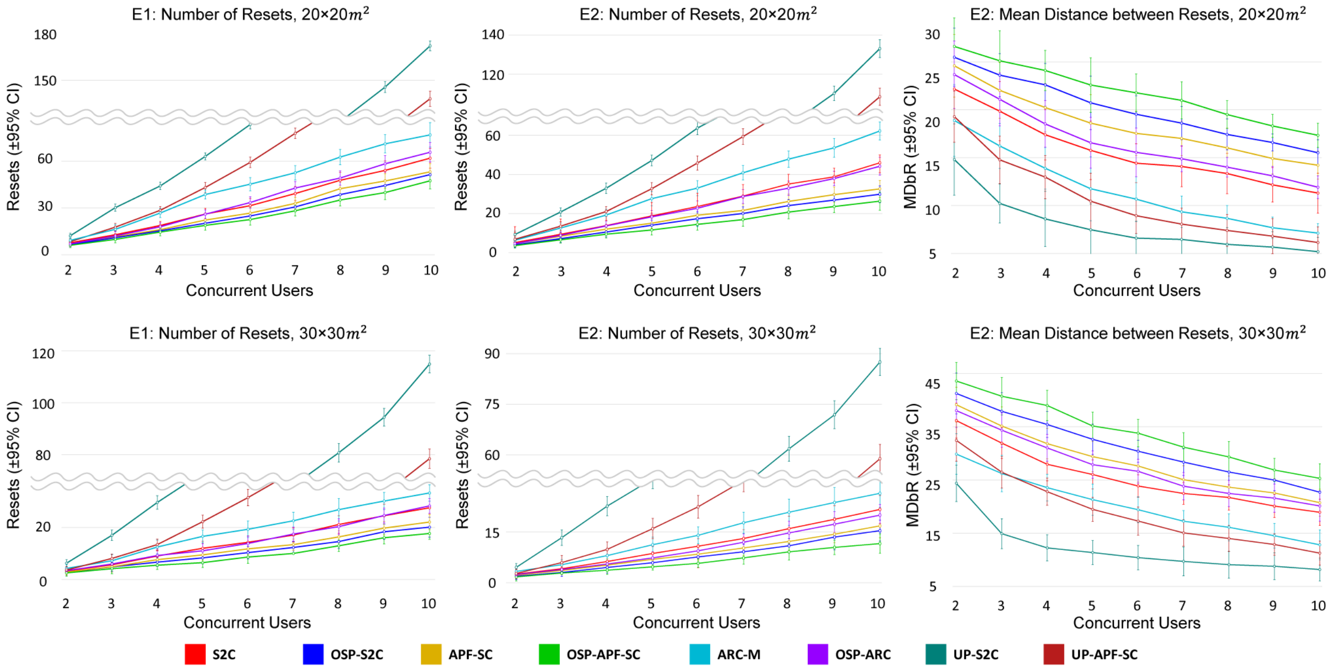


Fig. 8. Visualization of the results of **E1** and **E2** in simulation tests. In the graphs, OSP-APF-SC (light green) showed the best performance in terms of the total number of resets and the mean distance between resets in the virtual environment.

the data collected from the medium and large room. Please refer to our supplementary material for details.

4.3 Result of E2: performance on mixed exploration

4.3.1 Total resets and MDbR. Similarly to **E1**, we performed N-way ANOVA tests with the data collected from **E2**. Among these tests, we visualized the results of a three-way ANOVA test in Figure 8. Please refer to our supplementary material for details.

4.3.2 Regression. Similarly to **E1**, we performed multiple linear regression analyses with the data collected from **E2**. Please refer to our supplementary material for details.

4.4 Result of E3: environment with obstacles

4.4.1 Total reset and MDbR. We performed a one-way ANOVA test on the data collected from **E3**. Please refer to our supplementary material for details.

4.5 User Study Design

Based on the results from our simulation tests, we extended our experiment to user study to investigate the performance of OSP in real multi-user environments. In our user study, we used HTC VIVE Cosmos Elite HMDs to render a pair of identical binocular images of the virtual scenes. While HMDs were remotely connected to our desktop, we used four base stations that illuminate the shared physical space for inside-out tracking. The performance of all RDW methods, including OSP, were measured under this environment, maintaining an average of 90 fps. We used the RDW Toolkit Library [Azmandian et al. 2016] to construct our multi-user RDW environment and prepared a 6 × 6 m^2 physical space that could accommodate two users. Before conducting our main experiment, we led a pilot test to design our main experiment. We measured the performance of eight algorithms on eight participants (male:female = 6:2, $M = 25.29$, $SD = 2.81$) with two path patterns: large and mixed exploration.

Based on the results from our pilot test, we selected three RDW methods as the scope of our main experiment to prevent users from suffering side effects such as cybersickness after participating in excessive experiments. We selected OSP-APF-SC and its counterpart UP-APF-SC, since OSP-APF-SC showed the best performance at the simulation test. Moreover, we chose APF-SC to compare the performance of the space sharing method with OSP.

Same with the simulation test, we divided our user study into two experiments – **E4** and **E5** – with respect to the path patterns of users. We adopted a *between-subjects design* for our user study, i.e., we gathered different participants for each experiment **E4** and **E5**, while each experiment **E4** and **E5** followed a *within-subjects design*. In addition, we injected counterbalancing with a three by three (3 × 3) Latin square to remove any carryover effects. Then, we used G*power 3.1 program [Faul et al. 2007] to identify the minimum sample size needed to acquire statistically sound results from each experiment. According to the significance level (= 0.05), medium effect size (= 0.25), statistical power (= 0.80) from Cohen's guidelines, and the three RDW methods used as an independent

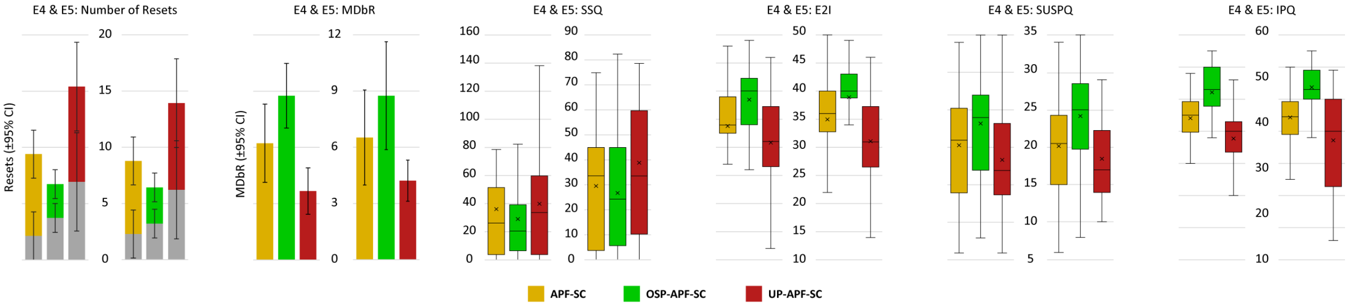


Fig. 9. Visualization of the results of **E4** and **E5** in our user study. In the graph that plots the number of resets (left), the gray bar represents the number of user resets or shutter resets.

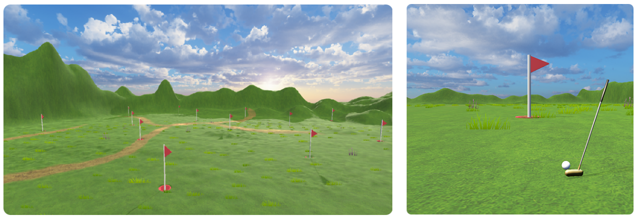


Fig. 10. Illustrations of the virtual environment for our user study. Users play VR golf putting games while walking around their own infinite virtual golf field.

variable, the minimum sample size required for **E4** and **E5** was both 28 participants.

Additionally, we used a set of verified questionnaires for a comprehensive evaluation of our user study. Since a rotation gain exceeding the detection threshold can be applied when using a reset technique, users may experience more cybersickness or feel a lower sense of presence and immersion if they encounter more resets [Langbehn et al. 2018a; Steinicke et al. 2009a,b]. Therefore, we used four questionnaires (7-point Likert scale) in order to measure these differences between different RDW methods: SSQ [Kennedy et al. 1993] (for motion sickness), E2I [Lin et al. 2002] (for immersion), SUSPQ [Usoh et al. 2000] and IPQ [Schubert et al. 2001] (for presence).

We recruited experiment participants to play a VR golf putting game and informed each participant that 10 US dollars ($) are given as a reward. The participants were asked to fill out a pre-SSQ before playing the game, and each participant was unaware that they would experience RDW during their engagement. Moreover, we gave instructions to each user (participant) with a video guide. We informed that a golf ball and the next hole cup are created in the virtual golf field (Figure 10) within three seconds after the user attempts putting at the current position, regardless of their success or failure, until the end of the game. The position where the golf ball reappears for the next target was determined by the current path pattern of the user. Note that a golf ball was created right in front of the virtual avatar's eyes when the current path pattern was idling in **E5**. A hole cup was placed 5m away from the ball.

To enhance user experience, we prompted haptic feedback (vibration) when the user make putts and included sound effects such as birds chirping and ball-putting sounds in our environment. Moreover, we applied some correction effects when users make putts to help them proceed to their next target without any difficulties. We repeated this process under three environmental conditions. Each iteration took 15 minutes: 3 minutes for preparation (device and server), 2 minutes for the experiment (users playing the VR golf putting game), and 10 minutes for surveying and break time.

4.6 Result of E4: performance on large exploration

A total of 30 people (male:female = 4:1, $M = 24.96, SD = 2.9$) participated in **E4**. All participants had a normal range of eyesight or wore eyeglasses for vision correction. The average number of prior experiences in VR contents among the participants was 1.95 trials ($SD = 1.08$) and each participant made an average of 7.13 golf putts ($SD = 0.85$) during the game. The average pre-SSQ score was 10.72 ($SD = 13.53$).

4.6.1 Total resets. The dataset passed the Shapiro-Wilk test ($p > 0.05$) but failed to pass the Brown-Forsythe test. Due to this reason, we conducted the Friedman test, which is a non-parametric alternative to a parametric statistical test such as one-way ANOVA with repeated measures (RM-ANOVA). We confirmed a significant difference in T for each RDW method ($\chi^2 = 131.38, p < .001$). The result of Wilcoxon signed-rank test for post-hoc analysis showed the order of performance of each RDW method as the following: OSP-APF-SC < APF-SC < UP-APF-SC ($p < p_c = .0166$).

4.6.2 MDbR. We also analyzed *MDbR* to consider the speed difference of each user in **E4**. We conducted an RM-ANOVA test to evaluate the effects of the three RDW methods on the dependent variable *MDbR*. We confirmed that the RDW method ($F_{2,42} = 112.984, p < .001, \eta_p^2 = .889$) has a large effect on *MDbR*. The result of Tukey's HSD test for post-hoc analysis showed that the order of performance is as follows: UP-APF-SC < APF-SC < OSP-APF-SC ($p < p_c = .0166$).

4.6.3 Questionnaires. We conducted the Friedman test and confirmed a significant difference in the measures used in our questionnaire, except SSQ ($p = .072$). The results are as follows: E2I ($\chi^2 = 27.235, p < .001$), SUSPQ ($\chi^2 = 10.855, p < .001$), IPQ ($\chi^2 = 29.448, p < .001$). The result of post-hoc analysis (Wilcoxon signed-rank test) showed that the order of performance are as follows: UP-APF-SC < APF-SC < OSP-APF-SC ($p < p_c = .0166$) (Figure 9).

4.7 Result of E5: the performance on mixed exploration

A total of 28 people (male:female = 5:2, $M = 25.1, SD = 3.17$) participated in **E5**. Same with **E4**, all participants had a normal range of eyesight or wore eyeglasses for vision correction. The average number of prior experiences in VR contents among the participants was 1.64 trials ($SD = 0.92$) and each participant made an average of 10.46 golf putts ($SD = 1.67$) during the game. The average pre-SSQ score was 11.96 ($SD = 13.59$). Since the duration of our game was 2 minutes in our user study (whereas it was 3 minutes in our simulation test), we adjusted the moment when the user starts changing its path pattern in mixed exploration to be when the user acquires 2-4 targets from the beginning.

4.7.1 Total resets. We confirmed the effect size of RDW method ($\chi^2 = 62.02, p < .001$) on T. The result of Wilcoxon signed-rank test for post-hoc analysis showed that the order of performance of each RDW method are as follows: OSP-APF-SC < APF-SC < UP-APF-SC ($p < p_c = .0166$).

4.7.2 MDbR. We confirmed that RDW method ($F_{2,39} = 58.93, p < .001, \eta_p^2 = .819$) has a large effect on *MDbR*. The result of Tukey's HSD test for post-hoc analysis showed that the order of performance are as follows: UP-APF-SC < APF-SC < OSP-APF-SC ($p < p_c = .0166$).

4.7.3 Questionnaires. Similarly to **E4**, we confirmed a significant difference in our questionnaire except SSQ ($p = .24$) as follows: E2I ($\chi^2 = 11.74, p < .001$), SUSPQ ($\chi^2 = 68.762, p < .001$), IPQ ($\chi^2 = 30.859, p < .001$). The result of post-hoc analysis (Wilcoxon signed-rank test) showed that the order of performance are as follows: UP-APF-SC < APF-SC < OSP-APF-SC ($p < p_c = .0166$) (Figure 9).

5 DISCUSSION

5.1 Results Analysis

We demonstrated the effectiveness of OSP by showing its outperformance compared to the conventional space subdivision method – uniform partitioning – under all experimental conditions. Furthermore, under all environmental conditions, each redirection controller showed better performance when it was used with OSP than when it was used with space sharing methods. In other words, OSP showed excellent compatibility with well-known redirection controllers referred to in this work.

Meanwhile, we observed a minor change in the order of performance of the RDW methods when the path pattern of all users was fixed to large exploration or mixed exploration. As shown in Figure 8 and Figure 9, the minor change originates from the increase in the performance gap between the OSP-based methods and its original redirection controller. This implies that the OSP model can redirect users within each sub-space, based on its analysis of the mechanisms of the designated redirection controller, to reduce spatial underutilization resulting from mixed exploration.

OSP showed higher scores than APF-SC and UP-APF-SC in terms of E2I (sense of immersion), SUSPQ and IPQ (sense of presence). These results are expected to be originated from reducing the total number of resets, allowing the users to immerse themselves in the VR content better and feel a higher sense of presence [Steinicke et al. 2009a,b; Suma et al. 2012a]. Although it is likely that the low average SSQ (motion sickness) score of OSP derives from fewer number of resets compared to other methods, it did not show any significant difference.

5.2 Ablation Study

We defined the reward function of OSP as Equation (4) through trial and error of observing to which value the cumulative reward converged. To examine the individual effectiveness of each component of the reward function, we led an ablation study by using simulation data collected under the following condition for 100 iterations: three simulated users with large exploration path pattern, using OSP-APF-SC as the RDW method, medium room for the physical environment, and 100 × 100 m^2 sized room for the virtual environment. We used the total number of resets as the performance metric and compared its value when each component was eliminated from the reward function. We conducted a one-way ANOVA test and post-hoc analysis using Tukey's HSD test. The results are as follows: Eliminate_*none* ($M = 9.85, SD = 2.23$) ≈ Eliminate_R_d ($M = 9.91, SD = 2.22$) < Eliminate_R_k ($M = 10.98, SD = 1.92$) < Eliminate_R_h ($M = 12.12, SD = 2.31$) ≈ Eliminate_R_b ($M = 12.8, SD = 2.12$) ($p < p_c = .01$). There was no significant difference between Eliminate_*none* and Eliminate_R_d ($p = .999$). Also, there was no significant difference between Eliminate_R_h and Eliminate_R_b ($p = .203$).

In addition, we compared Eliminate_*none* ($M = 1.33, SD = 1.59$) with Eliminate_R_d ($M = 3.7, SD = 3.13$) to evaluate the sample variance of the number of resets among users in each trial with the same data above. As a result of the Mann-Whitney U-test, Eliminate_R_d showed higher variance ($p < p_c = .025$). This implies that if R_d is eliminated from the reward function, the model is more likely to take biased actions towards a single or few users to simply maximize the cumulative reward. The results demonstrate that our final version of the reward function, which is equal to Eliminate_*none*, shows the best performance.

Moreover, we also analyzed the main driving force for the excellent performance of OSP: whether it is improved by reinforcement learning or simply enhanced by using a Voronoi diagram. By conducting a one-way ANOVA test and post-hoc analysis (Tukey's HSD test) under the same condition used in the ablation study above, we showed that the model trained with reinforcement learning outperformed the model that was not trained with such method: OSP-APF-SC ($M = 9.85, SD = 2.23$) < Voronoi-APF-SC ($M = 10.72, SD = 2.13$) ≈ APF-SC ($M = 11.51, SD = 2.08$) ($p < p_c = .0166$). Meanwhile, no significant difference between APF-SC and Voronoi-APF-SC was found ($p = .026$).

5.3 Limitation

First, although the OSP model operates in real-time within VR systems, it needs to be pre-trained in the given environmental condition (number of users, room size, and the given redirection controller used within each sub-space), which requires a pre-computation time. Second, note that even in a barricaded environment, we have only used a square shape for the physical room and a vast square area for the virtual room in our experiments. To extend the capacity of OSP as a dynamic space subdivision method, further studies on processing spaces with more irregular shapes are needed. Third, our user study is open for further experiments as it was only conducted in a $6 \times 6\ m^2$ physical space with two users due to limited space and lack of equipment. However, note that we have confirmed that the results from our user study still align with the tendency of the result obtained from the linear regression analysis of our simulation results. Moreover, as many previous RDW studies did, **E4** and **E5** both adopted a *within-subjects design* which may possibly trigger any carryover effects. To reduce such carryover effects and prevent cybersickness from participants, it was difficult to use more than two minutes per each RDW method in our experiment. Therefore, we believe that using a *between-subjects design* for our experiments with larger group of participants may produce more promising results. Lastly, since virtual shutters in OSP (also in UP) don't actually exist in the physical space, OSP may fail if the user completely ignores the given reset instructions and walks arbitrarily (not just a step or two). However, this is an obvious problem in any reset-based RDW methods that cannot be resolved other than educating the users properly before they start their exploration, just like driver education is crucial to novice drivers.

6 CONCLUSIONS

The conventional space subdivision method causes frequent resets to users due to smaller areas and suffers from severe space underutilization when the path pattern of each user differs from each other. In response to this, we proposed a reinforcement learning-based approach that can dynamically divide a shared physical space into optimal sub-spaces in a multi-user RDW environment. The OSP shows its strength in (1) enhancing space utilization by dividing the given space with dynamic shutters, based on its observation of the path pattern of each user and the properties of the designated redirection controller, and (2) providing a higher sense of immersion to users with better safety by precluding any user resets and reducing the total number of reset counts at the same time. We believe that our work will lay the groundwork for further studies on *space subdivision strategy* that uses dynamic shutters.

For future work, we first suggest studying dynamic space subdivision methods on environments with complex shapes and internal obstacles with irregular shapes. The findings from those studies will empower dynamic space subdivision methods like OSP to be more robust to various environments. Moreover, curved shutters can be introduced instead of simple linear shutters when using *space subdivision strategy*. Second, analyzing OSP's temporal coherence in detail and its impact on the performance can also be suggested for future study. In real environments, the decisions made by our trained agent are not always temporally coherent, which might lead to momentary unsteady redirection of users in the subtle technique. (However, note that our framework is applied with detection threshold [Hodgson and Bachmann 2013; Hodgson et al. 2014; Steinicke et al. 2009a, 2008a, 2009b, 2008b] so its impact is not recognized by users. Moreover, it is redundant to tune those small incoherences of virtual shutters in practice as they don't actually exist in the physical space and need not to be rendered.) Understanding the nature of this behavior will lay the groundwork for improving our framework. In addition, using more RDW methods [Li et al. 2021c] with *space subdivision strategy* and analyzing them with more diverse measures [Azmandian et al. 2022; Williams et al. 2022] can also be suggested. Lastly, although we have limited the scope of the path patterns to only three types in our training curriculum, we believe that using self-paced curriculum learning [Jiang et al. 2015] with reinforcement learning can facilitate our model to learn a broader range of path patterns in a more unsupervised setting. This will develop more efficient space partitioning methods that respect the various and fluid path patterns of users enjoying different VR contents.

ACKNOWLEDGMENTS

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (No. NRF-2020R1A2C2014622)

REFERENCES

Franz Aurenhammer. 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)* 23, 3 (1991), 345–405.

Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. 2016. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*. IEEE, 9–14.

Mahdi Azmandian, Timofey Grechkin, and Evan Suma Rosenberg. 2017. An evaluation of strategies for two-user redirected walking in shared physical spaces. In *2017 IEEE Virtual Reality (VR)*. IEEE, 91–98.

Mahdi Azmandian, Rhys Yahata, Timofey Grechkin, Jerald Thomas, and Evan Suma Rosenberg. 2022. Validating simulation-based evaluation of redirected walking systems. *IEEE Transactions on Visualization & Computer Graphics* 01 (2022), 1–1.

Eric R Bachmann, Eric Hodgson, Cole Hoffbauer, and Justin Messinger. 2019. Multi-user redirected walking and resetting using artificial potential fields. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2022–2031.

Eric R Bachmann, Jeanette Holm, Michael A Zmuda, and Eric Hodgson. 2013. Collision prediction and prevention in a simultaneous two-user immersive virtual environment. In *2013 IEEE Virtual Reality (VR)*. IEEE, 89–90.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

Luke Bölling, Niklas Stein, Frank Steinicke, and Markus Lappe. 2019. Shrinking circles: Adaptation to increased curvature gain in redirected walking. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2032–2039.

Yuchen Chang, Keigo Matsumoto, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2021. Redirection controller using reinforcement learning. *IEEE Access* 9 (2021), 145083–145097.

Ze-Yin Chen, Yi-Jun Li, Miao Wang, Frank Steinicke, and Qinping Zhao. 2021. A Reinforcement Learning Approach to Redirected Walking with Passive Haptic Feedback. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 184–192.

Yong-Hun Cho, Dae-Hong Min, Jin-Suk Huh, Se-Hee Lee, June-Seop Yoon, and In-Kwon Lee. 2021. Walking Outside the Box: Estimation of Detection Thresholds for Non-Forward Steps. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 448–454.

Emily Dao, Andreea Muresan, Kasper Hornbæk, and Jarrod Knibbe. 2021. Bad Breakdowns, Useful Seams, and Face Slapping: Analysis of VR Fails on YouTube. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

Tianyang Dong, Xianwei Chen, Yifan Song, Wenyuan Ying, and Jing Fan. 2020. Dynamic artificial potential fields for multi-user redirected walking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 146–154.

Tianyang Dong, Yue Shen, Tieqi Gao, and Jing Fan. 2021a. Dynamic Density-based Redirected Walking Towards Multi-user Virtual Environments. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 626–634.

Zhi-Chao Dong, Xiao-Ming Fu, Zeshi Yang, and Ligang Liu. 2019. Redirected smooth mappings for multiuser real walking in virtual reality. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 1–17.

Zhi-Chao Dong, Xiao-Ming Fu, Chi Zhang, Kang Wu, and Ligang Liu. 2017. Smooth assembled mappings for large-scale real walking. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Zhi-Chao Dong, Wenming Wu, Zenghao Xu, Qi Sun, Guanjie Yuan, Ligang Liu, and Xiao-Ming Fu. 2021b. Tailored Reality: Perception-aware Scene Restructuring for Adaptive VR Navigation. *ACM Transactions on Graphics (TOG)* 40, 5 (2021), 1–15.

Qiang Du, Maria Emelianenko, and Lili Ju. 2006. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM journal on numerical analysis* 44, 1 (2006), 102–119.

Qiang Du, Vance Faber, and Max Gunzburger. 1999. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review* 41, 4 (1999), 637–676.

Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. 2007. G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods* 39, 2 (2007), 175–191.

Steven Fortune. 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2, 1 (1987), 153–174.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

Matthew Hausknecht and Peter Stone. 2015. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Eric Hodgson and Eric Bachmann. 2013. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE transactions on visualization and computer graphics* 19, 4 (2013), 634–643.

Eric Hodgson, Eric Bachmann, and Tyler Thrash. 2014. Performance of redirected walking algorithms in a constrained virtual world. *IEEE transactions on visualization and computer graphics* 20, 4 (2014), 579–587.

Jeannette E Holm. 2012. *Collision prediction and prevention in a simultaneous multi-user immersive virtual environment.* Ph. D. Dissertation. Miami University.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. 2018. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* (2018).

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. 1993. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology* 3, 3 (1993), 203–220.

Dooyoung Kim, Jae-eun Shin, Jeongmi Lee, and Woontack Woo. 2021. Adjusting Relative Translation Gains According to Space Size in Redirected Walking for Mixed Reality Mutual Space Generation. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 653–660.

Vijay R Konda and John N Tsitsiklis. 2000. Actor-critic algorithms. In *Advances in neural information processing systems*. 1008–1014.

Bruce Krogh. 1984. A generalized potential field approach to obstacle avoidance control. In *Proc. SME Conf. on Robotics Research: The Next Five Years and Beyond, Bethlehem, PA, 1984*. 11–22.

Eike Langbehn, Paul Lubos, Gerd Bruder, and Frank Steinicke. 2017. Bending the curve: Sensitivity to bending of curved paths and application in room-scale vr. *IEEE transactions on visualization and computer graphics* 23, 4 (2017), 1389–1398.

Eike Langbehn, Paul Lubos, and Frank Steinicke. 2018a. Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In *Proceedings of the Virtual Reality International Conference-Laval Virtual*. 1–9.

Eike Langbehn, Frank Steinicke, Markus Lappe, Gregory F Welch, and Gerd Bruder. 2018b. In the blink of an eye: leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

Dong-Yong Lee, Yong-Hun Cho, and In-Kwon Lee. 2019. Real-time optimal planning for redirected walking using deep q-learning. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 63–71.

Dong-Yong Lee, Yong-Hun Cho, Dae-Hong Min, and In-Kwon Lee. 2020. Optimal planning for redirected walking based on reinforcement learning in multi-user environment with irregularly shaped physical space. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 155–163.

Changyang Li, Haikun Huang, Jyh-Ming Lien, and Lap-Fai Yu. 2021a. Synthesizing scene-aware virtual reality teleport graphs. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–15.

Rongpeng Li, Chujie Wang, Zhifeng Zhao, Rongbin Guo, and Honggang Zhang. 2020. The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility. *IEEE Communications Letters* 24, 9 (2020), 2005–2009.

Yi-Jun Li, De-Rong Jin, Miao Wang, Jun-Long Chen, Frank Steinicke, Shi-Min Hu, and Qinping Zhao. 2021b. Detection Thresholds with Joint Horizontal and Vertical Gains in Redirected Jumping. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 95–102.

Yi-Jun Li, Miao Wang, Frank Steinicke, and Qinping Zhao. 2021c. OpenRDW: A Redirected Walking Library and Benchmark with Multi-User, Learning-based Functionalities and State-of-the-art Algorithms. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 21–30.

JJ-W Lin, Henry Been-Lirn Duh, Donald E Parker, Habib Abi-Rached, and Thomas A Furness. 2002. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Proceedings ieee virtual reality 2002*. IEEE, 164–171.

Sebastian Marwecki, Maximilian Brehm, Lukas Wagner, Lung-Pan Cheng, Florian'Floyd' Mueller, and Patrick Baudisch. 2018. Virtualspace-overloading physical space with multiple virtual reality users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–10.

Justin Messinger, Eric Hodgson, and Eric R Bachmann. 2019. Effects of tracking area shape and size on artificial potential field redirected walking. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 72–80.

Thomas Nescher, Ying-Yin Huang, and Andreas Kunz. 2014. Planning redirection techniques for optimal free walking experience using model predictive control. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 111–118.

Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. 2018. 15 years of research on redirected walking in immersive virtual environments. *IEEE computer graphics and applications* 38, 2 (2018), 44–56.

Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. 2009. *Spatial tessellations: concepts and applications of Voronoi diagrams*. Vol. 501. John Wiley & Sons.

Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).

S. Razzaque. 2005. *Redirected walking*. Ph. D. Dissertation. The University of North Carolina at Chapel Hill.

Z. Kohn S. Razzaque and M. C. Whitton. 2001. Redirected walking. *Proceedings of the Eurographics, Manchester, UK* (2001), 5–7.

Thomas Schubert, Frank Friedmann, and Holger Regenbrecht. 2001. The experience of presence: Factor analytic insights. *Presence: Teleoperators & Virtual Environments* 10, 3 (2001), 266–281.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

Ana Serrano, Diego Gutierrez, Belen Masia, Daniel Martin, and Karol Myszkowski. 2020. *Imperceptible manipulation of lateral camera motion for improved virtual reality applications.* Technical Report.

Wataru Shibayama and Shinichi Shirakawa. 2020. Reinforcement Learning-Based Redirection Controller for Efficient Redirected Walking in Virtual Maze Environment. In *Computer Graphics International Conference*. Springer, 33–45.

Daniel J Simons and Daniel T Levin. 1997. Change blindness. *Trends in cognitive sciences* 1, 7 (1997), 261–267.

Frank Steinicke, Gerd Bruder, Klaus Hinrichs, Jason Jerald, Harald Frenz, and Markus Lappe. 2009a. Real walking through virtual environments by redirection techniques. *JVRB-Journal of Virtual Reality and Broadcasting* 6, 2 (2009).

Frank Steinicke, Gerd Bruder, Klaus Hinrichs, and Pete Willemsen. 2010. Change blindness phenomena for stereoscopic projection systems. In *2010 IEEE Virtual Reality Conference (VR)*. IEEE, 187–194.

Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. 2008a. Analyses of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. 149–156.

Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. 2009b. Estimation of detection thresholds for redirected walking techniques. *IEEE transactions on visualization and computer graphics* 16, 1 (2009), 17–27.

Frank Steinicke, Gerd Bruder, Timo Ropinski, and Klaus Hinrichs. 2008b. Moving towards generally applicable redirected walking. In *Proceedings of the Virtual Reality International Conference (VRIC)*. IEEE Press, 15–24.

Ryan R Strauss, Raghuram Ramanujan, Andrew Becker, and Tabitha C Peck. 2020. A steering algorithm for redirected walking using reinforcement learning. *IEEE transactions on visualization and computer graphics* 26, 5 (2020), 1955–1963.

Evan A Suma, Gerd Bruder, Frank Steinicke, David M Krum, and Mark Bolas. 2012a. A taxonomy for deploying redirection techniques in immersive virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*. IEEE, 43–46.

Evan A Suma, Seth Clark, David Krum, Samantha Finkelstein, Mark Bolas, and Zachary Warte. 2011. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*. IEEE, 159–166.

Evan A Suma, Zachary Lipps, Samantha Finkelstein, David M Krum, and Mark Bolas. 2012b. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (2012), 555–564.

Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie Kaufman. 2018. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

Jerald Thomas and Evan Suma Rosenberg. 2019. A general reactive algorithm for redirected walking using artificial potential functions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 56–62.

Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. 1999. Walking > walking-in-place > flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 359–364.

Martin Usoh, Ernest Catena, Sima Arman, and Mel Slater. 2000. Using presence questionnaires in reality. *Presence* 9, 5 (2000), 497–503.

Khrystyna Vasylevska, Hannes Kaufmann, Mark Bolas, and Evan A Suma. 2013. Flexible spaces: Dynamic layout generation for infinite walking in virtual environments. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 39–42.

Georges Voronoi. 1908. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1908, 134 (1908), 198–287.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.

Betsy Williams, Gayathri Narasimham, Bjoern Rump, Timothy P McNamara, Thomas H Carr, John Rieser, and Bobby Bodenheimer. 2007. Exploring large virtual environments with an HMD when physical space is limited. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization*. 41–48.

Niall L Williams, Aniket Bera, and Dinesh Manocha. 2021a. Arc: Alignment-based redirection controller for redirected walking in complex environments. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2535–2544.

Niall L Williams, Aniket Bera, and Dinesh Manocha. 2021b. Redirected walking in static and dynamic scenes using visibility polygons. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4267–4277.

Niall L Williams, Aniket Bera, and Dinesh Manocha. 2022. ENI: Quantifying Environment Compatibility for Natural Walking in Virtual Reality. *arXiv preprint arXiv:2201.01261* (2022).

Niall L Williams and Tabitha C Peck. 2019. Estimation of rotation gain thresholds considering fov, gender, and distractors. *IEEE transactions on visualization and computer graphics* 25, 11 (2019), 3158–3168.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

Jackie Yang, Christian Holz, Eyal Ofek, and Andrew D Wilson. 2019. Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1093–1107.

Jingxin Zhang, Eike Langbehn, Dennis Krupke, Nicholas Katzakis, and Frank Steinicke. 2018. Detection thresholds for rotation and translation gains in 360 video-based telepresence systems. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1671–1680.

Michael A Zmuda, Joshua L Wonser, Eric R Bachmann, and Eric Hodgson. 2013. Optimizing constrained-environment redirected walking instructions using search techniques. *IEEE transactions on visualization and computer graphics* 19, 11 (2013), 1872–1884.